Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 170 (2004) 317–336

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Adaptive stepsize based on control theory for stochastic differential equations

P.M. Burrage*, R. Herdiana, K. Burrage

Department of Mathematics, University of Queensland, St. Lucia, Brisbane 4072, Australia

Received 22 January 2003; received in revised form 27 January 2004

Abstract

The numerical solution of stochastic differential equations (SDEs) has been focussed recently on the development of numerical methods with good stability and order properties. These numerical implementations have been made with fixed stepsize, but there are many situations when a fixed stepsize is not appropriate. In the numerical solution of ordinary differential equations, much work has been carried out on developing robust implementation techniques using variable stepsize. It has been necessary, in the deterministic case, to consider the "best" choice for an initial stepsize, as well as developing effective strategies for stepsize control—the same, of course, must be carried out in the stochastic case.

In this paper, proportional integral (PI) control is applied to a variable stepsize implementation of an embedded pair of stochastic Runge–Kutta methods used to obtain numerical solutions of nonstiff SDEs. For stiff SDEs, the embedded pair of the balanced Milstein and balanced implicit method is implemented in variable stepsize mode using a predictive controller for the stepsize change. The extension of these stepsize controllers from a digital filter theory point of view via PI with derivative (PID) control will also be implemented. The implementations show the improvement in efficiency that can be attained when using these control theory approaches compared with the regular stepsize change strategy.

MSC: 65C20; 65C30

Keywords: Stochastic differential equations; Variable stepsize; Predictive control; PI(D) control

* Corresponding author. Tel.: +61-733656137; fax: +61-733656136.

E-mail addresses: pmb@maths.uq.edu.au (P.M. Burrage), ratna@maths.uq.edu.au (R. Herdiana), kb@maths.uq.edu.au (K. Burrage).

0377-0427/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2004.01.027

1. Introduction

In the simulation of ordinary differential equations (ODEs) by numerical integration methods, it has been evident that applying a variable stepsize strategy is more efficient than a fixed stepsize strategy. In the former strategy, smaller steps are used in regions where the solution varies considerably, whereas for slowly varying solutions larger steps are required. The superiority of adaptive stepsize for solving stochastic differential equations (SDEs) is also shown in [2,4] for explicit methods and recently in [3] for the balanced implicit (BI) method.

In a variable stepsize implementation, the selection of stepsize is crucial as it affects the local error. An estimation of the local error is calculated and compared to a specified accuracy requirement and a stepsize control algorithm then selects the next stepsize.

Various refinements for the stepsize control algorithm can be developed. Gustafsson et al. [7], Gustafsson [5] and Söderlind [13] have considered a control theory approach, and in particular we extend the proportional integral (PI) and PI with derivative control (PID) approaches to the numerical solution of nonstiff SDEs. Gustafsson [6] has shown the importance of a predictive controller when solving stiff ODEs by implicit Runge–Kutta methods, and we extend this to the stochastic case, using the balanced methods.

This paper is constructed as follows. In Section 2, we summarise the main theory for variable stepsize implementations for ODEs and then extend this to the stochastic arena. The technique of embedding is described for error estimation when applying the numerical methods in variable stepsize mode. In Section 3, we describe various elements of stepsize adaptivity based on control theory, including PI control, predictive control (PC) and PID. This is performed firstly in the deterministic case and then extended to the stochastic case. Numerical results for both nonstiff and stiff SDEs are presented in Section 4, and these demonstrate the improved results obtained with PI, PC and predictive PID control compared to results obtained using a basic stepsize change strategy. Lastly, the results of this research are summarised and suggestions for future research directions are given in Section 5.

2. Variable stepsize implementation

There are many differential equations (deterministic or stochastic) that cannot be solved efficiently in fixed stepsize mode. For example, if the solution has a turning point or sharp curve, it is often the case that a very small stepsize must be taken in order to understand the qualitative behaviour of the numerical solution in that region. Under a fixed stepsize mode of implementation, the stepsize for the entire range of integration is thus restricted to the smallest stepsize suitable for the ‘difficult’ areas. However, in variable stepsize mode, the stepsize can be made as small as necessary for such situations while being allowed to increase when integrating along the smoother parts of the numerical solution. This adaptive variable stepsize integration procedure thus results in a more efficient numerical solution than if the stepsize is fixed.

Clearly with any numerical implementation, it is desirable to minimise computational effort while maintaining a certain accuracy. This means that the truncation errors at the end of each step must be controlled, and such error estimation must be carried out with a minimum of extra work. In a variable stepsize implementation, the numerical method calculates a numerical approximation one step at a time; at the end of each step, the error in that step is estimated, and this information is used to determine a new stepsize—this stepsize is used either to advance the numerical approximation if the previous step was accepted, or it is used to repeat part of the failed step if the error was greater than the tolerance specified.

2.1. Embedding

Consider the general form of a SDE (in Stratonovich form) driven by d independent Wiener processes $W_j(t)$ (a stochastic process which is normally distributed as $N(0,t)$) given by

$$\mathrm{d}y = f(y)\,\mathrm{d}t + \sum_{j=1}^{d} g_j(y)\,\mathrm{d}W_j, \qquad y(t_0) = y_0, \;\; y \in \mathbb{R}^m. \tag{1}$$

A general class of s-stage stochastic Runge–Kutta (SRK) methods, which can attain global strong order 1.5, for solving (1) when $d = 1$ is given by

$$Y_i = y_n + h\sum_{j=1}^{s} a_{ij} f(Y_j) + J_1 \sum_{j=1}^{s} b_{ij}^{(1)} g(Y_j) + \frac{J_{10}}{h}\sum_{j=1}^{s} b_{ij}^{(2)} g(Y_j), \quad i = 1,\ldots,s,$$

$$y_{n+1} = y_n + h\sum_{j=1}^{s} \alpha_j f(Y_j) + J_1 \sum_{j=1}^{s} \gamma_j^{(1)} g(Y_j) + \frac{J_{10}}{h}\sum_{j=1}^{s} \gamma_j^{(2)} g(Y_j).$$

Here the method is characterised by the matrices A, $B^{(1)}$ and $B^{(2)}$ and by the update vectors α, $\gamma^{(1)}$ and $\gamma^{(2)}$. If A, $B^{(1)}$ and $B^{(2)}$ are all lower triangular then the method is explicit. The Stratonovich integrals

$$J_1 = \int_{t_n}^{t_{n+1}} \circ\,\mathrm{d}W(s) = W(t_{n+1}) - W(t_n) \sim N(0,h),$$

$$J_{10} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} \circ\,\mathrm{d}W(s)\,\mathrm{d}s$$

are simulated at each integration step. It is easily shown that if u and v are two independent $N(0,1)$ random variables then J_1 and J_{10}/h are efficiently calculated as

$$J_1 = \sqrt{h}u,$$

$$\frac{J_{10}}{h} = \frac{\sqrt{h}}{2}\left(u + \frac{v}{\sqrt{3}}\right).$$

There are a number of ways that the error at the end of each step can be estimated. The cheapest approach is to use the technique of embedding where the only extra calculations required are for the update stage of the embedded method. In this paper, the 2-stage SRK method R2 (of global strong order 1) given by

$$A=\begin{pmatrix}0 & 0\\ \frac{2}{3} & 0\end{pmatrix},\quad B^{(1)}=\begin{pmatrix}0 & 0\\ \frac{2}{3} & 0\end{pmatrix},\quad B^{(2)}=\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix},$$

$$\alpha^\top=\left(\frac{1}{4},\frac{3}{4}\right),\quad \gamma^{(1)\top}=\left(\frac{1}{4},\frac{3}{4}\right),\quad \gamma^{(2)\top}=(0,0)$$

is embedded in the 4-stage SRK E1 (of local strong order 1.5)

$$A=\begin{pmatrix}0 & 0 & 0 & 0\\ \frac{2}{3} & 0 & 0 & 0\\ \frac{3}{2} & -\frac{1}{3} & 0 & 0\\ \frac{7}{6} & 0 & 0 & 0\end{pmatrix},\quad \begin{aligned}\alpha^\top &= \left(\frac{1}{4},\frac{3}{4},-\frac{3}{4},\frac{3}{4}\right),\\ \gamma^{(1)\top} &= \left(-\frac{1}{2},\frac{3}{2},-\frac{3}{4},\frac{3}{4}\right),\\ \gamma^{(2)\top} &= \left(\frac{3}{2},-\frac{3}{2},0,0\right),\end{aligned}$$

$$B^{(1)}=\begin{pmatrix}0 & 0 & 0 & 0\\ \frac{2}{3} & 0 & 0 & 0\\ \frac{1}{2} & \frac{1}{6} & 0 & 0\\ -\frac{1}{2} & 0 & \frac{1}{2} & 0\end{pmatrix},\quad B^{(2)}=\begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ -\frac{2}{3} & 0 & 0 & 0\\ \frac{1}{6} & \frac{1}{2} & 0 & 0\end{pmatrix}.$$

Consequently, a reasonable error estimate is provided by

$$y_{n+1}-\hat{y}_{n+1},$$

where $\hat{y}_{n+1}$ and y_{n+1} are the two numerical solutions obtained at the end of the step using R2 (the embedded method) and E1 (the integration method), respectively. Then an error estimate of order $q+\frac{1}{2}$ is given by

$$\text{error}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{y_{n+1,i}-\hat{y}_{n+1,i}}{\text{tol}_i}\right)^2},$$

where m is the dimension of the SDE system, tol_i is the tolerance permitted for the ith component, and q is either $\hat{p}$ or p, $\hat{p}$ and p being the order, respectively, of the embedded and integration methods. In this paper, the interpretation is that the calculated error is an approximation to the error in the higher order method rather than the lower order method, and so $q=p$. An optimal stepsize (see [9], for example, in the deterministic case) is determined by comparing this error to 1:

$$\text{error}\approx Ch^{q+1/2},$$

$$1\approx Ch_{\text{opt}}^{q+1/2}$$

so that $h_{\text{opt}} = h(1/\text{error})^{1/(q+1/2)}$. For the (R2,E1)-embedded pair of methods,

$$h_{\text{opt}} = h\left(\frac{1}{\text{error}}\right)^{1/2}.$$

To avoid undue oscillatory behaviour, this optimal stepsize is often reduced by a safety factor (e.g. fac = 0.8), and there can also be safeguards built in to allow a minimal (facmin) and maximal (facmax) stepsize scaling factor (which may depend on the differential equation being solved). Thus Hairer et al. [9] (in the deterministic setting) use the stepsize control strategy defined by

$$h_{\text{new}} = h \min(\text{facmax}, \max(\text{facmin}, \text{fac}\,(1/\text{error})^{1/(q+1)})).$$

For SDEs, note that the power $1/(q+1)$ is replaced by $1/(q+1/2)$ to reflect the fact that, in the stochastic setting, orders increase by $\frac{1}{2}$ rather than 1.

The explicit SRKs described above can be used to solve nonstiff SDEs, but when the SDE system is stiff, an implicit method must be used.

2.2. The BI and balanced Milstein (BM) methods

As in the deterministic case, it is also necessary to use implicit schemes to solve stiff SDEs. An illustration of a stiff SDE in Itô form is given below. Consider the linear Itô version of (1) with $d = 1$ and $f(y) = G_0 y$, $g(y) = G_1 y$, where, for example,

$$G_0 = \begin{pmatrix} -a & a \\ a & -a \end{pmatrix} \quad \text{and } G_1 = \begin{pmatrix} b & 0 \\ 0 & b \end{pmatrix}. \tag{2}$$

Since the matrices G_0 and G_1 commute it can be shown that the solution is given by the Magnus formula (see [1]), namely

$$y(t) = \mathrm{e}^{((G_0 - \frac{1}{2}G_1^2)t + G_1 W(t))} y_0. \tag{3}$$

It is easily seen that the eigenvalues of $G_0 - \frac{1}{2}G_1^2$ are

$$-\frac{b^2}{2}, \quad -\frac{(4a+b^2)}{2}$$

and so the solution is asymptotically stable for $4a + b^2 > 0$. For this problem the solution can be stiff if a is large or b is large and in either case an implicit method is required.

Milstein et al. [12] derived a fully implicit numerical method for SDEs, called the BI method of strong order $\frac{1}{2}$, which is implicit in both the drift term and the stochastic term.

Consider (1) in Itô form; then the BI scheme (see [12]) is given by

$$y_{n+1} = y_n + h\, f(y_n) + \sum_{j=1}^{d} J_j g_j(y_n) + C_n(y_n - y_{n+1}),$$

where $h = t_{n+1} - t_n$, $J_j = W_j(t_{n+1}) - W_j(t_n)$,

$$C_n = hc_0(y_n) + \sum_{j=1}^{m} |J_j| c_j(y_n), \quad n = 0, 1, \ldots, N-1$$

and c_j, $j = 0, \ldots, m$ represent $(m \times m)$ matrix-valued functions.

For any sequence of real numbers γ_i satisfying $\gamma_0 \in [0, \bar{\gamma}]$, $\gamma_1 \geqslant 0, \ldots, \gamma_m \geqslant 0$ and $\bar{\gamma} \geqslant h$, and for all stepsizes h considered, the matrix

$$M(t,x) = I + \sum_{j=0}^{m} \gamma_j \ c_j(t,x)$$

should be nonsingular and satisfy

$$|(M(t,x))^{-1}| \leqslant K < \infty.$$

Some care has to be taken with implementing adaptive timestepping. Gaines and Lyons [4] showed with a particular timestepping mechanism that there is no guarantee of convergence to the correct solution, in variable step mode, for methods of order less than 1. On the other hand, in a recent paper by Szepessy et al. [15], they show correct convergence of the explicit Euler method which has strong order $\frac{1}{2}$ with two different types of adaptive timestepping mechanisms, based on a stochastic approach and a deterministic approach.

Inspired by the construction of the BI method, Burrage and Herdiana [3] introduced the BM method of strong order 1 and so an embedded pair (BM,BI) method of order $1(\frac{1}{2})$ is possible. The BM method, for the case $d = 1$, is of the form

$$y_{n+1} = y_n + (I + C_n)^{-1}(hf(y_n) + \Delta W_n g(y_n) + \tfrac{1}{2}[(\Delta W_n^2 - h)g'g(y_n)])$$

with

$$C_n = hc_0 + (\Delta W_n)^2 c_1, \tag{4}$$

and we may take the parameters $c_0 = -J_f$ and $c_1 = J_{g'g}$, where J_f and J_g are the Jacobians of $f(y_n)$ and $g(y_n)$, respectively. Also to avoid unbounded solutions it is assumed that (4) satisfies the condition

$$|(I + C_n)^{-1}| \leqslant K < \infty.$$

Numerical tests with variable stepsize in [3] show that, with the embedded method, the approximated solution converges to the correct solution.

2.3. Brownian paths

For a variable stepsize integration of SDEs, it is necessary to remain on the same Brownian path. This can be carried out by fixing the path with some stepsize h_{fix}, and then generating the appropriate random samples on each subinterval so that the eventual path followed balances with the path fixed at the start of the integration. Integration results can then be compared when the integration is repeated with a different initial stepsize, for example. Gaines and Lyons [4] created a Brownian tree that balanced with the initial fixed Brownian path. The structure of the Brownian tree was such that the stepsize could only be halved, doubled or remain unchanged at each step, and this can be rather restrictive. Mauthner [11] described an approach that allowed a completely flexible choice of stepsize, but then worked with the halving/doubling paradigm due to the ease of storing the simulated values in a binary tree as well as the reduced cost associated with their simulation.

However, in this present paper (as in [2]), the completely flexible stepsize control is the preferred approach due to its lack of restrictions. New random samples are generated as required as the integration progresses, and the samples on the subinterval $[t_1,t_2]\cup[t_2,t_3]$ satisfy

$$J_1(t_{1,}t_3)=J_1(t_1,t_2)+J_1(t_2,t_3),$$

$$J_{10}(t_1,t_3)=J_{10}(t_1,t_2)+J_{10}(t_2,t_3)+(t_3-t_2)J_1(t_1,t_2).$$

For the implementation of the (BM, BI) method, just the simple random variable $I_1=\int_{t_n}^{t_{n+1}}\mathrm{d}W(s)$ is required. (Note that the level-one Itô and Stratonovich stochastic integrals I_1 and J_1 are equivalent, but we use the notation I_1 when describing the implementation of the BM and BI methods to emphasise that this method is applied to the Itô form of the SDE.) To maintain the correct Brownian path, the implementation of the balanced methods proceeds as follows.

First generate a fixed Brownian path with fixed stepsize h and denote $i_1=\int_{t_0}^{t_0+h}\mathrm{d}W(s)$. Given the condition i_1 we need to simulate I_1 on the two subintervals $[t_0,t_0+h_1]$ and $[t_0+h_1,t_0+h]$. This involves using a new random variable Z as follows. Letting

$$I_1(t_0,t_0+h_1)=\frac{h_1}{h}\,i_1+Z,\qquad I_1(t_0+h_1,t_0+h)=\frac{h_2}{h}\,i_1-Z$$

(where $h_2=h-h_1$), then it is clear that $I_1(t_0,t_0+h_1)+I_1(t_0+h_1,t_0+h)=i_1$ as required; also it is necessary that

$$h_1:=E[(I_1(t_0,t_0+h_1))^2]=\frac{h_1^2}{h^2}E[i_1^2]+E[Z^2],$$

$$h_2:=E[(I_1(t_0+h_1,t_0+h))^2]=\frac{h_2^2}{h^2}E[i_1^2]+E[Z^2]$$

so that

$$E[Z^2]=\frac{h^2-h_1^2-h_2^2}{2h}=\frac{h_1h_2}{h}.$$

Similarly the requirement that $E[I_1(t_0,t_0+h_1)I_1(t_0+h_1,t_0+h)]=0$ also leads to $E[Z^2]=h_1h_2/h$. Therefore Z can be determined by sampling from $N(0,h_1h_2/h)$, and this will ensure that the I_1 samples on the subintervals have the correct distribution and are on the correct path.

For the balanced methods, simulation of the random variables is easier to implement than for higher order methods which may require, for example, a pair of random variables (J_1,J_{10}), hence increasing the computational complexity.

An algorithm for the stepsize progression is found in [2,3].

2.4. Implementation concerns

In the deterministic case various issues for effective variable stepsize implementations arise. These include selecting an appropriate initial stepsize h_0 (although in many cases an inappropriate choice of initial stepsize is not too disastrous); and ensuring tolerance proportionality, so that the error reduces at the same rate as a decreasing tolerance level. In the deterministic case there have been some developments to improve tolerance proportionality—in particular, the technique of Defect Control (see [10], for example) has helped improve this situation.

Another issue concerns possible oscillatory behaviour of stepsizes even though damping factors can be included in the stepsize selection process to minimise this. It is this behaviour that has motivated the development of PI control and PID control (which has a third component proportional to the derivative of the control error) for stepsize change strategies. In the next section, we will focus on control theory ideas, to determine whether or not these approaches produce smoother stepsize changes and reduce computational costs in the stochastic setting as they do in the deterministic setting.

3. Stepsize adaptivity based on control theory

Variable stepsize implementations have advantages over fixed stepsize implementations, as the stepsize used for the entire integration range is not restricted to the smallest stepsize necessary to give acceptable results over the range where the SDE may be stiff, for example. However one drawback with the standard variable stepsize implementation is the lack of a smooth progression of stepsizes (although the stepsizes will naturally vary based on the Wiener path). To a certain extent this lack of smoothness can be improved by including damping factors in the stepsize selection process. An alternative approach, however, is the control theory approach (see [7,5,8,13], for example). The benefits of this control theory approach are that a smoother sequence of stepsizes leads to fewer step rejections and hence less computational effort, and that the estimated error is also smoother and closer to the specified error tolerance.

For deterministic problems, the basic standard stepsize control (for a method of order p) is

$$h_{n+1} = \left(\frac{\theta\epsilon}{r_n}\right)^{1/k} h_n, \tag{5}$$

where $k = p + 1$ if local error per step is being used, r_n is the local error (normalised by dividing by the stepsize if error per unit step is required, in which case $k = p$), ϵ is the required tolerance, and θ is the safety factor between 0 and 1 (often set to 0.8 or 0.9). The step just completed is accepted if $r_n \leqslant \theta\epsilon$, and the new stepsize is chosen to maximise the error subject to the tolerance requirements.

For stepsize control using a Control Theory approach (as described in [7,5,8,13], for example), we can take logarithms of (5) to express the relationship between h_{n+1} and h_n:

$$\log h_{n+1} = \log h_n + \frac{1}{k}(\log(\theta\epsilon) - \log(r_n)). \tag{6}$$

The factor $1/k$ is called the integral gain.

Söderlind [13] analyses the closed-loop dynamics for both the controller and the controlled process. The factor $1/k$ in (6) is replaced by a parameter k_I representing ‘integral gain’, and, assuming asymptotic behaviour of r_n, namely

$$r_n = \varphi_n h_n^k, \tag{7}$$

the closed-loop dynamics can be expressed as

$$\log h_{n+1} = (1 - kk_\mathrm{I})\log h_n + k_\mathrm{I}(\log(\theta\epsilon) - \log(\varphi_n)).$$

The dynamics of this is governed by the root q of the characteristic equation ($q = 1 - kk_{\mathrm{I}}$), and so the general integral controller can be written

$$h_{n+1} = \left(\frac{\theta\epsilon}{r_n}\right)^{k_{\mathrm{I}}} h_n. \tag{8}$$

To obtain a PI controller (P for proportional, I for integral), the control $\log h_n$ is determined by

$$\log h_n = \log h_0 + k_{\mathrm{I}} \sum_{i=1}^{n} (\log(\theta\epsilon) - \log(r_i)) + k_{\mathrm{P}}(\log(\theta\epsilon) - \log(r_n));$$

now the formula includes a term proportional to the control error and indeed the parameter k_{P} is called the proportional gain. The PI controller (see [13] for details) can be expressed as

$$h_{n+1} = \left(\frac{\theta\epsilon}{r_n}\right)^{k_{\mathrm{I}}+k_{\mathrm{P}}} \left(\frac{\theta\epsilon}{r_{n-1}}\right)^{-k_{\mathrm{P}}} h_n \tag{9}$$

and the closed-loop dynamics of this control process can be described by

$$\begin{aligned}\log h_{n+1} = {} & (1 - kk_{\mathrm{I}} - kk_{\mathrm{P}})\log h_n + kk_{\mathrm{P}} \log h_{n-1} \\ & + k_{\mathrm{I}}(\log(\theta\epsilon) - \log(\varphi_n)) + k_{\mathrm{P}}(\log(\varphi_{n-1}) - \log(\varphi_n)),\end{aligned}$$

which has a second-order characteristic equation

$$q^2 - (1 - kk_{\mathrm{I}} - kk_{\mathrm{P}})q - kk_{\mathrm{P}} = 0. \tag{10}$$

There has been both theoretical and numerical experimentation to find good choices of k_{I} and k_{P}. For nonstiff ODEs, Gustafsson [5] analyses the control process models. The models must cover the situation when the error behaves asymptotically as well as when the stepsize may be limited by numerical stability. Using this analysis, Gustafsson [5] describes a methodology for choosing the k_{I} and k_{P} parameters, and concludes that

$$k_{\mathrm{I}} = \frac{0.3}{k}, \quad k_{\mathrm{P}} = \frac{0.4}{k}$$

is a good starting point for these parameters though the values may need refining depending on the particular integration method being used. Söderlind [13] suggests a choice from

$$\{(kk_{\mathrm{I}}, kk_{\mathrm{P}}) : kk_{\mathrm{I}} + kk_{\mathrm{P}} \leqslant 0.8, \quad kk_{\mathrm{I}} \geqslant 0.3, \quad kk_{\mathrm{P}} \geqslant 0.1\}.$$

In the case of stiff problems, instead of assuming $\varphi_n \approx \varphi_{n-1}$, Gustafsson [6] assumes that $\log\varphi$ is a linear function, that is $(\varphi_{n+1})/\varphi_n \approx \varphi_n/(\varphi_{n-1})$ or

$$\log\varphi_n = \log\varphi_{n-1} + \Delta\log\varphi_{n-1},$$

where $\Delta\log\varphi_{n-1} = \log\varphi_{n-1} - \log\varphi_{n-2}$.

By using an observer (see [6]) and observer gain $K = [k_1 k_2]^\top$ the prediction on φ can be written as

$$\log\varphi_n = \frac{(k_1 + k_2)q^2 - k_1 q}{q^2 + (-2 + k_1 + k_2)q + 1 - k_1} \log\varphi_{n-1}, \tag{11}$$

where q is the shift operator.

Inserting $r_n = \epsilon$ into (7) yields

$$\log h_n = k^{-1}(\log \epsilon - \log \varphi_n),$$

then substituting (11) and $\log \varphi_{n-1} = \log r_n - kq^{-1} \log h_n$ leads to the predictive controller:

$$\log h_n = \frac{1}{k} \frac{(k_1 + k_2)q^2 - k_1 q}{(q-1)^2} (\log \epsilon - \log r_n),$$

which can be rewritten as

$$\frac{h_{n+1}}{h_n} = \frac{h_n}{h_{n-1}} \left(\frac{\epsilon}{r_n}\right)^{(k_1+k_2)/k} \left(\frac{\epsilon}{r_{n-1}}\right)^{-k_1/k}.$$

The system is stable if the roots of the denominator

$$q^2 + (2 - k_1 + k_2)q + 1 - k_1 = 0$$

are inside the unit circle. Gustafsson [6] and Söderlind [13] suggest that taking $k_1 = 1$ and $k_2 = 1$ gives good performance.

Further extensions to the two control strategies discussed above are recently given in [14], where the time-stepping methods apply PID controllers. Now in general the *closed-loop dynamics* for $\log r$ and $\log h$ may be written in the form

$$\log r = R_\varphi(q) \log \varphi, \quad \log h = H_\varphi(q) \log \varphi \equiv -C(q) \log r,$$

where, for example, in the case of PI control $C(q) = k_\mathrm{I} q/(q-1) + k_\mathrm{P}$.

Here $H_\varphi(q)$ and $R_\varphi(q)$ are the closed-loop transfer functions which represent *digital filters* for stepsize and error, respectively. So the stepsize $\log h$ and error $\log r$ are results of digital signal processes of the external disturbance $\log \varphi$. The smoothness of the stepsize sequence is highly influenced by the spectral properties of the transfer map $H_\varphi(q)$.

The main idea in order to obtain smooth stepsize sequence is to make $H_\varphi(q) = 0$ at $q = -1$; this will reduce or suppress any high frequencies contained in $\log \varphi$. While $R_\varphi(q)$ is used in defining the order of adaptivity, the magnitude of the poles determines how fast $\log \varepsilon - \log r \to 0$. This leads to the following definitions given in [14].

Definition 3.1. Let all poles of $H_\varphi(q)$ be strictly inside the unit circle; then the stepsize filter order at $q = -1$ is p_F if $\|H_\varphi(q)\| = \mathrm{O}(\|q+1\|^{p_\mathrm{F}})$ as $q \to -1$.

Definition 3.2. Let all poles of $R_\varphi(q)$ be strictly inside the unit circle; then the controller's order of adaptivity is p_A if $\|R_\varphi(q)\| = \mathrm{O}(\|q-1\|^{p_\mathrm{A}})$ as $q \to 1$.

Söderlind [14] constructs a third-order control dynamics called the PID control (D for derivative) whose control transfer function is given by

$$C^{\mathrm{PID}}(q) = q^{-1} \left(k_\mathrm{I} \frac{q}{q-1} + k_\mathrm{P} + k_\mathrm{D} \frac{q-1}{q} \right),$$

where $k_\mathrm{I}, k_\mathrm{P}, k_\mathrm{D}$ are the integral, proportional and derivative gains, respectively. The general form of the PID controller is

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{k_\mathrm{I}+k_\mathrm{P}+k_\mathrm{D}} \left(\frac{\varepsilon}{r_{n-1}}\right)^{-(k_\mathrm{P}+2k_\mathrm{D})} \left(\frac{\varepsilon}{r_{n-2}}\right)^{k_\mathrm{D}} h_n. \tag{12}$$

The stepsize transfer function is then derived as

$$-kH_\varphi(q) = \frac{(kk_{\mathrm{I}} + kk_{\mathrm{P}} + kk_{\mathrm{D}})q^2 - (kk_{\mathrm{P}} + 2kk_{\mathrm{D}})q + kk_{\mathrm{D}}}{q^3 - (1 - kk_{\mathrm{I}} - kk_{\mathrm{P}} - kk_{\mathrm{D}})q^2 - (kk_{\mathrm{P}} + 2kk_{\mathrm{D}})q + kk_{\mathrm{D}}},$$

whereas $R_\varphi(q)$ has the same denominator with the numerator being $q^2(q-1)$. Assuming that the poles of $H_\varphi(q)$ are strictly inside the unit circle, then to have an order 1 stepsize filter at $q=-1$, the parameters must satisfy

$$k_{\mathrm{D}} = -\frac{k_{\mathrm{I}}}{4} - \frac{k_{\mathrm{P}}}{2}$$

and for order 2

$$k_{\mathrm{D}} = \frac{k_{\mathrm{I}}}{4} = -\frac{k_{\mathrm{P}}}{4}. \tag{13}$$

Let the notation *Hxyz* refer to the family of PID controllers with dynamic order x, adaptive order y and stepsize filter order z. A H312 PID controller will depend only on one parameter gain k_{I} due to condition (13), so that

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{k_{\mathrm{I}}/4} \left(\frac{\varepsilon}{r_{n-1}}\right)^{k_{\mathrm{I}}/2} \left(\frac{\varepsilon}{r_{n-2}}\right)^{k_{\mathrm{I}}/4} h_n. \tag{14}$$

Study on the roots (poles) of the denominator of $H_\varphi(q)$ for H312 in [14] leads to the suggestion of choosing the value $kk_{\mathrm{I}} = \frac{2}{9}$. Numerical results for deterministic problems show how the irregularity of the stepsize is smoothed out and replaced by significantly smoother sequences.

For a stiff system Söderlind [14] suggests using a predictive PID controller, with general form

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{k_{\mathrm{I}}+k_{\mathrm{P}}+k_{\mathrm{D}}} \left(\frac{\varepsilon}{r_{n-1}}\right)^{-(k_{\mathrm{P}}+2k_{\mathrm{D}})} \left(\frac{\varepsilon}{r_{n-2}}\right)^{k_{\mathrm{D}}} \frac{h_n}{h_{n-1}} h_n.$$

The predictive PID controllers do not have second-order filters at $q=-1$, so only order 1 stable filters are obtained. A H321 predictive PID controller is given by

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{3k_{\mathrm{I}}/4+k_{\mathrm{P}}/2} \left(\frac{\varepsilon}{r_{n-1}}\right)^{k_{\mathrm{I}}/2} \left(\frac{\varepsilon}{r_{n-2}}\right)^{-(k_{\mathrm{I}}/4+k_{\mathrm{P}}/2)} \frac{h_n}{h_{n-1}} h_n \tag{15}$$

and Söderlind [14] recommends the parameter values $kk_{\mathrm{I}} = 0.1$ and $kk_{\mathrm{P}} = 0.45$.

Another interesting controller is the H211 controller which has $k_{\mathrm{D}} = 0$ and $k_{\mathrm{P}} = -k_{\mathrm{I}}/2$, and can be written as

$$h_{n+1} = \left(\frac{\epsilon}{r_n}\right)^{k_{\mathrm{I}}/2} \left(\frac{\epsilon}{r_{n-1}}\right)^{k_{\mathrm{I}}/2} \frac{h_n}{h_{n-1}} h_n. \tag{16}$$

A variant of this (see [14]) is the one-parameter family of H211b given by

$$h_{n+1} = \left(\frac{\epsilon}{r_n}\right)^{1/(bk)} \left(\frac{\epsilon}{r_{n-1}}\right)^{1/(bk)} \left(\frac{h_n}{h_{n-1}}\right)^{-1/b} h_n. \tag{17}$$

Söderlind suggests that $b \in [2,8]$ with a recommended value of $b=4$, which gives increased smoothing properties.

In this paper we now extend these concepts of control theory as applied to ODEs to the numerical solution of SDEs. In the nonstiff case, we use PI Control via (9) to determine the new stepsize, and have chosen suitable values for k_{I} and k_{P} by numerical experimentation. Numerical testing suggested that the choices $kk_{\mathrm{I}} = 0.3, kk_{\mathrm{P}} = 0.1$ would give improvement over standard stepsize control in many situations, when using the SRK embedded pair of (R2,E1). Also the parameter pair $kk_{\mathrm{I}} = 0.125, kk_{\mathrm{P}} = 0.095$ gave reasonable results.

We also extend the analysis to consist of a number of other approaches based on predictive PID Control including H312 (with $kk_{\mathrm{I}} = 0.3$) and H321 (with $kk_{\mathrm{I}} = 0.3$, $kk_{\mathrm{P}} = 0.45$), along with H211b (with $b = 4$) and the purely integral controller (8) (with $kk_{\mathrm{I}} = 0.3$). These parameter choices were made after analysing numerical results on a number of problems averaged over 100 trajectories, to understand the trend of the solution.

The reason for choosing these controllers is that in the case of the adaptive simulation of SDEs the rationale for choosing a particular control is different to the situation occurring in the deterministic case. In the deterministic case the PI controllers and the second-order adaptive predictive controllers are constructed with the aim of getting improved stability in combination with methods having bounded stability regions and for coping with smooth stiff problems.

In the case of SDEs, we would like the stepsize control to extract the deterministic trend and disregard the noise. Thus it would be appropriate to use controllers that implement digital filters. In addition, since second-order adaptive predictive controllers extract linear trends, which can be sensitive to noise, these may be less effective in the presence of significant noise. Finally, since noise is mollified by using a low integral gain it is appropriate to use a controller with a low integral gain, say $kk_{\mathrm{I}} \leqslant 0.3$ rather than the conventional controller $kk_{\mathrm{I}} = 1$.

The numerical results presented in Section 4 are for a single trajectory and confirm the results of the parameter-search. Even in the deterministic case, the choice of tolerance level can have an effect on the robustness of the selected control parameters, but in the stochastic case, both the tolerance level and the intensity of noise in the SDE can play a role in how to choose the control parameters. The numerical results given in Section 4 demonstrate that, in the stochastic case also, PI and PID control can be very effective in terms of reducing computational effort and in 'smoothing' the stepsizes used.

4. Numerical results

In this section a number of SDEs are solved numerically to demonstrate the advantages of using PI control (using the embedded (R2,E1) method), and a predictive or a predictive PID controller (using the embedded (BM,BI) method) over a regular stepsize change strategy. In order to ensure that the integration remains on the same Brownian path (for both the nonPI and PI control techniques) the path is fixed with a steplength of 1. Subsequent subdivisions of the interval will then balance with the specified Brownian path (see [2] for more details on this, for example).

Example 4.1. This is a chemical kinetics problem (see [7, Problem 6]) which has been adapted to include stochasticity:

$$\begin{aligned}
\mathrm{d}y_1 &= (-(p_1 + y_3)y_1 + p_2 y_2)\,\mathrm{d}t + (-\beta_1 y_1 + \beta_2 y_2) \circ \mathrm{d}W,\\
\mathrm{d}y_2 &= p_3(y_1 - y_2)\,\mathrm{d}t + \beta_3(y_1 - y_2) \circ \mathrm{d}W,\\
\mathrm{d}y_3 &= p_4 y_1\,\mathrm{d}t + \beta_4 y_1 \circ \mathrm{d}W,
\end{aligned}$$

Table 1
Numerical results for Example 4.1

Stochasticity	Tol	No PI		PI set 1		PI set 2	
		Tried	Taken	Tried	Taken	Tried	Taken
$\beta_i = 0.1$	10^{-4}	2558	1835	2157	1495	2152	1606
	10^{-3}	603	447	529	374	509	395
	10^{-2}	218	157	202	144	194	147
$\beta_i = 0.5$	10^{-3}	1555	1116	1295	904	1246	945
	10^{-2}	404	294	350	246	342	256
$(1.0, 0.5, 0.5, 0.5)$	10^{-3}	3100	2214	2608	1789	2556	1769
	10^{-2}	776	557	662	453	648	446

where $(y_1(0), y_2(0), y_3(0))^\top = (1, 1, 0)^\top$ and $t \in [0, 5]$. The deterministic parameters are

$$p_1 = 55, \quad p_2 = 65, \quad p_3 = 0.0785, \quad p_4 = 0.1$$

and the stochasticity was firstly $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.1$, secondly $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.5$, and thirdly $\beta_1 = 1.0, \beta_2 = \beta_3 = \beta_4 = 0.5$. The initial stepsize is $h_0 = 0.02$.

For this example, two sets of PI control parameters are used to contrast the improvement of PI control over regular stepsize control. The values chosen for these parameters are $(kk_{\mathrm{I}}, kk_{\mathrm{P}}) = (0.3, 0.1)$—set 1, and $(0.125, 0.095)$—set 2. For these two sets of parameters, the characteristic equation (10) has roots $0.7359, -0.1359$ and $0.8871, -0.1071$, respectively, giving rise to ratios $\|q_1/q_2\|$ of 5.4153 and 8.2835, respectively. Table 1 presents the results of the integration, detailing attempted and successful steps.

It can be seen that PI control yields an improvement over regular stepsize control of approximately 15% (although the case with low stochasticity 0.1 with lax tolerance of 10^{-2} produces only an 8% improvement when using PI control).

For the case $\beta_i = 0.1$ (with tolerance 10^{-2} and $(kk_{\mathrm{I}}, kk_{\mathrm{P}}) = (0.3, 0.1)$), Fig. 1 shows a graph of work done in terms of number of steps versus time with approximately a 15% improvement if PI control is used (dashed line). Fig. 2 shows that the progression of successful stepsizes under PI control is smoother than that under the regular stepsize control strategy.

Example 4.2. This is a two-dimensional linear multiplicative Stratonovich noise SDE given by

$$\mathrm{d}y = G_0 y \, \mathrm{d}t + G_1 y \circ \mathrm{d}W, \quad y(0) = y_0,$$

where

$$G_0 = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}, \quad G_1 = \begin{pmatrix} 0 & -b \\ b & 0 \end{pmatrix}, \quad a < 0.$$

The exact solution to this is known since G_0 and G_1 commute, and is given by

$$y(t) = \exp(G_0 t + G_1 W(t)) y_0.$$

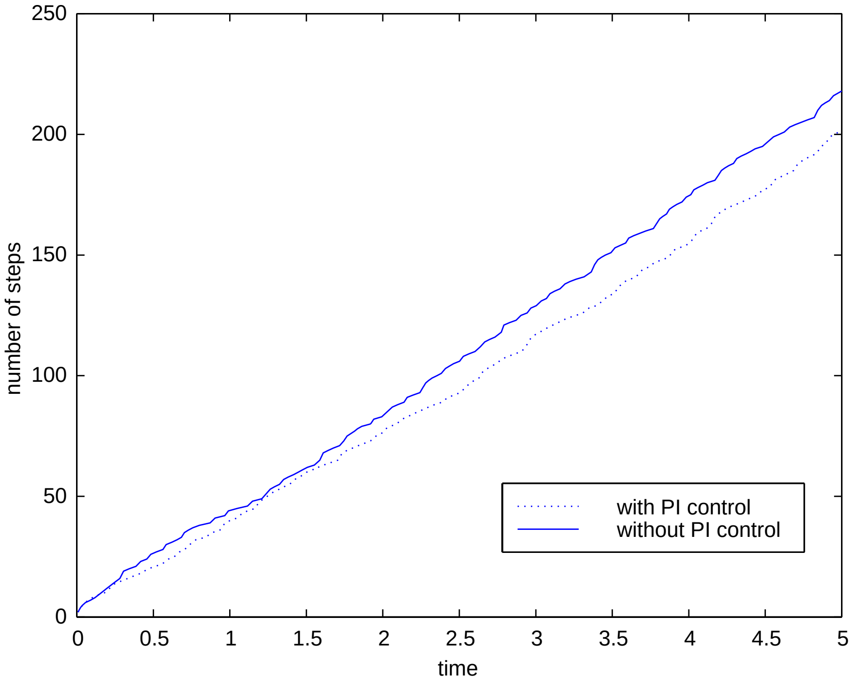

Fig. 1. Steps versus time—for Example 4.1.

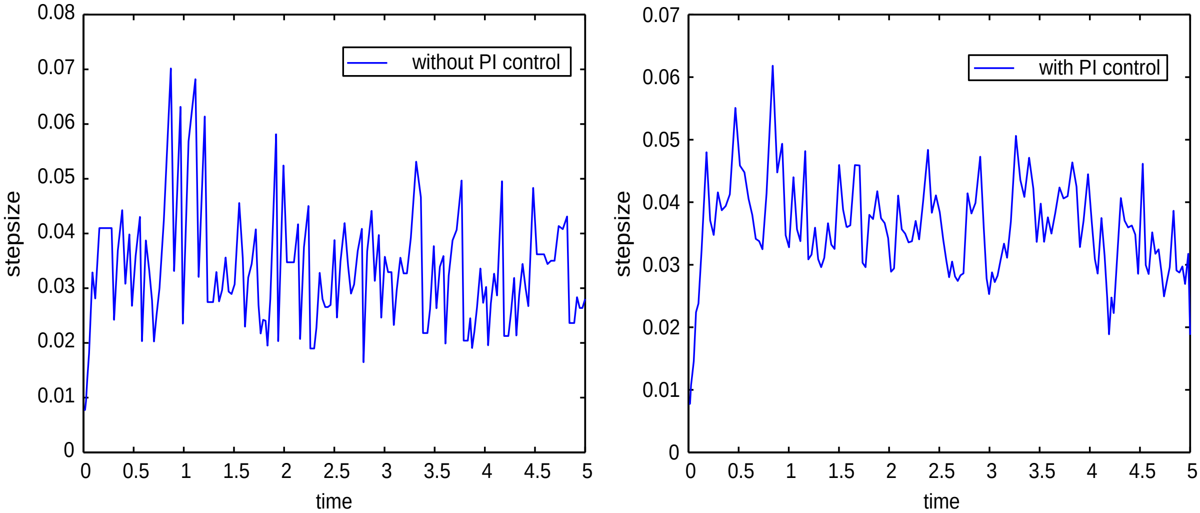

Fig. 2. Successful stepsizes versus time—for Example 4.1.

For this integration, $y_0 = (1,1)^\top$ and $t \in [0,2]$ with $a = -1$ or -2, and $b = 0.1, 0.5$ and 1.0 to represent the intensity of the noise. After numerical experimentation, again it turned out that values of $(kk_\mathrm{I}, kk_\mathrm{P})$ near to $(0.3, 0.1)$ and $(0.125, 0.095)$ produced reasonable results when compared with the regular stepsize control strategy, as presented in Table 2.

Table 2
Numerical results for Example 4.2

	Tol	No PI Tried	Taken	PI set 1 Tried	Taken	PI set 2 Tried	Taken
$a=-1,\ b=0.1$	10^{-6}	252	187	216	157	205	164
	10^{-5}	81	66	72	61	74	65
$a=-1,\ b=0.5$	10^{-5}	441	311	313	205	365	249
	10^{-4}	136	95	109	72	105	74
$a=-2,\ b=0.5$	10^{-5}	364	258	306	203	300	219
$a=-2,\ b=1.0$	10^{-5}	919	639	838	528	836	572
	10^{-4}	277	196	218	136	212	152

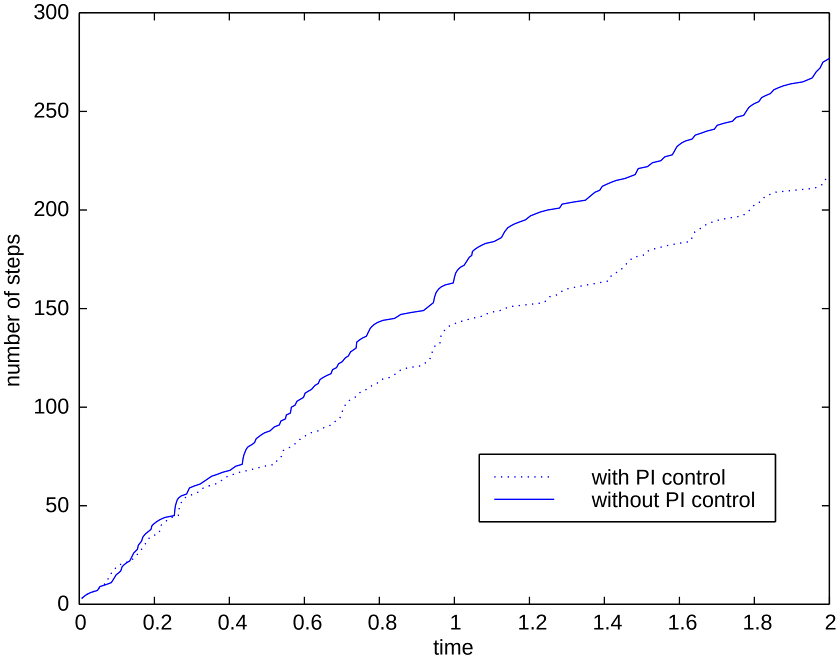

Fig. 3. Steps versus time—for Example 4.2.

In Fig. 3, the results with $a = -2,\ b = 1.0$ are presented graphically for the Without PI and the PI cases, for a tolerance of 10^{-4} and with $(kk_{\rm I}, kk_{\rm P}) = (0.3, 0.1)$. There is clearly a substantial improvement in terms of efficiency. Fig. 4 shows how the smoothness and size of successful stepsizes is improved in the PI control case.

The next two examples are stiff SDEs and they have been solved using the embedded pair (BM,BI) method with PC and also the H321 predictive PID controller (15). For these examples we use the three sets of PC control parameters recommended in [13]; these are set 1: $(kk_{\rm I}, kk_{\rm P}) = (0.4, 0.7)$; set 2: $(kk_{\rm I}, kk_{\rm P}) = (0.6, 0.9)$ and set 3: $(kk_{\rm I}, kk_{\rm P}) = (1, 1)$. For the H321 predictive PID controller we use

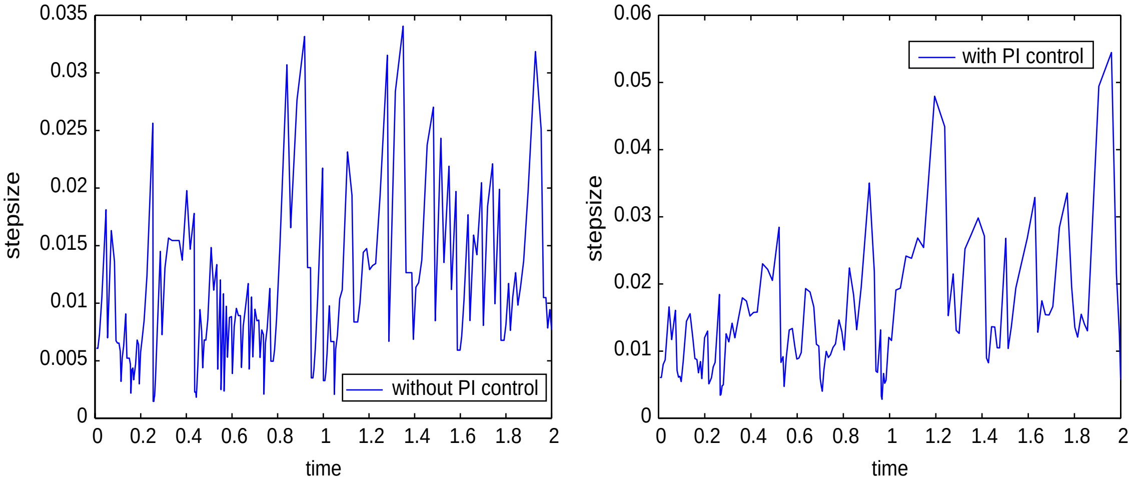

Fig. 4. Successful stepsizes versus time—for Example 4.2.

Table 3
Brusselator problem; standard, PC and PID controllers

	Tol = 10^{-2}			Tol = 10^{-3}		
	Tried	Failed	Avg. h	Tried	Failed	Avg. h
Standard	130	27	0.4854	340	61	0.1792
PC set 1	93	13	0.6250	337	55	0.1773
PC set 2	91	15	0.6579	298	46	0.1984
PC set 3	101	18	0.6024	320	60	0.1923
H321 PID	110	15	0.5263	239	41	0.2525

$(kk_{\mathrm{I}}, kk_{\mathrm{P}}) = (0.3, 0.45)$. Numerical performances are then compared to the simulations using standard control.

Example 4.3. This is the stochastic Brusselator equation in Itô form

$$\mathrm{d}y_1 = [(\alpha - 1)y_1 + \alpha y_1^2 + (y_1 + 1)^2 y_2]\,\mathrm{d}t + \sigma y_1(1 + y_1)\,\mathrm{d}W,$$

$$\mathrm{d}y_2 = [-\alpha y_1(1 + y_1) - (y_1 + 1)^2 y_2]\,\mathrm{d}t - \sigma y_1(1 + y_1)\,\mathrm{d}W$$

with $(y_1(0), y_2(0)) = (-0.1, 0), \alpha = 1.8$ and stochasticity level $\sigma = 0.8$; the integration is from $t = 0$ to $T = 50$; the fixed Wiener path is with $h_{\mathrm{fix}} = 0.1$ and initial stepsize is 0.05.

The numerical results are given in Table 3, where the advantage of applying the PC or PID controllers is shown in the significant reduction of the number of failed steps. In this example the tolerance level has some influence on which K values should be chosen. For tol $= 10^{-2}$, PC(0.4, 0.7) and PC(0.6, 0.9) give similar results with about 30% less work and a remarkable 44–50% less number of failed steps. Whereas when tol $= 10^{-3}$, PC(0.6, 0.9) provides the least amount of work

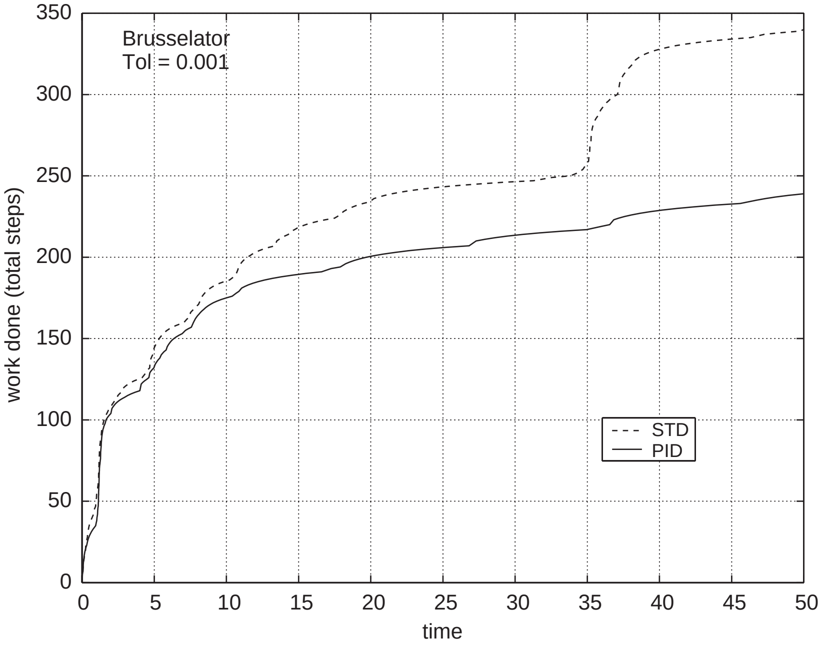

Fig. 5. Brusselator problem tol $= 10^{-3}$. Work versus time.

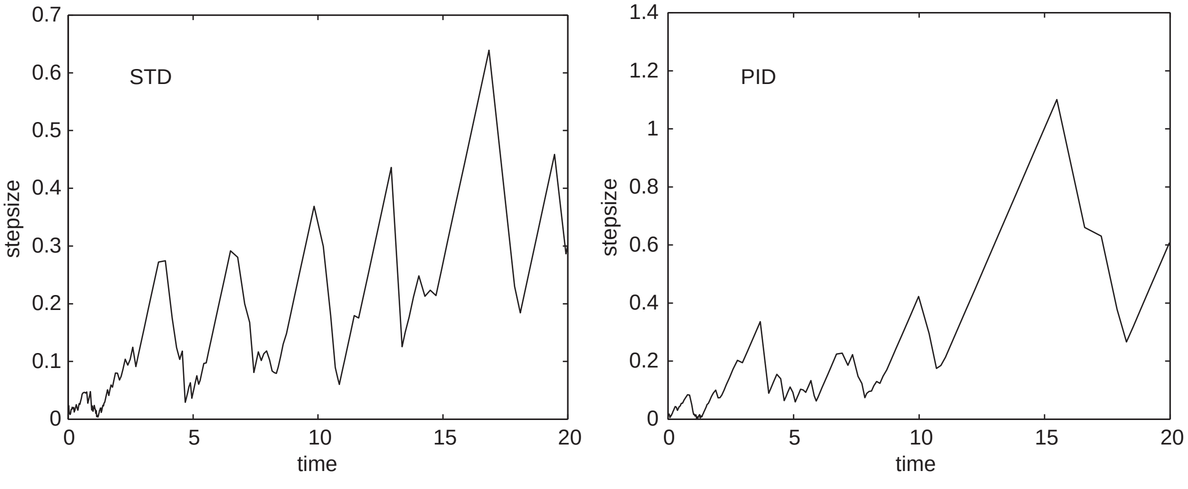

Fig. 6. Brusselator problem tol $= 10^{-3}$. Stepsize versus time.

required with 12% improvement in efficiency. The H321 predictive PID control performs very well at a nonlax tolerance. The graph in Fig. 5 shows the reduced amount of work done when the H321 predictive PID controller is applied in comparison to using standard control, and the graphs in Fig. 6 give a segment of the stepsize sequences taken.

Table 4
Example 4.4 ($a=25,\ b=5$): standard, PC and PID controllers

	Tol $= 10^{-3}$			Tol $= 10^{-4}$		
	Tried	Failed	Avg. h	Tried	Failed	Avg. h
Standard	2119	442	6.0(−4)	8148	1712	2.0(−4)
PC set 1	1503	283	8.0(−4)	7427	1282	2.0(−4)
PC set 2	1440	282	9.0(−4)	7509	1352	2.0(−4)
PC set 3	1336	279	9.0(−4)	7677	1466	2.0(−4)
H321 PID	1499	233	8.0(−4)	7565	1128	2.0(−4)

Table 5
Example 4.4 ($a=2,\ b=5$): various controllers

	Tol $= 10^{-2}$			Tol $= 10^{-3}$		
	Tried	Failed	Avg. h	Tried	Failed	Avg. h
Standard	268	52	4.6(−3)	1939	402	7.0(−4)
H321	643	44	1.7(−3)	2044	126	5.0(−4)
H312	334	59	3.6(−3)	1579	294	8.0(−4)
H211b	228	49	5.6(−3)	1549	345	8.0(−4)
Pure Integral	329	53	3.6(−3)	1450	242	8.0(−4)

Example 4.4. In this example we consider (2) with solution (3), with parameters $a=25$ and 2; and $b=5$. The initial stepsize is $h_0=0.05$. A fixed Brownian path of step length 0.1 is first generated.

Table 4 presents numerical results (for the very stiff case of $a=25$ and $b=5$) on the number of steps attempted and failed for the methods used in Example 4.3. Again the PC approach is more efficient than standard control. In particular all three set values of parameter K lead to less work than the regular standard control and reduce the number of failed steps significantly. PC set 3 provides the best performance for $\mathrm{tol}=10^{-3}$. The computational work is reduced to about 37% ($\mathrm{tol}=10^{-3}$) and 9% (for $\mathrm{tol}=10^{-4}$), along with a reduction of failed steps by 47% and 34%, respectively. Clearly, with PC and predictive PID controllers there are less failed steps and the latter controller has the least amount in all cases.

In Table 5 we compare the H321, H312, H211b (with b=4) controllers (as described in Section 3) and the purely integral control (with $kk_\mathrm{I}=0.3$) on problem (2) with $a=2$ and $b=5$. In looking at these results we see that

- H321 is worse than standard;
- H312 is considerably better than standard, especially at nonlax tolerances;
- H211b is robust at many levels of the tolerance, for $b=4$;
- pure integral control with a low integral gain has similar performance to H312.

Fig. 7 displays a segment of the stepsizes taken using PC(1,1) (right) and standard control (left) for the case $a=2,\ b=5$ and $\mathrm{tol}=10^{-2}$.

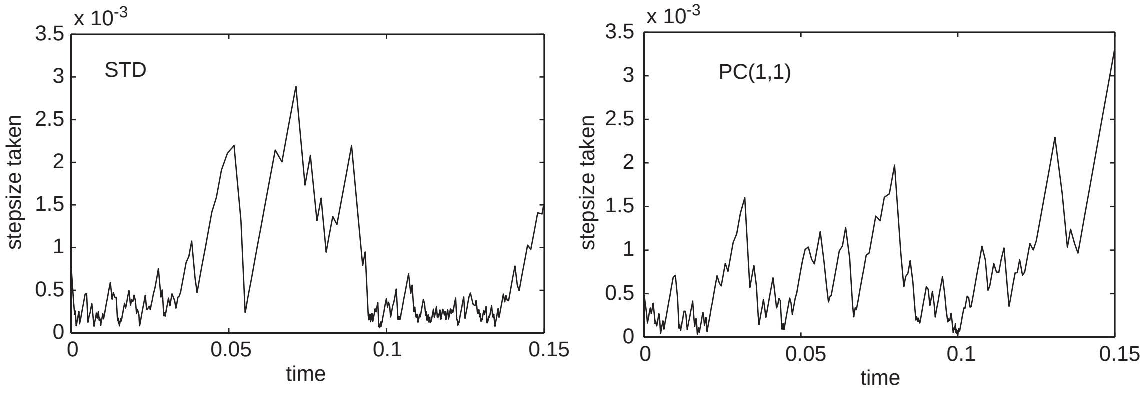

Fig. 7. Stepsizes versus time—for Example 4.4, $a = 2$, $b = 5$.

5. Conclusions and future directions

This paper has demonstrated the advantages of adaptive control for the stepsize change strategy when solving SDEs. While we have not presented any formal justification or theory for the choice of the k_I and k_P parameters in PI control, numerical experimentation demonstrates the advantages that this mode of stepsize control has. It appears, numerically, that the values used for k_I and k_P depend in some manner on the tolerance level and the intensity of noise in the particular SDE being solved, and such dependence may be hard to characterise formally.

Numerical experiments have also shown that some predictive and predictive PID controllers are alternatives that can enhance the performance of SDE methods even more than techniques based just on PI control. In particular, the first-order H211b controller with a value of $b = 4$ (which provides strong filtering) and the low gain purely integral controller both appear to offer very encouraging results. The last example in Section 4 shows that these two predictive controllers produce robust performance and the number of rejected steps is greatly decreased.

This paper represents a first attempt at applying the adaptive techniques of control theory used in the deterministic case to the stochastic case. While the results are promising, it seems clear that the types of control that are effective in the deterministic case may not necessarily be effective in the stochastic case.

Much work remains to be carried out in the development and fine-tuning of stochastic numerical methods. Some issues that should be investigated include:

- choice of initial stepsize;
- choice of tolerance level;
- choice of error criteria;
- issues of tolerance proportionality;
- further analysis of appropriate adaptive controllers;
- the development and implementation of methods and controllers suitable for noncommutative SDEs;
- the development of a sound theoretical basis for the convergence of adaptive stepsize strategies.

In conclusion, while much progress has already been made in the efficient numerical solution of SDEs in a variable stepsize setting, continuing development in the deterministic area indicate directions for further research in designing efficient techniques for stochastic differential equations.

Acknowledgements

We would like to thank Gustaf Söderlind for his very perceptive remarks which greatly improved the thrust of this paper. K. Burrage would like to thank the Australian Research Council for funding as a Federation Fellow.

References

[1] K. Burrage, P.M. Burrage, High strong order methods for non-commutative stochastic ordinary differential equation systems and the Magnus formula, Physica D 133 (1999) 34–48.

[2] P.M. Burrage, K. Burrage, A variable stepsize implementation for stochastic differential equations, SIAM J. Sci. Comput. 24 (3) (2002) 848–864.

[3] K. Burrage, R. Herdiana, The Balanced Milstein Method for Stochastic Differential Equations with a Variable Stepsize Implementation, Department of Mathematics, University of Queensland, 2003, submitted for publication.

[4] J.G. Gaines, T.J. Lyons, Variable stepsize control in the numerical solution of stochastic differential equations, SIAM J. Appl. Math. 5 (1997) 1455–1484.

[5] K. Gustafsson, Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods, ACM Trans. Math. Software 17 (1991) 533–554.

[6] K. Gustafsson, Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods, ACM Trans. Math. Software 20 (1994) 496–517.

[7] K. Gustafsson, M. Lundh, G. Söderlind, A PI stepsize control for the numerical solution of ordinary differential equations, BIT 28 (1988) 270–287.

[8] K. Gustafsson, G. Söderlind, Control strategies for the iterative solution of nonlinear equations in ODE solvers, SIAM J. Sci. Comput. 18 (1997) 23–40.

[9] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, 2nd Revised Edition, Springer, Berlin, 1993.

[10] D.J. Higham, Robust defect control with Runge–Kutta schemes, SIAM J. Numer. Anal. 26 (5) (1989) 1175–1183.

[11] S. Mauthner, Stepsize control in the numerical solution of stochastic differential equations, J. Comput. Appl. Math. 100 (1998) 93–109.

[12] G.N. Milstein, E. Platen, H. Schurz, Balanced implicit methods for stiff stochastic systems, SIAM J. Numer. Anal. 35 (1998) 1010–1019.

[13] G. Söderlind, The automatic control of numerical integration, CWI Quart. 11 (1) (1998) 55–74.

[14] G. Söderlind, Digital filters in adaptive time-stepping, ACM Trans. Math. Software 29 (1) (2003) 1–26.

[15] A. Szepessy, R. Tempone, G.E. Zouraris, Adaptive weak approximation of stochastic differential equations, Comm. Pure Appl. Math. 54 (2001) 1169–1214.