Simulation Modelling Practice and Theory 18 (2010) 1397–1411

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Simulation of Stochastic Hybrid Systems using probabilistic boundary detection and adaptive time stepping

Derek D. Riley [a,*], Xenofon Koutsoukos [a], Kasandra Riley [b]

[a] ISIS/EECS, Vanderbilt University, Nashville, TN, United States
[b] HHMI, Yale University, New Haven, CT, United States

ARTICLE INFO

Article history:
Received 4 June 2009
Received in revised form 6 April 2010
Accepted 31 May 2010
Available online 8 June 2010

Keywords:
Modeling
Simulation
Stochastic
Adaptive time stepping

ABSTRACT

Modeling and simulation of biochemical systems are important tasks because they can provide insights into complicated systems where traditional experimentation is expensive or impossible. Stochastic Hybrid Systems (SHS) are an ideal modeling paradigm for biochemical systems because they combine continuous and discrete dynamics in a stochastic framework. In this work we develop an advanced simulation method for SHS that explicitly considers switching and reflective boundaries and uses probabilistic crossing detection methods to improve accuracy. We also develop an adaptive time stepping algorithm for SHS to improve efficiency. We present case studies for a water/electrolyte balance system in humans and a biodiesel production model. Simulation results are presented to demonstrate the accuracy and efficiency of the improved simulation techniques.

1. Introduction

Modeling and analysis of biochemical systems are important tasks because they can provide insights into complicated systems where traditional experimentation is difficult or costly. Further, accurately simulating the individual interactions between the components of a biochemical system can shed light on the function of the entire system. Biological systems are often mixtures of continuous and discrete processes, so modeling and analysis methods ideally should be able to capture both types of dynamics. Biochemical processes are also inherently probabilistic because of the uncertainty of molecular motion, so models that incorporate stochasticity can provide new, more realistic insight into the dynamics of the system.

Stochastic Hybrid Systems (SHS) provide an ideal, formal framework for modeling biochemical systems because they can be used to capture complex continuous, discrete, and stochastic dynamics [1]. SHS models can be used to analyze and design complex systems that operate in the presence of uncertainty and variability because they incorporate complex dynamics, uncertainty, and multiple modes of operations. However, simulation of SHS models is a challenging task due to the complexities of the dynamics.

Simulation is a powerful analysis technique, but accurate simulation methods must be utilized to ensure reliable results are generated. Trajectories are tested for boundary crossings at each time step, and when a crossing is detected, the step is reversed and the discrete transition is fired. Simulation of systems with both discrete and continuous dynamics is especially challenging because of the error introduced near switching and reflecting boundaries.

Efficient simulation of stochastic dynamical systems is challenging because the stochastic dynamics require more computational effort to compute accurate approximations than deterministic simulation methods. Further, adaptive time stepping methods for stochastic dynamics are not as efficient as deterministic methods and much more difficult to compute efficiently for high-dimensional systems.

[*] Corresponding author. Tel.: +1 608 495 0653.
E-mail addresses: derek.riley@vanderbilt.edu (D.D. Riley), xenofon.koutsoukos@vanderbilt.edu (X. Koutsoukos), kasandra.riley@yale.edu (K. Riley).

1569-190X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2010.05.021

In this work we develop an advanced simulation technique for SHS that employs probabilistic boundary crossing detection methods for absorbing and reflecting boundaries utilizing probabilistic methods. We present a fixed time stepping method as well as an adaptive time stepping method for SHS. The methods we present improve the accuracy and efficiency of traditional SHS simulation methods. We demonstrate the improvements using SHS case studies including a small water/electrolyte balance in humans and a large, complex biodiesel production system. We developed SHS models for both biochemical systems to demonstrate our improved simulation methods when encountering discrete switching and reflecting boundaries. We present the models in this work as well as simulation results to highlight our improved simulation methods.

The rest of this paper is organized as follows: Section 2 describes the related work, Section 3 describes Stochastic Hybrid Systems, Section 4 covers fixed step simulation of SHS including the absorbing and reflecting boundary crossing detection techniques, Section 5 presents adaptive time stepping for SHS, Section 6 presents the case study of the water balance system and experimental simulation results, Section 7 presents the biodiesel production model and experimental simulation results, and Section 8 concludes the work.

2. Related work

2.1. Biochemical modeling

A recently renewed interest in the field of biochemical system modeling has increased the quality and diversity of the models created. Biological protein regulatory networks have been modeled with hybrid systems using linear differential equations to describe the changes in protein concentrations and discrete switches to activate or deactivate the continuous dynamics based on protein thresholds [2]. Biomolecular network modeling using hybrid systems is accomplished by using differential equations to model feedback mechanisms and discrete switches to model changes in the underlying dynamics [3]. A modeling technique that uses polynomial SHS to construct models for chemical reactions is presented in [4]. A SHS model of a genetic regulatory network is compared to a deterministic model in [5]. Switching thresholds for piecewise-affine models of genetic regulatory networks are studied in [6]. SHS models of biochemical systems using reaction rate analysis have been developed and simulated in [7]. An early stochastic model of the water/electrolyte balance system is presented in [8], and a non-stochastic model with experimental results is presented in [9].

2.2. Simulation of SHS

Simulation methods for SHS have been developed for the modeling language Charon, but the focus is on concurrency, and the behavior close to the boundaries is not studied [10]. A simulation engine for SHS is also implemented in Matlab using Simulink and Stateflow [11].

A technique for accurately detecting absorbing boundary crossing has been developed for one-dimensional systems [12], and extensions have been proposed that scale to higher-dimensional systems [13]. The boundary crossing detection algorithm presented in [14] uses analysis of moments to improve the accuracy of the approximation. Methods for approximating reflecting boundaries have also been studied previously [15], and an improved technique for approximating reflecting boundaries is presented in [16].

Adaptive time stepping has been used with great success for deterministic systems to improve accuracy and efficiency of simulators. Adaptive time stepping for stochastic differential equations (SDEs) is more difficult because the Brownian motion must be preserved, and high order methods must be used to guarantee the convergence of the solution [17]. Adaptive time stepping implementations for SDEs typically utilize halving and doubling of the time step based on approximations of the approximation error in the drift or diffusion terms [18]. An adaptive, parallel simulation method for stochastic systems is presented in [19], but adaptive simulation methods for SHS, to our knowledge, have not been presented.

Improved absorbing boundary crossing detection methods for SHS are presented in [20], and in this work we extend the previous work by adding an improved probabilistic reflecting boundary detection method. We also present an implementation of adaptive time stepping for SHS that incorporates methods for SDEs in a hybrid state space in this work.

3. Stochastic hybrid systems

We adopt the SHS model presented in [1]. To establish the notation, we let Q be a set of discrete states. For each $q \in Q$, we consider the Euclidean space $\mathbb{R}^{d(q)}$ with dimension $d(q)$ and we define an *invariant* as an open set $X^q \subseteq \mathbb{R}^{d(q)}$. The hybrid state space is denoted as $S = \bigcup_{q \in Q}\{q\} \times X^q$. Let $\overline{S} = S \cup \partial S$ and $\partial S = \bigcup_{q \in Q}\{q\} \times \partial X^q$ denote the completion and the boundary of S, respectively. The Borel σ-field in S is denoted as $\mathscr{B}(S)$.

Consider an $\mathbb{R}^p$-valued Wiener process $w(t)$ and a sequence of *stopping times* $\{t_0 = 0, t_1, t_2, \ldots\}$. Let the state at time t_i be $s(t_i) = (q(t_i), x(t_i))$ with $x(t_i) \in X^{q(t_i)}$. While the continuous state stays in $X^{q(t_i)}$, $x(t)$ evolves according to the stochastic differential equation (SDE)

$$dx = b(q, x)dt + \sigma(q, x)dw \tag{1}$$

where the discrete state $q(t) = q(t_i)$ remains constant. A sample path of the stochastic process is denoted by $x_t(\omega)$, $t > t_i$, $\omega \in \Omega$.

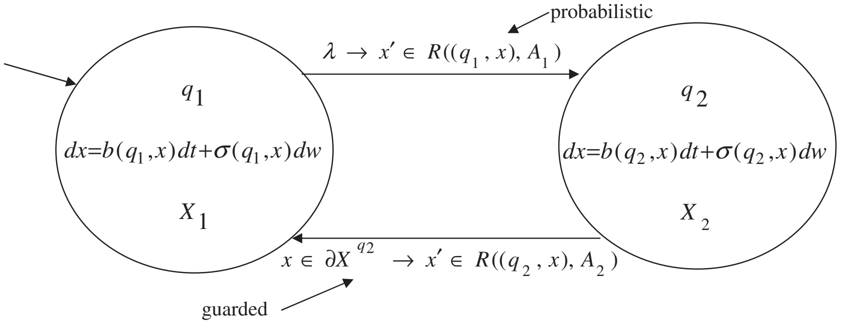
probabilistic
$\lambda \rightarrow x' \in R((q_1, x), A_1)$
q_1
$dx = b(q_1, x)dt + \sigma(q_1, x)dw$
X_1
q_2
$dx = b(q_2, x)dt + \sigma(q_2, x)dw$
X_2
$x \in \partial X^{q2} \rightarrow x' \in R((q_2, x), A_2)$
guarded

Fig. 1. Stochastic hybrid system.

The next stopping time t_{i+1} represents the time when the system transitions to a new discrete state. The discrete transition occurs either because the continuous state x exits the invariant $X^{q(t_i)}$ of the discrete state $q(t_i)$ (guarded transition) or based on an exponential distribution with non-negative transition rate function $\lambda : \overline{S} \rightarrow \mathbb{R}_+$ (probabilistic transition). At time t_{i+1} the system will transition to a new discrete state and the continuous state may jump according to the transition measure $R : \overline{S} \times \mathscr{B}(\overline{S}) \rightarrow [0, 1]$. The evolution of the system is then governed by the SDE (1) with $q(t) = q(t_{i+1})$ until the next stopping time. If $t_{i+1} = \infty$, the system continues to evolve according to (1) with $q(t) = q(t_i)$.

Fig. 1 shows a generic SHS model with two states and two transitions (one probabilistic and one guarded). The continuous dynamics of each state are defined by the associated stochastic differential equations. The probabilistic transition fires at the firing rate λ, and the guarded transition fires when x hits the boundary $x \in \partial X^{q_2}$. The logical condition $x \in \partial X^{q_2}$ is often referred to as the guard of the transition. Upon firing of a transition, the state resets according to the map $R((q,x), A)$. The following assumptions are imposed on the model. The functions $b(q,x)$ and $\sigma(q,x)$ are bounded and Lipschitz continuous in x for every q, and thus the SDE (1) has a unique solution for every q.

4. Fixed step SHS simulation

4.1. Fixed time step simulation

Simulation of SHS requires the combination of simulation methods for SDEs, detection of switching boundaries, approximation of reflecting boundaries, and detection of probabilistic transitions. At each time step, the values of the continuous variables must be updated, boundaries must be tested for crossings, and probabilistic transitions must be tested for firings. We present the individual methods and we describe their combination to create SHS simulation algorithms.

4.1.1. Numerical integration of SDEs

Simulation of SDEs can be performed using Taylor schemes of various orders. The simplest Taylor approximation scheme is the Euler–Maruyama (EM) method that is a first-order approximation. The kth component of the EM scheme is given by

$$X_{n+1}^k = X_n^k + b^k \Delta t + \sum_{j=1}^{m} \sigma^{k,j} \Delta W^j$$

for $k = 1, 2, \ldots, d$ where ΔW^j is the normally-distributed increment of the jth component of the d-dimensional Wiener process W assuming a d-dimensional drift coefficient b and a $d \times m$ diffusion coefficient σ.

Order of convergence is used to formalize the notion of accuracy. The order of convergence quantifies the quality of the approximation when considering simulation of stochastic systems. An approximation $X^{\Delta t}(T)$ at time T with step size Δt converges with order γ strongly to the actual trajectory $x(T)$ if there exists $c > 0$ such that $E(|x(T) - X^{\Delta t}(T)|) \leqslant c\Delta t^{\gamma}$. $X^{\Delta t}(T)$ converges with order γ weakly to $x(T)$ if there exists $c > 0$ such that $E(|f(x(T)) - f(X^{\Delta t}(T))|) \leqslant c\Delta t^{\gamma}$ for a given class of measurable functions f [21]. Strong convergence implies that the trajectory is a possible trajectory of the system, and weak convergence implies that the computed trajectory only preserves the moments of the actual trajectory. The EM method is simple to implement, but achieves a strong convergence of $\gamma = 0.5$ and weak convergence $\gamma = 1.0$, so small time steps must be used to generate accurate approximations.

The Milstein Method (MM) is a second-order Taylor scheme. The higher-order terms require additional computation; however, the approximation maintains acceptable efficiency for most systems. The kth component of the MM scheme is described by

$$X_{n+1}^k = X_n^k + b^k \Delta t + \sum_{j=1}^{m} \sigma^{k,j} \Delta W^j + \sum_{j_1, j_2=1}^{m} L^{j_1} \sigma^{k,j_2} I_{(j_1, j_2)}$$

where

$$L^j = \sum_{k=1}^{d} \sigma^{k,j} \frac{d}{dx^k} \quad \text{and} \quad I_{(j_1,j_2)} = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau_n}^{s_1} dw^{j_1}_{s_2} dw^{j_2}_{s}$$

For the cases where $j_1 = j_2$, the multiple stochastic (Stratonovich) integral can be calculated by

$$I_{(j_1,j_1)} = \frac{1}{2}((\Delta W^{j_1})^2 - \Delta t)$$

$I_{(j_1,j_2)}$ cannot, in general, be calculated using only ΔW^j values. To approximate $I_{(j_1,j_2)}$, multiple stochastic integrals are used in the following equation assuming $j_1 \neq j_2$.

$$I^p_{(j_1,j_2)} = \Delta t\left(\frac{1}{2}\xi_{j_1}\xi_{j_2} + \sqrt{p_p}(\mu_{j_1,p}\xi_{j_2} - \mu_{j_2,p}\xi_{j_1})\right) + \frac{\Delta t}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\left(\zeta_{j_1,r}\left(\sqrt{2}\xi_{j_2} + \eta_{j_2,r}\right) - \zeta_{j_2,r}\left(\sqrt{2}\xi_{j_1} + \eta_{j_1,r}\right)\right)$$

where

$$p_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2}$$

$$\xi_j = \frac{1}{\sqrt{\Delta t}}\Delta W^j$$

and $\mu_{j,p}$, $\eta_{j,r}$, and $\zeta_{j,r}$ are independent Gaussian random variables with mean 0 and standard deviation 1 for $j = 1, \ldots, m$ and $r = 1, \ldots, p$. The accuracy of the approximation $I^p_{(j_1,j_2)}$ of $I_{(j_1,j_2)}$ can be improved by using larger values of p. To obtain a strong convergence of order $\gamma = 1.0, p = p(\Delta t) \geqslant \frac{K}{\Delta t}$ must be chosen where K is some positive constant [21].

Taylor schemes for solving SDEs can have strong order of convergence of $\gamma = 0.5$ to $\gamma = 3.0$ and weak order of convergence of $\gamma = 1.0$ to $\gamma = 6.0$ depending on the number of approximating terms [21]. The computation of higher-order terms requires many more operations and can be prohibitively complicated and expensive; therefore, a trade off must be reached to achieve the appropriate accuracy and efficiency.

4.1.2. Absorbing boundaries

During the execution of a SHS, the process can hit a switching boundary defined by the invariants. At a switching boundary the continuous process is halted and re-started in a new state after executing any transition resets. Because the process is stopped when a boundary is encountered, switching boundaries can be treated as absorbing boundaries [16]. It is important to accurately estimate the time and location that the process is absorbed to minimize the error introduced into the approximation.

The easiest way to detect an absorbing boundary is to check the state against the invariants at each step of the approximation. Let us assume the state at time t is $X(t)$. If $X(t) \in X^q$, but $X(t + \Delta t) \notin X^q$, then the process is rolled back to time t and restarted in the new state. This method has a strong order of convergence of $\gamma = 0.5$ [14].

An improved method for absorbing boundary crossing detection based on stochastic sampling was developed in [16]. The approach can be used with boundaries that are hyperplanes or sufficiently smooth. The biochemical models we consider have boundaries that are hyperplanes, so this approach is valid for these systems. The probability that the state trajectory has hit the boundary between t and $t + \Delta t$ is

$$P(hit) = exp\left(\frac{-2(n \cdot (X_t - X_{ab}, Q_t))(n \cdot (X_{t+\Delta t} - X_{ab}, Q_t))}{n \cdot (\sigma(X_t, Q_t)\sigma^*(X_t, Q_t)n)\Delta t}\right)$$

where the switching boundaries are hyperplanes $\partial X^q = \{x \in \mathbb{R}^{d(q)} : n.(x - X_{ab}) = 0\}, n$ is the unit vector normal to the boundary $\partial X^q, X_{ab} \in \mathbb{R}^n$ is the position of the absorbing boundary, X_t is the computed continuous state at time t, and Q_t is the discrete state at time t. This improved method achieves a weak order of $\gamma = 1.0$ assuming that the boundary is sufficiently smooth [16].

4.1.3. Reflecting boundaries

Invariants can define reflective boundaries in addition to switching boundaries. Reflective boundaries are those where the process is reflected obliquely when it encounters the boundary [16]. For example, all biochemical systems are limited to non-negative concentrations of any chemicals, or some biochemical processes also have saturation limits that impose upper limits on concentrations. In both cases when the process reaches the boundary, it is reflected to mimic the behavior of the real system.

The traditional way to handle reflective boundaries is to detect the boundary crossing and reset the state to a position within the valid state space as shown in Fig. 2 [16]. The most common way to reset the state is to place it inside the state space the same distance that it covered after it crossed the boundary. This type of reflection guarantees that the invariants are always satisfied, but it is not always accurate for real systems. For example, biochemical systems require non-negative chemical concentrations, so if the simulation is reflected using the traditional method, the number of atoms involved in the

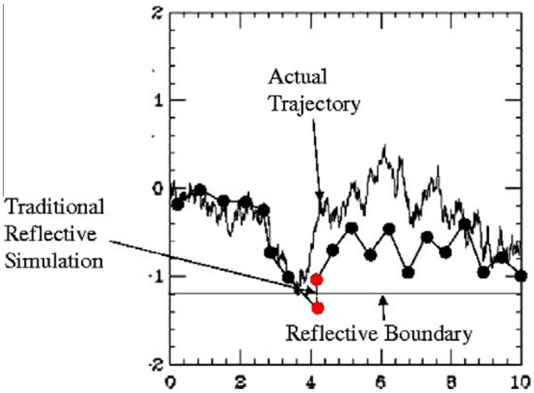

Fig. 2. Boundary reflection problem.

system can be changed significantly invalidating conservation of mass properties. While the use of SDEs inherently does not preserve conservation of mass in simulation (due to the noise terms), improved reflection techniques exist that better preserve conservation of mass by more accurately approximating the reflection.

We formally define the boundary reflection problem. Let us assume a system has an invariant X^q with a reflective boundary, and the state at time t is $X(t)$. If $X(t) \in X^q$, but $X(t + \Delta t)$ is computed to be outside of X^q, then the process is reflected in the direction normal to the boundary $X(t + \Delta t) \cdot n = X(t) \cdot n$, where n is the unit vector normal to the boundary. The traditional method of boundary reflection has a weak order of convergence of $\gamma = 0.5$ [16].

The improved method described in [16] defines a new diffusion process that adds the effect of the reflection to the original SDE to provide an exact simulation of obliquely-reflected processes:

$$dx = b(q,x)dt + \sigma(q,x)dw + n(q,x)dk$$

where $n(q,x)$ is a unit vector normal the boundary at state (q,x), $k = \int_0^t 1_{X \in \partial D}\, dk$ and ∂D is the reflective boundary.

The approximation of the process using the EM method for simplicity is calculated using:

$$X_{t+\Delta t} = X_t + b\Delta t + \sigma \Delta W + n\Delta k$$

where ΔW is a normally-distributed pseudo-random number and $\Delta k = k_t - k_{t+\Delta t}$. Approximating k_t is achieved using the technique described in [16]:

$$k_t = \max(0, z_t) \cdot n$$
$$z_t = X_t - X_{rb} + \frac{1}{2}\left(\sigma W + bt + \sqrt{|\sigma|^2 V + (\sigma W + bt)^2}\right)$$

where $V = \varepsilon(1/2t)$ is an exponentially-distributed random variable independent of W and X_{rb} is the position of the reflective boundary. This equation is extended to an order 1.0 method by including second-order term from the Milstein Method. It is derived from the solution to the Skorohod problem and results in a weak order 1.0 approximation of the reflecting boundary [16]. While simulation of conservation of mass is not possible for any technique based on SDEs, the improved accuracy of this technique ensures that the approximation closely follows the behavior of the real system, and therefore, more closely approximates the conservation of mass properties.

4.1.4. Probabilistic transitions

Firing of the probabilistic transitions (according to the transition rate λ) can be handled by the technique described in [10]. A graphical representation of this algorithm can be seen in Fig. 3. First, a new process Z must be defined

$$Z(t) = -U + e^{-\int_t^{t_{lt}} \lambda(x(s))ds}$$

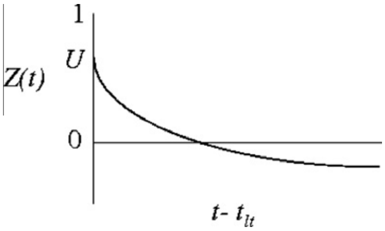

Fig. 3. Probabilistic transition firing method.

where $U \in [0,1]$ is a uniformly-distributed random number and t_{lt} is the time of the last probabilistic transition. A sample is drawn from the uniform distribution and Z is tested at each time step. When Z crosses 0, the transition is fired.

4.1.5. SHS simulation algorithms

Our first algorithm is the Hybrid Euler–Maruyama *HEM* method, which we will use as a baseline algorithm as presented in [22]. We use the Euler–Maruyama method for numerically integrating the SDEs, and we use the traditional (non-stochastic) methods for detecting boundaries. Discrete transitions are incorporated into the EM approximations by analyzing the state between steps of updating the continuous dynamics. If a guard condition on a transition is satisfied, then the transition is fired and resets are executed. Once the state is updated, the EM algorithm continues in the new state. Probabilistic transition firing is determined for *HEM* using the technique described in Section 4.1.4. We draw a sample from a uniform distribution and test the exponential decay at various times to determine the jump time for each probabilistic transition. When the exponential decay is greater than the random value, the transition is fired. The pseudocode for a step of *HEM* is given below where Δt is the step size, *guard* is the boolean guard on a single discrete transition, X_{rb} is the location of the reflective boundary, and t_{tl} is the time the last probabilistic transition fired. Multiple guarded or probabilistic transitions may be included by adding multiple tests.

Algorithm 4.1. HEMSTEP(X_t)

```
$X_{t+\Delta t} = X_t + b\Delta t + \sigma\Delta W$
if $guard == true$
   then FireGuardedTransition
if $X_{t+\Delta t} < X_{rb}$
   then ReflectBoundary
if $U_1 = rand(0,1) < exp(-\lambda(t - t_{tl}))$
   then FireProbabilisticTransition
$t = t + \Delta t$
return $(X_t)$
```

Our second algorithm *HMM* incorporates the MM, stochastic absorbing, and reflective boundary simulation methods to create a new simulation algorithm. To include the improved reflective boundary method we store the previous Δk value and calculate the new Δk at each step. If the trajectory is close to a reflecting boundary, we add Δk to the MM computation. Because there may be multiple absorbing boundaries that could be hit at the same time, at each time step we calculate the probability of hitting all nearby boundaries. We then select the boundary with the highest hitting probability and compare the probability to a uniformly-distributed number U_1. When $U_1 < max(P)$, then we consider the boundary to be hit, and we execute the transition resets and restart the process in the new state. The pseudocode for a step of *HMM* is as follows where $x_{ab,\xi}$ is the location of absorbing boundary ξ, n is the direction normal to the associated reflected or absorbing boundary, and V is an exponentially-distributed random variable independent of U_1.

Algorithm 4.2. HMMSTEP(X_t)

```
$k_{t+\Delta t} = X_t - X_{rb} + \frac{1}{2}\left(\sigma W + b(t+\Delta t) + \sqrt{|\sigma|^2 V + (\sigma W + b(t+\Delta t))^2}\right)$
$\Delta k = \max(k_{t+\Delta t},0) \cdot n - prev\Delta k$
$X_{t+\Delta t} = X_t + b\Delta t + \sigma\Delta W + \sum_{j_1,j_2=1}^{m} L^{j_1}\sigma^{k,j_2} I_{(j_1,j_2)} + n(\Delta k)$
$GuardedProb = max\left(exp\left(\frac{-2(n\cdot(X_t - X_{ab,\xi}))(n\cdot(X_{t+\Delta t} - X_{ab,\xi}))}{n.(\sigma\sigma^*(X_t)n)\Delta t}\right), \xi\right)$
if $U_1 = rand(0,1) < GuardedProb$
   then FireGuardedTransition
if $U_2 = rand(0,1) < exp(-\lambda(t - t_{tl}))$
   then FireProbabilisticTransition
$t = t + \Delta t$
return $(X_t)$
```

Error is introduced into the calculated SHS trajectory in several different ways. Approximation of the SDE introduces higher-order errors that are not calculated due to computational inefficiency. Error due to the use of pseudo-random numbers is typically not a concern for smaller simulations, but large simulations must use pseudo-random generators that do not repeat as often as the efficient generators to avoid this type of error. Finally, step size inherently introduces error in the SDE and boundary calculations as described earlier by the order of convergence γ.

The approximations using the EM or MM method, boundary methods, and probabilistic transitions converge to the actual solution individually as the step size is decreased to zero, so their combination also converges to the correct solution. By combining methods with higher-order convergence, we reduce approximation error more quickly than the lower order methods thereby improving efficiency and accuracy. The traditional absorbing and reflecting boundary algorithms have a weak order of convergence of $\gamma = 0.5$, while the improved methods both have a weak order of convergence of $\gamma = 1.0$ [16]. Therefore the *HEM* algorithm has a strong order of convergence of $\gamma = 0.5$, and the *HMM* algorithm has a strong order of convergence of $\gamma = 1.0$.

Accurate simulation of the trajectory near intersections of boundaries is a difficult problem, and must be handled carefully to minimize error. When the trajectory is in close proximity to multiple reflecting or absorbing boundaries, our algorithm considers the boundary with the highest hitting probability at each time step.

5. Adaptive time stepping for SHS

Error in a simulation method due to the step size can be decreased by decreasing the step size, but this comes at the cost of efficiency. Dynamically adjusting the time step of the simulation has been shown to increase the accuracy and efficiency of the approximation by allowing the step size to adjust to compensate for variable step size error. However, adaptive time stepping for stochastic systems is difficult because of the challenge of error approximation in the presence of stochastic dynamics [18]. Adaptive time stepping for SHS is further complicated by the discrete discontinuities, so additional care must be taken when a simulation trajectory approaches a boundary.

5.1. Background

Fixed step integration methods are easy to implement and are effective for generating approximations to differential equations, but they can be unnecessarily inefficient. Adaptive time stepping can improve efficiency by adjusting the time step of the approximation dynamically based on the error of the approximation.

Exact error cannot be determined for general systems, so error estimations must be used. Error estimation methods aim to determine the amount of error generated in a time step by examining the dynamics of the simulation. If the estimated error is too large, then the given time step should be decreased. Conversely, if the error is sufficiently small, the step size can be increased because the error introduced will be relatively small.

Accurate approximations of the error due to the step size must be made to ensure the step size is adjusted appropriately. For ordinary differential equations, error approximations and step size adjustments are fairly straight forward [18]. However, adaptive time stepping for SDEs is not as simple for multiple reasons. Not only is error introduced by several sources (that all must be accurately estimated), but also the Brownian path must be computed accurately when the step size changes to ensure randomness is preserved. Therefore, we begin by examining error estimates for SDEs.

5.2. SDE error approximation

Time discretization error for SDEs can be categorized into two types: drift and diffusion error. Neither type of error can be computed exactly because there are no analytical methods for computing the error of SDEs. However, both types of error can be estimated separately and decisions about the time step can be made based on the amount of either or both forms of error.

The error introduced by the diffusion term can be estimated by computing higher-order approximation terms. Given a SDE, we examine the higher-order terms of the Strantonovich–Taylor expansion of the original SDE $J_{10}b'\sigma, J_{01}\sigma'b, \frac{1}{6}J_1^3\sigma'\sigma\sigma$, and $\frac{1}{6}J_1^3\sigma'\sigma'\sigma$ where J_1, J_{10}, and J_{01} are multiple Strantonovich integrals [18]. Only the last term can be computed efficiently, so it is the best term to estimate the diffusion-influenced error:

$$E_\sigma = \Delta W^3 \sigma'^2 \sigma$$

where W^3 is an m-dimensional vector (corresponding to the number of Weiner processes) of cubed Gaussian terms. This method has been shown to be effective for estimating the diffusion error for systems with one noise dimension ($m = 1$) in [18]. The technique was extended to handle multiple multiplicative noise terms ($m > 1$) in [23] by computing a replacement for σ in the above equation based on a combination of the multiplicative noise terms. Our implementation uses the approach in [23] for multidimensional Wiener processes.

Error can also be introduced by the drift term, so we must also consider this error in our estimation methods. ODE-like error computation methods can be used to estimate the drift error using the SDE. Using the $O(h^2)$ error terms from the Milstein error expansion, the drift error can be estimated by

$$E_b = \Delta t^2 b' b$$

This method has been shown to be effective for estimating the drift error previously in [18].

5.3. Adaptive time stepping for SDEs

Before each step of the approximation, the drift and diffusion errors of the approximation are computed (E_b, E_σ), and the step is rejected or accepted depending on the amount of either type of error. If the step is rejected because either E_b or E_σ is too large, the step size is reduced to decrease the error until both error estimates are sufficiently small. If the two types of error are both determined to be smaller than a threshold, the step size can be increased to improve efficiency. Step sizes are typically halved and doubled in stochastic systems to simplify the computation of the Wiener process [18].

Brownian motion must be appropriately approximated for the variable time steps to ensure bias is not introduced. To simplify the process, a binary tree structure is used to store the noise values. The Wiener process is sampled at fixed intervals $\Delta w_k = w(k) - w(k-1)$ for $k = 1, \ldots, N$. Intermediate intervals on level j of the tree are calculated if needed by $\Delta w_{2k-1,j+1} = \frac{1}{2}\Delta w_{k,j} + y_{k,j}, \Delta w_{2k,j+1} = \frac{1}{2}\Delta w_{k,j} - y_{k,j}$ for $j = 1, 2, \ldots$ where $y_{k,j}$ is a normally-distributed random variable with mean 0 and variance 2^{-j} [17].

5.4. Adaptive time stepping for SHS

The dynamics of SHS introduce further complication into the estimation of the error for an approximation method. Significant error can be introduced for near boundaries as we showed in the previous section, and larger step sizes exacerbate this error. Traditional error estimation methods for SDEs do not consider boundaries, so large step sizes near boundaries are possible. Therefore, we must test and prevent large step sizes near reflecting or absorbing boundaries before a step size is accepted.

Error introduced near boundaries can be significantly reduced by shrinking the step size when a trajectory is near a boundary. Therefore, we test if $X_t - X_{ab} < \Psi$ or $X_t - X_{rb} < \Psi$ where Ψ is a minimum boundary distance threshold, and we decrease the step size to a small, predetermined value Δt_{min} if either condition is satisfied to ensure the most accurate boundary approximation. When the conditions are not satisfied, we allow the variable step algorithm to adjust the step size according to the traditional SDE adaptive algorithm. This method ensures that the adaptive time stepping methods do not increase the error from the boundary approximations in SHS.

5.5. Adaptive time stepping simulation algorithm

Adaptive time stepping extends the fixed step method by computing error estimates and adjusting the step size before a step is computed. We introduce a new algorithm *ATHMM* that incorporates adaptive time stepping into the *HMM* algorithm shown below. The algorithm tests both error estimates E_b and E_σ, and if either is above an upper threshold, the step size is cut in half. If either error estimate is below a lower threshold, the step size is doubled. If the error is between the two thresholds, then the step size is not adjusted. To avoid large step sizes near boundaries, the algorithm tests the distance to any reflecting or absorbing boundaries and changes the step size to Δt_{min} if it is sufficiently close to a boundary.

Algorithm 5.1. ATHMMSTEP(X_t)

```
while $\Delta t^2 b'b + J_1^3 \sigma'^2 \sigma > UpperThreshold$
  do $\Delta t = \frac{\Delta t}{2}$
while $\Delta t^2 b'b + J_1^3 \sigma'^2 \sigma < LowerThreshold$
  do $\Delta t = 2\Delta t$
if $|X_t - X_{ab,\xi}| < 1$ or $|X_t - X_{rb}| < 1$
  then $\Delta t = \Delta t_{min}$
$k_{t+\Delta t} = X_t - X_{rb} + \frac{1}{2}\left(\sigma W + b(t+\Delta t) + \sqrt{|\sigma|^2 V + (\sigma W + b(t+\Delta t))^2}\right)$
$\Delta k = \max(k_{t+\Delta t}, 0) \cdot n - prev\Delta k$
$X_{t+\Delta t} = X_t + b\Delta t + \sigma \Delta W + \sum_{j_1,j_2=1}^{m} L^{j_1} \sigma^{k,j_2} I_{(j_1,j_2)} + n(\Delta k)$
$GuardedProb = max\left(exp\left(\frac{-2(n \cdot (X_t - X_{ab,\xi}))(n \cdot (X_{t+\Delta t} - X_{ab,\xi}))}{n \cdot (\sigma\sigma^*(X_t)n)\Delta t}\right), \xi\right)$
if $U_1 = rand(0,1) < GuardedProb$
  then $FireGuardedTransition$
if $U_2 = rand(0,1) < exp(-\lambda(t - t_{TimeOfLastFire}))$
  then $FireProbabilisticTransition$
$t = t + \Delta t$
return ($X_t$)
```

6. Case study: water balance

We present the water/electrolyte balance system in humans as a small, realistic biochemical model. The relative simplicity of the model provides a platform for demonstration and comparison of the simulation methods in terms of accuracy and scalability.

6.1. Background

Water/electrolyte balance regulation in mammals is vital to life. If too much salt is present, dehydration occurs, leading to discomfort, performance degradation, and even death. If too much water is present, arterial pressure rises dangerously and the nervous system begins to malfunction. Therefore, virtually every living organism has a system that regulates water balance. In humans, this system includes blood pressure sensors, the kidneys, the hypothalamus, and other minor organs.

Anti-diuretic hormone (ADH) is a nine amino acid peptide hormone secreted by the hypothalamus. ADH is released when the body senses the intake of too much salt or a shortage of water. Upon these conditions ADH signals to the kidneys to retain water to compensate and bring the body back to equilibrium. Upon secretion, ADH travels through the bloodstream to exert the majority of its effects on specific receptors (arginine vasopressin receptor 2; AVPR2) in specialized cells within the kidney tubules. When ADH binds AVPR2, a chain of intracellular signaling events take place. The succession of signaling events ultimately results in additional insertion of extra water channels (aquaporins; AQP2) into the apical membrane of the cell. Aquaporins allow water to pass out of the nephrons and be re-collected into the cells. Once the water is reabsorbed, a smaller, more concentrated amount of urine is excreted.

The insertion of AQP2 channels into the cell's outer membrane is a highly regulated, multistep process. AQP2 is synthesized in the cell and inserted into intracellular membrane structures called vesicles. When called upon by ADH–AVPR2 interaction and resulting intracellular signaling, attachment and tethering proteins specifically direct the vesicles to fuse with the outer membrane of the cell. The fusion event results in the addition of the AQP2 molecules to the outer membrane. The total number of available AQP2-containing vesicles and the attachment and tethering proteins are both inherently limited in any given cell resulting in a saturation point for sensitivity of the cell to ADH [24].

When ADH is withdrawn, AQP2 accumulates in special membrane domains (clathrin-coated pits), which are subsequently engulfed (endocytosed) by the cell. Endocytosed AQP2 receptors are then recycled within the cells, ready for the next ADH signal. AQP2 is continuously and quickly recycled between the cell surface and intracellular compartments, rebounding between upper and lower limits for AQP2 cell surface localization. This behavior results in a reflection of the observed effects at the ADH saturation limit [24].

6.2. SHS model

We have developed a SHS model of the water/electrolyte balance system, seen in Fig. 4. The SHS has been adapted from the SDE model in [8] to include the hybrid thirst/dehydration mechanism described in [9]. The model includes two discrete states: normal and dehydrated. Transitions between the normal and dehydrated modes are defined by the transition guards in Fig. 4 and are based on the ratio of water to salt (or electrolyte concentration) in the body derived from data in [9].

We define three continuous states: total body water x_1, total body salt x_2, and ADH x_3 within each discrete state. The dynamics for the water and salt variables were based on simple input/output differences in the system with an added diffusion term that models uncertainty and system variability [8]. SDEs are used with constant diffusion because of the uncertainty of molecular interactions in these types of biochemical systems. Fluid output is directly dependent on the ADH concentration which is in turn affected by the fluid/salt ratio in the body.

The following SDEs describe the continuous dynamics in the normal state

$$\begin{aligned}
dx_1 &= (f_{in} - 45x_3^{-0.76})dt + 0.01dw_1\\
dx_2 &= (s_{in} - s_{out})dt + 0.01dw_2\\
dx_3 &= (-4.5)dt + 0.01dw_3
\end{aligned}$$

where f_{in} describes the amount of fluid input to the system per unit time, s_{in} describes the amount of salt input to the system per unit time, s_{out} describes the amount of salt output from the system per unit time, and $w = [w_1, w_2, w_3]^T$ is a three-dimensional Wiener process.

The next set of equations describe the dynamics when the body is in the dehydrated state determined by the electrolyte concentration.

$$\begin{aligned}
dx_1 &= (f_{in} - 45x_3^{-0.76})dt + 0.01dw_1\\
dx_2 &= (s_{in} - s_{out})dt + 0.01dw_2\\
dx_3 &= \left(\frac{80 * x_2}{x_1}\right)dt + 0.01dw_3
\end{aligned}$$

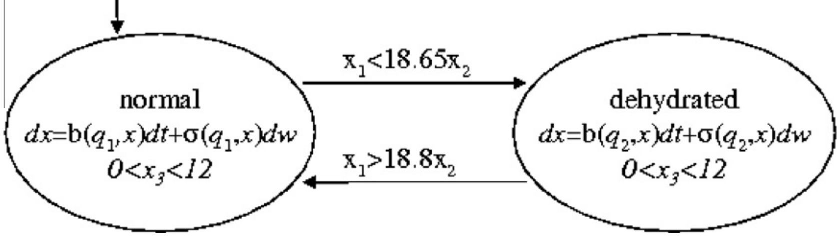

Fig. 4. SHS model of the water balance model.

The constants for the continuous dynamics were adapted from [8] to match the experimental data in [9]. We fit the experimental data to curves and determined appropriate adaptations for the dynamics when necessary. The values we used for our experiments are $f_{in} = 40$, $s_{in} = 2$, and $s_{out} = 2$. The fluid input f_{in} can be modeled as a continuous stream or discrete input, so for simplicity we consider only the continuous stream. Since our focus was primarily on the water balance, we modeled s_{in} and s_{out} as constant functions; however, these could be easily extended to model more realistic behavior if salt balance is the focus of the analysis.

Because ADH cannot have a negative value a reflective boundary is defined for x_3 at the value of zero. We also define a reflective boundary at $x_3 = 12$ to mimic the saturation limit of ADH in the kidneys. The limit is defined by the invariants in the system $x_3 \in 0, 12$. This range will not necessarily be the same for every person, but seems reasonable based on experimental data from [9]. Both the water concentration and salt concentration also must remain positive, but both of these concentrations maintain values sufficiently far from 0, so reflective boundaries are unnecessary.

Simulation of this model is important because it may help improve the understanding of the biological system and identify statistically significant aberrations in patients. Efficient, accurate simulation techniques are important to be able to refine the model and perform Monte Carlo analysis on the simulation data.

6.3. Boundary crossing detection

We implemented three SHS simulation algorithms: baseline (HEM with traditional boundary detection methods), Algorithm 5.1 (HMM with probabilistic boundary detection methods and adaptive timestepping), and Algorithm 5.1-fixedstep (HMM with probabilistic boundary detection methods and fixed time steps). We used the same Brownian motion for each set of simulation comparisons to highlight the algorithmic differences. In Fig. 5 the switching electrolyte boundary is presented for the baseline algorithm and Algorithm 5.1-fixedstep, and the difference between the detection times is shown by the gap between the indicated detection points. The proposed method anticipates the boundary crossing using probabilistic methods to avoid error incurred by over-shooting the crossing, while the baseline method re-starts the process only after the crossing is detected. It is evident that the anticipatory methods of the improved technique significantly alter the resulting trajectory thereby reducing the error incurred. In the water/electrolyte system this may mean that the actual system will react sooner than the model will indicate with the traditional simulation method predicts. For these simulations we used a step size of $\Delta t = .05$ and initial conditions shown in Table 1.

We consider the ADH concentration to demonstrate the reflecting boundary algorithm differences between the baseline algorithm and Algorithm 5.1-fixedstep. In Fig. 6 the reflecting boundary is represented by the dark line at $ADH = 12$. In the baseline method, the trajectory reaches the boundary and is kept within the valid state. However, the dynamics of the actual system are not well represented because the real system reaches a reflecting saturation level at the boundary. Algorithm 5.1-fixedstep accurately demonstrates the probabilistic effect of the reflected saturation boundary. The receptors in the real system cannot maintain the full concentration at 12 because the molecules of ADH have to be released to permit new molecules

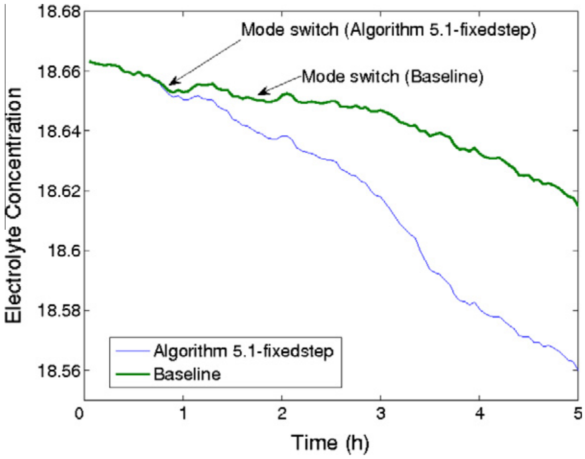

Fig. 5. Absorbing boundary in water balance model.

Table 1
Initial conditions.

Variable	x_1	x_2	x_3
Absorbing	39,790	2132	1
Reflecting	39,700	2132	11

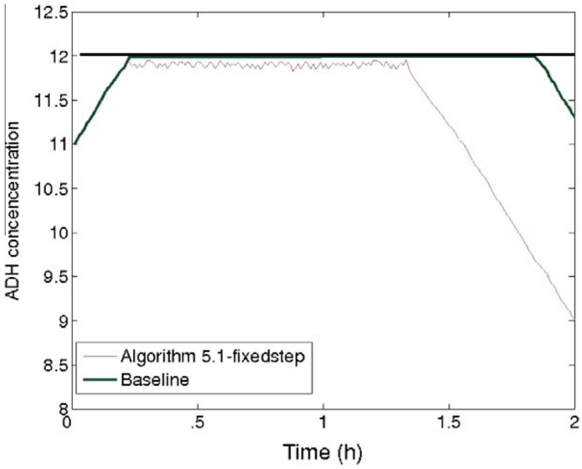

Fig. 6. Reflecting boundary in the water balance model.

to bind. The receptors cannot fire in an unbound state, so the influence of the *ADH* concentration must be reduced, as is evidenced by the small drops in concentration near the boundary. These drops eventually lead to a distinct difference between the outcomes of the two trajectories. While both trajectories eventually reach an equilibrium (not included in the figure), the difference in the dynamics leading to equilibrium may reveal new insights into the system. For these simulations, we used a step size of $\Delta t = 0.05$ and initial conditions shown in Table 1.

Performance results for the fixed step implementations are presented in Table 2. We ran 1000 sequential simulations of each algorithm at the given resolution. Algorithm 5.1-fixedstep increases the running time relative to the baseline method; however, the increase is small, the method scales well, and the accuracy improvement is significant. The simulations were performed on a 3 GHz desktop computer with 1 GB of RAM.

6.4. Step size results

The step size of the approximation directly influences the accuracy of the approximation. In Fig. 7, we compare four different step sizes of Algorithm 5.1-fixedstep and resulting trajectories with the same initial conditions as the absorbing

Table 2
Execution times (s).

Δt	Baseline	Algorithm 5.1-fixedstep
0.0001	352	374
0.0002	176	189
0.0005	70	75
0.001	37	38

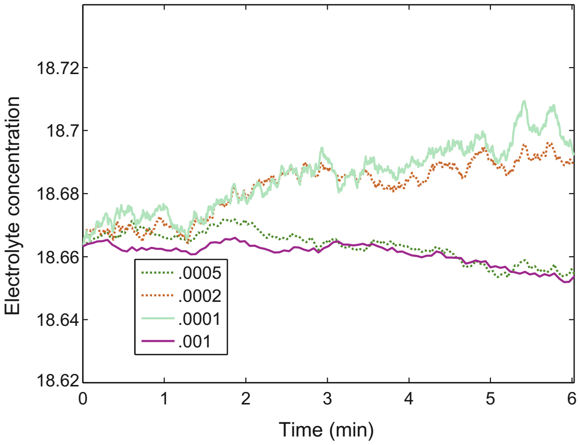

Fig. 7. Step size comparison.

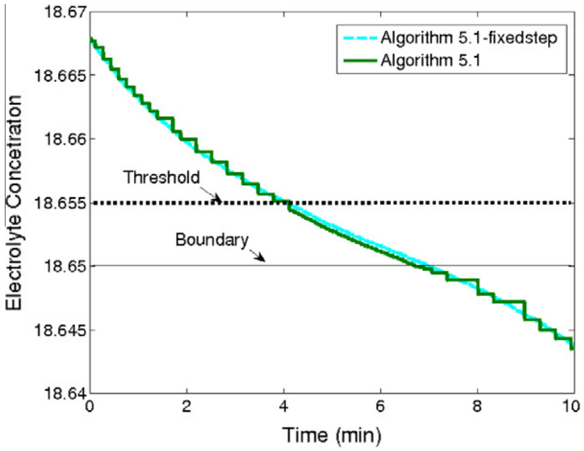

Fig. 8. Variable step example.

boundary example. We used the Wiener process from the highest-resolution trajectory with each lower-resolution simulation to ensure the comparison is accurate. At very fine resolutions the system becomes highly noisy. Thus, using more accurate approximation techniques with higher orders of convergence ensures that larger time steps can be used to maintain acceptable accuracy without having to approximate the highly noisy dynamics that lead to zeno behavior at boundaries if not handled carefully.

We compare simulations of Algorithm 5.1-fixedstep (with time step $\Delta t = 0.01$) with Algorithm 5.1 in Fig. 8. It can been seen in the figure that as the trajectory becomes less steep, the step size increases to improve efficiency, and when the trajectory crosses the threshold near the boundary, the variable time stepping algorithm uses the highest-resolution step to accurately estimate the boundary. The fixed step implementation required 1000 steps for the approximation while the variable step method with step error bounds of $|E_b| + |E_\sigma| < 0.01$ required only 295 steps, over tripling the efficiency and therefore reducing the runtime for this segment by almost one third. The figure shows that the accuracy is not significantly hindered by using the variable step size algorithm. Tighter error bounds can be used to create more accurate approximations at the cost of efficiency. Efficiency and execution time improvements are dependent on the dynamics of the model and error bounds, so significant efficiency gains can be expected but are not guaranteed.

7. Case study: biodiesel model

We present the biodiesel model as a large, complex model to demonstrate the scalability of our simulation methods for larger, complex systems. We use the biodiesel model to analyze the error and efficiency of both fixed and adaptive step methods. The biodiesel model has been developed and presented previously in [25], so we present a condensed description of the model.

7.1. Background

Biodiesel is made from vegetable oil and other chemicals by a process called transesterification [26]. The process involves six chemical species and six highly-coupled reactions. Vegetable oil in its purest form is made up of triglycerides (TG); however, it breaks down into diglycerides (DG) and monoglycerides (MG) as it is heated. An alcohol, methanol (M), is combined with the TGs, DGs, and MGs to convert them into biodiesel esters (E) and glycerol (GL). The chemical reactions are modeled using the reaction dynamics described in [25].

It is critical to determine whether or not a biodiesel processor can produce high quality biodiesel that passes the American Society for Testing and Materials (ASTM) biofuels tests. Studies show that the amounts of GLs and TGs that are less than one percent still allow the resulting fuel to meet ASTM specifications [27]. The ASTM requirements also limit the amount of methanol that is dissolved in the biodiesel; however, to meet this requirement most biodiesel production systems use post-processing washing techniques that clean the excess methanol from the biodiesel after the main reactions fully complete [28].

7.2. SHS model

We use the Variable Temperature BioDiesel (VTBD) model from [25]. The continuous dynamics in each state model the fluctuations in chemical concentrations and temperature. As seen in Fig. 9, the model has two discrete states. One models the system heating q_1, and the other models the cooling state q_2. The glycerol separation is modeled using the self-loop transitions in each discrete state.

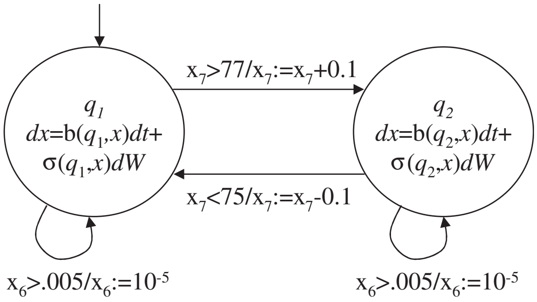

Fig. 9. SHS model of the VTBD system.

Since chemical dynamics are inherently stochastic, SDEs are an appropriate modeling paradigm for the chemical concentrations. We model the continuous dynamics for the system using SDEs as described in detail in [25].

Biodiesel is made in processors that use heaters and thermostats to regulate the temperature because the chemical reactions involved are highly sensitive to temperature. Heating the reacting liquid is necessary to ensure production of high quality biodiesel, but using too much heat wastes time and money. Therefore, processors generally have built-in thermostats that control the temperature. To model this, we use two discrete states of the system, one for heating and one for cooling. We model the thermostat controller using guarded transitions between the heating and cooling states given in [25].

7.3. Step size results

We use the VTBD model to demonstrate the performance and accuracy advantage of using adaptive time stepping methods. We consider the VTBD model with initial conditions: $x_1 = 0$, $x_2 = 0.8$, $x_3 = 3$, $x_4 = 0$, $x_5 = 10$, $x_6 = 0$, and $x_7 = 70$. We used Algorithm 5.1-fixedstep with time steps: $\Delta t = 0.001, 0.0001, 0.00001, 0.000001$. We compared the fixed step results with the adaptive Algorithm 5.1 with upper bounds: 0.01,0.001,0.0001,0.00001 and lower bounds: 0.0001,0.00001, 0.000001,0.0000001. Algorithm 5.1 starts with a small step size $\Delta t = 0.0000001$, and the algorithm quickly adjusts the step size according to the error, so little efficiency is lost and accuracy is preserved. We present the execution times and resulting overall error estimates for the fixed and adaptive methods in Fig. 10. It can be seen in the figure that the adaptive time step methods provide significant accuracy and efficiency improvements over fixed time step methods.

7.4. Monte Carlo results

Reachability for SHS can be computed using our improved ATHMM simulation algorithm (Algorithm 5.1) and Monte Carlo methods [29]. To evaluate the accuracy and efficiency of Monte Carlo methods and ATHMM algorithm, we tested the outcomes of the safety probability for the VTBD system using various numbers of iterations n. The performance and accuracy results of this analysis can be seen in Fig. 11. It is shown in the figure that increasing the number of iterations n increases the runtime therefore decreasing the efficiency, but the confidence in the solution improves significantly as the number of

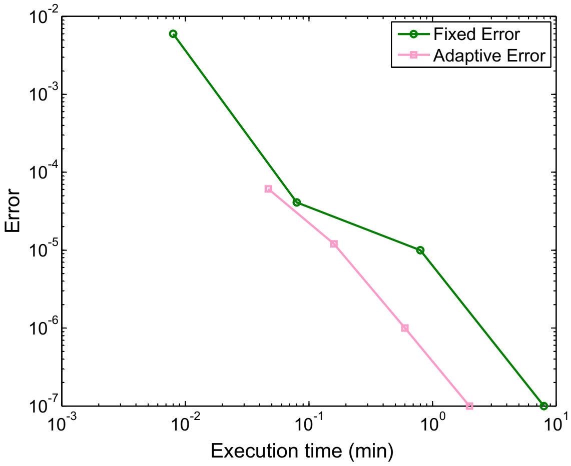

Fig. 10. Error comparison of time stepping methods for the VTBD model.

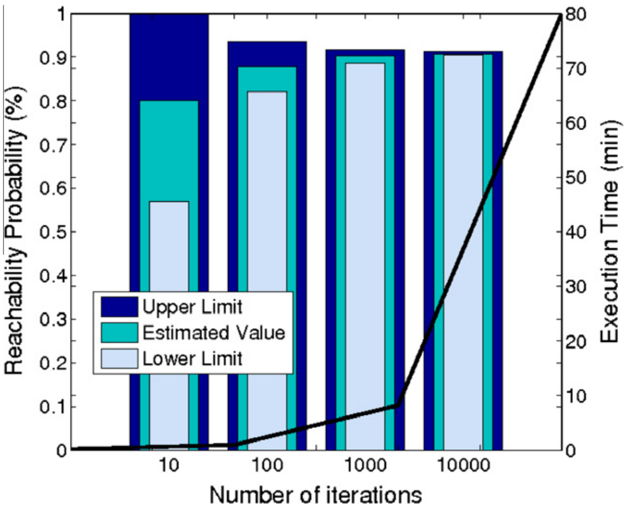

Fig. 11. 95% Confidence interval results with execution times for Monte Carlo methods using various numbers of iterations.

Monte Carlo iterations increases. The 95% confidence interval estimation, upper, and lower bounds for the four trials are shown in Fig. 11.

8. Conclusion

Accurate and efficient simulation of SHS is an important task because it is an important tool that can expose the intricacies of the complicated dynamics of highly-coupled systems like biochemical processes. The interplay between the continuous and discrete dynamics in SHS can introduce large errors into the simulations at the boundaries, so they must be approximated carefully. Our technique for simulating SHSs utilizing probabilistic absorbing boundary crossing detection and reflecting boundary calculation improves the accuracy and efficiency of the simulator when compared with the naive approaches. Further, the adaptive time stepping implementation we present further increases the ability of the simulator to take advantage of efficiency gains as well as provides the opportunity to bound the error of the approximation using the error estimates.

References

[1] M. Bujorianu, J. Lygeros, Theoretical foundations of general stochastic hybrid systems: modeling and optimal control, in: IEEE Conf. on Dec. and Cont., 2004.
[2] R. Ghosh, C. Tomlin, Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modeling: Delta–notch protein signalling, Syst. Biol. 1 (2004) 170–183.
[3] R. Alur, C. Belta, F. Ivanicic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, J. Schug, Hybrid modeling and simulation of biomolecular networks, in: Hybrid Systems Computation and Control, LNCS, vol. 2034, 2001, pp. 19–33.
[4] J. Hespanha, A. Singh, Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems, Int. J. Robust Control 15 (2005) 669–689 (Special Issue on Control at Small Scales).
[5] J. Hu, W. Wu, S. Sastry, Modeling subtilin production in bacillus subtilis using stochastic hybrid systems, in: Hybrid Systems Computation and Control, LNCS, vol. 2993, 2004, pp. 417–431.
[6] S. Drulhe, G. Ferrari-Trecate, H. de Jong, A. Viari, Reconstruction of switching thresholds in piecewise-affine models of genetic regulatory networks, in: Hybrid Systems Computation and Control, LNCS, vol. 3927, 2006, pp. 184–199.
[7] H. Salis, Y. Kaznessis, Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions, J. Chem. Phys. 122 (2005) 54–103.
[8] M. Leaning, R. Flood, D. Cramp, E. Carson, A system of models for fluid-electrolyte dynamics, IEEE Trans. Biomed. Eng. 32 (10) (1985) 856–864.
[9] O. Karanfil, A Dynamic Simulator for the Management of Disorders of the Body Water Metabolism, Master's Thesis, Bogazici University, 2005.
[10] M. Bernadskiy, R. Sharykin, R. Alur, Structured modeling of concurrent stochastic hybrid systems, FORMATS LNCS 3253 (2004) 309–324.
[11] Mathworks, Simbiology, <http://www.mathworks.com/products/simbiology/>.
[12] R. Mannella, Absorbing boundaries and optimal stopping in a stochastic differential equation, Phys. Lett. A 254 (1999) 257–262.
[13] G. Lamm, Extended Brownian dynamics. III: three dimensional diffusion, J. Chem. Phys. 80 (6) (1983) 2845–2855.
[14] E. Peters, T. Barenbrug, Efficient Brownian dynamics simulation of particles near walls. I: reflecting and absorbing walls, Phys. Rev. 66 (2002) 1–7.
[15] C. Constantini, B. Pacchiarotti, F. Sartoretto, Numerical approximation for functionals of reflecting diffusion processes, SIAM J. Appl. Math. 58 (1998) 73–102.
[16] E. Gobet, Euler schemes and half-space approximation for the simulation of diffusion in a domain, ESAIM: Probab. Stat. 5 (2001) 261–297.
[17] J. Gaines, T. Lyons, Variable step size control in the numerical simulation of stochastic differential equations, SIAM J. Appl. Math. 57 (1997) 1455–1484.
[18] H. Lamba, An adaptive timestepping algorithm for stochastic differential equations, J. Comput. Appl. Math. 161 (2003) 417–430.
[19] H. Salis, V. Sotiropoloulos, Y. Kaznessis, Multiscale Hy3S: hybrid stochastic simulation for supercomputers, BMC Bioinform. 7 (93) (2006).
[20] D. Riley, X. Koutsoukos, K. Riley, Modeling and simulation of biochemical processes using stochastic hybrid systems: the sugar cataract development process, in: Hybrid Systems Computation and Control, LNCS, vol. 4981, 2008, pp. 429–442.
[21] P. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, 1999.

[22] D. Riley, X. Koutsoukos, K. Riley, Modeling and analysis of the sugar cataract development process using stochastic hybrid systems, IET: Syst. Biol. 3 (3) (2009) 137–154.
[23] V. Sotiropoulos, Y. Kaznessis, An adaptive time step scheme for a system of stochastic differential equations with multiple multiplicative noise: chemical Langevin equation, a proof of concept, J. Chem. Phys. 128 (014103) (2008).
[24] Y. Noda, S. Sasaki, Regulation of aquaporin-2 trafficking and its binding protein complex, Biophys. Acta 1758 (2006) 1117–1125.
[25] D. Riley, X. Koutsoukos, K. Riley, Reachability analysis of stochastic hybrid systems: a biodiesel production system, Eur. J. Control, accepted for publication (Special Issue on Stochastic Hybrid Systems).
[26] H. Noureddini, D. Zhu, Kinetics of transesterification of soybean oil, JAOCS 74 (11) (1997) 1457–1463.
[27] S. Fernando, P. Karra, R. Hernandez, S. Jha, Effect of incompletely converted soybean oil on biodiesel quality, Energy 32 (2007) 844–851.
[28] Y. Zhang, M. Dube, D. McLean, M. Kates, Biodiesel production from waste cooking oil: 1. Process design and technological assessment, Bioresource Technology 89 (2003) 1–16.
[29] D. Riley, X. Koutsoukos, K. Riley, Reachability analysis for stochastic hybrid systems using multilevel splitting, in: Hybrid Systems Computation and Control, 2009.