Numerical Solution of Stochastic Differential Equations in Finance

Timothy Sauer

Department of Mathematics
George Mason University
Fairfax, VA 22030
tsauer@gmu.edu

Abstract. This chapter is an introduction and survey of numerical solution methods for stochastic differential equations. The solutions will be continuous stochastic processes that represent diffusive dynamics, a common modeling assumption for financial systems. We include a review of fundamental concepts, a description of elementary numerical methods and the concepts of convergence and order for stochastic differential equation solvers.

In the remainder of the chapter we describe applications of SDE solvers to Monte-Carlo sampling for financial pricing of derivatives. Monte-Carlo simulation can be computationally inefficient in its basic form, and so we explore some common methods for fostering efficiency by variance reduction and the use of quasi-random numbers. In addition, we briefly discuss the extension of SDE solvers to coupled systems driven by correlated noise, which is applicable to multiple asset markets.

1 Stochastic differential equations

Stochastic differential equations (SDEs) have become standard models for financial quantities such as asset prices, interest rates, and their derivatives. Unlike deterministic models such as ordinary differential equations, which have a unique solution for each appropriate initial condition, SDEs have solutions that are continuous-time stochastic processes. Methods for the computational solution of stochastic differential equations are based on similar techniques for ordinary differential equations, but generalized to provide support for stochastic dynamics.

We will begin with a quick survey of the most fundamental concepts from stochastic calculus that are needed to proceed with our description of numerical methods. For full details, the reader may consult Klebaner (1998); Oksendal (1998); Steele (2001).

A set of random variables X_t indexed by real numbers $t \geq 0$ is called a *continuous-time stochastic process*. Each instance, or *realization* of the stochastic process is a choice from the random variable X_t for each t, and is therefore a function of t.

Any (deterministic) function $f(t)$ can be trivially considered as a stochastic process, with variance $V(f(t)) = 0$. An archetypal example that is ubiquitous in models from physics, chemistry, and finance is the *Wiener process* W_t, a continuous-time stochastic process with the following three properties:

Property 1. For each t, the random variable W_t is normally distributed with mean 0 and variance t.

Property 2. For each $t_1 < t_2$, the normal random variable $W_{t_2} - W_{t_1}$ is independent of the random variable W_{t_1}, and in fact independent of all $W_t, 0 \leq t \leq t_1$.

Property 3. The Wiener process W_t can be represented by continuous paths.

The Wiener process, named after Norbert Wiener, is a mathematical construct that formalizes random behavior characterized by the botanist Robert Brown in 1827, commonly called Brownian motion. It can be rigorously defined as the scaling limit of random walks as the step size and time interval between steps both go to zero. Brownian motion is crucial in the modeling of stochastic processes since it represents the integral of idealized noise that is independent of frequency, called white noise. Often, the Wiener process is called upon to represent random, external influences on an otherwise deterministic system, or more generally, dynamics that for a variety of reasons cannot be deterministically modeled.

A typical *diffusion process* in finance is modeled as a differential equation involving deterministic, or *drift* terms, and stochastic, or *diffusion* terms, the latter represented by a Wiener process:

$$dX = a(t, X) \; dt + b(t, X) \; dW_t \tag{1}$$

Therefore the meaning of the SDE (1) is, by definition, the integral equation

$$X(t) = X(0) + \int_0^t a(s, y) \; ds + \int_0^t b(s, y) \; dW_s,$$

where the meaning of the last integral, called an **Ito integral**.

$\Delta t_i = t_i - t_{i-1}$ and $t_{i-1} \le t_i' \le t_i$. Similarly, the *Ito integral* is the limit

$$\int_c^d f(t) \; dW_t = \lim_{\Delta t \to 0} \sum_{i=1}^n f(t_{i-1}) \Delta W_i$$

where $\Delta W_i = W_{t_i} - W_{t_{i-1}}$, a step of Brownian motion across the interval. Note a major difference: while the t_i' in the Riemann integral may be chosen at any point in the interval (t_{i-1}, t_i), the corresponding point for the Ito integral is required to be the left endpoint of that interval.

Because f and W_t are rvs, so is the Ito integral $I = \int_c^d f(t) \; dW_t$. The *differential dI* is a notational convenience; thus

$$I = \int_c^d f \; dW_t$$

is expressed in differential form as

$$dI = f dW_t.$$

The differential dW_t of Brownian motion W_t is called *white noise*. A typical solution is a combination of drift and the diffusion of Brownian motion.

the chain rule for stochastic differentials, the *Ito formula*:

If $Y = f(t, X)$, then

$$dY = \frac{\partial f}{\partial t}(t, X) \; dt + \frac{\partial f}{\partial x}(t, X) \; dx + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, X) \; dx \; dx \qquad (2)$$

where the $dx \; dx$ term is interpreted by using the identities

$$dt \; dt = 0$$
$$dt \; dW_t = dW_t \; dt = 0$$
$$dW_t \; dW_t = dt \qquad (3)$$

Example. Some of the important features of typical SDEs can be illustrated using the following historically-pivotal example from finance, the **Black-Scholes diffusion equation**:

$$\begin{cases} dX = \mu X \; dt + \sigma X \; dW_t \\ X(0) = X_0 \end{cases} \qquad (4)$$

with constants μ and σ. Although the equation is comparatively simple, the fact that it can be exactly solved led to its central importance, by making a closed-form formula available for the pricing of simple options (Black and Scholes, 1973).

The solution of the Black-Scholes SDE is **geometric Brownian motion**

$$X(t) = X_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t}. \qquad (5)$$

Check , write $X = f(t, Y) = X_0 e^Y$, where $Y = (\mu - \frac{1}{2}\sigma^2)t + \sigma W_t$. By the Ito formula,

$$dX = X_0 e^Y \ dY + \frac{1}{2}e^Y \ dY \ dY$$

where $dY = (\mu - \frac{1}{2}\sigma^2) \ dt + \sigma \ dW_t$. $dY \ dY = \sigma^2 \ dt,$

==>

$$
\begin{aligned}
dX &= X_0 e^Y (r - \tfrac{1}{2}\sigma^2) \ dt + X_0 e^Y \sigma \ dW_t + \tfrac{1}{2}\sigma^2 e^Y \ dt \\
&= X_0 e^Y \mu \ dt + X_0 e^Y \sigma \ dW_t \\
&= \mu X \ dt + \sigma X \ dW_t
\end{aligned}
$$

Fig. 1 shows a realization of geometric Brownian motion with constant drift coefficient μ and diffusion coefficient σ. Similar to the case of ODEs, relatively few SDEs have closed-form solutions . It is often necessary to use numerical approximation techniques.

2 Numerical methods for SDEs.

The simplest effective computational method for the approximation of ODEs is Euler's method (Sauer, 2006). The Euler-Maruyama method (Maruyama, 1955) is the analogue of the Euler method for ODEs. To develop an approximate solution on the interval $[c, d]$, assign a grid of points

$$c = t_0 < t_1 < t_2 < \ldots < t_n = d.$$

Approximate x values

$$w_0 < w_1 < w_2 < \ldots < w_n$$

will be determined at the respective t points. Given the SDE initial value problem

$$
\begin{cases}
dX(t) = a(t, X)dt + b(t, X)dW_t \\
X(c) = X_c
\end{cases}
\tag{6}
$$

Euler-Maruyama Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + a(t_i, w_i)\Delta t_{i+1} + b(t_i, w_i)\Delta W_{i+1} \tag{7}$$

where

$$\Delta t_{i+1} = t_{i+1} - t_i$$
$$\Delta W_{i+1} = W(t_{i+1}) - W(t_i). \tag{8}$$

The crucial question is how to model the Brownian motion ΔW_i. Each random number ΔW_i is computed as

$$\Delta W_i = z_i\sqrt{\Delta t_i} \tag{9}$$

where $z_i \sim N(0,1)$.

Note the departure from the deterministic ODE case. Each set of $\{w_0, \ldots, w_n\}$ produced by the Euler-Maruyama method is an approximate realization of the solution stochastic process $X(t)$ which depends on the random numbers z_i that were chosen. Since Wt is a stochastic process, each realization will be different and so will our approximations.

Example, we show how to apply the Euler-Maruyama method to the Black Scholes SDE (4). The Euler-Maruyama equations (7) have form

$$w_0 = X_0 \tag{10}$$
$$w_{i+1} = w_i + \mu w_i \Delta t_i + \sigma w_i \Delta W_i.$$

We will use the drift coefficient $\mu = 0.75$ and diffusion coefficient $\sigma = 0.30$, which are values inferred from the series of market close share prices of Google, Inc. (NYSE ticker symbol GOOG) during the 250 trading days in 2009. To calculate the values μ and σ^2, respectively, mean and var of the daily stock price returns were converted to an annual basis, assuming independence of the daily returns.

An exact realization, generated from the solution (5), along with the corresponding Euler-Maruyama approximation, are shown in Fig. 1. By corresponding, we mean that the approximation used the same Brownian motion realization as the true solution. Note the close agreement between the solution and the approximating points, plotted as small circles every 0.2 time units. In addition, the original time series of Google share prices is shown for comparison. Both the original time series (grey curve) and the simulation from (5) (black curve) should be considered as realizations from the same diffusion process, with identical μ, σ and initial price $X_0 = 307.65$.
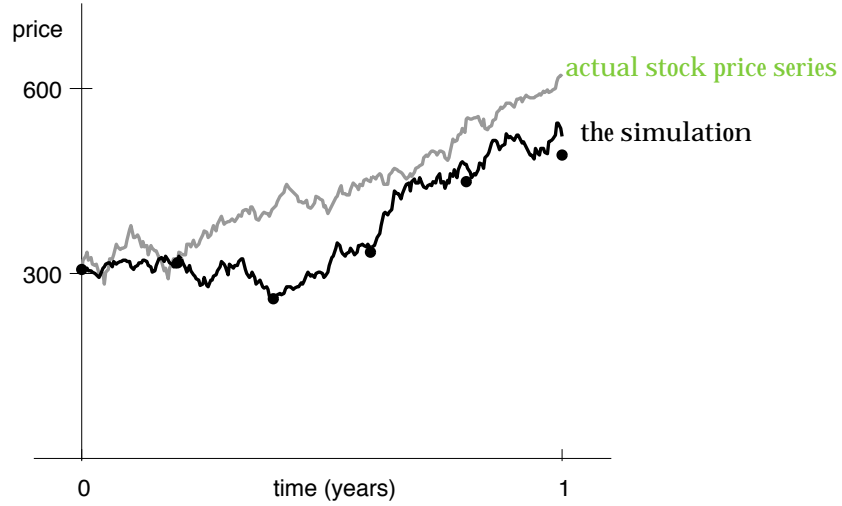
Fig. 1. Solution to the Black Scholes SDE (4). The exact solution (5) is plotted as a black curve. The Euler-Maruyama approximation with time step $\Delta t = 0.2$ is plotted as circles. The drift and diffusion parameters are set to $\mu = 0.75$ and $\sigma = 0.30$, respectively. Shown in grey is the actual stock price series, from which μ and σ were inferred.

Example, consider the *Langevin equation*

$$dX(t) = -\mu X(t) \ dt + \sigma \ dW_t \tag{11}$$

where μ and σ are positive constants. In this case, it is not possible to analytically derive the solution to this equation in terms of simple processes. The solution of the Langevin equation is a stochastic process called the *Ornstein-Uhlenbeck process*. Fig. 2 shows one realization of the approximate solution. It was generated from an Euler-Maruyama approximation:

$$w_0 = X_0 \tag{12}$$
$$w_{i+1} = w_i - \mu w_i \Delta t_i + \sigma \Delta W_i$$

for $i = 1, \ldots, n$. This SDE is used to model systems that tend to revert to a particular state, in this case the state $X = 0$, in the presence of a noisy background. Interest-rate models, in particular, often contain mean-reversion assumptions.

3 Strong convergence of SDE solvers.

The definition of convergence is similar to the concept for ODE solvers, aside from the differences caused by the fact that a solution
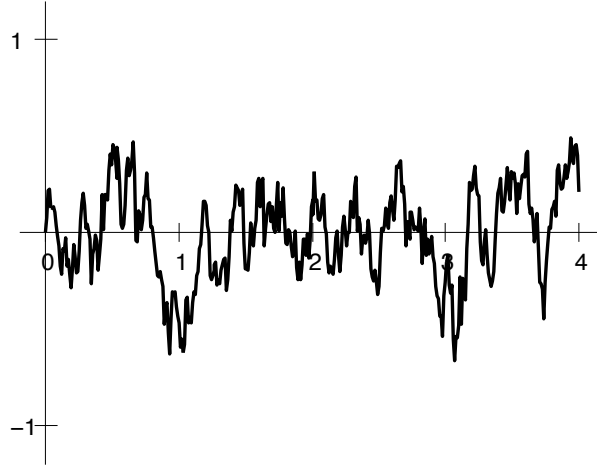
Fig. 2. Solution to Langevin equation (11). The upper path is the solution approximation for parameters $\mu = 10, \sigma = 1$, computed by the Euler-Maruyama method.

to an SDE is a stochastic process, and each computed trajectory is only one realization of that process. Each computed solution path $w(t)$, using Euler - Maruyama for example, gives a random value at T, so that $w(T)$ is a rv as well . The difference between the values at time T, $e(T) = X(T) - w(T)$, is therefore a rv.

A discrete-time approximation is said to *converge strongly* to the solution $X(t)$ at time T if

$$\lim_{\Delta t \to 0} E\{|X(T) - w_{\Delta t}(T)|\} = 0$$

where $w_{\Delta t}$ is the approximate solution computed with constant stepsize Δt, and E denotes expected value. For strongly convergent approximations, we further quantify the rate of convergence by the concept of order. An SDE solver *converges strongly with order m* if the expected value of the error is of mth order in the stepsize, i.e. if for any time T,

$$E\{|X(T) - w_{\Delta t}(T)|\} = O((\Delta t)^m)$$

for sufficiently small stepsize Δt. This definition generalizes the standard convergence criterion for ODEs, reducing to the usual definition when the stochastic part of the equation $\to 0$.

Although the Euler method for ODEs has order 1, the strong order for the Euler-Maruyama method for SDEs is $1/2$. This fact was proved in Gikhman and Skorokhod (1972), under appropriate conditions on the functions a and b in (6).

In order to build a strong order 1 method for SDEs , another term in the "stochastic Taylor series" must be added to the method. Consider the SDE

$$\begin{cases} dX(t) = a(X,t)dt + b(X,t)dW_t \\ X(0) = X_0. \end{cases} \tag{13}$$

Milstein Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + a(w_i, t_i)\Delta t_i + b(w_i, t_i)\Delta W_i$$
$$+ \tfrac{1}{2}b(w_i, t_i)\frac{\partial b}{\partial x}(w_i, t_i)(\Delta W_i^2 - \Delta t_i) \tag{14}$$

The Milstein Method has order one. Note that the Milstein Method is identical to the Euler-Maruyama Method if there is no X term in the diffusion part $b(X, t)$ of the equation. In case there is, Milstein will in general converge to the correct stochastic solution process more quickly than Euler-Maruyama as $\Delta t_i \to 0$.

Comparison of the Euler-Maruyama and Milstein methods, we apply them to the Black Scholes SDE

$$dX = \mu X \ dt + \sigma X \ dW_t. \tag{15}$$

The Milstein Method becomes

$$w_0 = X_0 \tag{16}$$
$$w_{i+1} = w_i + \mu w_i \Delta t_i + \sigma w_i \Delta W_i + \tfrac{1}{2}\sigma(\Delta W_i^2 - \Delta t_i)$$

Applying the Euler-Maruyama Method and the Milstein Method with decreasing stepsizes Δt results in successively improved approximations, as Table 1 shows:

The two columns represent the average, over 100 realizations, of the error $|w(T) - X(T)|$ at $T = 8$. The orders $1/2$ for Euler-Maruyama and 1 for Milstein are clearly visible in the table. Cutting the stepsize by a factor of 4 is required to reduce the error by a factor of 2 with the Euler-Maruyama method. For the Milstein method, cutting the stepsize by a factor of 2 achieves the same result. The data in the table is plotted on a log-log scale in Fig. 3.

The Milstein method is a Taylor method, meaning that it is derived from a truncation of the stochastic Taylor expansion of the solution. This is in many cases a disadvantage, since the partial derivative appears in the approximation method, and must be provided explicitly by the user. This is analogous to Taylor methods for solving ODEs, which are seldom used in practice for that reason. To counter this problem, Runge-Kutta methods were developed for ODEs, which trade these extra partial derivatives in the Taylor expansion for extra function evaluations from the underlying equation.

Table 1. Average error at $T = 8$ of approximate solutions of (4). The error scales as $\Delta t^{1/2}$ for Euler-Maruyama and Δt for Milstein.

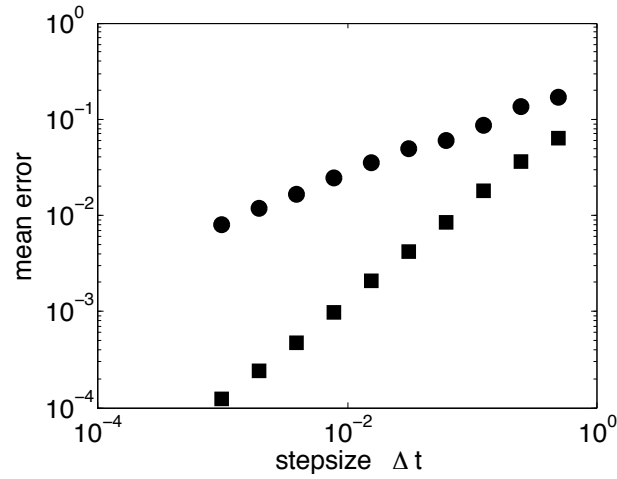| Δt	Euler-Maruyama	Milstein
2^{-1}	0.169369	0.063864
2^{-2}	0.136665	0.035890
2^{-3}	0.086185	0.017960
2^{-4}	0.060615	0.008360
2^{-5}	0.048823	0.004158
2^{-6}	0.035690	0.002058
2^{-7}	0.024277	0.000981
2^{-8}	0.016399	0.000471
2^{-9}	0.011897	0.000242
2^{-10}	0.007913	0.000122

Fig. 3. Error in the Euler-Maruyama and Milstein methods. Solution paths are computed for the geometric Brownian motion equation (15) and are compared to the correct $X(T)$ given by (5). The absolute difference is plotted versus stepsize h for the two different methods. The Euler-Maruyama errors are plotted as circles and the Milstein error as squares. Note the slopes $1/2$ and 1, respectively, on the log-log plot.

In the SDE context, the same trade can be made with the Milstein method, resulting in a strong order 1 method that requires evaluation of $b(X)$ at two places on each step. A heuristic derivation can be carried out by making the replacement

$$b_x(w_i) \approx \frac{b(w_i + b(w_i)\sqrt{\Delta t_i}) - b(w_i)}{b(w_i)\sqrt{\Delta t_i}}$$

in the Milstein formula (14), which leads to the following method (Rumelin, 1982):

Strong Order 1.0 Runge-Kutta Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + a(w_i)\Delta t_i + b(w_i)\Delta W_i$$
$$+ \tfrac{1}{2}[b(w_i + b(w_i)\sqrt{\Delta t_i}) - b(w_i)](\Delta W_i^2 - \Delta t_i)/\sqrt{\Delta t_i}$$

The orders of the methods introduced here for SDEs, $1/2$ for Euler-Maruyama and 1 for Milstein and the Runge-Kutta counterpart, would be considered low by ODE standards. Higher-order methods can be developed for SDEs, but become much more complicated as the order grows. As an example, consider the strong order 1.5 scheme for the SDE (13) proposed in Platen and Wagner (1982):

Strong Order 1.5 Taylor Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + a\Delta t_i + b\Delta W_i + \tfrac{1}{2}bb_x(\Delta W_i^2 - \Delta t_i)$$
$$+ a_y\sigma\Delta Z_i + \tfrac{1}{2}(aa_x + \tfrac{1}{2}b^2 a_{xx})\Delta t_i^2$$
$$+ (ab_x + \tfrac{1}{2}b^2 b_{xx})(\Delta W_i\Delta t_i - \Delta Z_i)$$
$$+ \tfrac{1}{2}b(bb_{xx} + b_x^2)(\tfrac{1}{3}\Delta W_i^2 - \Delta t_i)\Delta W_i \tag{17}$$

where partial derivatives are denoted by subscripts, and where the additional rv ΔZ_i is normally distributed with mean 0, variance $E(\Delta Z_i^2) = \tfrac{1}{3}\Delta t_i^3$ and correlated with ΔW_i with covariance $E(\Delta Z_i\Delta W_i) = \tfrac{1}{2}\Delta t_i^2$. Note that

$$\Delta Z_i = \tfrac{1}{2}\Delta t_i(\Delta W_i + \Delta V_i/\sqrt{3})$$

where ΔV_i is chosen independently from $\sqrt{\Delta t_i}N(0, 1)$.

Whether higher-order methods are needed in a given application depends on how the resulting approximate solutions are to be used. In the ODE case, the usual assumption is that the initial condition and the equation are known with accuracy. Then it makes sense to calculate the solution as closely as possible to the same accuracy, and higher-order methods are called for. In the context of SDEs, in particular if the initial conditions are chosen from a probability distribution, the advantages of higher-order solvers are often less compelling, and if they come with added computational expense, may not be warranted.

4 Weak convergence of SDE solvers

Strong convergence allows accurate approximations to be computed on an individual realization basis. For some applications, such detailed pathwise information is required. In other cases, the goal is to ascertain the probability distribution of the solution $X(T)$, and single realizations are not of primary interest.

Weak solvers seek to fill this need. They can be simpler than corresponding strong methods, since their goal is to replicate the probability distribution only.

A discrete-time approximation $w_{\Delta t}$ with step-size Δt is said to *converge weakly* to the solution $X(T)$ if

$$\lim_{\Delta t \to 0} E\{f(w_{\Delta t}(T))\} = E\{f(X(T))\}$$

for all polynomials $f(x)$. According to this definition, all moments converge as $\Delta t \to 0$. If the stochastic part of the equation is zero and the initial value is deterministic, the definition agrees with the strong convergence definition, and the usual ODE definition.

Weakly convergent methods can also be assigned an order of convergence. We say that a the solver *converges weakly with order m* if the error in the moments is of mth order in the stepsize, or

$$|E\{f(X(T))\} - E\{f(w_{\Delta t}(T))\}| = O((\Delta t)^m)$$

for sufficiently small stepsize Δt.

In general, the rates of weak and strong convergence do not agree. Unlike the case of ordinary differential equations, where the Euler method has order 1, the Euler-Maruyama method for SDEs has strong order $m = 1/2$. However, Euler-Maruyama is guaranteed to converge weakly with order 1.

Higher order weak methods can be much simpler than corresponding strong methods, and are available in several different forms. The most direct approach is to exploit the Ito-Taylor expansion (Kloeden and Platen, 1992), the Ito calculus analogue of the Taylor expansion of deterministic functions. An example SDE solver that converges weakly with order 2 is the following:

Weak Order 2 Taylor Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + a\Delta t_i + b\Delta W_i + \tfrac{1}{2}bb_x(\Delta W_i^2 - \Delta t_i)$$
$$+ a_x b\Delta Z_i + \tfrac{1}{2}(aa_x + \tfrac{1}{2}a_{xx}b^2)\Delta t^2$$
$$+ (ab_x + \tfrac{1}{2}b_{xx}b^2)(\Delta W_i\Delta t_i - \Delta Z_i) \tag{18}$$

where ΔW_i is chosen from $\sqrt{\Delta t_i}N(0,1)$ and ΔZ_i is distributed as in the above Strong Order 1.5 Method.

A second approach is to mimic the idea of Runge-Kutta solvers for ODEs. These solvers replace the explicit higher derivatives in the Ito-Taylor solvers with extra function evaluations at interior points of the current solution interval. Platen (1987) proposed the following weak order 2 solver of Runge-Kutta type:

Weak Order 2 Runge-Kutta Method

$$w_0 = X_0$$
$$w_{i+1} = w_i + \tfrac{1}{2}[a(u) + a(w_i)]\Delta t_i$$
$$+ \frac{1}{4}[b(u_+) + b(u_-) + 2b(w_i)]\Delta W_i$$
$$+ \frac{1}{4}[b(u_+) - b(u_-)](\Delta W_i^2 - \Delta t)/\sqrt{\Delta t_i} \qquad (19)$$

where

$$u = w_i + a\Delta t_i + b\Delta W_i$$
$$u_+ = w_i + a\Delta t_i + b\sqrt{\Delta t_i}$$
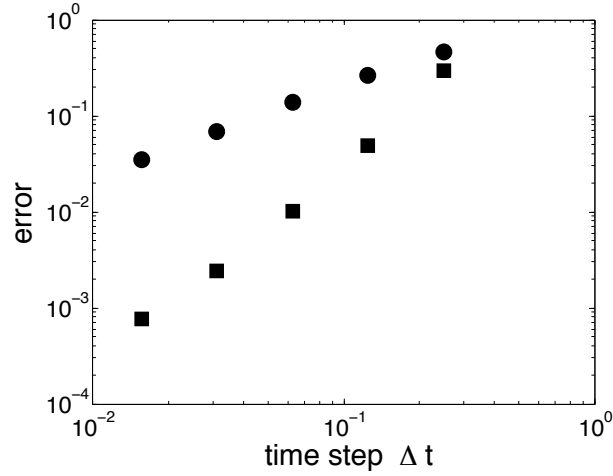$$u_- = w_i + a\Delta t_i - b\sqrt{\Delta t_i}.$$

Fig. 4. The mean error of the estimation of $E(X(T))$ **for SDE (15).** The plot compares the Euler-Maruyama method (circles) which has weak order 1, and the weak order 2 Runge-Kutta type method (squares) given in (19). Parameter used were $X(0) = 10, T = 1, \mu = -3, \sigma = 0.2$.

Fig. 4 compares the Euler-Maruyama method, which converges with order 1 in the weak sense, to the Weak Order 2 Runge-Kutta-Type Method. Note the difference between strong and weak convergence. In the previous Fig. 3, which considers strong convergence, the mean error of the estimate of a point $X(T)$ on the solution curve was plotted. In Fig. 4, on the other hand, the mean error of the estimate of the expected value $E[X(T)]$ is plotted, since we are comparing weak convergence of the methods. The weak orders are clearly revealed by the log-log plot.

Several other higher-order weak solvers can be found in Kloeden and Platen (1992). Weak Taylor methods of any order can be constructed, as well as Runge-Kutta analogues that reduce or eliminate the derivative calculations. In addition, standard Richardson extrapolation techniques (Sauer, 2006) can be used to bootstrap weak method approximations of a given order to the next order. See Kloeden and Platen (1992) for full details.

Weak solvers are often an appropriate choice for financial models, when the goal is to investigate the probability distribution of an asset price or interest rate, or when Monte-Carlo sampling is used to price a complicated derivative. In such cases it is typical to be primarily interested in one of the statistical moments of a stochastically-defined quantity, and weak methods may be simpler and still sufficient for the sampling purpose. In the next section we explore some of the most common ways SDE solvers are used to carry out Monte-Carlo simulations for derivative pricing.

5 Monte-Carlo sampling of SDE paths for option pricing

As an illustrative example of the use of SDE solvers for option pricing, consider the European call, whose value at expiration time T is $\max\{X(T) - K, 0\}$, where $X(t)$ is the price of the underlying stock, K is the strike price. The no-arbitrage assumptions of Black-Scholes theory imply that the present value of such an option is

$$C(X_0, T) = e^{-rT} E(\max\{X(T) - K, 0\}) \tag{20}$$

where r is the fixed prevailing interest rate during the time interval $[0, T]$, and where the underlying stock price $X(t)$ satisfies the SDE
$$dX = rX \; dt + \sigma X \; dW_t.$$

The value of the call option can be determined by calculating the expected value (20) explicitly. Using the Euler-Maruyama method for following solutions to the Black-Scholes SDE, the value $X(T)$ at the expiration time T can be determined for each path, or realization of the stochastic process. For a given n realizations, the average $\langle\max\{X(T) - K, 0\}\rangle$ can be used as an approximation to the expected value in (20). Carrying this out and comparing with the exact solution from the Black-Scholes formula

$$C(X, T) = XN(d_1) - Ke^{-rT}N(d_2) \tag{21}$$

where

$$d_1 = \frac{\log(X/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \quad d_2 = \frac{\log(X/K) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}},$$

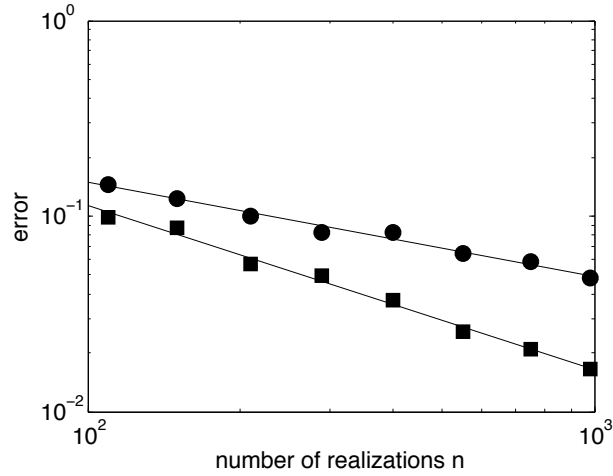yields the errors plotted as circles in Fig. 5.

Fig. 5. Option pricing comparison between pseudo-random and quasi-random numbers. Circles (squares) represent error in Monte-Carlo estimation of European call by following SDE paths using pseudo-random (quasi-random) numbers to generate increments. Settings were $X(0) = 10, K = 12, r = 0.05, \sigma = 0.5$, expiration time $T = 0.5$. The number of Wiener increments per trajectory was $m = 8$.

The results above were attained using pseudo-random numbers to generate the Wiener increments ΔW in the Euler-Maruyama method. An improvement in accuracy can be achieved by using *quasi-random* numbers instead.

By definition, standard normal pseudo-random numbers are created to be independent and identically-distributed, where the distribution is the standard normal distribution. For many Monte-Carlo sampling problems, the independence is not crucial to the computation. If that assumption can be discarded, then there are more efficient ways to sample, using what are called *low-discrepancy* sequences. Such quasi-random sequences are identically-distributed but not independent. Their advantage is that they are better at self-avoidance than pseudo-random numbers, and by essentially reducing redundancy they can deliver Monte-Carlo approximations of significantly reduced variance with the same number of realizations.

Consider the problem of estimating an expected value like (20) by calculating many realizations. By Property 2 of the Wiener process, the m increments $\Delta W_1, \ldots, \Delta W_m$ of each realization must be independent. Therefore along the trajectories, independence must be preserved. This is accomplished by using m different low-discepancy sequences along the trajectory. For example, the base-p lowdiscrepancy sequences due to Halton (1960) for m different prime numbers p can be used along the trajectory, while the sequences themselves run across different realizations.

Fig. 5 shows a comparison of errors for the Monte-Carlo pricing of the European call, using this approach to create quasi-random numbers. The low-discrepancy sequences produce nonindependent uniform random numbers, and must be run through the Box-Muller method (Box and Muller, 1958) to produce normal quasi-random numbers. The pseudo-random sequences show error proportional to $n^{-0.5}$, while the quasi-random appear to follow approximately $n^{-0.7}$.

More sophisticated low-discrepancy sequences, due to Faure, Niederreiter, Xing, and others, have been developed and can be shown to be more efficient than the Halton sequences. The chapter in this volume by Niederreiter (Niederreiter, 2010) describes the state of the art in generating such sequences.
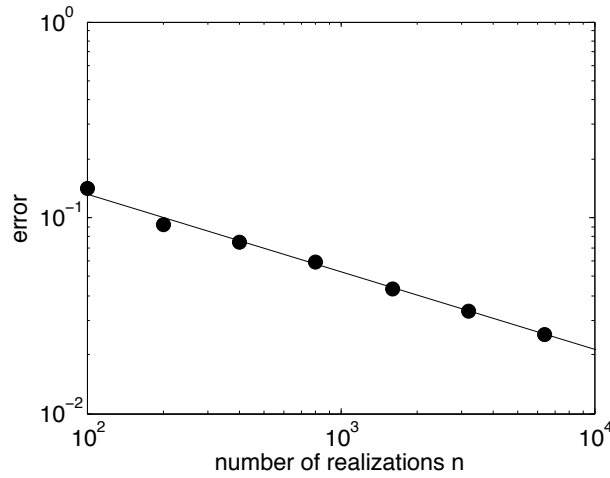

Fig. 6. Pricing error for barrier down-and-out call option. Error is proportional to the square root of the number of Monte-Carlo realizations.

The quasi-random approach can become too cumbersome if the number of steps m along each SDE trajectory becomes large. As an example, consider a *barrier* option, whose value is a function of the entire trajectory. For a *down-and-out* barrier call, the payout is canceled if the underlying stock drops below a certain level during the life of the option. Therefore, at time T the payoff is $\max(X(T) - K, 0)$ if $X(t) > L$ for $0 < t < T$, and 0 otherwise. For such an option, accurate pricing is dependent on using a relatively large number of steps m per trajectory. Results of a Monte-Carlo simulation of this modified call option are shown in Fig. 6, where the error was computed by comparison with the exact price

$$V(X,T) = C(X,T) - \left(\frac{X}{L}\right)^{1-\frac{2r}{\sigma^2}} C(L^2/X, T)$$

where $C(X, t)$ is the standard European call value with strike price K. The trajectories were generated with Euler-Maruyama approximations with pseudo-random number increments, where $m = 1000$ steps were used.

Other approaches to making Monte-Carlo sampling of trajectories more efficient fall under the umbrella of variance reduction. The idea is to calculate the expected value more accurately with fewer calls to the random number generator. The concept of *antithetic variates* is to follow SDE solutions in pairs, using the Wiener increment in one solutions and its negative in the other solution at each step. Due to the symmetry of the Wiener process, the solutions are equally likely. For the same number of random numbers generated, the standard error is decreased by a factor of $\sqrt{2}$.

A stronger version of variance reduction in computing averages from SDE trajectories can be achieved with *control variates*. We outline one such approach, known as variance reduction by delta-hedging. In this method the quantity that is being estimated by Monte-Carlo is replaced with an equivalent quantity of smaller variance. For example, instead of approximating the expected value of (20), the cash portion of the replicating portfolio of the European call can be targeted, since it must equal the option price at expiration.
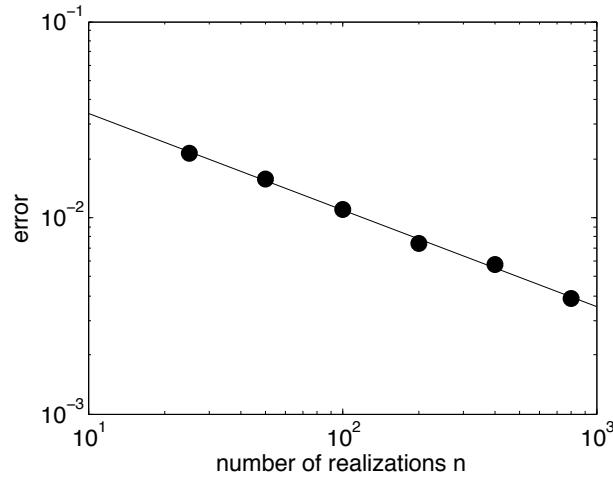

Fig. 7. Estimation errors for European call using control variates. Error is proportional to the square root of the number of Monte-Carlo realizations. Compare absolute levels of error with Fig. 5.

Let C_0 be the option value at time $t = 0$, which is the goal of the calculation. At the time $t = 0$, the seller of the option hedges by purchasing $\Delta = \frac{\partial C}{\partial X}$ shares of the underlying asset. Thus the cash account, valued forward to time T, holds

$$[C_0 - \frac{\partial C}{\partial X}(t_0)X_{t_0}]e^{r(T-t_0)}.$$

At time step $t = t_1$, the seller needs to hold $\Delta = \frac{\partial C}{\partial X}(t_1)$ shares, so after purchasing $\frac{\partial C}{\partial X}(t_1) - \frac{\partial C}{\partial X}(t_0)$ shares, the cash account (valued forward) drops by

$$\left[\frac{\partial C}{\partial X}(t_1) - \frac{\partial C}{\partial X}(t_0)]X_{t_1}\right] e^{r(T-t_1)}.$$

Continuing in this way, the cash account of the replicating portfolio at time T, which must be C_T, equals

$$C_0 e^{r(T-t_0)} - \sum_{k=0}^{N} \left[\frac{\partial C}{\partial X}(t_k) - \frac{\partial C}{\partial X}(t_{k-1})\right] X_{t_k} e^{r(T-t_k)}$$

$$= C_0 e^{r(T-t_0)} + \sum_{k=0}^{N-1} \frac{\partial C}{\partial X}(t_k)(X_{t_{k+1}} - X_{t_k} e^{r\Delta t}) e^{r(T-t_{k+1})}$$

and so

$$C_0 = e^{-r(T-t_0)} \left[C_T - \sum_{k=0}^{N-1} \frac{\partial C}{\partial X}(t_k)(X_{t_{k+1}} - X_{t_k} e^{r\Delta t}) e^{r(T-t_{k+1})}\right]$$

$$= e^{-r(T-t_0)} \left[C_T - \mathrm{cv}\right]$$

where cv denotes the control variate. Estimating the expected value of this expression yields fast convergence, as demonstrated in Fig. 7. Compared to Fig. 5, the errors in pricing of the European call are lower by an order of magnitude for a similar number of realizations. However, the calculation of the control variate adds significantly to the computational load, and depending on the form of the derivative, may add more overhead than is gained from the reduced variance in some cases.

6 Multifactor models

Financial derivatives that depend on a variety of factors should be modeled as a stochastic process that is driven by a multidimensional Wiener process. The various random factors may be independent, but more realistically, there is often correlation between the random inputs.

For multifactor Wiener processes $(W_t^1, \ldots, W_t^k)$, the generalization of Ito's Formula requires that (3) is replaced with

$$dt\ dt = 0$$
$$dt\ dW_t^i = dW_t^i\ dt = 0$$
$$dW_t^i\ dW_t^j = \rho_{ij}\ dt \tag{22}$$

where ρ_{ij} represents the statistical correlation between W_t^i and W_t^j. As usual, correlation ρ of two random variables X_1 and X_2 is defined as

$$\rho(X_1, X_2) = \frac{\text{cov}(X_1, X_2)}{\sqrt{V(X_1)}\sqrt{V(X_2)}}.$$

Note that $\rho(X_1, X_1) = 1$, and X_1 and X_2 are uncorrelated if $\rho(X_1, X_2) = 0$.

To construct discretized correlated Wiener processes for use in SDE solvers, we begin with a desired correlation matrix

$$R = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1k} \\ \vdots & & \vdots \\ \rho_{k1} & \cdots & \rho_{kk} \end{bmatrix}$$

that we would like to specify for Wiener processes $W^1, \ldots, W^k$. The matrix R is symmetric with units on the main diagonal. A straightforward way to create noise processes with a specified correlation is through the singular value decomposition (SVD) (see Sauer (2006) for a description). The SVD of R is

$$R = \Gamma \Lambda \Gamma^\top$$

where Γ is an orthogonal matrix ($\Gamma^{-1} = \Gamma^\top$), and Λ is a diagonal matrix with nonzero entries on the main diagonal.

Begin with k independent, uncorrelated Wiener processes $Z_1, \ldots, Z_k$, satisfying $dZ_i dZ_i = dt, dZ_i dZ_j = 0$ for $i \neq j$. Define the column vector $\mathbf{dW} = \Gamma \Lambda^{1/2} \mathbf{dZ}$, and check that the covariance matrix, and therefore the correlation matrix, of $\mathbf{dW}$ is

$$\begin{aligned} \mathbf{dW dW}^\top &= \Gamma \Lambda^{1/2} \mathbf{dZ} (\Gamma \Lambda^{1/2} \mathbf{dZ})^\top \\ &= \Gamma \Lambda^{1/2} \mathbf{dZ dZ}^\top \Lambda^{1/2} \Gamma^\top \\ &= \Gamma \Lambda \Gamma^\top dt = R \ dt \end{aligned}$$

For example, a two-asset market has correlation matrix

$$R = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = \begin{bmatrix} \text{corr}(W^1, W^1) & \text{corr}(W^1, W^2) \\ \text{corr}(W^2, W^1) & \text{corr}(W^2, W^2) \end{bmatrix}.$$

Since the SVD of this 2×2 correlation matrix is

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1+\rho & 0 \\ 0 & 1-\rho \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix},$$

we calculate

$$\begin{aligned} dW^1 &= \frac{\sqrt{1+\rho}}{\sqrt{2}} \ dZ^1 + \frac{\sqrt{1-\rho}}{\sqrt{2}} \ dZ^2 \\ dW^2 &= \frac{\sqrt{1+\rho}}{\sqrt{2}} \ dZ^1 - \frac{\sqrt{1-\rho}}{\sqrt{2}} \ dZ^2. \end{aligned} \tag{23}$$
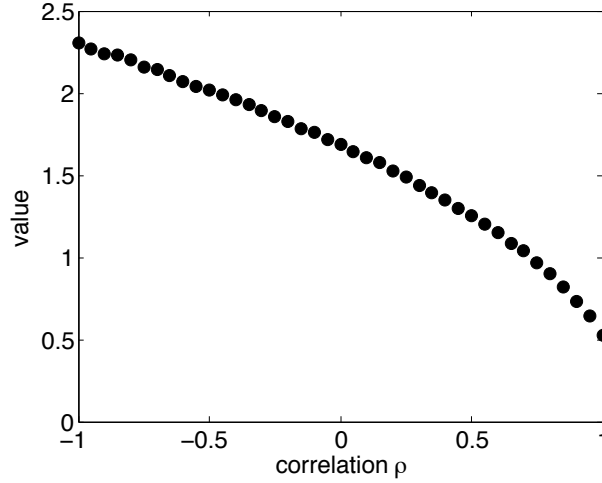
Fig. 8. European spread call value as a function of correlation. The Euler-Maruyama solver was used with multifactor correlated Wiener processes. The initial values of the underlying assets were $X_1(0) = 10, X_2(0) = 8$, the interest rate was $r = 0.05$, strike price $K = 2$, and expiration time $T = 0.5$.

With a change of variables, the correlation ρ can be generated alternatively as

$$dW^1 = dZ^1$$
$$dW^2 = \rho \, dZ^1 + \sqrt{1 - \rho^2} \, dZ^2. \tag{24}$$

As a simple example, we calculate the value of a European spread call using Monte-Carlo estimation of noise-coupled stochastic differential equations using a two-factor model. Assume there are two assets X_1 and X_2 satisfying arbitrage-free SDEs of form

$$dX_1 = rX_1 \, dt + \sigma_1 X_1 \, dW^1$$
$$dX_1 = rX_2 \, dt + \sigma_2 X_3 \, dW^2 \tag{25}$$

where $dW^1 dW^2 = \rho \, dt$, and that the payout at expiration time T is $\max\{X_1(T) - X_2(T) - K, 0\}$ for a strike price K. The Monte-Carlo approach means estimating the expected value

$$E(e^{-rT} \max\{X_1(T) - X_2(T) - K, 0\}).$$

Using either form (23) or (24) for the coupled Wiener increments in the Euler-Maruyama paths, the correct price can be calculated. Fig. 8 shows the dependence of the price on the two-market correlation ρ. As can be expected, the more the assets move in an anticorrelated fashion, the more probable the spread call will land in the money.

7 Summary

Numerical methods for the solution of stochastic differential equations are essential for the analysis of random phenomena. Strong solvers are necessary when exploring characteristics of systems that depend on trajectory-level properties. Several approaches exist for strong solvers, in particular Taylor and Runge-Kutta type methods, although both increase greatly in complication for orders greater than one.

In many financial applications, major emphasis is placed on the probability distribution of solutions, and in particular mean and variance of the distribution. In such cases, weak solvers may suffice, and have the advantage of comparatively less computational overhead, which may be crucial in the context of Monte-Carlo simulation.

Independent of the choice of stochastic differential equation solver, methods of variance reduction exist that may increase computational efficiency. The replacement of pseudorandom numbers with quasirandom analogues from low-discrepancy sequences is applicable as long as statistical independence along trajectories is maintained. In addition, control variates offer an alternate means of variance reduction and increases in efficiency in Monte-Carlo simulation of SDE trajectories.

References

Black F, Scholes M (1973) The pricing of options and corporate liabilities. Journal of Political Economy 81:637–654

Box GEP, Muller M (1958) A note on the generation of random normal deviates. Ann Math Stat 29:610–611

Burrage K, Burrage PM, Mitsui T (2000) Numerical solutions of stochastic differential equations - implementation and stability issues. J Comp Appl Math 125:171–182

Burrage K, Burrage PM, Tian T (2004) Numerical methods for strong solutions of stochastic differential equations: an overview Proc Roy Soc London A 460:373-402

Fishman GS (1996) Monte Carlo: concepts, algorithms, and applications. Springer, Berlin Heidelberg NewYork

Gentle JE(2003) Random number generation and Monte Carlo methods, 2nd ed. Springer, Berlin Heidelberg NewYork

Gikhman I, Skorokhod A (1972) Stochastic differential equations. Springer, Berlin.

Glasserman P (2004) Monte Carlo methods in financial engineering. Springer, NewYork

Halton JH (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik 2:84–90

Hellekalek P (1998) Good random number generators are (not so) easy to find. Math and Comp in Simulation 46:485-505

Higham DJ (2001) An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Review 43:525–546

Higham DJ, Kloeden P (2005) Numerical methods for nonlinear stochastic differential equations with jumps. Num Math 101:101-119

Higham DJ, Mao X, Stuart A (2002) Strong convergence of Euler-type methods for nonlinear stochastic differential equations. SIAM J Numer Anal 40:1041–1063

Hull JC (2002) Options, futures, and other derivatives, 5th ed. Prentice Hall, Upper Saddle River, NJ

Jentzen A, Kloeden P, Neuenkirch A (2008) Pathwise approximation of stochastic differential equations on domains: higher order convergence rates without global Lipschitz coefficients. Num Math 112:41–64

Klebaner F (1998) Introduction to stochastic calculus with applications. Imperial College Press, London

Kloeden P, Platen E (1992) Numerical solution of stochastic differential equations. Springer, Berlin

Kloeden P, Platen E, Schurz H (1994) Numerical solution of SDE through computer experiments. Springer, Berlin

Lamba H, Mattingly JC, Stuart A (2007) An adaptive Euler-Maruyama scheme for SDEs: convergence and stability. IMA J Num Anal 27:479-506

Marsaglia G, Zaman A (1991) A new class of random number generators. Ann Appl Prob 1:462–480

Marsaglia G, Tsang WW (2000) The ziggurat method for generating random variables. J Stat Soft 5:1–7

Maruyama G (1955) Continuous Markov processes and stochastic equations. Rend Circ Math Palermo 4:48–90

Milstein G (1988) A theorem on the order of convergence of mean-square approximations of solutions of stochastic differential equations. Theory Prob Appl 32:738–741

Milstein G (1995) Numerical integration of stochastic differential equations. Kluwer, Dordrecht

Milstein G, Tretyakov M (1997) Mean-square numerical methods for stochastic differential equations with small noises. SIAM J Sci Comp 18:1067–1087

Milstein G, Tretyakov M (2004) Stochastic numerics for mathematical physics. Springer, Berlin Heidelberg New York

Milstein G, Tretyakov M (2005) Numerical integration of stochastic differential equations with nonglobally Lipschitz coefficients. SIAM J Num Anal 43:1139-1154

Niederreiter H (1992) Random number generation and quasi-Monte Carlo methods. SIAM Publications, Philadelphia

Niederreiter H (2010) Low discrepancy simulation. This volume.

Oksendal B (1998) Stochastic differential equations: an introduction with applications, 5th ed. Springer, Berlin Heidelberg New York

Park S, Miller K (1988) Random number generators: good ones are hard to find. Comm ACM 31:1192–1201

Platen E (1987) Derivative-free numerical methods for stochastic differential equations. Lec Notes Control Inform Sci 96:187–193

Platen E (1999) An introduction to numerical methods for stochastic differential equations. Acta Num 8:197-246

Platen E, Wagner W (1982) On a Taylor formula for a class of Ito processes. Prob Math Stat 3:37–51

Romisch W, Winkler R (2006) Stepsize control for mean-square numerical methods for stochastic differential equations with small noise. SIAM J Sci Comp 28:604–625

Rubinstein RY (1981) Simulation and the Monte Carlo method. Wiley, New York

Rumelin W (1982) Numerical treatment of stochastic differential equations. SIAM J Num Anal 19:604–613

Saito Y, Mitsui T (1996) Stability analysis of numerical schemes for stochastic differential equations. SIAM J Num Anal 33:2254-2267

Sauer T (2006) Numerical Analysis. Pearson, Boston

Steele JM (2001) Stochastic calculus and financial applications. Springer, New York

Talay D, Tubaro L (1990) Expansion of the global error for numerical schemes solving stochastic differential equations. Stoch Anal Appl 8:483–509

Tocino A, Ardanuy R (2002) Runge-Kutta methods for numerical solution of stochastic differential equations. J Comp Appl Math 138:219-241