Open access • Journal Article • DOI:10.1137/S1064827500376922

A Variable Stepsize Implementation for Stochastic Differential Equations

Pamela Burrage, Kevin Burrage

Published on: 01 Mar 2002 - SIAM Journal on Scientific Computing (Society for Industrial and Applied Mathematics)

Topics: Adaptive stepsize, Cash–Karp method, Stochastic differential equation, Runge–Kutta methods and Differential equation

SIAM J. SCI. COMPUT.
Vol. 24, No. 3, pp. 848–864

A VARIABLE STEPSIZE IMPLEMENTATION FOR STOCHASTIC DIFFERENTIAL EQUATIONS*

P. M. BURRAGE† AND K. BURRAGE†

Abstract. Stochastic differential equations (SDEs) arise from physical systems where the parameters describing the system can only be estimated or are subject to noise. Much work has been done recently on developing higher order Runge–Kutta methods for solving SDEs numerically. Fixed stepsize implementations of numerical methods have limitations when, for example, the SDE being solved is stiff as this forces the stepsize to be very small. This paper presents a completely general variable stepsize implementation of an embedded Runge–Kutta pair for solving SDEs numerically; in this implementation, there is no restriction on the value used for the stepsize, and it is demonstrated that the integration remains on the correct Brownian path.

Key words. SDEs, Runge–Kutta, variable stepsize, embedding

AMS subject classifications. 65C30, 65L05, 65L06

PII. S1064827500376922

1. Introduction.

There has been much work on the solution of ODEs using variable stepsize implementations, but the situation becomes much more complex when SDEs are involved. Nevertheless, the techniques used in the ODE case form a solid basis for their extension to the stochastic setting. In particular, this paper uses the technique of embedding to provide a variable stepsize implementation; with this approach, an estimate of the error at each step can be obtained cheaply with the information then being used to determine the appropriate value for the next stepsize.

An important factor in a variable stepsize implementation for SDEs is the necessity of remaining on the correct Brownian path; for example, when a stepsize is rejected or when the integration must be repeated with a different initial value or initial stepsize, the same Brownian path must be followed. Gaines and Lyons (1997) have proved that a stochastic numerical method must have a strong order of at least 1 to guarantee convergence to the correct solution if a variable stepsize implementation is used. In their paper, they develop a variable stepsize implementation using a Brownian tree approach. This ensures that the correct Brownian path is followed, but it restricts any change of stepsize to either double or half of the previous stepsize. Mauthner (1999) has also worked on a variable stepsize implementation and has described a way of computing the Stratonovich integrals $J_1 = \int \circ dW_s$ and $J_{10} = \int \int \circ dW_s\, ds$ on the subintervals $[t_1, t_2], [t_2, t_3]$ of $[t_1, t_3]$ for any value t_2 $(t_1 < t_2 \leq t_3)$. However, in Mauthner (1999), the advantage of a completely general stepsize is not exploited; instead, the halving or doubling strategy is the one preferred. In this paper, there is no such restriction imposed on the stepsize, and it is demonstrated that the correct Brownian path is maintained.

The structure of this paper is as follows. For completeness, some background on the numerical solution of SDEs is given in section 2. Then, in section 3, the technique of embedding is described, and two methods (R2 embedded in E1) are presented. The method E1 (of strong local order 1.5) is suitable for SDEs with one Wiener process

*Received by the editors August 17, 2000; accepted for publication (in revised form) June 19, 2001; published electronically November 22, 2002.
http://www.siam.org/journals/sisc/24-3/37692.html
†Department of Mathematics, University of Queensland, Brisbane 4072, Australia (pmb@maths.uq.edu.au, kb@maths.uq.edu.au).

or where the SDE is fully commutative. The strategies for error control and stepsize change are outlined along with a discussion on tolerance proportionality. In section 4, Brownian trees are defined and constructed for a particular Brownian path of Wiener increments. As well as the Wiener increments in the Brownian tree, Lévy areas are used to generate higher order Stratonovich integrals that correspond to the same path. This section also includes an alternative approach to following the Brownian path, where there is no restriction on the stepsize (compared to the halving or doubling of stepsize that is required in the Brownian tree construction). This approach derives a relationship between the Stratonovich integrals on the subintervals and those on the original interval, ensuring that all of those Stratonovich integrals have correct covariances. Implementation issues are discussed in section 5. Section 6 is used to present numerical results, comparing a fixed stepsize implementation with that of an unrestricted variable stepsize implementation on a selection of test problems. The advantages of a variable stepsize implementation for SDEs are readily apparent.

2. Background results. SDEs describe physical systems where noise is present, with the noise being modelled by a Wiener process that is nowhere differentiable. The general form of an autonomous SDE is

$$dy(t) = f(y(t))dt + g(y(t))dW(t), \qquad y \in \mathbb{R}^m, \qquad y(t_0) = y_0, \tag{2.1}$$

where f is the slowly varying continuous component called the drift coefficient (an m-vector-valued function), g is the rapidly varying continuous component called the diffusion coefficient (an $m \times d$ matrix-valued function), and $W(t)$ is a d-dimensional process having independent scalar Wiener process components ($t \geq 0$). A Wiener process W is a Gaussian process with the property that

$$E(W(t)) = 0, \qquad E(W(t)W(s)) = \min\{t, s\}.$$

The Wiener increments $W(t) - W(s)$ are independent Gaussian processes with mean 0 and variance $|t - s|$.

Equation (2.1) can be written as a stochastic integral equation

$$y(t) = y(t_0) + \int_{t_0}^{t} f(y(s))ds + \int_{t_0}^{t} g(y(s))dW(s),$$

where the first integral is a regular Riemann–Stieltjes integral and the second integral is a stochastic integral, commonly interpreted in either Itô or Stratonovich form. The Stratonovich interpretation follows the usual rules of Riemann–Stieltjes calculus and for this reason is the form used in this paper. (The symbol $\circ$ in front of $dW(s)$ will serve to confirm a Stratonovich integral.) However, an SDE presented in Itô form can be converted to Stratonovich form using a simple formula which relates the two interpretations. Indeed, the solution of (2.1) and its related Stratonovich SDE

$$dy(t) = \overline{f}(y(t)) + g(y(t)) \circ dW(t), \tag{2.2}$$

where

$$\overline{f}_i(y(t)) = f_i(y(t)) - \frac{1}{2}\sum_{j=1}^{m}\sum_{k=1}^{d} g_{jk}(y(t))\frac{\partial g_{ik}(y(t))}{\partial y_j}, \qquad i = 1, \ldots, m, \tag{2.3}$$

are exactly the same.

A multiple Stratonovich integral is given by

$$J_{j_1 j_2 \cdots j_l}(t_0, t) = \int_{t_0}^{t} \int_{t_0}^{s_l} \cdots \int_{t_0}^{s_2} \circ \, dW_{s_1}^{j_1} \circ \cdots \circ dW_{s_l}^{j_l},$$

where $j_l \in \{0, 1, \cdots, d\}$ for d Wiener processes. Note that the integral $J_0(t_0, t) = \int_{t_0}^{t} \circ dW_{s_1}^{0} = \int_{t_0}^{t} ds_1$. For ease of notation, the written dependence on t_0 and t will be dropped when the meaning is clear from the context.

There are relationships between stochastic integrals—that is, between the Itô integrals, between the Stratonovich integrals, and also between the Itô and Stratonovich integrals. These relationships can be derived directly from the definition of the integral; there are also relationships proved in Kloeden and Platen (1992). For example,

$$\begin{aligned} J_0 J_1 &= J_{01} + J_{10}, \\ J_{11} &= \frac{1}{2} J_1^2, \\ J_i J_j &= J_{ij} + J_{ji}, \\ J_i J_{kl} &= J_{ikl} + J_{kil} + J_{kli}. \end{aligned}$$

These relationships are important in the evaluation of expectations of products of Stratonovich integrals, which arise when deriving order conditions for determining stochastic Runge–Kutta (SRK) methods.

There are two ways of measuring the accuracy of a numerical solution of an SDE. These are strong convergence and weak convergence. Strong convergence is required when each trajectory of the numerical solution must be close to the exact solution.

DEFINITION 2.1. *Let $\overline{y}_N$ be the numerical approximation to $y(t_N)$ after N steps with constant stepsize $h = \frac{t_N - t_0}{N}$; then $\overline{y}$ is said to* converge strongly *to y with strong global order p if $\exists C > 0$ (independent of h) and $\delta > 0$ such that*

$$E\left(\|\overline{y}_N - y(t_N)\|\right) \leq C h^p, \quad h \in (0, \delta).$$

This definition is for global order; the local error can behave as $O(h^{p+1/2})$; fractional orders arise as the root mean square order of the Wiener process is $h^{1/2}$.

In contrast to strong convergence, some SDEs just require the estimation of their moments, and for this weaker condition there is the definition of weak convergence.

DEFINITION 2.2. *The discrete time approximation $\overline{y}_N$ is said to* converge weakly *to y with order p, if for each polynomial q (which is $2(p+1)$ times continuously differentiable), $\exists C > 0$ (independent of h) and $\delta > 0$ such that*

$$||E\left[q(\overline{y}_N)\right] - E\left[q(y(t_N))\right]|| \leq C h^p, \quad h \in (0, \delta).$$

Numerical methods for SDEs are derived by comparing the stochastic Taylor series expansion of the numerical solution with that of the exact solution, over one step, assuming exact initial values. This comparison results in a set of order conditions to be satisfied; see Burrage and Burrage (2000) for the development of these order conditions using rooted tree theory in the case of Stratonovich problems.

The general explicit SRK method (with s stages) for solving the Stratonovich problem considered in this paper is given by

$$\begin{aligned} Y_i &= y_n + h \sum_{j=1}^{i-1} a_{ij} f(Y_j) + \sum_{j=1}^{i-1} \left(J_1 b_{ij}^{(1)} + \frac{J_{10}}{h} b_{ij}^{(2)} \right) g(Y_j), \qquad i = 1, \ldots, s, \\ y_{n+1} &= y_n + h \sum_{j=1}^{s} \alpha_j f(Y_j) + \sum_{j=1}^{s} \left(J_1 \gamma_j^{(1)} + \frac{J_{10}}{h} \gamma_j^{(2)} \right) g(Y_j). \end{aligned} \tag{2.4}$$

If the method does not include J_{10}, then the maximum strong order is 1.0; the inclusion of this Stratonovich integral allows methods with strong order greater than 1 to be developed (see, for example, Burrage and Burrage (1996)). Methods formulated from (2.4) can be extended for use in the d-Wiener process case (as long as the SDE system coefficients are fully commutative—otherwise, the order of the method is reduced to 0.5) by sampling additionally from $2, \ldots, d$ random number generators.

However, in the noncommutative case, a new style of method is required to avoid the order reduction mentioned above. The stochastic Taylor series expansion up to trees with three nodes is given, for example, in Kloeden and Platen (1992) or Burrage and Burrage (2000) as

$$y(t) = y_0 + \sum_{j=0}^{d} g_j(y_0) J_j + \sum_{i,j=0}^{d} g_j'(y_0)\left(g_i(y_0)\right) J_{ij} + \sum_{i,j,k=0}^{d} g_k'(y_0)\left(g_j'(y_0)\right)\left(g_i(y_0)\right) J_{ijk} + \sum_{i,j,k=0}^{d} g_k''(y_0)\left(g_j(y_0), g_i(y_0)\right) J_{ijk},$$

where g_i is the SDE coefficient associated with the ith Wiener process. Consequently, a numerical method (for solving noncommutative SDEs) needs representation from terms that correspond to $g_j'(y_0)\left(g_i(y_0)\right) J_{ij}$, as otherwise the order will be damped by the effect of the Stratonovich integrals J_{ij} (and $(E(J_{ij}^2))^{1/2} = O(h)$). A style of method to circumvent this problem is developed and then implemented in a variable stepsize setting in Burrage and Burrage (2002).

This section has provided an overview of the basic definitions required for studying numerical methods for solving SDEs; in the next section, the particular technique of embedding is described, as this implementation mode provides cheap error analysis at the end of each step.

3. Embedded SRK methods. When implementing a numerical method to solve an initial value problem

$$dy(t) = f(y(t))dt + g(y(t)) \circ dW(t), \qquad y(t_0) = y_0,$$

it is necessary to be able to control the truncation errors; however, any extra work in estimating these errors should be minimized. Having estimated the error at each step, it is then possible to adjust the stepsize being used based on the order of accuracy required.

In the deterministic case, the technique of embedding is a particularly efficient way of estimating the error at the end of a numerical step. In this paper, this approach is extended to the stochastic case, and, in particular, a 2-stage SRK method (called R2, with strong order 1) is embedded within a 4-stage SRK method (E1) of strong local order 1.5, thus enabling an error estimate to be obtained cheaply (with only 2 extra function evaluations required to calculate the update value from the 2-stage method).

Let $\hat{y}_{n+1}$ be the numerical result obtained from the implementation of an s-stage SRK method, and let y_{n+1} be that obtained from a higher stage SRK method (where the methods have order $\hat{p}$ and p, respectively). Then y_{n+1} is used to advance the numerical computation in the next step, while both $\hat{y}_{n+1}$ and y_{n+1} are used to

estimate the error. Here it is absolute error that is under consideration. For an m-dimensional system, let tol_i be the tolerance permitted for the ith component; then an error estimate of order $q + \frac{1}{2}$ is given by

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

where q is taken to be either $\hat{p}$ or p. 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 $q = p$. (Note that, in the deterministic case (with $q = \min(\hat{p}, p)$), the error estimate would behave as $O(h^{q+1})$, but, in the stochastic setting, order increases in increments of $\frac{1}{2}$.) As it is desirable that $y_{n+1,i} - \hat{y}_{n+1,i} \approx tol_i$, the step just completed is rejected if $error > 1$ and is accepted otherwise. An optimal stepsize (see Hairer, Nørsett, and Wanner (1993), for example, for ODEs) is determined by comparing this error to 1:

$$error \approx Ch^{q+\frac{1}{2}},$$
$$1 \approx Ch_{opt}^{q+\frac{1}{2}}$$

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

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

For an efficient implementation using a variable stepsize strategy, Hairer, Nørsett, and Wanner (1993), in a deterministic setting, decrease the optimal stepsize by a safety factor (for example, $fac = 0.8$) to avoid oscillatory behavior in the stepsize, and they also require that the stepsize does not increase or decrease too quickly:

$$h_{new} = h * \min(facmx, \max(facmn, fac * (1/error)^{1/(q+1/2)})), \tag{3.1}$$

where $facmx$ and $facmn$ are the maximal and minimal stepsize scaling factors allowed, respectively, for the problem being solved.

In P. M. Burrage (1999), a two-stage method (R2) of strong order 1 was derived; this method is optimal in terms of minimizing the principal error coefficients. However, in order to get a higher strong order of convergence it is necessary to use both the Stratonovich integrals J_1 and J_{10} and to go to four stages. For an embedding implementation, P. M. Burrage (1999) has shown that it is not possible to embed a three-stage Runge–Kutta method of order 3 in a four-stage Runge–Kutta method of order 4 even in the deterministic sense, and so, in the stochastic case, the best situation is to embed a two-stage SRK method into a four-stage SRK method. Method R2 has parameters

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array},$$

while E1 (of strong local order 1.5) is defined by (2.4) with

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

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

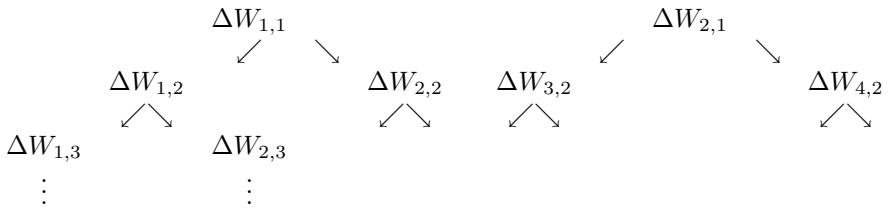
This embedded pair is most suitable for SDEs with only one Wiener process or, for $d > 1$, for SDEs that are fully commutative in the system coefficients. For other SDEs, it has been shown (see Burrage and Burrage (1998), for example) that these SRK methods suffer a severe order reduction down to 0.5; one way of overcoming this reduction is to include commutators in the method formulation (see Burrage and Burrage (1999)). However, implementation costs are increased for methods with commutators, due to the expense of calculating derivatives, and this has led to the development of suitable methods without commutators (see Burrage and Burrage (2002)).

4. Brownian trees. Most implementations of numerical methods for solving SDEs use a fixed stepsize, and, indeed, convergence of the method was demonstrated only for such stepsizes. However, recently Gaines and Lyons (1997) have proved that a method must have strong order of at least 1 to guarantee convergence to the correct solution if variable stepsizes are used and the stepsize does not vary too much. This result demonstrates that the embedded pair (R2, E1) is appropriate for a variable stepsize implementation.

In solving an SDE numerically, the Wiener process is approximated by sampling Wiener increments from the normal distribution with mean 0 and variance 1; these increments (scaled for the stepsize h being used) form the Brownian path for the trajectory under consideration.

The approach by Gaines and Lyons (1997) to ensure that the integration remains on the correct Brownian path is as follows: one Brownian path that represents the equally spaced increments for $W(t_i)$ along the required time interval $\{t_0, t_1, \ldots, t_N = T\}$ is fixed; then further Wiener increments on subdivisions of these intervals are generated recursively, with the subintervals being subdivided further if the accuracy of the numerical solution demands h to be still smaller. This approach ensures that the same Brownian path can be traversed if the numerical calculations are repeated with a different initial value or a different initial stepsize. The term *Brownian tree* is used to describe the set of Brownian paths that have been constructed.

Thus a Brownian tree is made up of Brownian (or Wiener) increments as follows:



By using a method due to Lévy (1948), the increments for level $j+1$ are computed for $j = 1, 2, \ldots$ as

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

where $y_{k,j}$ is normally distributed with mean 0 and variance 2^{-2j}. Such a tree can continue down to any level, and it is only the top level that needs to be complete to define the fixed Brownian path. Using Lévy areas, higher order stochastic integrals can also be generated to correspond to these Brownian increments. The Lévy area on the interval $(t, t+h)$ is defined for the ith and jth Wiener processes by

$$A_{ij}(t,t+h) = \frac{1}{2}\left(\int_t^{t+h}\int_t^s \circ dW^{(i)}(r) \circ dW^{(j)}(s) - \int_t^{t+h}\int_t^s \circ dW^{(j)}(r) \circ dW^{(i)}(s)\right),$$

and this can be approximated by

$$\hat{A}_{ij} = \frac{1}{2}\left(\sum_{1\le q<p\le 2^k} \Delta W_q^{(i)}\Delta W_p^{(j)} - \sum_{1\le p<q\le 2^k} \Delta W_q^{(i)}\Delta W_p^{(j)}\right)$$

(see Gaines and Lyons (1997), for example, for more details).

Thus an approximation to J_{ij} can be determined by first calculating $\hat{A}_{ij}$ and then computing

$$J_{ij} = \hat{A}_{ij} + \frac{1}{2}J_iJ_j$$

as the Lévy area $A_{ij} = \frac{1}{2}(J_{ij} - J_{ji})$. In particular, J_{10} can be computed as $\hat{A}_{10} + \frac{1}{2}J_1J_0$ (where $J_0 = h$).

To solve an SDE by halving or doubling the stepsize using the increments stored in the Brownian tree, the procedure is as follows. First, the top level of the tree is generated for the required number N of unit time intervals. The first numerical step is computed, and its success is judged by some acceptance criterion. If the step has failed, it is repeated with half the stepsize. Each time the step fails, the stepsize is halved, while, when the step succeeds, the numerical computation proceeds either with the same stepsize (if doubling h is not possible at this point) or with $2 * h$. Doubling of the stepsize is permitted as long as the increment about to be read in the tree is at a position that allows movement up the tree. For example, a sequence of stepsizes $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$, $\frac{1}{4}$ is permitted, but, for the sequence $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, the next stepsize must stay at $\frac{1}{8}$ as the alignment is not correct to allow progression up the tree to the next "$\frac{1}{4}$"-increment. While this approach guarantees comparable repeated calculations along the same fixed path, the imposition of a stepsize change strategy that allows only a halving or doubling of the stepsize can be too restrictive.

An alternative approach is to allow for a completely flexible change of stepsize while still guaranteeing that the correct Brownian path is followed. Suppose that g_1

and g_2 were the two $N(0,1)$ samples that were used to determine $J_1(t_0, t_0+h)$ and $J_{10}(t_0, t_0+h)$; that is,

$$J_1(t_0, t_0+h) = \sqrt{h} g_1,$$
$$J_{10}(t_0, t_0+h) = \frac{h\sqrt{h}}{2}\left(g_1 + \frac{g_2}{\sqrt{3}}\right).$$

Now let the interval $[t_0, t_0+h]$ be partitioned into $[t_0, t_0+h_1] \cup [t_0+h_1, t_0+h]$ so that the first subinterval is of length $h_1 < h$ and the second subinterval has length $h_2 = h - h_1$. Then J_1 and J_{10} on each of the two subintervals can be expressed as a combination of g_1, g_2 and two new samples on $N(0,1)$ (say, N_1 and N_2) in the following way:

$$\begin{pmatrix} J_{1a} \\ J_{10a} \\ J_{1b} \\ J_{10b} \end{pmatrix} \equiv \begin{pmatrix} J_1(t_0, t_0+h_1) \\ J_{10}(t_0, t_0+h_1) \\ J_1(t_0+h_1, t_0+h) \\ J_{10}(t_0+h_1, t_0+h) \end{pmatrix} = X \begin{pmatrix} N_1 \\ N_2 \\ g_1 \\ g_2 \end{pmatrix}. \tag{4.1}$$

The elements of the 4×4 matrix X are determined by requiring that the covariances of any combination of J_1 and J_{10} (on either subinterval) satisfy the known rules for these random samples. Thus it is necessary that

$$E\begin{bmatrix} J_{1a}^2 & J_{1a}J_{10a} & J_{1a}J_{1b} & J_{1a}J_{10b} \\ J_{10a}J_{1a} & J_{10a}^2 & J_{10a}J_{1b} & J_{10a}J_{10b} \\ J_{1b}J_{1a} & J_{1b}J_{10a} & J_{1b}^2 & J_{1b}J_{10b} \\ J_{10b}J_{1a} & J_{10b}J_{10a} & J_{10b}J_{1b} & J_{10b}^2 \end{bmatrix} = \begin{pmatrix} h_1 & \frac{1}{2}h_1^2 & 0 & 0 \\ \frac{1}{2}h_1^2 & \frac{1}{3}h_1^3 & 0 & 0 \\ 0 & 0 & h_2 & \frac{1}{2}h_2^2 \\ 0 & 0 & \frac{1}{2}h_2^2 & \frac{1}{3}h_2^3 \end{pmatrix}.$$

In addition, the direct rules of integration dictate that

$$\begin{aligned} J_1(t_0, t_0+h) &= J_1(t_0, t_0+h_1) + J_1(t_0+h_1, t_0+h) \\ &= J_{1a} + J_{1b}, \\ J_{10}(t_0, t_0+h) &= J_{10}(t_0, t_0+h_1) + J_{10}(t_0+h_1, t_0+h) + h_2 J_1(t_0, t_0+h_1) \\ &= J_{10a} + J_{10b} + h_2 J_{1a}. \end{aligned}$$

Thus, writing

$$\begin{pmatrix} J_1(t_0, t_0+h_1) \\ J_{10}(t_0, t_0+h_1) \\ J_1(t_0+h_1, t_0+h) \\ J_{10}(t_0+h_1, t_0+h) \end{pmatrix} = \begin{pmatrix} X_{11}N_1 + X_{12}N_2 + X_{13}g_1 + X_{14}g_2 \\ X_{21}N_1 + X_{22}N_2 + X_{23}g_1 + X_{24}g_2 \\ X_{31}N_1 + X_{32}N_2 + X_{33}g_1 + X_{34}g_2 \\ X_{41}N_1 + X_{42}N_2 + X_{43}g_1 + X_{44}g_2 \end{pmatrix},$$

it can be seen that the direct rules of integration are satisfied if

$$X_{11} + X_{31} = 0, \tag{4.2}$$
$$X_{12} + X_{32} = 0, \tag{4.3}$$
$$X_{13} + X_{33} = \sqrt{h}, \tag{4.4}$$
$$X_{14} + X_{34} = 0, \tag{4.5}$$
$$X_{21} + X_{41} + h_2 X_{11} = 0, \tag{4.6}$$
$$X_{22} + X_{42} + h_2 X_{12} = 0, \tag{4.7}$$
$$X_{23} + X_{43} + h_2 X_{13} = \frac{1}{2}h\sqrt{h}, \tag{4.8}$$
$$\frac{2\sqrt{3}}{h\sqrt{h}}\left(X_{24} + X_{44} + h_2 X_{14}\right) = 1. \tag{4.9}$$

In addition, the covariances are satisfied if

$$X_{11}^2 + X_{12}^2 + X_{13}^2 + X_{14}^2 = h_1, \tag{4.10}$$

$$X_{11}X_{31} + X_{12}X_{32} + X_{13}X_{33} + X_{14}X_{34} = 0, \tag{4.11}$$

$$X_{11}X_{21} + X_{12}X_{22} + X_{13}X_{23} + X_{14}X_{24} = \frac{1}{2}h_1^2, \tag{4.12}$$

$$X_{11}X_{41} + X_{12}X_{42} + X_{13}X_{43} + X_{14}X_{44} = 0, \tag{4.13}$$

$$X_{21}X_{31} + X_{22}X_{32} + X_{23}X_{33} + X_{24}X_{34} = 0, \tag{4.14}$$

$$X_{31}X_{41} + X_{32}X_{42} + X_{33}X_{43} + X_{34}X_{44} = \frac{1}{2}h_2^2, \tag{4.15}$$

$$X_{31}^2 + X_{32}^2 + X_{33}^2 + X_{34}^2 = h_2, \tag{4.16}$$

$$X_{21}^2 + X_{22}^2 + X_{23}^2 + X_{24}^2 = \frac{1}{3}h_1^3, \tag{4.17}$$

$$X_{21}X_{41} + X_{22}X_{42} + X_{23}X_{43} + X_{24}X_{44} = 0, \tag{4.18}$$

$$X_{41}^2 + X_{42}^2 + X_{43}^2 + X_{44}^2 = \frac{1}{3}h_2^3. \tag{4.19}$$

The first four equations give expressions for X_{1j}, while (4.10) and (4.16) (together with the identity $h_1 + h_2 = h$) result in

$$X_{33} = \frac{h_2}{\sqrt{h}},$$

and hence

$$X_{13} = \frac{h_1}{\sqrt{h}}.$$

The second four equations determine X_{2j}, and the application of all of these expressions to the remaining equations eventually yields the solution, with X given by

$$\begin{pmatrix} 0 & -\sqrt{\frac{h_1h_2}{h^3}(h_1^2 - h_1h_2 + h_2^2)} & \frac{h_1}{\sqrt{h}} & \frac{\sqrt{3}h_1h_2}{h\sqrt{h}} \\ \frac{h_1\sqrt{h_1}h_2\sqrt{h_2}}{2\sqrt{3}\sqrt{h}\sqrt{h_1^2-h_1h_2+h_2^2}} & \frac{h_1h_2\sqrt{h_1h_2}(h_1-h_2)}{2h\sqrt{h}\sqrt{h_1^2-h_1h_2+h_2^2}} & \frac{h_1^2}{2\sqrt{h}} & \frac{h_1^2(h_1+3h_2)}{2\sqrt{3}h\sqrt{h}} \\ 0 & \sqrt{\frac{h_1h_2}{h^3}(h_1^2 - h_1h_2 + h_2^2)} & \frac{h_2}{\sqrt{h}} & \frac{-\sqrt{3}h_1h_2}{h\sqrt{h}} \\ \frac{-h_1\sqrt{h_1}h_2\sqrt{h_2}}{2\sqrt{3}\sqrt{h}\sqrt{h_1^2-h_1h_2+h_2^2}} & \frac{h_2\sqrt{h_1}\sqrt{h_2}(h_1^2-h_1h_2+2h_2^2)}{2h\sqrt{h}\sqrt{h_1^2-h_1h_2+h_2^2}} & \frac{h_2^2}{2\sqrt{h}} & \frac{h_2^2(h_2-3h_1)}{2\sqrt{3}h\sqrt{h}} \end{pmatrix}. \tag{4.20}$$

Not all of the equations to be solved were independent, and, during the solution process, the free parameter X_{31} was set to zero.

Note that the X-matrix above has also been derived by Mauthner (1999) but from a completely different approach.

From a programming perspective, it is sensible to record the values obtained for J_1 and J_{10} on each interval rather than recording the $N(0,1)$ samples used to obtain these values. Hence, in the formula given by (4.1), g_1 is replaced by the equivalent $\frac{1}{\sqrt{h}}J_1(t_0, t_0+h)$ and g_2 by $-\frac{\sqrt{3}}{\sqrt{h}}J_1(t_0, t_0+h) + \frac{2\sqrt{3}}{h\sqrt{h}}J_{10}(t_0, t_0+h)$.

In this paper, it is the approach defined by (4.1) that is followed when implementing a numerical method in variable stepsize mode, and thus an arbitrary stepsize change is permitted.

First, the Brownian path is fixed for a nominated stepsize h_{fix}; this can represent a series of output points, for example. If this stepsize is the maximum allowed for the integration, then all subsequent simulations are generated "downward"; however,

if the integration requires $h > h_{fix}$, the simulated Stratonovich integrals can just as easily be generated "upward" from the fixed path. Given the fixed Brownian path, the integration proceeds using the desired stepsize h_1; the values of J_1 and J_{10} on these subintervals do not need to be stored—they can be merely generated as required based on the fixed path. At the end of the integration, the sum of the J_1-values along the path actually followed equals the sum of the J_1-values along the fixed path. Similarly, the J_{10}-values adhere to the definition

$$\begin{aligned} J_{10}(t_1,t_3) &= \int_{t_1}^{t_3}\int_{t_1}^{s} \circ dW_{s_1} ds \\ &= \int_{t_1}^{t_2}\int_{t_1}^{s} \circ dW_{s_1} ds + \int_{t_2}^{t_3}\int_{t_1}^{s} \circ dW_{s_1} ds \\ &= J_{10}(t_1,t_2) + \int_{t_2}^{t_3}\left(\int_{t_1}^{t_2} \circ dW_{s_1} + \int_{t_2}^{s} \circ dW_{s_1}\right) ds \\ &= J_{10}(t_1,t_2) + \int_{t_2}^{t_3} J_1(t_1,t_2) ds + J_{10}(t_2,t_3) \\ &= J_{10}(t_1,t_2) + J_{10}(t_2,t_3) + (t_3 - t_2) J_1(t_1,t_2) \end{aligned}$$

for the subintervals $[t_1,t_3] = [t_1,t_2] \cup [t_2,t_3]$.

For the situation where higher order Stratonovich integrals are required by the numerical method, the Lévy area approach is used so that, given a J_1-value belonging to the path, this subinterval can be further subdivided and the Lévy area $\hat{A}_{ij}$ can be approximated.

In more detail, the progression along the Brownian path is as follows:

$$\begin{array}{lll} |-----------|-----------| & & \\ t_n & t_n+h & t_n+2h \\ |------|------| & & \\ t_n & t_n+h_1 \quad t_n+h_1+h_2 & \end{array}$$

So $(J_1(t_n,t_n+h_1), J_{10}(t_n,t_n+h_1), J_1(t_n+h_1,t_n+h), J_{10}(t_n+h_1,t_n+h))^\top$ are simulated according to the formula (4.1), where the numerical method is evaluated with a stepsize of h_1. The error for that step is then computed as the difference between the two numerical methods (see, e.g., R2 and E1). If $error > 1$, then the step is rejected; otherwise, it is accepted. In either case, the new stepsize is determined via (3.1), where, for the (R2, E1) pair, $q = \frac{3}{2}$. If the step has been rejected, then the interval $[t_n, t_n+h]$ is resubdivided into $[t_n, t_n+h_{new}], [t_n+h_{new}, t_n+h]$, where $h_{new} < h_1$; however, for a successful step, it is the subinterval $[t_n+h_1, t_n+h]$ that is now subdivided: $[t_n+h_1, t_n+h_1+h_3], [t_n+h_1+h_3, t_n+h]$. If $h_3 > h_2$ (where $h_2 = h - h_1$), then this current step overlaps two subintervals of the fixed Brownian path, and the correct values of J_1 and J_{10} on the larger interval $[t_n+h_1, t_n+2h]$ must be used in the calculation of J_1 and J_{10} on $[t_n+h_1, t_n+h_1+h_3]$. In this case,

$$J_1(t_n+h_1, t_n+2h) = J_1(t_n+h_1, t_n+h) + J_1(t_n+h, t_n+2h)$$

and

$$J_{10}(t_n+h_1, t_n+2h) = J_{10}(t_n+h_1, t_n+h) + J_{10}(t_n+h, t_n+2h) + h\, J_1(t_n+h_1, t_n+h).$$

When $h = h_2$, the simulated values have already been calculated from the step of length h_1, while, for $h < h_2$, the new Stratonovich integrals are generated from the values on the interval $[t_n+h_1, t_n+h]$.

Using these interrelationships from subinterval to subinterval, the integration can proceed from t_0 to T using any value of h dictated by the performance of the numerical method during the preceding step while remaining on the correct Brownian path.

5. Implementation details. When implementing a numerical method in variable stepsize mode, a number of potential issues must be considered.

First, an appropriate tolerance level should be selected. Too severe a tolerance will result in many steps being required, while too lax a tolerance may allow an inaccurate or even an incorrect numerical solution. In this paper, the tolerance values chosen for variable stepsize control are purely experimental; there should be some correlation between the tolerance level and the degree of stochasticity of the SDE, but there has been no rigorous testing resulting in guidelines for the choice of tolerance, and we do not address tolerance proportionality (by which it is meant that a decrease in tolerance should lead to a proportional increase in the accuracy of the numerical solution).

Another important point is the choice of initial stepsize. If the initial stepsize is too large for the problem being solved, then several steps at the beginning of the numerical integration are wasted until the stepsize is reduced to an acceptable level. Alternatively, an initial stepsize that is too small also wastes computational time while it is steadily increased to an appropriate level. However, the analysis involved in selecting an "optimal" initial stepsize for a stochastic problem is considerable, and a strategy for this selection needs further development.

In a Brownian tree structure, where the stepsize is halved or doubled or remains unchanged, the integration steps will always match at the alignment points specified by the fixing of the path. In an unrestricted variable stepsize implementation (as in this paper), it is possible for the new stepsize to overlap the boundaries/alignment points under which the Brownian path was constructed. In fact, it is this feature which makes this new variable stepsize approach very desirable and indeed necessary in the numerical integration of an SDE.

Another feature of this stochastic variable stepsize implementation that differs from that in the deterministic setting is the possibility of stepsize acceleration. The situation arises when a step fails, possibly due to an "extreme" random sample; if the integration had been proceeding with larger values of h, the step failure will force the new stepsize to be quite small (particularly if the error is large); the acceleration option recognizes this situation and, after the small step has been implemented, the program code tries a new stepsize of $0.9 *$ previously acceptable h. If this is not successful, then the integration proceeds with a regular upgrading of h; however, if it was successful, then the integration can continue with the larger h-value, thus minimizing the amount of computational work required. This same strategy is coded for an implementation where output is required at multiple output points within the interval $[t_0, T]$. Again, the necessity of output at a particular point may have forced h to be artificially small, and the program allows for the possibility of regaining the previous h-level as soon as possible. The following diagram demonstrates the choice of h:

h_{old}	h_f	h_n	h?
[-----]	[-----]	[-----]	[-----]
OK	failed	OK	

$$h = \max(h_n, h_{old} * 0.9).$$

If the stepsize acceleration turns out to be unsuccessful, then the implementation reverts to choosing h based on h_n and the current error.

In an alternative to the stepsize acceleration feature the use of bands is introduced to prevent too-frequent stepsize changes. This problem has sometimes arisen in the variable stepsize solution of ODEs.

Four different bands have been tried in this paper; the bands are defined by

$$\begin{array}{ll} \text{Band 0:} & \text{no restrictions,} \\ \text{Band 1:} & [\tfrac{1}{2}, 1, 2], \\ \text{Band 2:} & [\tfrac{1}{2}, \tfrac{3}{4}, 1, \tfrac{3}{2}, 2], \\ \text{Band 3:} & [\tfrac{1}{2}, \tfrac{3}{4}, 1, \tfrac{5}{4}, \tfrac{3}{2}, \tfrac{7}{4}, 2]. \end{array}$$

To implement the stepsize control subject to these bands, the stepsize change factor is calculated and then compared to the band specified. For a general band $[b_0, b_1, \ldots, b_v]$ and for $0 \le i < v$, if $b_i \le$ stepsize change factor $< b_{i+1}$, then the factor is replaced by b_i. The maximum factor is b_v. For example, if the band is $[\frac{1}{2}, 1, 2]$, then the stepsize will only be halved, doubled, or remain the same (so this corresponds to the Brownian tree restrictions discussed previously). Band 2 is more lenient, with the stepsize not needing such a drastic decrease or increase, while Band 3 allows some additional flexibility. Band 0 is the completely unrestricted case. Some numerical testing of the use of bands for SDEs is presented in Example 6.1 in the next section.

6. Numerical results. In this section, Examples 6.1–6.3 compare the results obtained from a fixed stepsize implementation (using method E1) with those from a variable stepsize implementation (using the (R2, E1)-embedded pair of methods). For Example 6.1, where the exact solution is known, the errors obtained for each implementation are compared with the computational effort required to achieve that accuracy. The advantages of a variable stepsize implementation for Examples 6.2 and 6.3 are demonstrated pictorially.

Example 6.1. This Stratonovich SDE is two-dimensional with one Wiener process and is defined by

$$dy = Ay\,dt + By \circ dW_t, \qquad y(0) = y_0,$$

where

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

The explicit solution is

$$y(t) = \exp(At + B\,W(t))\,y_0,$$

and, for this integration (from 0 to 2), the initial value is $y_0 = (1, 1)^\top$, and the initial stepsize is $h_0 = 0.1$. The Brownian path is fixed with a length of 1, and the results are averaged over 100 trajectories to indicate the trend of the solution. Parameter a is set to -1, while the intensity of the noise is determined by the three cases $b = 0.1, 0.5, 1.0$. The results are presented in Tables 6.1–6.4. The figures for the average steps tried/taken are from Band 0, where h was not restricted during the integration. These figures were usually higher in the Band 0 case than for other bands.

Note that, as the stochasticity increases, the tolerance needs to be relaxed. In most cases, the use of bands to prevent too-frequent stepsize changes results in the integration taking fewer steps, although the restriction on h can sometimes lead to an

TABLE 6.1
Fixed stepsize solutions.

a, b	h	Error	# steps
-1,0.1	0.02	2.54(-5)	100
	0.015	1.91(-5)	134
	0.005	6.11(-6)	400
-1,0.5	0.02	6.27(-4)	100
	0.01	3.10(-4)	200
	0.0025	7.53(-5)	800
-1,1.0	0.05	6.32(-3)	40
	0.015	1.83(-3)	134
	0.005	6.02(-4)	400

TABLE 6.2
Global errors—variable stepsize, $a = -1, b = 0.1$.

Band \ Tol:	1.0(-3)	1.0(-4)	1.0(-5)	1.0(-6)
0	4.01(-4)	8.79(-5)	2.34(-5)	6.66(-6)
1	4.02(-4)	8.36(-5)	2.18(-5)	6.57(-6)
2	3.86(-4)	8.81(-5)	2.31(-5)	6.51(-6)
3	3.99(-4)	8.61(-5)	2.24(-5)	6.46(-6)
Avg steps tried	15	37	97	267
Avg steps taken	14	30	74	194

TABLE 6.3
Global errors—variable stepsize, $a = -1, b = 0.5$.

Band \ Tol:	1.0(-2)	1.0(-3)	1.0(-4)	1.0(-5)
0	6.19(-3)	1.35(-3)	3.26(-4)	7.60(-5)
1	5.67(-3)	1.34(-3)	3.28(-4)	7.65(-5)
2	6.26(-3)	1.35(-3)	3.32(-4)	7.76(-5)
3	6.23(-3)	1.39(-3)	3.33(-4)	7.65(-5)
Avg steps tried	15	42	143	494
Avg steps taken	12	30	100	345

TABLE 6.4
Global errors—variable stepsize, $a = -1, b = 1.0$.

Band \ Tol:	1.0(-2)	1.0(-3)	1.0(-4)
0	7.81(-3)	1.76(-3)	3.85(-4)
1	8.07(-3)	1.79(-3)	3.99(-4)
2	8.38(-3)	1.86(-3)	4.09(-4)
3	8.74(-3)	1.88(-3)	4.09(-4)
Avg steps tried	30	105	399
Avg steps taken	22	73	278

increased number of step rejections. (For example, h_1 may have been an acceptable stepsize, but $2h_1$ or even $\frac{3h_1}{2}$ may be too large an increase in stepsize, thus causing the new stepsize to be rejected.) In conclusion, as the numerical results do not indicate a consistent and obvious improvement in either the accuracy of the solution or the amount of computational effort required, it appears preferable to allow the integration to proceed without any artificial restriction on the stepsize calculation in terms of bands. Comparing the results from the fixed stepsize integration with those computed with variable stepsize, the variable stepsize results can be seen to produce equivalent accuracy for considerably reduced computational effort. It is also clear that some

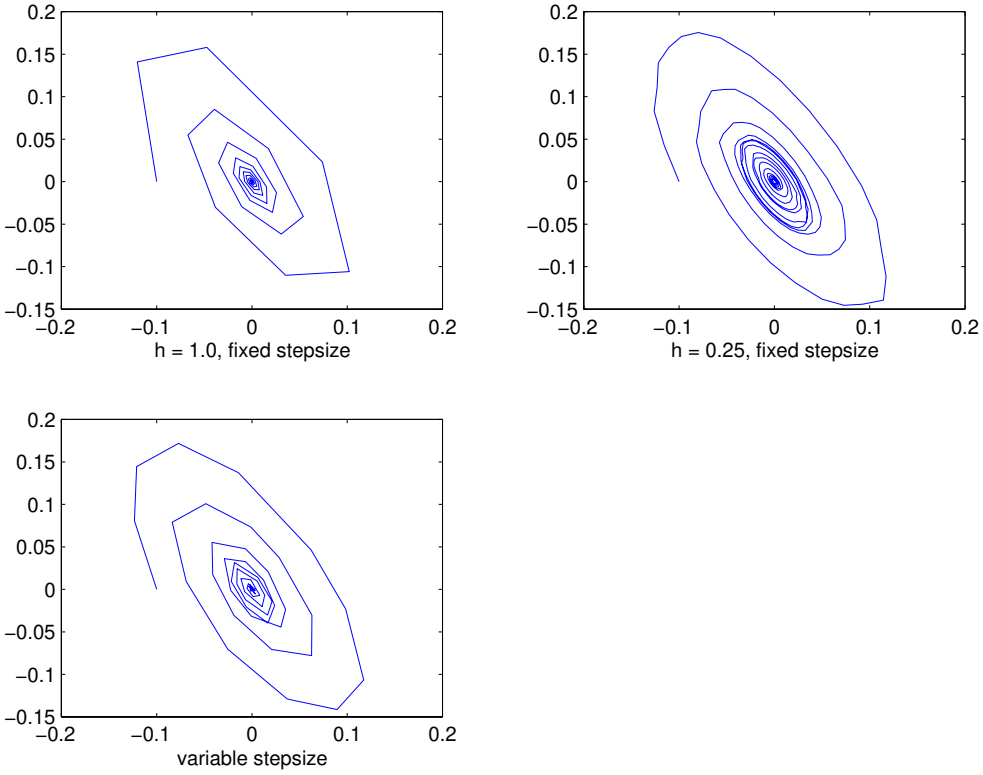


FIG. 6.1. *Stochastic Brusselator example.*

form of tolerance proportionality is taking place in that, as the tolerance reduces by a factor of 10, the global error reduces approximately by a factor of 5.

Example 6.2. This example is a stochastic version of the Brusselator system (see Kloeden and Platen (1992)) with no analytical solution. In Stratonovich form, the two-dimensional system is

$$\begin{aligned} dy_1(t) &= \big((\alpha-1)y_1(t) + \alpha y_1^2(t) + (y_1(t)+1)^2 y_2(t) \\ &\qquad - \sigma^2 y_1(t)(1+y_1(t))(1+2y_1(t))/2\big)\,dt + \sigma y_1(t)(1+y_1(t)) \circ dW(t), \\ dy_2(t) &= \big(-\alpha y_1(t) - \alpha y_1^2(t) - (y_1(t)+1)^2 y_2(t) \\ &\qquad + \sigma^2 y_1(t)(1+y_1(t))(1+2y_1(t))/2\big)\,dt - \sigma y_1(t)(1+y_1(t)) \circ dW(t). \end{aligned}$$

When the parameter α is less than 2, the zero solution $(y_1, y_2) \equiv (0,0)$ is globally asymptotically stable, while, for $\alpha > 2$, there is a limit cycle. In this example, $\alpha = 1.9$, $\sigma = 0.1$, and the initial value is $[-0.1, 0.0]^\top$; the numerical solutions are presented in Figure 6.1.

The graphs in the top two quadrants demonstrate the convergence to the fixed point $(0,0)$ for a fixed stepsize solution using method E1 and stepsizes of $h = 1.0$ and $h = 0.25$, respectively (where, with integration from 1 to 100, 100 and 400 steps were required). In quadrant 3 is the variable stepsize implementation using a tolerance of 0.01; the initial stepsize was 0.5, 84 steps were attempted, and 76 of these were successful; the successful stepsizes ranged from 0.4911 to 2.0 (which was

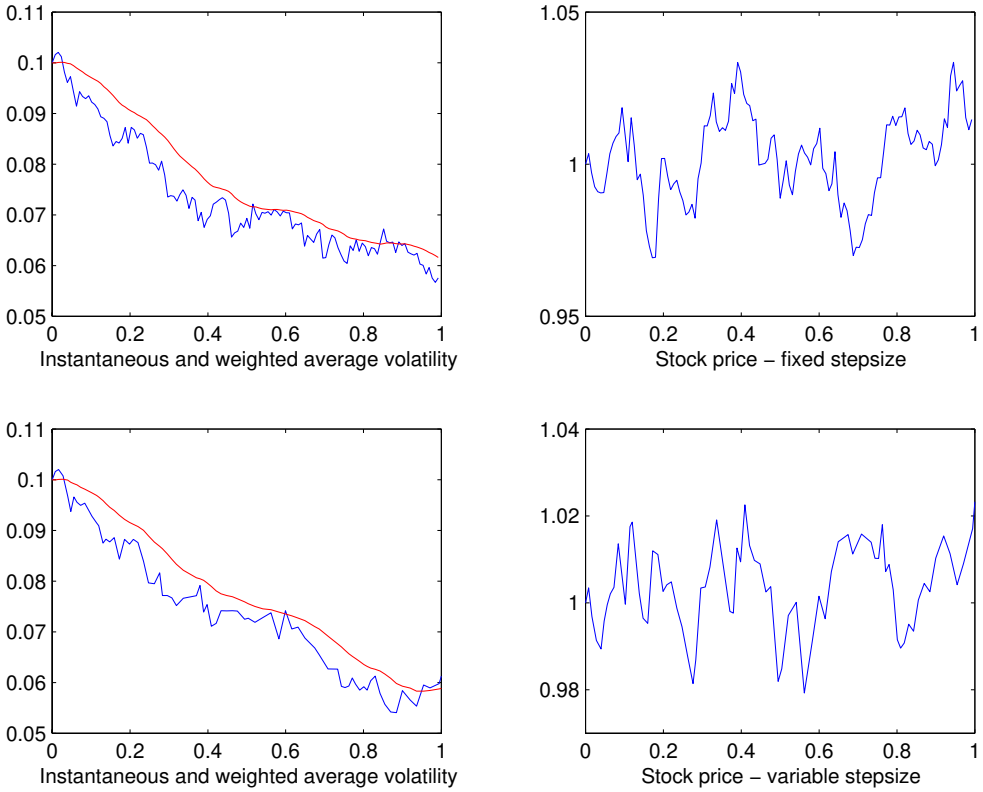


FIG. 6.2. *Finance example.*

the maximum stepsize specified in this run). There were no band restrictions on the change of stepsize. Subsequent runs with Bands 1, 2, and 3 required between 95 and 118 attempted steps with between 74 and 89 being successful. The tolerance was set to 0.01. Clearly the variable stepsize strategy shows great advantage over the fixed; larger stepsizes can be taken some distance away from the fixed point, with the variable stepsize implementation strategy allowing smaller steps to be taken as the numerical solution converges to $(0, 0)$.

Example 6.3. The final example in this paper (see Figure 6.2) is taken from the field of mathematical finance (Hofmann, Platen, and Schweizer (1992)). The SDE is three-dimensional with two Wiener processes and models the stock price (S_t), the instantaneous volatility of the stock (σ_t), and the weighted average volatility of the stock (ζ_t):

$$\begin{aligned} dS_t &= (r - \sigma_t^2/2)S_t dt + S_t\sigma_t \circ dW_1(t), \\ d\sigma_t &= (-q(\sigma_t - \zeta_t) - p^2\sigma_t/2)dt + p\sigma_t \circ dW_2(t), \\ d\zeta_t &= \frac{1}{\alpha}(\sigma_t - \zeta_t)dt, \end{aligned}$$

where $r = 0$, $q = 1$, $p = 0.3$, $\alpha = \frac{1}{10}$, and $W_1(t)$ and $W_2(t)$ are independent Wiener processes. The initial value is the vector $[1.0,0.1,0.1]^\top$, and the integration is from 0 to $T = 1.0$.

For the variable stepsize implementation (where the tolerance was set to 0.0002), 108 steps were attempted with 76 steps accepted; the stepsizes ranged from 0.0036 to 0.0234, with the initial stepsize chosen to be $\frac{1}{128}$, and the path was fixed with a steplength of 1.0. The results are depicted in quadrants 3 and 4, while quadrants 1 and 2 give the results obtained over the same Brownian path but in fixed stepsize mode (with stepsize $\frac{1}{128}$). The change in stock price (for each of fixed and variable stepsize implementations) is given in quadrants 2 and 4, plotted against time, while the instantaneous and weighted average volatilities are plotted against time in quadrants 1 and 3. The variable stepsize solution clearly gives the same qualitative results as for fixed stepsize but with less computational effort.

7. Conclusions. In this paper, the advantages of a variable stepsize implementation (not restricted to doubling or halving) over a fixed stepsize case have been demonstrated. As in the deterministic case, it is seen that this fully flexible stepsize implementation can be much more robust and effective than a fixed step or halving and doubling mode of implementation.

There is still work to be done on selecting an appropriate tolerance value as well as an appropriate initial stepsize, and (as in the deterministic case) there is scope for refining the stepsize change strategy. It is the authors' intention to investigate the concept of proportional integration (PI) control as applied to ODEs and extended to cover the stochastic case to see whether the amount of computational effort can be reduced along with a smoothing of the successful stepsizes (see Burrage, Herdiana, and Burrage (2002)). Tolerance proportionality (or, rather, the lack of it) is also an issue to be investigated in the future. When all of these factors have been taken into account, it is hoped that a robust stochastic variable stepsize integrator (suitable for a wide range of SDEs) can be developed.

REFERENCES

K. Burrage and P. M. Burrage (1996), *High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations*, Appl. Numer. Math., 22, pp. 81–101.

K. Burrage and P. M. Burrage (1998), *General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems*, Appl. Numer. Math., 28, pp. 161–177.

K. Burrage and P. M. Burrage (1999), *High strong order methods for non-commutative stochastic ordinary differential equation systems and the Magnus formula*, in Predictability: Quantifying Uncertainty in Models of Complex Phenomena, S. Chen, L. Margolin, and D. Sharp, eds., Phys. D, 133, pp. 34–48.

K. Burrage and P. M. Burrage (2000), *Order conditions of stochastic Runge–Kutta methods by B-series*, SIAM J. Numer. Anal., 38, pp. 1626–1646.

K. Burrage and P. M. Burrage (2002), *Without-Commutator Methods for Stochastic Ordinary Differential Equations*, in preparation.

P. M. Burrage (1999), *Runge-Kutta Methods for Stochastic Differential Equations*, Ph.D. Thesis, Department of Mathematics, University of Queensland, Brisbane, Australia.

P. M. Burrage, R. Herdiana, and K. Burrage (2002), *PI-Control for the Numerical Solution of Stochastic Differential Equations*, in preparation.

J. G. Gaines and T. J. Lyons (1997), *Variable step size control in the numerical solution of stochastic differential equations*, SIAM J. Appl. Math., 57, pp. 1455–1484.

E. Hairer, S. P. Nørsett, and G. Wanner (1993), *Solving Ordinary Differential Equations* I. *Nonstiff Problems*, 2nd revised ed., Springer-Verlag, Berlin.

N. Hofmann, E. Platen, and M. Schweizer (1992), *Option pricing under incompleteness and stochastic volatility*, Math. Finance, 2, pp. 153–187.

P. E. Kloeden and E. Platen (1992), *The Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, Berlin.

P. Lévy (1948), *Processus stochastiques et mouvement Brownien*, Monographies des Probabilités, Gauthier-Villars, Paris.

S. Mauthner (1999), *Schrittweitensteuerung bei der numerischen Lösung stochastischer Differentialgleichungen*, Thesis, Fortschr.-Ber. VDI Reihe 10 Nr 578, Düsseldorf, Germany.