An adaptive time step scheme for a system of stochastic differential equations with multiple multiplicative noise: Chemical Langevin equation, a proof of concept

Vassilios Sotiropoulos and Yiannis N. Kaznessis

Department of Chemical Engineering and Materials Science, University of Minnesota, 421 Washington Ave. SE, Minneapolis, Minnesota 55455, USA

(Received 15 December 2006; accepted 23 October 2007; published online 2 January 2008)

Models involving stochastic differential equations (SDEs) play a prominent role in a wide range of applications where systems are not at the thermodynamic limit, for example, biological population dynamics. Therefore there is a need for numerical schemes that are capable of accurately and efficiently integrating systems of SDEs. In this work we introduce a variable size step algorithm and apply it to systems of stiff SDEs with multiple multiplicative noise. The algorithm is validated using a subclass of SDEs called chemical Langevin equations that appear in the description of dilute chemical kinetics models, with important applications mainly in biology. Three representative examples are used to test and report on the behavior of the proposed scheme. We demonstrate the advantages and disadvantages over fixed time step integration schemes of the proposed method, showing that the adaptive time step method is considerably more stable than fixed step methods with no excessive additional computational overhead. © 2008 American Institute of Physics.

[DOI: 10.1063/1.2812240]

I. INTRODUCTION

In recent years there has been an increasing interest in stochastic modeling. Fields such as computational system biology have been impacted greatly by the observation that biological systems are inherently stochastic.1 Studying the effects of the intrinsic fluctuations of cell species in the overall cell phenotype requires the development of sophisticated stochastic models,2 including stochastic-discrete and stochastic-continuous. Both have been introduced by Gillespie.3,4 Recent literature has focused on developing algorithms that integrate multiple time scales present in the system kinetics. Building on the work of Gillespie there have been many algorithms that continue and improve its initial work for simulating stochastic chemical systems.5–15 The majority of these algorithms have been incorporated in software suites that are able to simulate multiscale models of biological reacting networks, but are not limited to them.16–19

In this work we focus on chemical reacting systems that are modeled by a set of stochastic differential or chemical Langevin equations. Importantly we present an adaptive time step algorithm that numerically integrates stochastic differential equations (SDEs) involving multiple time scales.

The challenge is to develop integration schemes for SDEs that are both accurate and as fast as their deterministic equivalents. While SDEs can be numerically integrated in a similar fashion as ordinary differential equations (ODEs), there are significant differences in the two approaches. The major one stems from the fact that the classic chain rule found in the deterministic case is substituted by the well known Itô formula in stochastic calculus. This complicates the extraction of numerical methods from an Itô-Taylor expansion, since extra terms are introduced. The latter reduces to the chain rule formula only for linear systems. Moreover, the theory behind SDEs becomes complicated and differs from that of ODEs for adaptive and implicit integration methods. However, there is a sufficient number of numerical schemes for SDEs, starting from the simple Euler-Maruyama method, going on with the Milstein method, and continuing with higher order schemes such as Runge-Kutta methods. To that we can add the explicit and implicit, partial implicit or fully implicit, versions of the methods.20–22 A detailed description and analysis of these methods can be found in the well written book of Kloeden and Platen.23

Similar to ODEs, multiple time scales in the underlying models cause a system of SDEs to become mathematically stiff. Conventional fixed step methods, in both the stochastic and the deterministic regime, require a small time step for integrating stiff systems. Therefore they become computationally slow. In addition, stiffness may arise during some parts of the simulation allowing for a larger time step in the remaining time interval. Hence the need for an adaptive time step method that will adjust the time step accordingly is evident. To our knowledge, and in contrast to the deterministic cases, adaptive time stepping strategies for SDEs are significantly less developed and limited to special cases. Although recently there has been a considerable effort to develop adaptive time stepping schemes for SDEs, the majority of them deals only with cases where there is only a single Wiener process and where SDE terms are commutative.24–27 In the literature there are variable step size algorithms which in their majority use higher order methods to numerically integrate the system.24,25 This implies that multiple Itô integrals
have to be approximated increasing the computational cost. An adaptive scheme based on the Euler-Maruyama method is also available, but with a significant implementation cost.28

In this paper we present an adaptive time stepping scheme for integrating systems of stiff SDEs with multiple multiplicative noise. These types of SDEs are the most difficult to numerically integrate due to the intense coupling of the noise terms and the existing stiffness. We use the Milstein method as our numerical method and combine it with local error criteria originating from the work of Lamba.27 These determine when the adaptive time stepping selection mechanism should be introduced. For the variable step size scheme we choose to use the methodology of Gaines and Lyons26 that introduced the notion of Brownian trees originated from the work of Lévy.29 Brownian trees are based on a binary logic; the step can be either halved or doubled.

Finally, the developed framework is applied and tested to a particular class of SDEs, the chemical Langevin equations (CLEs) that arise in the description of dilute chemical reacting systems far from the thermodynamic limit.4 Such systems can be described as continuous Markov processes governed by a chemical master equation that reduces to a Fokker-Plank equation.14 The CLE is an Itô stochastic differential equation with multiplicative noise and represents one possible solution of the Fokker-Plank equation. These kinds of systems have recently appeared in the field of computational biology, modeling cell processes and interactions of cell species, where fluctuations play a key role. Three examples are chosen, the first is a system of linear SDEs with multiple multiplicative noise and the second is a nonlinear system. In the final example we use an actual stiff biological example to test the performance of the proposed algorithm in a realistic biological network. Therefore the adaptive scheme is integrated into HY3S, a collection of multiscale algorithms that use CLEs to propagate system of reactions that belong in the continuous Markov process regime.16 HY3S is capable of simulating the stochastic dynamics of networks of biochemical reactions and has been used to study the dynamic behavior of gene regulatory networks.30

The present paper is organized as follows. In the Theory section (Sec. II) we formulate the problem and then discuss convergence properties of numerical solutions for SDEs. Next we outline the Milstein method, the numerical scheme used predominantly in CLE integration. Then we describe the binary adaptive time step selection based on the notion of Brownian trees and continue with the introduction of the local error criteria. Next, we briefly provide some necessary background information about CLEs, and finally we discuss implementation issues. In the Examples section (Sec. III) we test and report on the behavior of the proposed scheme, and finally in the Discussion section (Sec. IV) we conclude and argue about the contribution of the present scheme.

Very recently, a new method was published by Lamba et al. that demonstrates strong convergence of an adaptive time stepping scheme for SDEs based on the Euler-Maruyama method instead of the Milstein method used in the present work.31 While their method would be likely faster for a majority of SDEs, since it has less implementation requirements, mainly in that it does not require the approximation of a two dimensional Itô integral, it fails to work for most CLEs because they do not always satisfy Assumption 5.1 in their manuscript. This assumption requires that the number of reactants be equal to the number of reactions which is not the case in the majority of CLEs and also requires the CLEs to follow a dissipativity condition which is not always necessarily met.

For the first time an adaptive time stepping method is presented for integrating systems of chemical Langevin equations accurately and efficiently. Chemical Langevin equations are stochastic differential equations with multiple multiplicative noises and belong to the subclass of SDEs that are among the hardest to numerically integrate due to the noncommutativity and the multiple noise terms. To our knowledge there are no simple and relative easy to numerically implement schemes that accurately and efficiently overcome noncommutativity terms and multiplicative noise. Most previously reported schemes rely on higher order Runge-Kutta algorithms and are harder to implement. Importantly, the scheme is simple to implement and can be potentially applied to any system of SDEs beyond CLEs.

II. THEORY

We consider a system of Itô SDEs with multiple multiplicative noise,

$$dX_i = f_i(X(t))dt + \sum_{j=1}^{M} g_{i,j}(X(t))dW_j(t) \quad i = 1, \ldots, N,$$

where $X(t)$ is a N-dimensional state vector. In the case of biomolecular systems, for example, $X(t)$ can be the vector with the concentration of the N species. $W(t)$ is an M-dimensional Wiener process. A Wiener process $W$ is a Gaussian process with the following properties:

$$E(W(t) = 0), \quad E(W(t)W(s)) = \min(t,s),$$

where $E(y)$ is the expectation value of variable $y$. Additionally, $f_i(X(t))$, $g_{i,j}(X(t))$ are scalars with values depending on the state vector of the system. The first part on the right hand side of Eq. (1) is called the drift term, while the second is usually referred to as the diffusion part. For chemical reaction networks $f_i(X(t))$ is associated with the deterministic reaction rates and $g_{i,j}(X(t))$ are the terms of the fluctuations for a system away from the thermodynamic limit.

There are two ways to write down a SDE, using either the Itô or the Stratonovich formulation. Equation (1) uses the Itô formulation. The main difference between the two lies in the way the stochastic integrals are computed. The Itô formulation computes the integral in the beginning of each sub-interval, while Stratonovich computes it in the middle. The two forms are equivalent, and we can obtain the Stratonovich from the Itô form by using a simple formula.23 From a purely mathematical point of view, both representations are appropriate. From a physical point of view, Itô SDEs are more appropriate describing systems in which intrinsic noise is important.23 In the remainder of the paper we will consider SDEs in the Itô form, since CLEs are also cast in Itô form.
A. Strong and weak convergence

Before dealing with the actual numerical integration schemes, there is an important concept we have to consider: convergence of the numerical solution. Although these definitions are found in the literature we present these definitions here for completeness. In numerical methods used to integrate SDEs, two definitions of convergence are present, the weak and the strong convergence. The weak convergence deals with convergence in the probability distribution of the actual and numerical solution. On the other hand, strong convergence deals with convergence between the trajectories of the actual and numerical solution. In equation form the above definitions are formulated as follows.

**Strong convergence.** A time discrete approximation \( X_d(t) \) of the Itô process \( X_t \), converges strongly to \( X_t \) with order \( \gamma \geq 0 \) at time \( T \), if there exists a positive constant \( C \), independent of \( \delta \) and \( \delta_0 > 0 \), such that,

\[
e(\delta) = E(|E(X_T - X_d(T))|) \leq C \delta^\gamma, \quad \delta \in (0, \delta_0).
\]  (3)

**Weak convergence.** A time discrete approximation \( X_d(t) \), of the Itô process \( X_t \), converges weakly to \( X_t \) with order \( \beta \geq 0 \) at time \( T \), if for each \( q \in C[0,T] \) [\( q \) is \( 2(\beta+1) \) times differentiable] there exists a positive constant \( C \), independent of \( \delta \) and \( \delta_0 > 0 \), such that,

\[
e(\delta) = |E(q(X_T)) - E(q(X_d(T)))| \leq C \delta^\beta, \quad \delta \in (0, \delta_0).
\]  (4)

In practical situations when algorithms are used to numerically compute the solution of SDEs, Eqs. (3) and (4) can be used to calculate the error. Though Eq. (3) is more appropriate, its implementation for computing the strong mean and variance errors requires fixing the Brownian paths that the Wiener process follows. Practically this means we have to use the same sequence of random numbers to compute the actual solution, analytically if possible, and the approximate one, which is not always applicable.

B. Milstein method

The simplest scheme to numerically integrate Eq. (1) is the explicit Euler-Maruyama method. It is derived from first order truncation of the Itô-Taylor expansion. The Euler-Maruyama method has a strong order of 0.5 and weak order of 1.0. The scheme for Eq. (1) has the form

\[
X^{k+1}_i = X^k_i + f(i)(X^k)\Delta t + \sum_{j=1}^{M} g_{ij}(X^k)I(j)
i = 1, \ldots, N,
\]  (5)

where

\[
I(j) = \int_{t \Delta t}^{t \Delta t} dW_j(t) = N_j(0, \Delta t)
\]  (6)

is one dimensional (1D) Itô integral. In practical applications the 1D Itô integral of Eq. (6) can be approximated as a Gaussian random number with zero mean and variance \( \Delta t \), \( N(0, \Delta t) \).

The method we concentrate on is the explicit Milstein method, which is similar to the Euler-Maruyama method with the only difference being the addition of an extra term containing a two dimensional Itô integral. This extra term is of order \( O(\Delta t^2) \) and is responsible for increasing the strong order convergence to 1.0 compared to the 0.5 order of the Euler-Maruyama, while the weak order is also 1.0. The Milstein scheme is derived also from an Itô-Taylor expansion. Applying it directly to Eq. (1),

\[
X^{k+1}_i = X^k_i + f(i)(X^k)\Delta t + \sum_{j=1}^{M} g_{ij}(X^k)I(j) + \sum_{j=1}^{M} \sum_{j=1}^{M} L_{ij} g_{ij}(X^k)I(j,j_2), \quad i = 1, \ldots, N,
\]  (7)

where \( I(j,j_2) \) is a two dimensional Itô integral,

\[
I(j,j_2) = \int_{t \Delta t}^{t \Delta t} dW_j(t) dW_j(t_2),
\]  (8)

and \( L_{ij} \) is an operator defined as follows:

\[
L_{ij} = \sum_{d=1}^{M} g_{id,j} \frac{\partial}{\partial X_d}.
\]  (9)

The difficult part in the implementation of this method is the approximation of the two dimensional (2D) Itô integral [Eq. (8)]. When \( j_1 = j_2 \) the expression for the Itô integral simplifies to

\[
I(j_1,j_1) = \frac{1}{2}((\Delta W_{j_1})^2 - \Delta t).
\]  (10)

When \( j_1 \neq j_2 \) with \( j_1,j_2 = 1,\ldots,M \) the Itô and Stratonovic versions of the integral are equal, and thus we can use the following approximation which is based on the Stratonovic definition of the integral. The approximation is based on a Fourier expansion \(^{23}\) of the Stratonovic version of Eq. (8).

\[
P(j_1,j_2) = \Delta t \left\{ \frac{1}{2} \xi_{j_1} \xi_{j_2} + \sqrt{\rho_p} \left( \mu_{j_1} \xi_{j_2} - \mu_{j_2} \xi_{j_1} \right) \right\}
+ \frac{\Delta t}{2\pi} \sum_{r=1}^{p} \left\{ \xi_{j_1,r} \left( \sqrt{2} \xi_{j_2} + \eta_{j_2,r} \right) - \xi_{j_2,r} \left( \sqrt{2} \xi_{j_1} 
+ \eta_{j_1,r} \right) \right\},
\]  (11)

where

\[
\rho_p = \frac{1}{12} - \frac{1}{2\pi^2} \sum_{r=1}^{p} \frac{1}{r^2}
\]  (12)

for \( j = 1, \ldots, M \), \( r = 1, \ldots, p \), and \( p = 1,2, \ldots \) determines the number of Fourier expansion terms retained in the solution. \( \xi_j \), \( \mu_{j,p} \), \( \eta_{j,r} \), and \( \xi_{j,r} \), being all independent normal Gaussian random numbers, \( N(0,1) \). In particular \( \xi_j \) is correlated to \( \Delta W_j \) by the following relation:

\[
\xi_j = \frac{1}{\sqrt{\Delta t}} \Delta W_j,
\]  (13)

where \( \Delta W_j \) is again approximated as a Gaussian random number with zero mean and variance \( \Delta t \), \( N(0, \Delta t) \). The choice of \( p \) value determines the accuracy of the 2D Itô integral approximation. In this work it is chosen so it guar-
The notion of Brownian trees introduced by Gaines and Lyons based on results of Lévy becomes very useful in selecting the new Wiener increments ($\Delta W_j$) conditioned on the previous ones. Brownian trees are based on the following binary logic: the time integration step can be either halved or doubled. Generalizations of the Brownian trees where the time step is increased in multiples between 0.0 and 2.0 exist in the literature, but require extra effort to compute the new Wiener increments. A Brownian tree is made up of increments of Wiener processes, $\Delta W_{i,j}$, where $j$ indicates the row and $i$ the branch, where $\Delta W$ is located on the tree. The values of each row are conditioned to the corresponding values of the previous row. Also each $\Delta W_{i,j}$ corresponds to a time step that is equal to the initial time step divided by $2^{j-1}$.

In Fig. 1, a schematic representation of a Brownian tree is presented. The top row corresponds to the initial time step of the simulation $\Delta t_{ini}$ and “houses” the corresponding Gaussian random numbers (for a single SDE, $\Delta W_{1,1}$ is a scalar value, while for a system of SDEs $\Delta W_{1,1}$ is a vector). Additional branches and rows are created when halving the time step. The number of branches of each row is $2^{j-1}$ and the corresponding time step for each $\Delta W_{i,j}$ is $\Delta t_{ini}/2^{j-1}$, where $j$ is the row number. The construction of the Brownian tree, i.e., evaluating the Wiener increments of each branch, utilizes a relationship introduced by Lévy,

\[
\Delta W_{2k-1,j+1} = \frac{1}{2} \Delta W_{k,j} + y_{k,j},
\]

\[j = 1, 2, \ldots \text{ number of rows},\]

\[
\Delta W_{2k,j+1} = \frac{1}{2} \Delta W_{k,j} - y_{k,j}, \quad k = 1, 2, \ldots, 2^{j-1},\]  

where $y_{k,j}$ follows a normal distribution with zero mean and variance 2$^{-j}$ and $k$ denotes the number of nodes with each node having two branches. This procedure assures that the integration will remain on the proper path initially defined by $\Delta W_{1,1}$. It is important to note that for the definition of the Brownian tree only $\Delta W_{1,1}$ is necessary, since all other rows follow from Eq. (14). This allows for a convenient way to dynamically generate the tree during simulations.

Going up and down in the Brownian tree or equivalently altering the time step depends on whether or not a set of predefined constraints are met. For example, if we try an initial time step $\Delta t_{ini}$ with $\Delta W_{1,1}$ and one or more of the constraints are not met, the system is “rewound” and half the initial step size is tried where now $\Delta W_{1,2}$ is used. If that fails then $\Delta t_{ini}/4$ with $\Delta W_{1,3}$ is tried, but if the criteria are met the system is propagated with the remaining half step and $\Delta W_{2,2}$. The procedure continues until the final time has been reached. Every time the step size is accepted the algorithm checks if it is possible to climb up the tree, otherwise the time step is kept the same. The step can be doubled when-

\[
\Delta W_{2k-1,j+1} = \frac{1}{2} \Delta W_{k,j} + y_{k,j},
\]

\[j = 1, 2, \ldots \text{ number of rows},\]

\[
\Delta W_{2k,j+1} = \frac{1}{2} \Delta W_{k,j} - y_{k,j}, \quad k = 1, 2, \ldots, 2^{j-1},\]

where $y_{k,j}$ follows a normal distribution with zero mean and variance 2$^{-j}$ and $k$ denotes the number of nodes with each node having two branches. This procedure assures that the integration will remain on the proper path initially defined by $\Delta W_{1,1}$. It is important to note that for the definition of the Brownian tree only $\Delta W_{1,1}$ is necessary, since all other rows follow from Eq. (14). This allows for a convenient way to dynamically generate the tree during simulations.

Going up and down in the Brownian tree or equivalently altering the time step depends on whether or not a set of predefined constraints are met. For example, if we try an initial time step $\Delta t_{ini}$ with $\Delta W_{1,1}$ and one or more of the constraints are not met, the system is “rewound” and half the initial step size is tried where now $\Delta W_{1,2}$ is used. If that fails then $\Delta t_{ini}/4$ with $\Delta W_{1,3}$ is tried, but if the criteria are met the system is propagated with the remaining half step and $\Delta W_{2,2}$. The procedure continues until the final time has been reached. Every time the step size is accepted the algorithm checks if it is possible to climb up the tree, otherwise the time step is kept the same. The step can be doubled when-

\[
\Delta W_{2k-1,j+1} = \frac{1}{2} \Delta W_{k,j} + y_{k,j},
\]

\[j = 1, 2, \ldots \text{ number of rows},\]

\[
\Delta W_{2k,j+1} = \frac{1}{2} \Delta W_{k,j} - y_{k,j}, \quad k = 1, 2, \ldots, 2^{j-1},\]  

where $y_{k,j}$ follows a normal distribution with zero mean and variance 2$^{-j}$ and $k$ denotes the number of nodes with each node having two branches. This procedure assures that the integration will remain on the proper path initially defined by $\Delta W_{1,1}$. It is important to note that for the definition of the Brownian tree only $\Delta W_{1,1}$ is necessary, since all other rows follow from Eq. (14). This allows for a convenient way to dynamically generate the tree during simulations.

Going up and down in the Brownian tree or equivalently altering the time step depends on whether or not a set of predefined constraints are met. For example, if we try an initial time step $\Delta t_{ini}$ with $\Delta W_{1,1}$ and one or more of the constraints are not met, the system is “rewound” and half the initial step size is tried where now $\Delta W_{1,2}$ is used. If that fails then $\Delta t_{ini}/4$ with $\Delta W_{1,3}$ is tried, but if the criteria are met the system is propagated with the remaining half step and $\Delta W_{2,2}$. The procedure continues until the final time has been reached. Every time the step size is accepted the algorithm checks if it is possible to climb up the tree, otherwise the time step is kept the same. The step can be doubled when-

\[
\Delta W_{2k-1,j+1} = \frac{1}{2} \Delta W_{k,j} + y_{k,j},
\]

\[j = 1, 2, \ldots \text{ number of rows},\]

\[
\Delta W_{2k,j+1} = \frac{1}{2} \Delta W_{k,j} - y_{k,j}, \quad k = 1, 2, \ldots, 2^{j-1},\]  

where $y_{k,j}$ follows a normal distribution with zero mean and variance 2$^{-j}$ and $k$ denotes the number of nodes with each node having two branches. This procedure assures that the integration will remain on the proper path initially defined by $\Delta W_{1,1}$. It is important to note that for the definition of the Brownian tree only $\Delta W_{1,1}$ is necessary, since all other rows follow from Eq. (14). This allows for a convenient way to dynamically generate the tree during simulations.

Going up and down in the Brownian tree or equivalently altering the time step depends on whether or not a set of predefined constraints are met. For example, if we try an initial time step $\Delta t_{ini}$ with $\Delta W_{1,1}$ and one or more of the constraints are not met, the system is “rewound” and half the initial step size is tried where now $\Delta W_{1,2}$ is used. If that fails then $\Delta t_{ini}/4$ with $\Delta W_{1,3}$ is tried, but if the criteria are met the system is propagated with the remaining half step and $\Delta W_{2,2}$. The procedure continues until the final time has been reached. Every time the step size is accepted the algorithm checks if it is possible to climb up the tree, otherwise the time step is kept the same. The step can be doubled when-
ever the branch number is divisible by two. This procedure allows for a flexible time step size that will decrease if necessary and increase when able.

The procedure described above corresponds to a SDE with a single Wiener process. For a system of SDEs with multiple noise terms ($M$ Wiener increments) the above mentioned scheme can be easily generalized. The main difference is that at every branch of the Brownian tree instead of $\Delta W_{i,j}$ being a scalar, $\Delta W_{i,j}$ is a vector containing the $M$ Wiener increments. As in the single Wiener process case, we use relation (14) to generate the corresponding vector elements of each branch. Relationship (14) is applied at each vector element. Again we only need to know the elements of vector $\Delta W_{1,1}$ in order to construct all subsequent rows and branches of the tree. Finally the strategy for climbing up and down the tree is independent of the number of Wiener processes and therefore is identical in either the single or multiple Wiener processes cases.

In Fig. 2 we use MATLAB to generate two different visualizations of the Brownian tree. In both figures an initial value of the random process $W \sim N(0, \Delta t)$ is chosen and through Eq. (14) the underlying rows are created. $\Delta t_{ini}$ is chosen to be 0.01 s for simplicity, i.e., $\Delta W \sim N(0, \Delta t)$. In Fig. 2(a) the values between each dot depict a Wiener increment corresponding to $\Delta W_{i,j}$, $i=1,2,3,4$. In other words, Fig. 2(a) is an example realization of Fig. 1, where the bold (blue), dashed (green), and bold with dot (red) lines depict the first, second, and third rows, respectively. Note that all three lines have the same start and end points which basically summarizes the idea of Brownian trees, remain on the correct Brownian path. From Fig. 2(b) we observe that as the row number increases the Wiener increment become finer and finer.

D. Error criteria

In order for the adaptive scheme to decide whether or not to alter the time step local error criteria are necessary. Lambda proposed a set of criteria that determine the local error on both the drift and diffusion terms. In that work both criteria are present for a SDE with one dimensional Wiener process. Based on these derivations, we extend the criteria so that they apply to a system of SDEs with multiple multiplicative noise as in Eq. (1). The drift local error ($E_d$) is, namely,

$$E_d(X^k, \Delta t) = \frac{\Delta t}{2} \left\| f(X^k + \Delta t f(X^k)) - f(X^k) \right\|_\infty. \quad (15)$$

The infinity norm corresponds to the maximum absolute sum along the row dimension. The drift local error is of order $O(\Delta t^2)$. The local error of the diffusion term ($E_d$) is

$$E_d(X^k, \Delta t) = \frac{1}{2} \|\Delta W\|_{\infty} \cdot \|h^t \cdot h\|_\infty, \quad (16)$$

where $\Delta W$ is an $M$-dimensional vector with cubed Gaussian random numbers as elements, $h$ ($N \times 1$ matrix) contains the sum of the corresponding row elements of matrix $g(X(t))$, meaning $h_j = \sum_i g_{ij}$, $h^t$ ($N \times N$ matrix) is the Jacobian of $h$, and $\cdot$ symbolizes matrix vector multiplication. Again the infinity norm corresponds to the maximum absolute sum along the row dimension. Both errors correspond to estimates of the leading error component in the drift and diffusion terms.

E. Chemical Langevin equations

A subset of SDEs with multiple multiplicative noise terms is the chemical Langevin equations (CLEs). In what follows, we will briefly outline how CLEs emerge given a chemical kinetics model. Consider a well-mixed volume $V$,
containing $N$ distinct chemical species $S_i$ ($i=1,\ldots,N$) participating in $M$ chemical reactions. The state vector $X(t) = (X_1(t),\ldots,X_N(t))$ contains the time evolution of the system, i.e., the number of molecules of each species at a certain time. An $M \times N$ matrix $y$ is defined, containing all stoichiometric coefficients, where $v_{ji}$ is the change in the number of $S_i$ molecules caused by the $j$th reaction ($j=1,2,\ldots,M$). Reaction propensities $q_j(X(t))$ form an $M$-vector denoting the probabilistic rates of a reaction. In particular, $\alpha_j(X(t))dt$ gives the probability that the $j$th reaction occurs in a small time interval $[t,t+dt]$.

If the following conditions are met the system of reactions can be described as a continuous time Markov process governed by multidimensional Fokker-Plank equation.

(i) The reaction occurs many times in a small time interval.
(ii) The effect of each reaction on the numbers of reactants and products species is small, when compared to the total numbers of reactant and product species.

Or in equation form, respectively,

$$\alpha_j(X(t)) \geq \lambda \gg 1,$$

$$X(t) > \varepsilon \cdot |v_{ji}|,$$ (17)

where the $ith$ species is either a product or a reactant in the $j$th reaction.

The two parameters $\lambda$ and $\varepsilon$ define, respectively, the number of reactions occurring within time $\Delta t$ and what is the upper limit for the effect of a reaction to be negligible in the number of molecules of the reactants and products. This approximation becomes valid when both $\lambda$ and $\varepsilon$ become infinite, i.e., in the thermodynamic limit. In practice, typical values for $\lambda$ and $\varepsilon$ are 10 and 100, respectively.

The multidimensional Fokker-Plank equation describes the evolution of the probability distribution of the reactions. The solution is a distribution, not necessarily Gaussian, depicting the state occupancies. If the interest is in obtaining the evolution of the probability distribution of the reactions.

The solution is a distribution, not necessarily Gaussian, depicting the state occupancies. If the interest is in obtaining the evolution of the probability distribution of the reactions. The effect of each reaction on the numbers of reactants and products species is small, when compared to the total numbers of reactant and product species.

Or in equation form, respectively,

$$\alpha_j(X(t)) \geq \lambda \gg 1,$$

$$X(t) > \varepsilon \cdot |v_{ji}|,$$ (17)

where the $ith$ species is either a product or a reactant in the $j$th reaction.

The two parameters $\lambda$ and $\varepsilon$ define, respectively, the number of reactions occurring within time $\Delta t$ and what is the upper limit for the effect of a reaction to be negligible in the number of molecules of the reactants and products. This approximation becomes valid when both $\lambda$ and $\varepsilon$ become infinite, i.e., in the thermodynamic limit. In practice, typical values for $\lambda$ and $\varepsilon$ are 10 and 100, respectively.

The multidimensional Fokker-Plank equation describes the evolution of the probability distribution of the reactions. The solution is a distribution, not necessarily Gaussian, depicting the state occupancies. If the interest is in obtaining one of the possible trajectories of the solution, the proper course of action is to solve a system of CLEs. The CLE is an Itô stochastic differential equation with multiplicative noise and represents one possible solution of the Fokker-Plank equation. From a multidimensional Fokker-Plank equation we end up with a system of CLEs,

$$dX_i = \sum_{j=1}^{M} v_{ji} \alpha_j(X(t))dt + \sum_{j=1}^{M} v_{ji} \sqrt{\alpha_j(X(t))} dW_j,$$ (18)

where $\alpha_j, v_{ji}$ are the propensities and the stoichiometric coefficients, respectively, $M$ is the number of fast reactions, and $W$ is a Wiener Process with dimension $M$, producing the Gaussian white noise.

In order to validate the proposed algorithm we will need to compare our numerical solution with the “actual” solution. In general, for systems of SDEs with multiple multiplicative noise, like Eq. (18), analytical solutions do not exist. Therefore there is the need to find an alternative way to estimate the error. In the case of chemical kinetics models we are able to compute an accurate numerical solution using the fact that the original system is a discrete time Markov process governed by a chemical master equation. We use the stochastic simulation algorithm (SSA) to obtain the trajectories which we consider as accurate as the actual solution. Note that the trajectories produced by the system of CLEs are an approximation to the solution of SSA when the above mentioned conditions are met with the error being marginal.

As a final point we would like to discuss under what condition systems in the form of Eq. (18) are considered to be stiff. While in the deterministic case we can simply judge by looking into the eigenvalues of the system, in the stochastic regime there is no similar criterion. Nevertheless, one can judge whether or not the system is stiff by looking into the values of the reaction propensities. The larger the absolute ratio of the maximum propensity over the minimum propensity, the larger the stiffness of the system. A measure of stiffness $S_f$ can then be defined as

$$S_f = \frac{\max(q(t))}{\min(q(t))},$$ (19)

where $q(t)$ corresponds to the vector of reaction propensities.

F. Implementation details

In this section we apply the adaptive time stepping algorithm to the system of CLEs [Eq. (18)] and present a brief description of how the algorithm functions.

1. Milstein method

We start by rewriting the basic relations of the numerical integration scheme as they apply in the system of CLEs. Applying the Milstein method to the system of CLEs,

$$X_i^{k+1} = X_i^k + \sum_{j=1}^{M} v_{ji} \alpha_j(X^k) \Delta t + \sum_{j=1}^{M} v_{ji} \sqrt{\alpha_j(X^k)} I(j) + \frac{1}{2} \sum_{j=1}^{M} \sum_{j_2=1}^{M} v_{ji} v_{j_2} \alpha_j \alpha_{j_2} \frac{\partial \alpha_{j_2}}{\partial X_i} I(j,j_2),$$

$$i = 1,2,\ldots,N,$$ (20)

where $I(j), I(j_1,j_2)$ represent 1D and 2D Itô integrals, respectively.

2. Local error criteria

For the local error criteria, Eqs. (15) and (16), we have for the vector $f$ and matrix $g$, respectively,
which when substituted in the respective equations yield the local error criteria applicable to our case. In our algorithm we introduce two user defined parameters that are used to control the error tolerance in the adaptive scheme. The first variable called SDE tolerance represents the tolerance in the drift term is calculated by multiplying SDE coefficient with SDE tolerance. The higher the SDE coefficient value the less important the drift error becomes and vice versa. While the standard procedure is to use the same error tolerance for both error terms, we found that by using this extra weight coefficient the algorithm can be better tuned to produce both accurate and fast results. Therefore we choose to use a value of 1000 for the SDE coefficient.

\[ \Delta_{\text{mean}}(t) = \frac{|E[X_{i}^{\text{actual}}(t)] - E[X_{i}^{\text{numerical}}(t)]|}{E[X_{i}^{\text{actual}}(t)]}, \]

\[ \Delta_{\text{var}}(t) = \frac{|\text{var}[X_{i}^{\text{actual}}(t)] - \text{var}[X_{i}^{\text{numerical}}(t)]|}{\text{var}[X_{i}^{\text{actual}}(t)]}, \]

where \( i \) corresponds to the \( i \)th component of the state vector, \( E[X_{i}(t)] \) denotes the mean of \( X_{i}(t) \), and \( \text{var}[X_{i}(t)] \) stands for variance of \( X_{i}(t) \).

### III. EXAMPLES

The subject of this section is to validate the adaptive scheme. Three examples are used. The first is a chemical kinetics model that leads to a system of stiff linear CLEs, while the second one leads to a system of stiff nonlinear CLEs. In the final example the adaptive scheme is integrated into HY3S, a multiscale algorithm that uses CLEs to propagate the system of reactions that belong in the continuous Markov process regime. Results are compared with the actual solution obtained through SSA and with the simple Milstein scheme.

#### A. A reversible dimerization reaction

Consider the following reaction network, used to study the implicit and explicit tau-leap methods, after minor changes:

\[ S_1 \xrightarrow{c_1} S_2, \]

\[ S_2 \xrightarrow{c_2} S_1, \]

\[ S_1 \xrightarrow{c_3} S_1, \]

with kinetic parameters \( c_1 = 10^5 \text{ s}^{-1}, c_2 = 10^4 \text{ molecules s}^{-1}, \) and initial conditions \( S_1(0) = 10^3 \) and \( S_2(0) = 10^3 \) molecules. The initial propensities of the two reactions are

\[ \alpha_1 = c_1 \times S_1 = 10^8 \text{ molecules s}^{-1}, \]

\[ \alpha_2 = c_2 \times S_2 = 10^8 \text{ molecules s}^{-1}, \]

\[ \alpha_3 = c_3 = 9.998 \times 10^4 \text{ molecules s}^{-1}. \]

The system initially and during the course of simulation satisfies conditions (17), which means that its time evolution can be described through a system of CLEs. Moreover, fluctuations and the large values of the propensities, initially and during the course of simulation, are responsible for the mathematical stiffness arising in the system of CLEs.

In order to validate the adaptive scheme we use SSA to simulate the system on the time interval \([0, 0.01]\) s. We use the SSA realization available in HY3S. As we mentioned in Sec. II the trajectories generated by both the SSA and the proposed scheme sample the underlying distribution. In order to accurately sample the distribution we need a large sample of trajectories. Here we run 10,000 independent trials. The results obtained through SSA are presented in Fig. 3.
Figure 3(b) presents the probability distribution of the number of molecules of $S_1$ and $S_2$ at time $t=0.005$ s. The mean and variance [Figs. 3(c) and 3(d)] computed in this section will be used to validate the adaptive Milstein scheme in the following section.

In what follows we propagate the system [Eq. (23)] in time through a system of CLEs and compare the results with SSA so that we can determine the accuracy of the adaptive scheme. The system of CLEs is

$$dX_1 = [\nu_{11}\alpha_1(X(t)) + \nu_{21}\alpha_2(X(t)) + \nu_{31}\alpha_3(X(t))]dt + \nu_{11}\sqrt{\alpha_1(X(t))}dW_1 + \nu_{21}\sqrt{\alpha_2(X(t))}dW_2 + \nu_{31}\sqrt{\alpha_3(X(t))}dW_3,$$

$$dX_2 = [\nu_{12}\alpha_1(X(t)) + \nu_{22}\alpha_2(X(t))]dt + \nu_{12}\alpha_1(X(t))dW_1 + \nu_{22}\alpha_2(X(t))dW_2,$$

where $X_1$ corresponds to the state of species $S_1$, meaning the number of molecules of $S_1$, and $X_2$ corresponds to the state of species $S_2$, $\nu_{ij}$ is the stoichiometric coefficient of the $ij$th reaction, and $\alpha_i(X(t)) = c_iX_i(t)$, $\alpha_2(X(t)) = c_2X_2(t)$, and $\alpha_3(X(t)) = c_3$ are the reaction propensities.

We first examine how well the fixed step Milstein method captures the complex dynamics. The 2D Itô integral is approximated with a parameter $p$ set to 10. Its effect on the error will be discussed below. We run again 10 000 independent trials and calculate the solution in the interval of $[0, 0.01]$ s. For comparison with the actual solution (Fig. 3) we calculate the normalized weak mean and variance errors of $S_1$ using Eq. (22) and report on their average. The results are depicted in Fig. 4.

From Fig. 4 we observe that the fixed step method fails to produce solutions when the step size is larger than $5.0 \times 10^{-6}$ s. This happens because during the integration species concentrations attain negative values, which is physically unacceptable. Additionally, there is a slight decrease in
the normalized weak mean error as the time step decreases [Fig. 4(a)], but still the order of the error remains $O(10^{-4})$. On the other hand, the error in the variance decreases continuously as the time step decreases [Fig. 4(b)]. The variance error decreases roughly by two orders of magnitude. In summary, as the time step decreases significantly the fixed step method produces accurate solutions. By comparing the corresponding probability distributions we can infer that an acceptable convergence in distribution is observed when $t$ is equal to or less than $5.0 \times 10^{-7}$ s (figures are not presented for brevity).

Next we study the accuracy of the proposed adaptive scheme. First, we examine how the error generated by the scheme is affected by the user defined error tolerance (SDE tolerance). On the other hand SDE coefficient is kept constant at 1000 during the course of all simulations involving the adaptive scheme. Second, we examine the effect of parameter $p$ on the approximation of the 2D Itô integral and consequently in the error of the solution.

For analyzing the effect of SDE tolerance we again run 10 000 independent simulations and fix the initial time step to be $\Delta t_{ini} = 10^{-4}$ s and $p=10$ for varying SDE tolerance values, ranging from $10^{-2}$ to $10^{-6}$. Subsequently, we calculate the average normalized weak mean and variance errors of $S_1$ and $S_2$ and the probability distribution of $S_1$ at $t=0.005$ s (Fig. 5). It is evident from Figs. 5(a) and 5(b) that as we decrease the error tolerance the weak mean error decreases slightly, while the decrease is more extreme in the variance where the change is two orders of magnitude between SDE tolerances $10^{-2}$ and $10^{-6}$. The most affected quantity by the variation of SDE tolerance is the variance as was the case in Fig. 4. If we further compare Figs. 4, 5(a), and 5(b) we observe that for the fixed step Milstein method to reach the same accuracy as the adaptive step Milstein with SDE tol-
ance of $10^{-4}$ the needed step size is approximately $\Delta t=5.0 \times 10^{-7}$ s. Recall that all adaptive time step runs were initialized with $\Delta t_{\text{ini}}=10^{-4}$ s for which the fixed step method failed.

From the definition of the weak error we know that it quantifies the convergence in the probability distribution of the actual and numerical solutions. Figures 5(c) and 5(d) compare the distribution of the solution between SSA (actual) and the adaptive Milstein method with varying SDE tolerance. Results show that an acceptable convergence is observed for SDE tolerance values of $10^{-4}$ or less. Concluding, both the mean and variance errors have to be “small” in order to have convergence in distribution. This fact is not always obvious from figures similar to Figs. 5(a) and 5(b).

Now we turn our attention to the effect of parameter $p$ in the solution error. 10,000 independent trajectories with initial time step of $\Delta t_{\text{ini}}=10^{-4}$ s and SDE tolerance of $10^{-4}$ are used to obtain solutions with varying $p$. The values of $p$ we tested are 1, 10, 50, 100, and 1000. Recall that $p$ determines the truncation order of the Fourier expansion of the 2D Itô Integral. Average normalized weak mean and variance errors of $S_i$ for the different values of $p$ are plotted in Fig. 6. Figures 6(a) and 6(b) reveal that the value of $p$ does not have a significant effect on the weak mean and variance error for given initial $\Delta t$ and given SDE tolerance. The argument is further supported by examining the corresponding distributions which differ only slightly (figures not shown for brevity). This result was more or less expected since we examine the behavior of the weak error. Both the Milstein and the Euler-Maruyama methods are of weak order 1.0. On the other hand the Euler-Maruyama is of strong order 0.5, while the Milstein is of strong order 1.0. The inclusion of the 2D Itô integral is the reason why the latter method has a higher strong order. Hence, when we compute the weak error, the value of $p$ should not make any difference, which is indeed clear, looking at Fig. 6. The value of $p$ should play a role when computing the strong error.

Finally, since computational costs are also important we briefly present and compare the simulation times. Note that simulation times refer to the overall time frame of 0.01 s. All realizations were obtained using dual-core 2.6 GHz AMD Opteron processors. In Fig. 7 we display how the simulation time of each trial depends on the weak variance error introduced by either the fixed step or adaptive Milstein method. We only display results using the variance since both methods yield similar behavior and accuracy for the weak mean error and thus the error in the variance is more indicative.

For the variable step size method we chose $\Delta t_{\text{ini}}=10^{-4}$ s and $p=10$.

From Fig. 7 we note that as the desired accuracy increases the two schemes converge in execution time. In general, the fixed step method is slightly faster for a given error tolerance. We believe that this is directly correlated with the constant calculation of the two error criteria and the need to reapproximate the 2D Itô integral using a series of random number that add up to the execution time. In our opinion the latter is the main reason for the existing slow down. Obviously the anticipated speedup observed in the case of adaptive schemes in ODEs, where a substantial decrease in computational costs is the usual case, is not present. That was something we could have anticipated given the fact that adaptive methods are more intensive in the stochastic regime. However, the adaptive scheme manages to integrate the system regardless of the initial time step, adding stability to the integrator. Finally, for comparison purposes we note that SSA required approximately 2.5 s per trial to simulate the system [Eq. (23)]. This makes both the fixed and adaptive step methods faster than SSA except if we use time steps less than $5 \times 10^{-8}$ s for the fixed step method or SDE tolerance less than $10^{-5}$ for the adaptive method, which will not be necessary since an acceptable convergence is observed for larger values in both cases.

B. A system of stiff, nonlinear CLEs

As a second example we exploit a more elaborate one. Consider the following reaction network:

$$S_1 + S_2 \rightleftharpoons S_3,$$

$$S_1 + S_3 \rightleftharpoons S_2,$$

$$S_2 + S_3 \rightleftharpoons S_1,$$

(26)

with kinetic parameters $c_1=10^3$ molecules$^{-1}$ s$^{-1}$, $c_2=10^3$ s$^{-1}$, $c_3=10^{-5}$ molecules$^{-1}$ s$^{-1}$, $c_4=10$ s$^{-1}$, $c_5 = 1.0$ molecule$^{-1}$ s$^{-1}$, $c_6=10^6$ s$^{-1}$, and initial conditions $S_1(0)=10^3$ molecules, $S_2(0)=10^3$ molecules, and $S_3(0)=10^6$ molecules. During the length of the simulation interval ([0, 0.01] s) reactions (26) satisfy the conditions presented in Eq. (17), which means that the systems’ time evolution can be described through a system of CLEs. The corresponding system of CLEs is
where \( X_i \) corresponds to the state of \( i \)th species and the corresponding propensities are

\[
\alpha_1(X(t)) = c_1 X_1(t) X_2(t), \quad \alpha_2(X(t)) = c_2 X_2(t), \\
\alpha_3(X(t)) = c_2 X_1(t) X_3(t), \quad \alpha_4(X(t)) = c_4 X_4(t), \\
\alpha_5(X(t)) = c_3 X_3(t) X_5(t), \quad \alpha_6(X(t)) = c_6 X_6(t).
\]

(28)

By substituting the kinetic parameters and initial conditions in the above equations we note there is a five order separation in the reaction scales, denoted by the propensities values.

Again we use SSA as our actual solution. We simulate the system on the time interval of \([0, 0.01]\) s and conduct 10 000 independent trials. For brevity and clarity we only present the results for \( S_1 \) (Fig. 8). Similar behavior is observed for \( S_2 \) and \( S_3 \) where both their values fluctuate over time around their initial conditions. Finally, the mean and the variance evaluated through SSA are used for the comparison and the evaluation of the adaptive scheme.

Figure 9 examines how well the fixed step method integrates the system of CLEs. The 2D Itô integral is approximated with parameter \( p \) set to 10. We run again 10 000 independent trials and calculate the solution in the interval of \([0, 0.01]\) s. For comparison with the actual solution (Fig. 8) we calculate the average normalized weak mean and variance errors of all species using Eq. (22). The results are presented in Fig. 9. As in the previous example, the fixed step method fails to integrate the system for large time step values. In particular, we could not obtain solutions for step size larger than \(5.0 \times 10^{-7}\) s. This occurs because species concentrations attain negative values. Notice that the behavior of the scheme is similar to the one of the previous example, meaning that the error in the variance is the one mostly affected by the decrease in the time step [Fig. 9(b)]. While the normalized weak mean error remains for practical purposes the same [Fig. 9(a)]. A notable exception is the behavior of \( S_3 \) which appears to have a very small error in the mean. This can be explained if we take into the account the large in comparison with the other species initial and hence equilibrium values. Its large concentrations allow for a small influence by the noisy environment, meaning it is minimally affected by the evolution of the noise terms.

Subsequently we look into the accuracy of the proposed scheme. We examine the effect on the user defined parameter SDE tolerance in the accuracy of the solution. SDE coefficient is kept at 1000. Results are shown in Fig. 10, obtained from 10 000 independent trajectories and fixed initial time step of \( \Delta t_{\text{int}} = 10^{-4}\) s and \( p = 10 \). The values of SDE tolerance range from \(10^{-2}\) to \(10^{-3}\). We calculate the average normal-
ized weak mean and variance errors of all species and look into the probability distribution of species $S_1$ at $t=0.005$ s. Obviously as we decrease the error tolerance, the weak variance error decreases. The decrease is approximately one order of magnitude between SDE tolerances of $10^{-2}$ and $10^{-5}$ [Fig. 10(b)]. For the weak mean error we observe small if any decrease in its value as SDE tolerance decreases. It is noteworthy to point out that the adaptive scheme achieves to integrate the system even though it uses a high initial time step if compared with the fixed step counterpart. Finally, from Figs. 10(c) and 10(d) we infer that an acceptable convergence in distribution is feasible for values of SDE tolerance equal to or less than $10^{-4}$.

Figure 11 shows a different behavior from the one noted in Fig. 7. In this example both approaches seem to produce results with the same accuracy while approximately requiring the same computational effort. This means that there is no apparent advantage of the fixed step scheme over the adaptive. We believe that this is the case because from a two variable system in the first example we went to a three variable system which is also nonlinear that allowed the adaptive scheme to gain slightly over the fixed step method. Nonetheless, perhaps the important benefit is larger time steps while retaining stability. Finally, it is interesting to note that SSA required approximately 65.81 s per trial to simulate system [Eq. (26)]. Apparently, both methods are faster except if we use very small values for either the time step of the fixed method or SDE tolerance of the adaptive method.

C. An actual stiff biochemical network

The third example involves a larger chemical kinetics network, in particular, an actual biochemical network that

\[ \text{FIG. 9. (Color online) Solution of Eq. (27) using fixed step Milstein ($p=10$) method. (A) Comparison of the average normalized weak mean error of $S_1$, $S_2$, and $S_3$ generated by the fixed step Milstein method. (B) Comparison of the average normalized weak mean variance of $S_1$, $S_2$, and $S_3$ generated by the fixed step Milstein method.} \]

\[ \text{FIG. 10. (Color online) Solution of Eq. (27) using the adaptive scheme with variable SDE tolerance (SDE coeff=1000, $p=10$, $\Delta t_{ini}=10^{-4}$ s). (A) Average normalized weak mean error of $S_1$, $S_2$, and $S_3$ for different error tolerances. (B) Average normalized weak variance error of $S_1$, $S_2$, and $S_3$ for different error tolerances. (C) Comparison of the probability distribution of species $S_1$ between SSA and solutions with SDE tolerances of $10^{-2}$ and $10^{-3}$ at $t=0.005$ s. (D) Comparison of the probability distribution of species $S_1$ between SSA and solutions with SDE tolerances of $10^{-4}$ and $10^{-5}$ at $t=0.005$ s.} \]
experiences stiffness. The reason we choose to use this last example is to highlight the advantages of using the present algorithm in conjunction with our Hybrid multiscale algorithm, called HY3S (hybrid stochastic simulations for supercomputers) capable of fast and accurately simulate in time biochemical networks that are far from the thermodynamic limit. HY3S is based on a hybrid approach that separates the reactions into two subsets, fast/continuous and slow/discrete. The first are propagated in time using the chemical Langevin equation (CLE) and the later using differential “jump equations.” When the underlying biological network is stiff, the fixed step integration in HY3S fails, mainly because species populations become negative. Therefore we incorporated into HY3S our adaptive time step selection scheme in order to add stability and better error control in the time integration process.

As an example we use a previously studied stochastic Petri model proposed by Srivastava et al. that quantifies the heat shock response of *E. Coli.* The model involves 17 linear and nonlinear reactions with 14 participating species. The reactions with their corresponding kinetic parameters are presented in Table I, while the initial values of each species are depicted in Table II. The volume of the cell is considered to be $V=1.5 \times 10^{-15}$ l. All the kinetic and initial data are chosen in accordance with Ref. 35 except for the initial value for species DnaJ where we followed an approach similar to Ref. 11 in order for the system to be further from the equilibrium state. As it becomes apparent from Tables I and II not all reactions can be initially (and during the course of the simulation) classified as fast [see Eq. (17)], hence not all species are propagated using CLEs. Reactions (15)–(17) are the ones classified as fast throughout the course of the simulation. This is the shortcoming of using an actual biological example since in nature slow and fast reactions coexist, interact, and control the rates of each other. Still the existence of fast reactions will allow examining the behavior of the present algorithm in the context of HY3S.

### Table I. Reactions that quantify the heat shock response of *E. Coli* together with their kinetic parameters chosen according to Ref. 35.

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DNA $\cdot \sigma^{32}$ $\rightarrow$ mRNA $\cdot \sigma^{32}$</td>
<td>$1.4 \times 10^{-3}$ s$^{-1}$</td>
</tr>
<tr>
<td>2 mRNA $\cdot \sigma^{32}$ $\rightarrow$ mRNA $\cdot \sigma^{32}$ + mRNA $\cdot \sigma^{32}$</td>
<td>$0.07$ s$^{-1}$</td>
</tr>
<tr>
<td>3 mRNA $\cdot \sigma^{32}$ $\rightarrow$ δ</td>
<td>$1.4 \times 10^{-6}$ s$^{-1}$</td>
</tr>
<tr>
<td>4 $\sigma^{32}$ $\rightarrow$ RNA $\cdot \sigma^{32}$</td>
<td>$0.7$ s$^{-1}$</td>
</tr>
<tr>
<td>5 RNA $\cdot \sigma^{32}$ $\rightarrow$ $\sigma^{32}$</td>
<td>$0.13$ s$^{-1}$</td>
</tr>
<tr>
<td>6 DNA $\cdot$ DnaJ + RNA $\cdot \sigma^{32}$ $\rightarrow$ DnaJ + DNA $\cdot$ DnaJ + $\sigma^{32}$</td>
<td>$4.41 \times 10^{-6}$ M$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>7 DnaJ $\rightarrow$ δ</td>
<td>$6.4 \times 10^{-10}$ s$^{-1}$</td>
</tr>
<tr>
<td>8 $\sigma^{32}$ + DnaJ $\rightarrow$ $\sigma^{32}$ + DnaJ</td>
<td>$3.27 \times 10^{-3}$ M$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>9 $\sigma^{32}$ + DnaJ $\rightarrow$ $\sigma^{32}$ + DnaJ</td>
<td>$4.4 \times 10^{-4}$ s$^{-1}$</td>
</tr>
<tr>
<td>10 DNA $\cdot$ FtsH + RNA $\cdot \sigma^{32}$ $\rightarrow$ FtsH + DNA $\cdot$ FtsH + $\sigma^{32}$</td>
<td>$4.41 \times 10^{-6}$ M$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>11 FtsH $\rightarrow$ δ</td>
<td>$7.4 \times 10^{-11}$ s$^{-1}$</td>
</tr>
<tr>
<td>12 $\sigma^{32}$ + DnaJ + FtsH $\rightarrow$ DnaJ + FtsH</td>
<td>$1.28 \times 10^{-3}$ s$^{-1}$</td>
</tr>
<tr>
<td>13 DNA $\cdot$ GroEL + RNA $\cdot \sigma^{32}$ $\rightarrow$ GroEL + DNA $\cdot$ FtsH + $\sigma^{32}$</td>
<td>$5.69 \times 10^{5}$ M$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>14 GroEL $\rightarrow$ δ</td>
<td>$1.8 \times 10^{-8}$ s$^{-1}$</td>
</tr>
<tr>
<td>15 Protein $\rightarrow$ Unfolded Protein</td>
<td>$0.2$ s$^{-1}$</td>
</tr>
<tr>
<td>16 DnaJ + Unfolded Protein $\rightarrow$ DnaJ + Unfolded Protein</td>
<td>$9.7256 \times 10^{6}$ M$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>17 DnaJ + Unfolded Protein $\rightarrow$ Protein + DnaJ</td>
<td>$0.2$ s$^{-1}$</td>
</tr>
</tbody>
</table>

![FIG. 11. (Color online) Comparison of execution times between the adaptive and fixed step methods for a single trial. Average normalized weak variance error of $S_1$ as a function of the execution time. Both methods have $p$ set to 10 and the adaptive scheme uses an initial time step of $\Delta t_{ini} = 10^{-4}$ s.](http://jcp.aip.org/jcp/copyright.jsp)
As in the previous two examples we use SSA as our actual solution. We simulate the system on the time interval of \([0, 100]\) s and conduct 1000 independent trials. Again the mean and the variance evaluated through SSA are used for the comparison and the evaluation of the adaptive scheme. Note that we conduct ten times fewer trials than in the previous two examples. In order to study the effect of the algorithm on a bigger time frame, 100 s compared to 0.01 s, we had to reduce the number of total trials. Still the number of trials is enough to produce an accurate sampling of the underlying distribution. Our results will be reported based on two species, DnaJ and \(\sigma^{32}\). The first is chosen because it is involved in fast reactions and the second because it is only affected by slow reactions, hence we will be able to judge whether or not the algorithm introduces error in both the fast and slow subspaces. A sample trajectory for both species is shown in Fig. 12. Note in Fig. 12(a) the steep initial decrease in the concentration of DnaJ.

As in the previous two examples the fixed step Milstein method fails to integrate the system when the integration time step is larger than 10\(^{-4}\) s. This is shown in Fig. 13, where the average normalized weak mean and variance errors for different time steps are compared for the two species. All results were obtained through 1000 independent trials with the parameter \(p\) set at 10. Focusing on the results for DnaJ we note that as the time step decreases, the error in the mean remains practically unaltered, while the error in the variance decreases, a scenario seen in the previous two examples. On the other hand the error in both the mean and variance for \(\sigma^{32}\) remain the same as the time step changes. This implies that the time step used in the integration of the fast species does not affect the propagation of the slow species. The last observation is something we anticipated and also expected to see when instead of the fixed step we use the adaptive method.

By substituting the fixed step integrator with the adaptive one, the results report a similar performance but instead of the varying time step we vary the SDE tolerance. Figure 14 shows the results obtained from 1000 independent trials using a fixed initial time step of \(\Delta t_{ini}=0.1\) s, \(p=10\), and SDE coefficient=1000. We calculate the average normalized weak mean and variance errors of the two species of interest using values of SDE tolerance that range from 10\(^{-2}\) to 10\(^{-5}\). As in Fig. 13 the error in the mean is mainly unaffected for both DnaJ and \(\sigma^{32}\), while a decrease is only observed in the variance error for \(\sigma^{32}\) as SDE tolerance decreases. Importantly we note that first the adaptive scheme does not affect the propagation of the slow species, since \(\sigma^{32}\) mean and variance error are not affected. And second the adaptive algorithm manages to integrate the system starting from an initial time step of \(\Delta t_{ini}=0.1\) s. Comparing the probability distributions obtained through SSA and HY3S we report that an acceptable convergence is obtained for SDE tolerance equal or less to 10\(^{-4}\). Results are not shown for brevity.

Finally reporting on the execution times we note that the adaptive scheme requires more time to produce accurate re-
sults. Results shown in Fig. 15 resemble those in Fig. 7. We think this is because the complexity of the system of CLEs in this example, two linear and one nonlinear reactions, is closer to that of the first example and hence the same reasons apply. Still the adaptive scheme retained its stability by integrating the system even though the starting time step was large. As an additional comment we want to point out the use of the variance error in DnaJ to report the execution times, since it is the most indicative.

IV. DISCUSSION

We have demonstrated that the proposed adaptive scheme can produce accurate results when stiff systems of Itô SDEs with multiple multiplicative noise arise, such as the system of CLEs in Eq. (25) and (27). The SDE tolerance, a user defined parameter, can be used to tune the scheme and balance precision with simulation times. The use of the weight coefficient (SDE coefficient) allows balancing the importance between the drift and diffusion error controls. The higher the SDE coefficient value, the less important the drift error becomes and vice versa.

We also noted that the proposed scheme is more stable than its fixed step counterpart. While the fixed step method fails to produce results, because of numerical instabilities, the adaptive scheme is able to produce stable solutions even when the initial time step is large. This feature becomes important when trying to integrate systems of SDEs for which we do not know \textit{a priori} whether or not they are stiff or if they become during the course of integration. The adaptive scheme adds the necessary stability in the integration scheme allowing the method to avoid incorrect integration paths.

In terms of computational efficiency the developed adaptive time step method slightly underperforms compared to the fixed time step method. In the first and third examples the margin of the fixed step method was present but not significant, while in the second example, fixed and adaptive scheme produced the same accuracy in the same amount of computational time. This means that an adaptive time step scheme is considerably more stable than fixed step analogs with no excessive additional computational overhead. It is interesting to note that in all the examples the adaptive scheme is initialized with a time step that is far from the “optimum” time step of each SDE tolerance value. In general, the larger the distance between the two, the more computational intensity is necessary.

In conclusion, we developed an adaptive scheme which numerically solves stiff systems of SDEs with multiple multiplicative noise by appropriately adjusting the time step, dynamically decreasing the time step of the numerical integration when stiffness exists, but dynamically increasing it when the system is no longer stiff. While this procedure adds stability in the integration algorithm, it does not appear to have a significant negative impact on the speedup of the fixed Milstein method. This is important since for many applications the stability part of an algorithm is a crucial component of the modeling effort. The presented algorithm has been embedded successfully in a hybrid multiscale algorithm we have developed, called HY3S, used to propagate in time chemical kinetics models that are far from the thermodynamic limit.\textsuperscript{16} HY3S is available for download at http://hysss.sourceforge.net/.

ACKNOWLEDGMENTS

This work was supported by a grant from the National Science Foundation (BES-0425882). Computational support from the Minnesota Supercomputing Institute (MSI) is grate-
fully acknowledged. This work was also supported by the National Computational Science Alliance under TG-MCA04N033.

30 V. Sotiropoulos and Y. Kaznessis, BMC Systems Biology 1, 7 (2007).