Numer Algor (2015) 68:791–812
DOI 10.1007/s11075-014-9872-6

ORIGINAL PAPER

Adaptive time-stepping for the strong numerical solution of stochastic differential equations

Silvana Ilie · Kenneth R. Jackson · Wayne H. Enright

Received: 23 January 2014 / Accepted: 1 May 2014 / Published online: 23 May 2014

Abstract Models based on stochastic differential equations are of high interest today due to their many important practical applications. Thus the need for efficient and accurate numerical methods to approximate their solution. In this paper, we propose several adaptive time-stepping strategies for the strong numerical solution of stochastic differential equations in Itô form, driven by multiple Wiener processes satisfying the commutativity condition. The adaptive schemes are based on I and PI control, and allow arbitrary values of the stepsize. The explicit Milstein method is applied to approximate the solution of the problem and the adaptive implementations are based on estimates of the local error obtained using Richardson extrapolation. Numerical tests on several models arising in applications show that our adaptive time-stepping schemes perform better than the fixed stepsize alternative and an adaptive Brownian tree time-stepping strategy.

Keywords Stochastic differential equations · Adaptive time-stepping · PI control · Milstein method · Strong numerical approximation · Commutative noise

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

S. Ilie (✉)
Department of Mathematics, Ryerson University, Toronto, ON, M5B 2K3, Canada
e-mail: silvana@ryerson.ca

K. R. Jackson · W. H. Enright
Department of Computer Science, University of Toronto, Toronto, ON, M5S 3G4, Canada

K. R. Jackson
e-mail: krj@cs.toronto.edu

W. H. Enright
e-mail: enright@cs.toronto.edu

1 Introduction

Stochastic modeling and simulation have become areas of intense research in recent years, as more sophisticated mathematical models of physical phenomena became available. Stochastic differential equations arise in many applications. Examples include molecular biology, epidemiology, population dynamics, optimal control theory, hydrology, theoretical physics and finance [1–3].

Stochastic models are computationally much more challenging than deterministic models. The high computational cost of the numerical simulations of stochastic models arising in applications motivated the search for more efficient approaches [1, 4]. One way to reduce the computational cost of an approximation algorithm is to use adaptive time-stepping schemes to advance the numerical solution. In the framework of ordinary differential equations, stepsize adapting strategies have been proved to be essential in generating optimal algorithms. While for the numerical solution of ordinary differential equations such strategies have been well-developed [5], much less work has been done on designing adaptive algorithms for approximating the solution to stochastic differential equations (SDE).

The design of an adaptive time-stepping technique depends on whether the SDE is in the Itô or the Stratonovich form, and whether strong or weak numerical solution of the SDE is required. Weak numerical solutions are acceptable when only the moments of the exact solution need to be estimated accurately, while strong numerical solutions are required when individual trajectories of the exact solution need to be well approximated. Both weak and strong numerical methods are crucial and the choice between the two depends on the application. According to Burrage, Burrage & Tian [6], "in genetic regulation, for example, where the behaviour of just one molecule can be highly significant, strong solutions can be important". While strong numerical solutions can be critical for stochastic models in molecular biology and biochemistry [6–9], weak numerical solutions are sufficient for models in financial mathematics.

Adaptive time-stepping for the strong (pathwise) solution of stochastic differential equations driven by one Wiener process was considered by Lamba [10], Mauthner [11], Hofmann, Müller-Gronbach & Ritter [12]. Adaptivity for Stratonovich stochastic differential equations with multidimensional Wiener processes was studied by Burrage & Burrage [13], Burrage, Burrage & Tian [6]. In Hofmann et al. [12], adaptive time-stepping strategies in the mean-square sense were developed, which were optimal for asymptotically small stepsizes. Adaptive discretization schemes for the weak solution of stochastic differential equations were discussed in Szepessy et al. [14]. Gaines & Lyons [15] showed that to guarantee convergence for variable stepsize schemes applied to the pathwise solution of stochastic differential equations a strong order at least one discretization scheme is needed. An important difficulty in integrating the pathwise solution of a system of stochastic differential equations is that, when rejection of stepsizes is allowed, the solution should remain on the same Brownian path. Otherwise a bias in the numerical solution is introduced. Mauthner [11, 16] developed a general adaptive strategy for the numerical pathwise solution of Stratonovich stochastic differential equations with one driving Wiener process.

This strategy works as follows: once a stepsize is rejected and a smaller stepsize is considered, the necessary Wiener integrals for the smaller stepsize are conditioned on the previously generated Wiener integrals, corresponding to the rejected stepsize, such that the same Brownian path is traversed. Burrage & Burrage [13] introduced a general adaptive algorithm for Stratonovich SDEs, when Runge-Kutta methods are considered and an embedded method is used to estimate the local error. The Runge-Kutta methods employed have strong order one for problems with commutative noise. Gaines & Lyons [15] proposed a Brownian tree structure for adapting the stepsize for stochastic differential equations, where only halving and doubling of the stepsize is allowed. This is the standard adaptive time-stepping method for Itô SDEs driven by multidimensional Wiener processes, however it is a very restrictive strategy.

In this paper, we propose some alternative adaptive time-stepping strategies for the strong numerical solution of multidimensional Itô stochastic differential equations, driven by multiple Wiener processes which satisfy the commutativity condition. The case of the strong numerical solution of SDE driven by one Wiener process is well understood and we do not advocate the use of our approach for this class of problems. Stochastic differential equation models with commutative noise arise in a wide range of application areas such as genetics (e.g., Shiga model [1]), finance (e.g., LIBOR Market Models [17]), physics (e.g., stochastic Lorenz equations) and biological sciences (e.g., Duffing-van der Pol oscillator [1]), etc.

We employ the Milstein method to compute the numerical solution of the stochastic differential equation and use extrapolation to cheaply estimate the local error of the Milstein scheme. We propose some adaptive time-stepping strategies based on I and PI controllers. On each individual path, the adaptive method adjusts the time-step such that the local error during one step is below the user-prescribed tolerance. Such a time-stepping scheme is convergent, since the Milstein method has strong order of convergence one. This method works very well for problems which are non-stiff or mildly stiff. Our approach extends to Itô SDEs the work of Söderlind [18, 19], who developed PI-controllers which behave well for the numerical solution of ordinary differential equations. Moreover, in our adaptive time-stepping strategy, when a stepsize is rejected only the Wiener increments corresponding to the smaller stepsize are conditioned on the previously generated Wiener increments. This reduces the computational cost of adapting the time-step. Therefore, our direct method for Itô SDE is less expensive than transforming the problem into a Stratonovich SDE and applying the variable time-step technique proposed by Burrage, Herdiana and Burrage in [20] where, in addition, higher order Wiener integrals need to be conditioned.

The paper is organized as follows. In Section 2 we describe the necessary background for the numerical solution of stochastic differential equations. In Section 3, some new adaptive algorithms for selecting the stepsize are presented. In Section 4, we give some numerical results for several stochastic models arising in applications. We compare the proposed adaptive schemes with the fixed stepsize ones as well as to the existing adaptive methods based on a Brownian tree and show that our adaptive methods perform better than both of these alternative approaches.

2 Numerical solution of stochastic differential equations

Let us consider a system of stochastic differential equations in Itô form

$$d\mathbf{X} = \mathbf{f}(t, \mathbf{X})dt + \sum_{j=1}^{m} \mathbf{g}^j(t, \mathbf{X})dW^j \tag{1}$$

where $\mathbf{f} : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$, $\mathbf{g} : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^m$, and $W^j(t)$ are independent scalar Wiener processes for $j = 1, \ldots, m$. A Wiener process $W = \{W(t), t \geq 0\}$ is a Gaussian process satisfying

$$W(0) = 0 \ \text{ w.p.1} , \quad W(t) - W(s) \sim \sqrt{t - s} \cdot N(0, 1) ,$$

and having independent increments, $W(t) - W(s)$ and $W(u) - W(v)$, for any $0 \leq s < t < v < u \leq T$.

We assume that the drift coefficient $\mathbf{f}$ and each diffusion coefficient $\mathbf{g}^j$ in (1) satisfy the Lipschitz condition: there exists some constant $M > 0$ such that

$$\|\mathbf{f}(t, \mathbf{y}) - \mathbf{f}(t, \mathbf{z})\| \leq M\|\mathbf{y} - \mathbf{z}\| , \quad \|\mathbf{g}^j(t, \mathbf{y}) - \mathbf{g}^j(t, \mathbf{z})\| \leq M\|\mathbf{y} - \mathbf{z}\|$$

for all $\mathbf{y}, \mathbf{z} \in \mathbb{R}^d$ and $t \in [0, T]$. This condition [1] ensures that an initial value problem for the stochastic differential (1) has a pathwise unique solution on $[0, T]$.

For any $j = 1, \ldots, m$, let us denote the following differential operator by

$$L^j = \sum_{k=1}^{d} \mathbf{g}^{k,j} \frac{\partial}{\partial X^k} . \tag{2}$$

A stochastic differential equation in Itô form (1) can be written in Stratonovich form as

$$d\mathbf{X} = \mathbf{F}(t, \mathbf{X})dt + \sum_{j=1}^{m} \mathbf{g}^j(t, \mathbf{X}) \circ dW^j$$

where $\mathbf{F} = \left(F^1, \ldots, F^d\right)'$ and

$$F^k(t, \mathbf{X}) = f^k(t, \mathbf{X}) - \frac{1}{2} \sum_{j=1}^{m} L^j g^{k,j}(t, \mathbf{X})$$

for $k = 1, \ldots, d$, and the differential operator L^j is defined by (2).

Since in most cases a closed form solution does not exist, numerical methods are needed to approximate the solution of a stochastic differential equation, at some grid points of the time-interval $[0, T]$, $0 = t_0 < t_1 < \cdots < t_n < \cdots < t_N = T$.

Definition 1 A time-discretization of $\mathbf{X}(t)$ on $[0, T]$, $\hat{\mathbf{X}}_n$, is said to have *strong order of convergence* $\gamma > 0$ if there exist a constant $C > 0$, independent of h, and $\delta > 0$ such that

$$E\left(|\mathbf{X}(t_n) - \hat{\mathbf{X}}_n|\right) \leq Ch^\gamma ,$$

for any fixed $t_n = n \cdot h \in [0, T]$ and any $h \in (0, \delta)$.

2.1 Milstein method

A widely used numerical method of strong order 1 is the Milstein scheme, which may be written as

$$X_{n+1}^k = X_n^k + f^k(t_n, \mathbf{X}_n)h_n + \sum_{j=1}^{m} g^{k,j}(t_n, \mathbf{X}_n)\Delta W_n^j + \sum_{j_1,j_2=1}^{m} L^{j_1} g^{k,j_2}(t_n, \mathbf{X}_n) I_{(j_1,j_2)}^{(n)}$$

for $k = 1, \ldots, d$. The Wiener increments can be generated numerically by $\Delta W_n^j = W_{n+1}^j - W_n^j = \sqrt{h_n} \cdot r_{j,n}$ where $t_{n+1} = t_n + h_n$ and $r_{j,n}$ are realizations of the standard normal random variable $N(0, 1)$. We denote the double Itô integrals by

$$I_{(j_1,j_2)}^{(n)} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW^{j_1}(s_2) dW^{j_2}(s_1) \,, \tag{3}$$

where $j_1, j_2 = 1, \ldots, m$. Note that for $j_1 \neq j_2$ the double Itô integrals cannot be expressed in terms of the increments $\Delta W_n^{j_1}$ and $\Delta W_n^{j_2}$. Their numerical approximation is computationally intensive [1, Chapter 5]. Such expensive simulations can be avoided when the stochastic differential equation has commutative noise.

The stochastic differential equation is said to have *commutative noise* if

$$L^{j_1}\mathbf{g}^{j_2} = L^{j_2}\mathbf{g}^{j_1} \text{ for any } j_1, j_2 = 1, \ldots, m, \text{ with } j_1 \neq j_2 \,,$$

where the differential operators L^j are defined by (2). Then, the following general property can be used to simplify the numerical scheme,

$$I_{(j_1,j_2)}^{(n)} + I_{(j_2,j_1)}^{(n)} = \Delta W_n^{j_1} \Delta W_n^{j_2}$$

for $j_1 \neq j_2$ and $j_1, j_2 = 1, \ldots, m$. In addition, for any $j_1 = 1, \ldots, m$

$$I_{(j_1,j_1)}^{(n)} = \frac{1}{2}\left((\Delta W_n^{j_1})^2 - h_n\right) .$$

In the case of stochastic differential equations with commutative noise, the Milstein scheme reduces to

$$\begin{aligned} X_{n+1}^k = {} & X_n^k + f^k(t_n, \mathbf{X}_n)h_n + \sum_{j=1}^{m} \mathbf{g}^{k,j}(t_n, \mathbf{X}_n)\Delta W_n^j \\ & + \frac{1}{2}\sum_{j_1 \neq j_2}^{m} L^{j_1} g^{k,j_2}(t_n, \mathbf{X}_n)\Delta W_n^{j_1}\Delta W_n^{j_2} \\ & + \frac{1}{2}\sum_{j=1}^{m} L^j \mathbf{g}^{k,j}(t_n, \mathbf{X}_n)\left((\Delta W_n^j)^2 - h_n\right) \end{aligned}$$

for $k = 1, \ldots, d$. Therefore the simulation of higher dimensional Itô integrals is no longer necessary. It is interesting to observe that, similarly to the Milstein scheme, the simulation of stochastic differential equations with non-commutative noise by Runge-Kutta schemes of strong order 1 or higher is not possible if only the Wiener increments ΔW_n^j are employed. Thus, the advantage of using numerical schemes of

strong order greater than 1 is reduced, as then it is required to sample the expensive higher dimensional Itô integrals.

2.2 Local error estimation

By analogy to ordinary differential equations, assuming the same starting point of the current step, the (pathwise) local error is the difference between the exact and the numerical solution on the same Brownian path, at the end of the step. For a strong method, the numerical solution approximates the exact solution on each Brownian path. On such a path, while the local error is a random quantity, it has an Itô-Taylor series structure associated with it [1].

A variable stepsize implementation depends on the measure of the error generated by the numerical method under consideration. In the following, we use extrapolation [5] to estimate the local error obtained after each step with the Milstein method. Extrapolation is a cheap method to approximate the error if the problem is not stiff. The numerical solution is computed first in one step, say on $[t_n, t_{n+1})$, where $t_{n+1} = t_n + h_n$. Then, the solution is approximated in two steps over the same interval: on $[t_n, t_n + h_n/2)$ and again on $[t_n + h_n/2, t_{n+1})$. Finally, local extrapolation is applied, that is the local error in the numerical solution is approximated by taking the difference between the approximated values in two steps, $X_{n+1,2}$, and in one step, $X_{n+1,1}$.

If the user-prescribed tolerance is Tol, then the local error is approximated by

$$err(X_n, h_n) = \sqrt{\frac{1}{d}\sum_{k=1}^{d}\left(\frac{X^k_{n+1,2} - X^k_{n+1,1}}{Tol}\right)^2} \tag{4}$$

and should satisfy the condition

$$err(X_n, h_n) \leq 1\ . \tag{5}$$

We note that, for stochastic differential equations, if a numerical method is of strong (global) order γ, then the local error has order $(\gamma + 1/2)$ rather than $(\gamma + 1)$, as is the case for ordinary differential equations. The fractional order is a result of the property that the root mean square order of each Wiener process is $h_n^{1/2}$. Thus the local error approximation for a step h_n behaves as

$$err(X_n, h_n) \approx \frac{\phi(t_n, X_n)}{Tol} \cdot h_n^{\gamma+1/2}\ ,$$

where $\phi(t_n, X_n) = \phi_n$ is the principal error function. The function ϕ depends on both the drift and the diffusion coefficients. In general, the dependency of the principal error function on these coefficients is quite complex and it is difficult to decompose the local error estimate into the drift-dominating and the diffusion-dominating components. Extrapolation is a simple and inexpensive method to estimate the error for a general class of stochastic differential equations. In particular, it avoids the evaluations of quite complicated error terms.

3 An adaptive time-stepping scheme

In this section we discuss a variable time-stepping strategy which allows rejection of a stepsize, while guaranteeing that the correct Brownian path is followed. This ensures that the correct statistics of the numerical solution are maintained. According to Gaines & Lyons [15], a numerical method of strong order 1 guarantees that an adaptive time-stepping method converges to the strong solution of a stochastic differential equation. Gaines & Lyons [15] used a stepsize selection strategy which allowed only preserving the step or doubling/halving of the previous stepsize. More precisely, the choice of the time intervals was such that a tree structure was generated, called a *Brownian tree*. Following Gaines & Lyons [15], we assume, without loss of generality and for ease of discussion, that the interval of integration is [0, 1]. This binary tree may have only intervals of the form $[k/2^n, (k+1)/2^n]$ for $k,\ n \in \mathbb{N}$. The first level of the tree is obtained by computing the Wiener increments on N unit length subintervals of the initial interval,

$$\Delta W_k = W(k) - W(k-1), \text{ for } k = 1, \ldots, N\ .$$

Therefore each Wiener increment ΔW_k is normally distributed with mean zero and variance one. When the accuracy is not satisfied on one of the intervals $[k-1, k]$, then this interval is divided in two subintervals of the same length and the accuracy is verified on each. If a smaller stepsize is needed to satisfy the accuracy requirement, then such a stepsize is obtained with a recursive generation of smaller subintervals by taking the midpoint of the current interval. The Wiener increments corresponding to level j are constructed as

$$\begin{aligned} \Delta W_{2k-1,j+1} &= \frac{1}{2}\Delta W_{k,j} + z_{k,j}\ , \\ \Delta W_{2k,j+1} &= \frac{1}{2}\Delta W_{k,j} - z_{k,j}\ , \text{ for } j = 1, 2, \ldots, \end{aligned}$$

where $z_{k,j-1}$ are normally distributed with mean zero and variance 2^{-j}. The Brownian path has to pass through all the points generated for the smaller subintervals before progressing to an upper lever in the tree. This condition imposes serious restrictions on the selection of a stepsize, and may slow down the integration process significantly.

In the generic case of a non-commutative SDE, Gaines & Lyons show that it is sufficient to include the approximation of the Lévy areas $A_{(i,j)}(t, t+h) = I_{(i,j)}(t, t+h) - I_{(j,i)}(t, t+h)$, in addition to the Wiener increments, to guarantee convergence of the variable stepsize method. A different technique is to discretize directly the double Itô integrals (3) by using the Karhunen-Loève or Fourier expansions of the Brownian bridge processes corresponding to the Wiener processes in the SDE. Both approximations, by Lévy areas and by Karhunen-Loève expansions, are quite expensive.

The strategy we consider in this paper is based on an alternative method introduced by Mauthner [11, 16] for stochastic differential equations with a single driving Wiener process and which was extended in Burrage [21], Burrage & Burrage [13] and Burrage, Herdiana and Burrage [20] for multiple driving Wiener processes. These

approaches apply to Stratonovich stochastic differential equations. They employ embedded stochastic Runge-Kutta (SRK) methods to advance the solution and the difference between the higher order and the lower order approximations is used to estimate the local error. The SRK methods are of global order 1, as required for the convergence of an adaptive scheme, provided that the noise is commutative. Otherwise the global order is 1/2.

The idea is as follows: assume that a step h is computed and the Wiener increment

$$\Delta W_h = W(t+h) - W(t) = \int_t^{t+h} dW(s)$$

is sampled ($\Delta W_h = i_h$, with $i_h \in \mathbb{R}$ a realization of the Wiener increment). If the step h is rejected, then i_h is stored and a smaller stepsize $0 < h_1 < h$ is tried. Then the Wiener increments on the subintervals $[t, t+h_1]$ and $[t+h_1, t+h]$ are evaluated, conditioned on the Wiener increments on the entire interval $[t, t+h]$, i.e. $\Delta W_h = i_h$. This ensures maintaining the same Brownian path.

Let us denote the Wiener increments on $[t, t+h_1]$ and on $[t+h_1, t+h]$ by

$$\Delta W_{h_1} = W(t+h_1) - W(t)\,, \qquad \Delta W_{h_2} = W(t+h) - W(t+h_1)\,,$$

where $h_2 = h - h_1$. The increments should satisfy the additivity condition given by the direct integration

$$\Delta W_{h_1} + \Delta W_{h_2} = \int_t^{t+h} dW(s) = i_h\,. \tag{6}$$

Moreover, the Wiener increments are normally distributed and they obey the conditions for expectation

$$E((\Delta W_{h_1}, \Delta W_{h_2})) = (0,0) \tag{7}$$

and for covariances

$$\mathrm{Cov}((\Delta W_{h_1}, \Delta W_{h_2})) = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix} \tag{8}$$

which guarantee that the correct Brownian path is followed. Then, it can be shown that the Wiener increments satisfying the conditions (6), (7) and (8) can be calculated as [13, 16]

$$\Delta W_{h_1} = \frac{h_1}{h} i_h + \sqrt{\frac{h_1 h_2}{h}}\, z\,, \qquad \Delta W_{h_2} = \frac{h_2}{h} i_h - \sqrt{\frac{h_1 h_2}{h}}\, z\,, \tag{9}$$

where $z = N(0,1)$ is a new random variable.

A stepsize h_{n+1} is accepted if $err(X_{n+1}, h_{h+1}) \leq 1$, otherwise it is rejected and the procedure above is applied. Since the error estimate computed for the previous successful stepsize is $err(X_n, h_n) = (\phi_n / Tol) h_n^{\gamma+1/2}$, then an optimal stepsize h_{n+1} should satisfy $(\phi_{n+1}/Tol) h_{n+1}^{\gamma+1/2} = 1$. Based on these relations, the standard approach from ODE is to choose the next stepsize h_{n+1} to satisfy

$$h_{n+1} = h_n \left(\frac{fac}{err(X_n, h_n)} \right)^{1/(\gamma+1/2)}$$

where the safety factor $fac \leq 1$ is introduced to reduce the chance of rejecting the next stepsize. Since the stepsize should not increase or decrease too much, the following stepping scheme is employed

$$h_{n+1} = h_n \min\left(facmax, \max(facmin, (fac/err(X_n, h_n))^{1/(\gamma+1/2)})\right) \quad (10)$$

which is known as the integral stepsize controller in the deterministic framework [5]. The maximal stepsize increase allowed $facmax > 1$ and the minimal stepsize decrease $facmin < 1$ are chosen depending on the problem. We remark that SDEs seem to be more sensitive than ODEs to the choice of $facmax$ and $facmin$. For example, if $facmax$ is too large, the algorithm may lead to many step rejections. On the other hand, if $facmax$ is too small, then the stepping strategy may take more steps than necessary to traverse the integration interval, while satisfying the tolerance. A smaller sensitivity arises for $facmin$.

3.1 Proportional-integral stepsize control

We present below the approach to adaptive time-stepping based on control theory, and in particular using predictive-integral (PI) controllers, proposed by Söderlind for ODE [18, 19] and by Burrage, Herdiana and Burrage [20] for Stratonovich SDE. We are interested in the PI-controllers which behave well for the numerical solution of Itô SDEs. Let us take k to be the order of the local error, that is $k = \gamma + 1/2$ where γ is the order of the global error. Thus, in the asymptotic regime, the local error per step can be written as $e(X_n, h_n) = \phi_n h_n^k$, where ϕ_n may vary significantly. The standard adaptive stepsize algorithm based on this error is

$$h_{n+1} = h_n \left(\frac{fac \cdot Tol}{e(X_n, h_n)}\right)^{1/k}. \quad (11)$$

By taking the logarithm in (11) we obtain

$$\log(h_{n+1}) = \log(h_n) + \frac{1}{k}\big(\log(fac \cdot Tol) - \log(e(X_n, h_n))\big) \quad (12)$$

which is called a discrete-time integral (I) controller in control theory. Its name comes from the observation that the solution of the difference equation (12) is

$$\log(h_n) = \log(h_0) + \frac{1}{k}\sum_{j=0}^{n-1}(\log(fac \cdot Tol) - \log(e(X_j, h_j))), \quad (13)$$

which is similar to a discrete representation of an integral.

The process that needs to be controlled is modeled by the local error, for which we derive that

$$\log(e(X_n, h_n)) = k\log(h_n) + \log(\phi_n). \quad (14)$$

We wish to study now the closed loop-dynamics [18] which considers the interaction of the controller and of the process it controls. Thus, we can substitute (14) into (12) to obtain

$$\log(h_{n+1}) = \frac{1}{k}(\log(fac \cdot Tol) - \log(\phi_n)). \quad (15)$$

This is a difference equation in $\log(h_n)$. Its characteristic equation, $q = 0$, has as unique root the origin. The factor $k_I = 1/k$ is called the integral gain of the controller. Though, in control theory k_I is viewed as a design parameter, its value being determined depending on the desired properties of the controller. In the particular case $k_I = 1/k$, we obtain the dead-beat controller, which predicts a stepsize as non-smooth as the principal error function. We note that the principal error function ϕ_n depends on the Wiener processes, thus it takes random values. This makes the dead-beat controller less attractive when solving numerically stochastic differential equations. If k_I is an arbitrary parameter, then the closed loop-dynamics become

$$\log(h_{n+1}) = (1 - kk_I)\log(h_n) + k_I(\log(fac \cdot Tol) - \log(\phi_n)) .$$

The difference equation above has the characteristic equation $q - (1 - kk_I) = 0$ with the root $q = 1 - kk_I$. The controller is stable if and only if its root is inside the unit circle, thus $kk_I \in (0, 2)$. Hence, we derived the integral controller

$$h_{n+1} = h_n \left(\frac{fac \cdot Tol}{e(X_n, h_n)} \right)^{k_I} .$$

However, it is well-known in control theory that more robust controllers may be designed by inserting a proportional component in the integral controller. Such controllers are called proportional-integral, or PI. We may compute them by adding a term proportional to the control error $(\log(fac \cdot Tol) - \log((e(X_{n-1}, h_{n-1}))$ to the integral controller (13). Consequently,

$$\begin{aligned} \log(h_n) \;=\; & \log(h_0) + k_I \textstyle\sum_{j=0}^{n-1} (\log(fac \cdot Tol) - \log(e(X_j, h_j))) \\ & + \; k_P(\log(fac \cdot Tol) - \log((e(X_{n-1}, h_{n-1})) , \end{aligned}$$

where k_P is the proportional gain. This leads to the following recursion

$$\begin{aligned} \log(h_{n+1}) \;=\; & \log(h_n) + k_I(\log(fac \cdot Tol) - \log(e(X_n, h_n))) \\ & + \; k_P(\log(e(X_{n-1}, h_{n-1}))) - \log(e(X_n, h_n))) \end{aligned} \tag{16}$$

Hence, we get the PI-controller

$$h_{n+1} = h_n \left(\frac{fac \cdot Tol}{e(X_n, h_n)} \right)^{k_I} \left(\frac{e(X_{n-1}, h_{n-1})}{e(X_n, h_n)} \right)^{k_P}$$

or, equivalently,

$$h_{n+1} = h_n \left(\frac{fac \cdot Tol}{e(X_n, h_n)} \right)^{k_I + k_p} \left(\frac{e(X_{n-1}, h_{n-1})}{fac \cdot Tol} \right)^{k_P} . \tag{17}$$

If we substitute the asymptotic model of the local error (14) into the difference equation of the controller (16), we obtain the closed loop-dynamics

$$\begin{aligned} \log(h_{n+1}) \;=\; & (1 - kk_I - kk_P)\log(h_n) + kk_P \log(h_{n-1}) \\ & + \; k_I(\log(fac \cdot Tol) - \log(\phi_n)) + k_P(\log(\phi_{n-1}) - \log(\phi_n)) \end{aligned}$$

The characteristic equation of this difference equation is

$$q^2 - (1 - kk_I - kk_P)q - kk_P = 0 . \tag{18}$$

The PI-controller is stable provided that the roots of characteristic equation (18) are inside the unit circle. We are interested in controllers which are suited for solving numerically non-stiff SDEs.

We note that the controller PI-1 with parameters satisfying $(kk_I, kk_P) = (0.3, 0.1)$ gives good results for the Milstein scheme. The controller is stable.

After performing many simulations, we propose an improved controller, PI-2, with parameters $(kk_I, kk_P) = (0.101, 0.009)$. The PI-2 controller shows an enhanced performance over the other PI-controllers tested, including PI-1 and the standard PI-controllers for ODEs, when Milstein's method is employed. Its roots are $[0.9, -0.01]$, therefore it is stable. It has the advantage that its negative root is quite small while the magnitude of the ratio of the positive to negative root is large, thereby reducing the risk of step rejections.

In Fig. 1, we show in the top plot the values, inside the unit circle, of the roots q_1 (*continuous line*) and q_2 (*dashed line*) of the (18) as functions of the parameters kk_I and kk_P; the black dots show the values of the roots for the controller PI-2. The bottom plot gives a view from above, with the black lines delimiting the region in the (kk_I, kk_P) plane where both roots of the (18) are within the unit circle.

We remark that there is a trade-off between finding the largest ratio of the positive to the negative value of the roots inside the unit circle and minimizing the total computational cost of the method, that is the total number of attempted steps for the same tolerance. A very large ratio of the roots is represented by points close to (0, 0) (blue point in Fig. 1, *bottom plot*) and reduces the number of rejected steps but requires a large total number of steps, being too conservative. Our choice for the PI-2 controller (*black point* in Fig. 1, bottom plot) is determined experimentally, as a very good compromise.

The results for the PI-1, PI-2 and dead-beat controllers are given below. We note that, when applied to stochastic differential equations, the standard PI-controllers developed for non-stiff ODEs require more work to achieve the same accuracy than the adaptive controllers proposed in this paper. This is because their design objectives are different in the deterministic framework. In the case of ODEs, the objective is to choose controllers which lead to smooth stepsize sequences. In the case of SDEs, one important challenge when designing pathwise-adaptive methods is to decrease the (typically large) number of step rejections, since such rejections add significantly to the computational effort. For mildly stiff Itô SDEs simulated with Milstein's method, we propose adaptive controllers for the strong (pathwise) numerical solution, that reduce the number of rejected steps and the total computational cost. The adaptive schemes introduced are recommended for problems with non-negligible noise. If, however, the focus is on the accurate approximation of the expected value $E(F(X(T)))$, where F is a function and $X(\cdot)$ is the solution of the SDE (1), not on the accurate simulation of each individual trajectory, then a different approach is more efficient. This approach, called a multi-level Monte-Carlo (MLMC) method, was introduce by Giles in [22, 23]. The MLMC methods [24] reduce the number of paths that need to be generated in order to accurately approximate the expectation, by averaging over a hierarchy of trajectories ranging from coarser to finer grids. The coarser grid paths are more efficient to simulate but they are less accurate, while the finer grid trajectories are more accurate but they are also more expensive to generate.

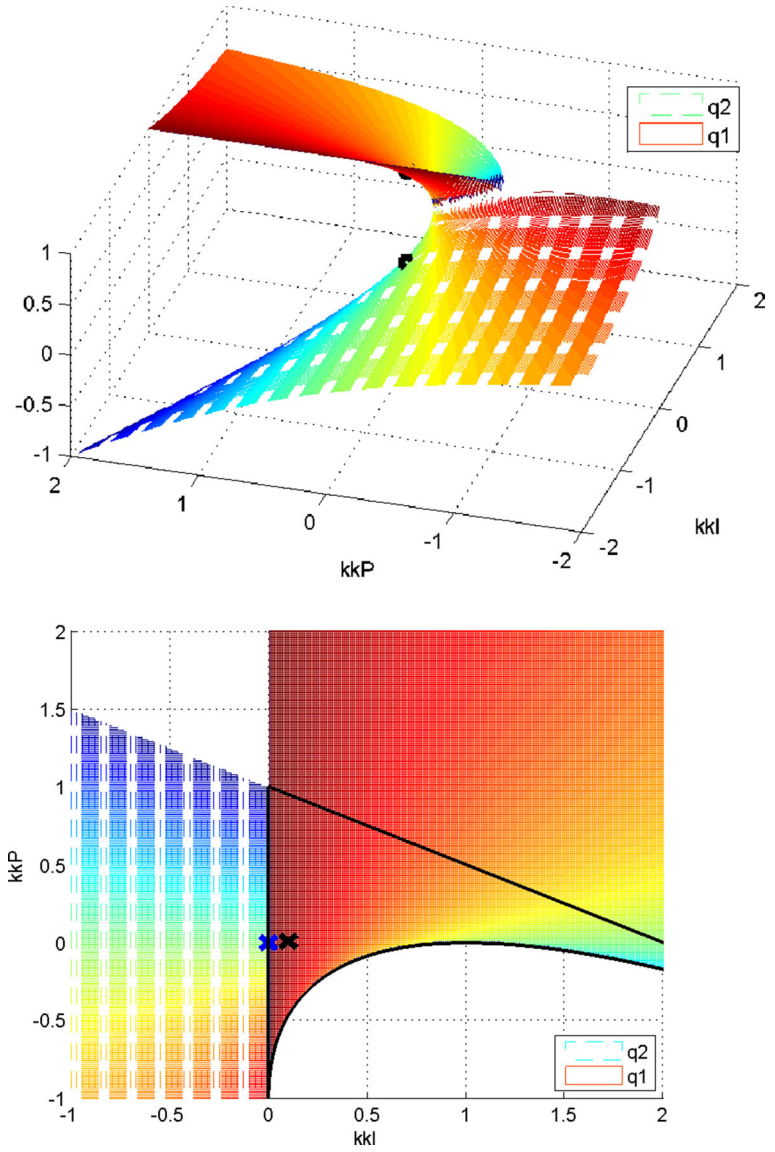

Fig. 1 The roots of equation (18) as functions of kk_I and kk_P. The *black dots* in the *top plot* show the roots used in PI-2. The *bottom plot* shows the region in the (kk_I, kk_P) plane where both roots of the equation (18) are within the unit *circle*

On a range of problems, the MLMC strategies maintain the accuracy associated with the finer grids by using also some of the coarser grid simulations, such that the overall computational cost is significantly reduced. This is achieved by ensuring a close coupling between the successive simulation levels, thus obtaining a low variance between these levels of approximation. An interesting direction of research would be

to design adaptive I and PI-controllers, based on good error estimates, for the MLMC methods, which would increase the efficiency of these techniques. The hierarchy of grids would be obtained by increasing the accuracy of the numerical solution.

In the next section, we test the proposed adaptive techniques against the two other currently existing strategies for the strong numerical solution of generic Itô SDE driven by multiple Wiener processes (with or without commutative noise), the fixed stepsize scheme and the Brownian tree adaptive strategy. Our methods show an improved performance over both of these alternative schemes for the commutative case, which suggests that a similar improvement may be possible in the non-commutative case.

4 Numerical results

In this section we test the adaptive algorithms proposed above on several examples of Itô stochastic differential equations arising in applications. The adaptive schemes are compared to fixed stepsize schemes and adaptive schemes using Brownian trees for the strong numerical solution of the models under consideration. The procedure for the comparison is as follows: for each tolerance Tol we ran the adaptive algorithms first, for the I and PI controllers and recorded the total numbers of steps attempted by each (the number of accepted and of rejected steps). Next, we ran the fixed stepsize algorithm with the same number of steps as the maximum between the number of steps taken by the I and the PI-based methods for the tolerance Tol. In addition, we ran the existing adaptive schemes based on Brownian trees for the same tolerance and record the total number of attempted steps. In each case, we estimated the error obtained according to the formula for the local error (4). We remark that the error recorded in the last column of Tables 1, 2 and 3 is computed as the mean over the maximum of the local errors on each individual trajectory. However, we should note that the I and the PI adaptive schemes ensure that the local error on each step and on each trajectory is below the given tolerance.

4.1 Marine bacteriophage infection model

A dimensionless deterministic model for the epidemics induced by the virulent phages on marine bacteria was given by Beretta & Kuang [25]. Below, we consider a stochastic extension of the model introduced in Carletti [26] and Carletti et al. [27]

$$\begin{aligned} ds(t) &= (as(t)(1-(i(t)+s(t)))-s(t)p(t))dt+\sigma_1(s(t)-s^*)dW^1(t) \\ di(t) &= (s(t)p(t)-\ell i(t))dt+\sigma_2(i(t)-i^*)dW^2(t) \\ dp(t) &= (-s(t)p(t)-mp(t)+b\ell i(t))dt+\sigma_3(p(t)-p^*)dW^3(t)\,. \end{aligned} \tag{19}$$

Here s represents the susceptible bacteria, i the infected bacteria and p the phage (viruses). The parameters a, ℓ, m and b are the bacteria logistic growth, the bacteria lysis death rate, the phage death rate and the virus replicating factor, respectively. They have the values observed for the infection by viruses of the bacteria *Cytophage*

Table 1 The phage-bacteria interaction model (1000 trajectories) $facmin = 0.2$, $facmax = 1.4$

Tol	Method	Ratio times	# Attempted steps (mean)	# Accepted steps (mean)	# Rejected steps (mean)	Error
10^{-2}	Adaptive PI-2	1.00	592	511	81	0.993
	Adaptive PI-1	1.06	612	499	113	0.994
	Adaptive I	1.14	650	492	158	0.995
	Brownian tree	–	760	501	259	0.995
	Fixed		650	–	–	28.74
$2 \cdot 10^{-3}$	Adaptive PI-2	1.00	1123	962	161	0.996
	Adaptive PI-1	1.06	1147	930	217	0.996
	Adaptive I	1.10	1195	899	296	0.997
	Brownian tree	–	1438	951	487	0.997
	Fixed		1195	–	–	43.66
10^{-3}	Adaptive PI-2	1.00	1582	1352	230	0.997
	Adaptive PI-1	1.02	1595	1290	305	0.998
	Adaptive I	1.07	1655	1243	412	0.998
	Brownian tree	–	2043	1352	991	0.998
	Fixed		1655	–	–	46.79
$2 \cdot 10^{-4}$	Adaptive PI-2	1.00	3940	3354	586	0.999
	Adaptive PI-1	1.02	3951	3185	766	0.999
	Adaptive I	1.05	4002	2994	1008	0.999
	Brownian tree	–	5067	3360	1707	0.999
	Fixed		4002	–	–	48.40

marinoflava, that is $a = 10$, $\ell = 24.628$, $m = 14.925$ and $b = 60$. The noise perturbs the positive equilibrium position $E_+ = (s^*, i^*, p^*)$ where

$$s^* = \frac{m}{b-1}, \; i^* = \frac{as^*(1-s^*)}{\ell + as^*}, \; p^* = \frac{a\ell(1-s^*)}{\ell + as^*} \; .$$

Note $b \gg b^* = 1 + m = 15.925$. The initial conditions are $(s_0, i_0, p_0) = (0.3, 0.2, 5)$, while the noise coefficients have the values $\sigma_i = 0.4$ for $i = 1, 2, 3$.

The evolution of the three interacting species is given in Fig. 2 and a plot of the stepsizes taken by the integral adaptive scheme vs. time for $Tol = 2 \cdot 10^{-3}$ on a Brownian path is shown in Fig. 3. The behaviors of the I, PI-1, PI-2 adaptive, the adaptive Brownian tree, and the fixed step size algorithms are reported in Table 1. The table shows the ratio of the computation times of the adaptive algorithms to the running time of the PI-2 adaptive scheme, the number of attempted steps, accepted steps and the accuracy of the numerical method for a sequence of imposed tolerances. The Brownian tree method fails on at least 10% of the trajectories tried, therefore its average computational times were not meaningful and were not reported. For the Brownian tree method, Table 1 presents the average number of attempted, accepted and rejected steps only on the successful trials. For the same total work, the fixed

Table 2 The stochastic Brusselator model (2000 trajectories), $facmax = 1.1$, $facmin = 0.2$

Tol	Method	Ratio times	# Attempted steps (mean)	# Accepted steps (mean)	# Rejected steps (mean)	Error
$2 \cdot 10^{-2}$	Adaptive PI-2	1.00	464	415	49	0.990
	Adaptive PI-1	1.01	464	415	49	0.990
	Adaptive I	1.02	464	410	54	0.991
	Brownian tree	1.07	531	356	175	0.994
	Fixed		464	–	–	5.147
10^{-2}	Adaptive PI-2	1.00	590	529	61	0.990
	Adaptive PI-1	1.01	593	530	63	0.993
	Adaptive I	1.02	592	521	71	0.993
	Brownian tree	1.14	727	485	242	0.997
	Fixed		593	–	–	5.724
$2 \cdot 10^{-3}$	Adaptive PI-2	1.00	1165	1048	117	0.996
	Adaptive PI-1	1.01	1178	1060	118	0.996
	Adaptive I	1.03	1177	1033	143	0.997
	Brownian tree	1.62	2197	1459	738	0.998
	Fixed		1178	–	–	7.135
10^{-3}	Adaptive PI-2	1.00	1643	1476	167	0.997
	Adaptive PI-1	1.01	1666	1502	164	0.998
	Adaptive I	1.02	1668	1465	203	0.998
	Brownian tree	1.62	3049	2024	1025	0.998
	Fixed		1668	–	–	7.978
$2 \cdot 10^{-4}$	Adaptive PI-2	1.00	4083	3662	421	0.999
	Adaptive PI-1	1.01	4198	3802	396	0.999
	Adaptive I	1.03	4131	3641	490	0.999
	Brownian tree	1.65	7267	4807	2461	0.999
	Fixed		4198	–	–	11.56

stepsize scheme gives an error up to 48 times larger than the adaptive algorithms for the tolerances tested. The ratio between the number of rejected steps and the total number of attempted steps for the adaptive method based on the I-controller is around 25%, for the PI-1 controller is about 19%, for the PI-2 controller is less than 15%, while, for the adaptive Brownian tree scheme, the ratio is between 33% and 48% on the successful trials. The total work performed by the PI methods was less than that of the I scheme, with the PI-2 adaptive scheme taking slightly fewer steps in total than PI-1 for the same tolerance. The adaptive Brownian tree algorithm took considerably more attempted steps on its successful trajectories than the I, PI-1, and PI-2 methods for the same tolerance. Moreover, the Brownian tree approach leads to a biased numerical solution due to the large number of rejected trajectories.

Table 3 The stochastic chemical model (1000 trajectories) for $facmax = 1.5$, $facmin = 0.2$

Tol	Method	Ratio times	# Attempted steps (mean)	# Accepted steps (mean)	# Rejected steps (mean)	Error
$2 * 10^{-2}$	Adaptive PI-2	1.00	1342	1145	197	0.9758
	Adaptive PI-1	1.03	1356	1067	289	0.9888
	Adaptive I	1.05	1386	1002	384	0.9919
	Brownian tree	1.12	1556	1025	531	0.9978
	Fixed	–	1386	–	–	111.6604
10^{-2}	Adaptive PI-2	1.00	2109	1802	307	0.9840
	Adaptive PI-1	1.03	2123	1667	456	0.9926
	Adaptive I	1.05	2164	1564	600	0.9948
	Brownian tree	1.12	2418	1597	821	0.9987
	Fixed	–	2164	–	–	101.3402
$2 * 10^{-3}$	Adaptive PI-2	1.00	6108	5286	822	0.9929
	Adaptive PI-1	1.03	6146	4845	1301	0.9975
	Adaptive I	1.05	6217	4493	1724	0.9982
	Brownian tree	1.10	6916	4576	2340	0.9995
	Fixed	–	6217	–	–	87.2380
10^{-3}	Adaptive PI-2	1.00	9911	8646	1265	0.9956
	Adaptive PI-1	1.03	9961	7909	2052	0.9985
	Adaptive I	1.06	10133	7332	2801	0.9989
	Brownian tree	1.10	11008	7285	3723	0.9997
	Fixed	–	10133	–	–	80.3096

On these trajectories, if a step satisfies the accuracy criteria, it is accepted even if it produces negative population numbers. However, the integration will not recover after such a step is accepted. By contrast, the proposed adaptive methods preserve the positive amounts of species throughout the integration. Therefore, for our adaptive strategies no additional positivity condition needs to be imposed. Also, we remark that the fixed stepsize strategy is considerably less accurate than the variable stepsize implementations.

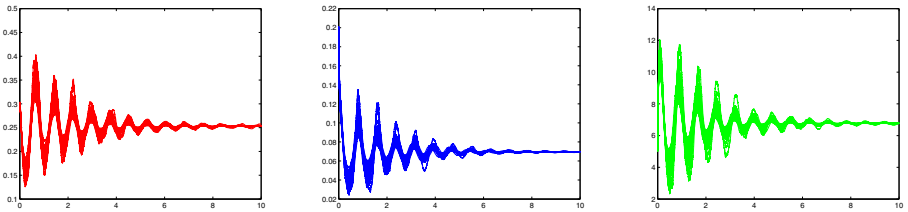

Fig. 2 The phage-bacteria interaction model: susceptible bacteria (*left*) and infected bacteria (*center*) and phage (*right*). Only 40 trajectories are shown

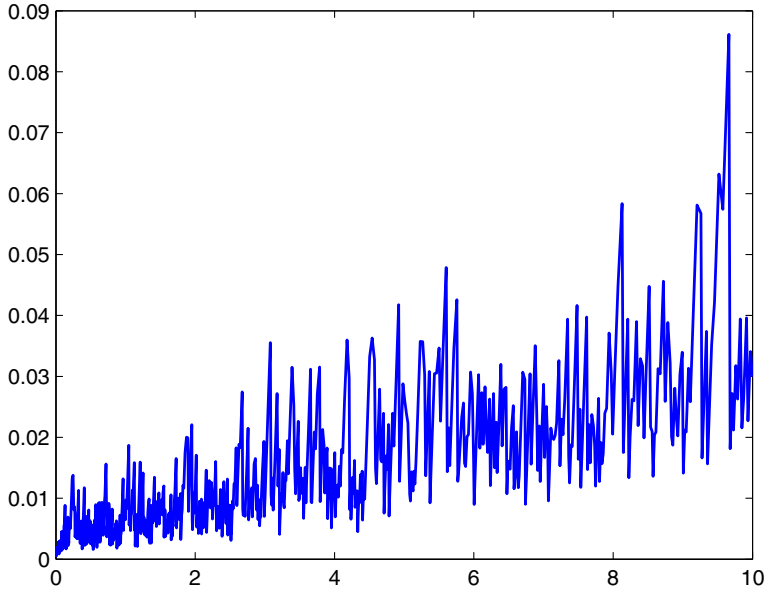

Fig. 3 The phage-bacteria interaction model: stepsize vs. time for Tol=$2 \cdot 10^{-3}$ and the I-adaptive scheme

4.2 Stochastic Brusselator

Another example of interesting qualitative behavior is the stochastic Brusselator [1]. The mathematical model is

$$\begin{aligned} dx_1 &= \big((\alpha-1)x_1 + \alpha x_1^2 + (1+x_1)^2 x_2\big)dt + \sigma x_1(1+x_1)dW_t \\ dx_2 &= \big(-\alpha x_1 - \alpha x_1^2 - (1+x_1)^2 x_2\big)dt - \sigma x_1(1+x_1)dW_t \end{aligned} \tag{20}$$

for $t \in [0, 100]$. The system (20) is subject to the initial conditions $x(0) = [0.01, 0.01]^T$ and the parameters are $\alpha = 2.1,\ \sigma = 0.2$. The deterministic version of the Brusselator exhibits unforced oscillations. Indeed, the problem exhibits a Hopf bifurcation when $\alpha = 2$ and for $\alpha > 2$ becomes a limit cycle. The stochastic version is obtained from the deterministic model by allowing the parameter α to be perturbed by noise, $\alpha \to \alpha + \sigma W_t$.

The dynamics of the two components of the Brusselator are shown in Fig. 4 for a typical trajectory. The sequence of I-adaptive stepsizes on an individual Brownian path, corresponding to a tolerance of 10^{-4}, is plotted against time in Fig. 5. In Table 2, we give the ratio of the simulation times of the adaptive algorithms to that of the PI-2 adaptive scheme, the total number of attempted, successful and rejected steps and the accuracy of the I, PI-1 and PI-2, and Brownian tree adaptive methods applied to the Brusselator system, for a range of prescribed tolerances. The performance of the fixed stepsize algorithm is also included. We observe that the error committed by using the fixed stepsize algorithm, for the same tolerance and for the same work, is up to 12 times higher than that obtained with the adaptive schemes. The ratio of rejected

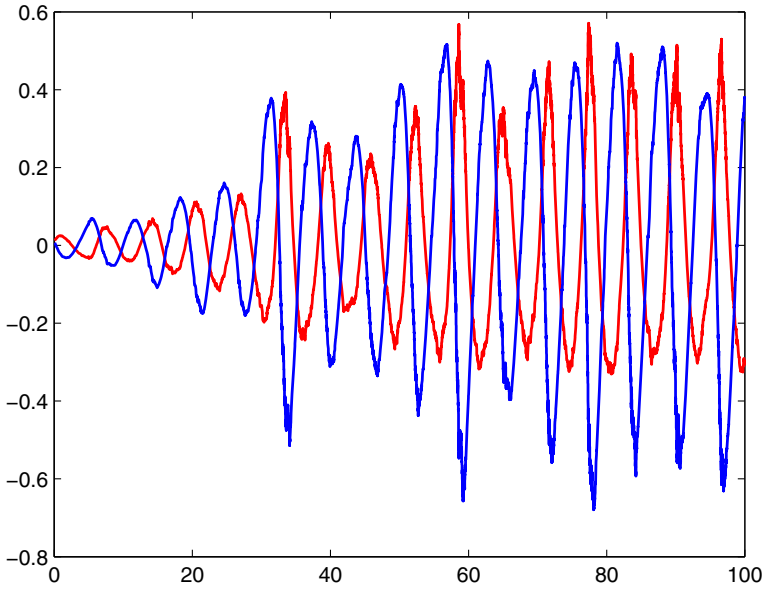

Fig. 4 The stochastic Brusselator model: for Tol=10^{-4}

to attempted steps is remarkably low, below 12% for the I-controller, below 10% for the PI-1 and PI-2 controllers, for the tolerances tried. However, for the adaptive Brownian tree algorithm, it is at least 33%. The work taken by the I, PI-1, and PI-2

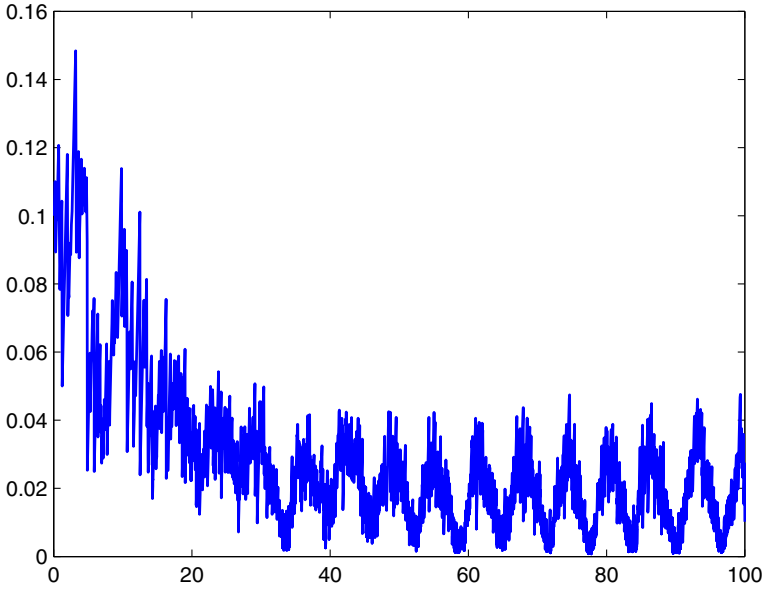

Fig. 5 The stochastic Brusselator model: stepsize vs. time for Tol=10^{-4} and the I-adaptive scheme

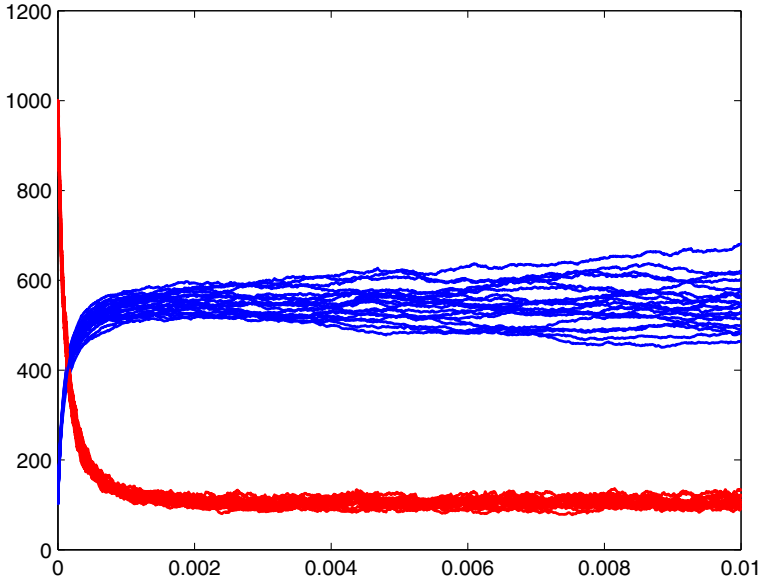

Fig. 6 The stochastic chemical reaction model: for Tol=10^{-2} Only 20 trajectories are shown

algorithms is similar. We note that the adaptive Brownian tree algorithm takes up to a remarkable 90% more steps than the PI algorithms for the same tolerance. Thus, the proposed I and PI adaptive methods perform much better than both the Brownian tree adaptive scheme and the fixed stepsize one.

4.3 Chemical reaction model

The last example we consider is a chemical reaction model [28], which was modified to include external commutative noise

$$\begin{aligned} dx_1 &= (-c_1x_1 - c_2x_1(x_1-1) + 2c_3x_2)dt + x_1\left(\alpha_1 dW_t^1 + \alpha_2 dW_t^2\right) \\ dx_2 &= \left(\frac{c_2}{2}x_1(x_1-1) - c_3x_2 - c_4x_2\right)dt + x_2\left(\beta_1 dW_t^1 + \beta_2 dW_t^2\right). \end{aligned} \tag{21}$$

The system parameters take the values $c_1 = c_2 = 10$, $c_3 = 100$ and $c_4 = 0.1$ while the stochastic coefficients are $\alpha_1 = 5$, $\alpha_2 = \beta_1 = 0.5$ and $\beta_2 = 0.001$. The initial conditions are $x_1(0) = 1000$ and $x_2(0) = 100$. The integration is performed on the interval $[0, 0.01]$.

The graph of 20 trajectories is depicted in Fig. 6, while Fig. 7 shows the plot of the evolution in time of the sequence of PI-2 adaptive stepsizes on an individual Brownian path for $Tol = 2 \cdot 10^{-2}$. The ratio of the computation times of the variable stepsize algorithms compared to the PI-2 adaptive strategy and the comparison between the fixed stepsize scheme and the adaptive schemes are presented in Table 3, for several tolerances. The error for the fixed stepsize algorithm is between approximately 80 and 110 times larger than that obtained with the proposed adaptive

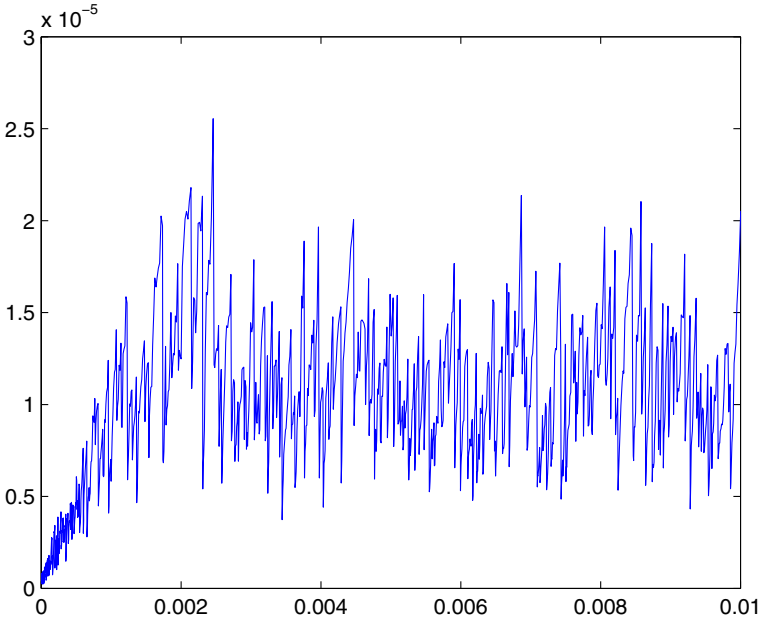

Fig. 7 The stochastic chemical reaction model: stepsize vs. time for Tol=$2 \cdot 10^{-2}$ and the PI-2 adaptive scheme

methods. For a tolerance ranging from $2 \cdot 10^{-2}$ to 10^{-3} the number of rejected steps is approximately 28% of the number of attempted steps for the I-adaptive implementation, approximately 21% for the PI-1 adaptive scheme and below 15% for the PI-2 adaptive method. For the adaptive Brownian tree method the ratio of rejected to attempted steps is between 33% and 35%. Brownian tree adaptive methods attempted more steps than the I, PI-1, and PI-2 methods, but accepted fewer steps than the PI-schemes for the same tolerance. The PI-controllers required less work than the I-controller. The results show again the advantage of using the variable stepsize methods based on PI-control over a constant step scheme or an adaptive Brownian tree method.

5 Conclusion

This paper provides a strategy for adapting the stepsize in the strong numerical solution of Itô stochastic differential equations with commutative noise. The strategy allows for a flexible stepsize selection, which is much less restrictive than the adaptive Brownian tree strategy (where only doubling and halving of the stepsize are allowed). The stepsizes may be occasionally rejected, but such rejections are guaranteed not to introduce bias in the approximated solution. The underlying numerical method used is the Milstein scheme and the local error is estimated at a low cost by extrapolation. The proposed adaptive time-stepping strategies, based on integral

and proportional-integral controllers, are tested on three interesting models arising in applications and are shown to perform significantly better than the fixed stepsize scheme and better than the adaptive Brownian tree method.

In the future, we will consider efficient and reliable adaptive schemes for stochastic differential equations with non-commutative noise and optimal initial time-step selection. In addition, future research may be done to extend the adaptive PI control strategies to the weak numerical solution of stochastic differential equations. Weak higher order stochastic Runge-Kutta schemes were developed by Komori [29], Komori & Burrage [30] and Rössler [31], and embedded stochastic Runge-Kutta methods may be used to estimate the local error.

Acknowledgments The authors would like to thank the referees for their comments which helped improve the presentation.

References

1. Kloeden, P.E., Platen, E.: Numerical solution of stochastic differential equations. Springer-Verlag, Berlin (1992)
2. Gardiner, C.W.: methods, Stochastic a handbook for the natural and social sciences. Springer, Berlin (2009)
3. van Kampen, N.G.: Stochastic processes in physics and chemistry. North-Holland, Amsterdam (2007)
4. Higham, D.J.: An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Review **43**(3), 525–546 (2001)
5. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I, 2nd revised edition. Springer, Berlin (2009)
6. Burrage, K., Burrage, P.M., Tian, T.: Numerical methods for strong solutions of stochastic differential equations: an overview. Proc. R. Soc. Lond. A **460**(2041), 373–402 (2004)
7. Burrage, K., Hancock, J., Leier, A., Nicolau, D.V.: Modelling and simulation techniques for membrane biology. Brief. Bioinform. **8**(4), 234–244 (2007)
8. Salis, H., Kaznessis, Y.N.: An equation-free probabilistic steady-state approximation: dynamic application to the stochastic simulation of biochemical reaction networks. J. Chem. Phys. **123**, 214106 (2005)
9. Salis, H., Sotiropoulos, V., Kaznessis, Y.N.: Multiscale Hy3S: Hybrid stochastic simulation for supercomputers. BMC Bioinform **7**, 93 (2006)
10. Lamba, H.: An adaptive time-stepping algorithm for stochastic differential equations. J. Comput. Appl. Math. **161**, 417–430 (2003)
11. Mauthner, S.: Step size control in the numerical solution of stochastic differential equations. J. Comput. Appl. Math. **100**, 93–109 (1998)
12. Hofmann, N., Müller-Gronbach, T., Ritter, K.: The optimal discretization of stochastic differential equations. J. Complexity **17**, 117–153 (2001)
13. Burrage, K., Burrage, P.M.: 3. SIAM J. Sci. Comput. **24**, 848–864 (2002)
14. Szepessy, A., Tempone, R., Zouraris, G.: Adaptive weak approximation of stochastic differential equations. Commun. Pure Appl. Math. **54**(10), 1169–1214 (2001)
15. Gaines, J.G., Lyons, T.J.: Variable step size control in the numerical solution of stochastic differential equations. SIAM J. Appl. Math. **57**, 1455–1484 (1997)
16. Mauthner, S.: Schrittweitensteuerung bei der numerischen Lösung stochastischer Differentialgleichungen, PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany (1999)
17. Glasserman, P.: Monte Carlo methods in financial engineering. Springer, New York (2010)
18. Söderlind, G.: Automatic control and adaptive time-stepping. Numer. Algoritm **31**, 281–310 (2002)
19. Söderlind, G.: Digital filters in adaptive time-stepping. ACM Trans. Math. Software **29**(1), 1–26 (2003)

20. Burrage, P.M., Herdiana, R., Burrage, K.: Adaptive stepsize based on control theory for stochastic differential equations. J. Comput. Appl. Math. **170**, 317–336 (2004)
21. Burrage, P.M.: Runge-Kutta methods for stochastic differential equations Ph.D. dissertation, University of Queensland (1999)
22. Giles, M.B.: Multi-level Monte Carlo path simulation. Oper. Res. **56**(3), 607–617 (2008)
23. Giles, M.B.: Improved multilevel Monte Carlo convergence using the Milstein scheme. In: Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 343–358. Springer (2008)
24. Giles, M.B.: Multilevel Monte Carlo methods. In: Monte Carlo and Quasi-Monte Carlo methods 2012, pp. 79–98. Springer (2013)
25. Beretta, E., Kuang, Y.: Modeling and analysis of a marine bacteriophage infection. Math. Biosci. **149**, 57–76 (1998)
26. Carletti, M.: Numerical solution of stochastic differential problems in the biosciences. J. Comput. Appl. Math. **185**, 422–440 (2006)
27. Carletti, M., Burrage, K., Burrage, P.: Numerical simulation of stochastic ordinary differential equations in biomathematical modelling. Math. Comput. Simulation **64**, 271–277 (2004)
28. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. **115**, 1716 (2001)
29. Komori, Y.: Weak second order stochastic Runge-Kutta methods for commutative stochastic differential equations. J. Comput. Appl. Math **203**, 57–79 (2007)
30. Komori, Y., Burrage, K.: Supplement: Efficient weak second order stochastic Runge-Kutta methods for non-commutative Stratonovich stochastic differential equations. J. Comput. Appl. Math. **235**, 5326–5329 (2011)
31. Rössler, A.: Second order Runge-Kutta methods for Itô stochastic differential equations. SIAM J. Num. Anal. **47**(3), 1713–1738 (2009)