Weak Approximation of Stochastic Differential Equations and Application to Derivative Pricing

SYOITI NINOMIYA* & NICOLAS VICTOIR**

*Center for Research in Advanced Financial Technology, Tokyo Institute of Technology, 2-12-1 Ookayama Meguro-ku, Tokyo, Japan, **Mathematical Institute, 24-29 St Giles, Oxford, UK

(Received 24 April 2006; in revised form 5 February 2007)

ABSTRACT A new, simple algorithm of order 2 is presented to approximate weakly stochastic differential equations. It is then applied to the problem of pricing Asian options under the Heston stochastic volatility model.

2000 Mathematics Subject Classification, 65C30, 65C05

KEY WORDS: Heston model, numerical methods for stochastic differential equations, mathematical finance, quasi-Monte Carlo method

1. Introduction

1.1 The Problem and its Motivation

We consider a stochastic differential equation written in the Stratonovich form

\[ Y(t, x) = x + \int_0^t V_0(Y(s, x))ds + \sum_{i=1}^d \int_0^t V_i(Y(s, x)) \cdot dB_i^s, \]

where \( B = (B^1, ..., B^d) \) is a standard Brownian motion, and \( C^\infty_p(\mathbb{R}^N; \mathbb{R}^N) \) denotes the set of \( \mathbb{R}^N \)-valued smooth functions defined over \( \mathbb{R}^N \) whose derivatives of any order are bounded. In particular, we will use the classical notation \( Vf(x) = \sum_{i=1}^N V_i(x)(\partial f/\partial x_i)(x) \) for \( V \in C^\infty_p(\mathbb{R}^N; \mathbb{R}^N) \) and \( f \) a differentiable function from \( \mathbb{R}^N \) into \( \mathbb{R} \). This stochastic differential equation can be written in Itô form:
\[ Y(t, x) = x + \int_0^t \tilde{V}_0(Y(s, x))ds + \sum_{i=1}^d \int_0^t V_i(Y(s, x))dB^i_s, \]

where

\[ \tilde{V}_0^i(y) = V_0^i(y) + \frac{1}{2} \sum_{j=1}^d V_j^i(y). \]

Now, given a function \( f \) with some regularity, how can one approximate efficiently \( E[f(Y(1, x))] \)? It is equivalent to the following deterministic problem: if \( L \) is the differential operator \( V_0 + (1/2) \sum_{i=1}^d V_i^2 \) and \( u \) is the solution of the heat equation

\[ \frac{\partial u}{\partial t}(t, x) = Lu, \quad u(0, x) = f(x), \]

how does one approximate \( u(1, x) \) (which is equal to \( E[f(Y(1, x))] \)) by Feynman-Kac theorem (Ikeda and Watanabe, 1981)).

This problem has had a lot of attention because of its practical importance: it gives the evolution of the temperature in some media, and also represents the price of financial derivatives under stochastic financial models such as Black–Scholes (Black and Scholes, 1973).

Non-probabilistic methods to solve the PDE (such as finite difference methods) seem to work well only when \( L \) is elliptic and of low dimension. We refer the reader to (Lapeyre et al., 1998) for a more detailed discussion on the subject. We will focus in this paper on probabilistic methods.

Before continuing the general discussion, we give an example in finance to illustrate the problem.

**Example 1.1 (Asian Option under Heston model).** We consider the price of an Asian call option with maturity \( T \) and strike \( K \) written on an asset whose price process \( Y_1 \) satisfies the following two factor stochastic volatility model (Heston model (Heston, 1993)):

\[
\begin{align*}
Y_1(t, x) &= x_1 + \int_0^t \mu Y_1(s, x)ds + \int_0^t Y_1(s, x) \sqrt{Y_2(s, x)} dB^1(s), \\
Y_2(t, x) &= x_2 + \int_0^t \alpha \left( \theta - Y_2(s, x) \right) ds \\
&\quad + \int_0^t \beta \sqrt{Y_2(s, x)} \left( \rho dB^1(s) + \sqrt{1-\rho^2} dB^2(s) \right),
\end{align*}
\]

where \( x=(x_1, x_2) \in (\mathbb{R}_{>0})^2 \), \((B^1(t), B^2(t))\) is a 2-dimensional standard Brownian motion, and \( \alpha, \theta, \mu \) are some positive coefficients such that

\[ 2\alpha \theta - \beta^2 > 0 \]

to ensure the existence and uniqueness of a solution to our SDE which never hits 0 (Feller, 1951). \( \rho \) is the correlation between two factors \( B^1(t) \) and \( B^2(t) \) and \( |\rho| \leq 1 \). The
payoff of this option is \( \max(Y_3(T, x)/T-K, 0) \), where

\[
Y_3(t, x) = \int_0^t Y_1(s, x) \, ds.
\]

(3)

The price of this option becomes \( D \times E[\max(Y_3(T, x)/T-K, 0)] \) where \( D \) is the appropriate discount factor. Let \( Y(t, x) = (Y_1(t, x), Y_2(t, x), Y_3(t, x)) \). The SDEs (2) and (3) can be transformed into a Stratonovich form SDE:

\[
Y(t, x) = \sum_{i=0}^{2} \int_0^t V_i(Y(s, x)) \circ dB_i(s),
\]

(4)

where

\[
V_0((y_1, y_2, y_3)) = (y_1 \left( \mu - \frac{y_2}{2} - \frac{\rho \beta}{4}, x(0-y_2) - \frac{\beta^2}{4}, y_1 \right)
\]

\[
V_1((y_1, y_2, y_3)) = (y_1 \sqrt{y_2}, \rho \beta \sqrt{y_2}, 0)
\]

\[
V_2((y_1, y_2, y_3)) = (0, \beta \sqrt{(1-\rho^2)y_2}, 0).
\]

(5)

1.2 Notation

If \( V \) is a smooth vector field, i.e. an element of \( C_c^\infty(\mathbb{R}^N; \mathbb{R}^N) \), \( \exp(V)x \) denotes the solution at time 1 of the ordinary differential equation

\[
\frac{dz_t}{dt} = V(z_t), \quad z_0 = x.
\]

For \( x \in \mathbb{R}, \lfloor x \rfloor \) denotes the integer part of \( x \). For a random variable \( X \), \( \text{Var}[X] \) denotes the variance of \( X \).

1.3 Probabilistic Methods

1.3.1 Order 1. The most popular probabilistic method to approximate \( E[f(Y(1, x))] \) is called the Euler–Maruyama method (Kloeden and Platen, 1999). We first fix \( n \) independent \( d \)-dimensional random variables \( Z_1, ..., Z_n \) such that, if \( X \) denotes a standard normal random variables,

\[
E[p(Z_k)] = E[p(X)]
\]

(6)

for all polynomial \( p \) of degree less than or equal to 3. Then one defines recursively the following random variables:

\[
X_0^{(EM), n} = x,
\]

\[
X_{(k+1)/n}^{(EM), n} = X_{k/n}^{(EM), n} + \frac{1}{n} \sum_{i=1}^{d} V_i \left( X_{k/n}^{(EM), n} \right) Z_k + \frac{1}{n} \sum_{i=1}^{d} V_i \left( X_{k/n}^{(EM), n} \right) Z_k + 1.
\]
Then, one can show (Kloeden and Platen, 1999; Talay and Tubaro, 1990) that for an arbitrary $C^4$ function $f$

$$\left\| E\left[f\left(X^{(\text{EM}),n}_1\right)\right] - E[f(Y(1,x))]\right\| \leq C_f \frac{1}{n}.$$  

(7)

Of course, one needs an algorithm to compute $E\left[f\left(X^{(\text{EM}),n}_1\right)\right]$. If the $Z_k$ are constructed from Bernoulli random variables, $E\left[f\left(X^{(\text{EM}),n}_1\right)\right]$ is a discrete sum, but one would need to do $2^{nd}$ additions, which can be rather lengthy when $nd$ is large (one is then forced to use some Monte Carlo on a discrete measure). If the $Z_k$ are normal random variables, one then is forced to use Monte Carlo or quasi-Monte Carlo techniques. When $nd$ is large, quasi-Monte Carlo methods become less effective than Monte Carlo, but if $nd$ is not too high, quasi-Monte Carlo methods can be very efficient.

Another method with the same rate of convergence appeared in (Lyons and Victoir, 2004), and is called cubature on Wiener space of degree 3. It is defined with the following recursive formula:

$$X^{(\text{cub3}),n}_0 = x,$$

$$X^{(\text{cub3}),n}_{(k+1)/n} = \exp\left(\frac{1}{n} V_0 + \frac{1}{\sqrt{n}} \sum_{i=1}^d Z^i_{k+1} V_i\right) X^{(\text{cub3}),n}_{k/n}$$

Such an algorithm can be seen as a practical application of the Wong–Zakai theorem (Ikeda and Watanabe, 1981; Wong and Zakai, 1965), when the $Z_k$ are normal random variables.

If $B^n = (B^n_1, \ldots, B^n_d) (n \in \mathbb{N})$ is the piecewise linear approximation of the Brownian motion defined by $B^n = (\lfloor nt \rfloor + 1 - nt) B_{nt/n} + (nt - \lfloor nt \rfloor) B_{\lfloor nt \rfloor + 1/n}$, and $Y^n$ denotes the solution of the ordinary differential equation

$$Y^n_t = x + \int_0^t V_0(Y^n_s) ds + \sum_{i=1}^d \int_0^t V_i(Y^n_s) dB^{n,i}_s,$$

then the Wong–Zakai theorem states that $Y^n$ converges almost surely to $Y^x$. It is easy to see that $X^{(\text{cub3}),n}$ and $Y^n_1$ are equal in law, proving the convergence of the weak algorithm cubature on Wiener space of degree 3 (but this argument does not provide the rate of convergence).

**Remark 1.1.** In the algorithm cubature on Wiener space of degree 3, one has to solve numerically ODEs (unless one is lucky and one has a closed form solution!). One possibility is to take its Taylor approximation of order 1 for the approximation of $\exp(V)x$ (i.e. the Euler method for ODEs) and we fall back on the Milstein scheme. Not spending enough care on the approximating method of the ODEs to be solved can result in some catastrophic situations. A general case where that happens is when the diffusion is almost surely on a subset of $\mathbb{R}^N$, that is, does not fill the
whole space. If one has an approximation scheme which at some time provides an answer outside this set (which is what happen if one approximates badly the ODEs), the algorithm may go very wrong or even bug. Increasing \( n \) (which is costly) or artificial techniques can be implemented to solve this problem, while this can be overcome by taking an appropriately good approximation of the ODEs which have to be solved (we usually recommend a high order Runge–Kutta scheme, or an adaptive step size scheme, but this may depend on the particular SDE to approximate). We will give an example of this problem in Section 3.

**Remark 1.2.** Random variables which satisfy (6) are easy to find. One can take, for a fixed \( i, Z_i^j \) to be \( d \) independent Bernoulli or Gaussian random variables. A more elaborate choice of such random variables appeared in Lyons and Victoir (2004) and Stroud (1971).

**Remark 1.3.** Here, we have used the subdivision \((kln)_k \in \{0, \ldots, n\}\) of \([0, 1]\). It is not clear whether taking equal time steps is optimal or not. Recently, Kusuoka (Kusuoka, 2005) proved that the partitioning into equal time steps is optimal when we use the algorithm proposed in this paper. We do not want to address this problem in this paper, and we will always take subdivisions with equal time steps.

1.3.2 *Higher order.* A way to obtain approximations of higher order is based on the understanding of more terms in the stochastic Taylor formula (see Castell, 1993; Kloeden and Platen, 1999, for example). When the vector fields \( V_i \) commute, it is relatively easy to find a scheme of high order, see Kloeden and Platen (1999) and the references within. In the general case, one needs to understand how to approximate weakly the increments of the Brownian motion together with its first few iterated integrals. This was first successfully done, to our knowledge, in Kusuoka (2001), Liu and Li (2000), Talay (1990, 1995), and Kusuoka and Ninomiya (2004), and then generalized with the method cubature on Wiener space (Lyons and Victoir, 2004).

1.4 *Romberg Extrapolation*

Consider a nice scheme of order \( p \), that is, a scheme \( X_{k/n}^{(\text{ord } p), n} \) such that for smooth \( f \), there exists a constant \( K_f \) such that

\[
\left| E[f\left(X_{1}^{(\text{ord } p), n}\right)] - E[f(Y(1, x))] - K_f \frac{1}{n^p} \right| \leq C_f \frac{1}{n^{p+1}}.
\]

Then,

\[
\frac{2^p}{2^p - 1} E[f\left(X_{1}^{(\text{ord } p), 2n}\right)] - \frac{1}{2^p - 1} E[f\left(X_{1}^{(\text{ord } p), n}\right)]
\]

provides a scheme of order \( p+1 \). We refer once again to Talay and Tubaro (1990) for more details and the proof that the Euler-Maruyama scheme and its successive Romberg extrapolations are “nice” schemes. Recently, it was proved that our new algorithm presented below is a “nice” scheme (Kusuoka, 2005).
1.5 A Remark on the Monte Carlo Method

Let $W$ be a random variable. When we compute $E[W]$ by Monte Carlo method with $M$ samples, we consider a random variable $P_{i} \sim W_{i}$ where $W_{i}$'s are independent random variables whose distributions are identical to $W$'s. We denote this random variable by $MC(W, M)$. By virtue of the central limit theorem, we can consider that $MC(W, M)$ behaves as a normal random variable of mean $E[W]$ and variance $\text{Var}[W]/M$.

Let $X_{i}^{(\text{ord } p), n}$ denotes a scheme of order $p$ of the type above. To calculate $X_{i}^{(\text{ord } p), n}$ numerically, one needs to approximate an integral over a $nC(d)$ dimensional space ($C(d)$ denoting a function depending on $d$; for Euler or Cub3, $C(d)=d$. As we will see later, $C(d)=d+1$ for our new algorithm). If one uses the Monte-Carlo method to approximate this integral, and uses $M$ samples, the random variable $MC\left(f\left(X_{i}^{(\text{ord } p), n}\right), M\right)$ is considered. The situation is summarized by the following relations:

\begin{equation}
E[f(Y(1, x))] = E[f\left(X_{i}^{(\text{ord } p), n}\right)] + O(n^{-p}),
\end{equation}

\begin{equation}
MC\left(f\left(X_{i}^{(\text{ord } p), n}\right), M\right) \sim N\left(E[f\left(X_{i}^{(\text{ord } p), n}\right)], \frac{\text{Var}[f\left(X_{i}^{(\text{ord } p), n}\right)]}{M}\right).
\end{equation}

Two types of approximation errors are involved in this calculation. One is the difference between $E[f(Y(1, x))]$ and $E[f\left(X_{i}^{(\text{ord } p), n}\right)]$ and the other is the difference between $MC\left(f\left(X_{i}^{(\text{ord } p), n}\right), M\right)$ ($\omega$) and $E[f\left(X_{i}^{(\text{ord } p), n}\right)]$. In this paper, we call the former error the discretization error and the latter error integration error. (10) shows that we can consider the integration error of Monte Carlo method to be a normal random variable of mean 0 and variance $\text{Var}[f\left(X_{i}^{(\text{ord } p), n}\right)]/M$.

Because the difference between $\text{Var}[f\left(X_{i}^{(\text{ord } p), n}\right)]$ and $\text{Var}[f(Y(1, x))]$ is very small, we make the following remark.

**Remark 1.4.** As long as we use the Monte Carlo method for numerical approximation of $E[f(Y(1, x))]$, the number of sample points needed to attain the given accuracy is independent of the dimension of integration, namely the number $n$ of partitions and the order $p$ of the approximation scheme.

1.6 A Remark on the Quasi-Monte Carlo Method

Although there are some results which justify the quasi-Monte Carlo method and give theoretical error with respect to the number $M$ of sample points and the dimension of the integral domain, they provide little help with error estimation in practice when we apply the quasi-Monte Carlo method to weak approximation of SDEs (see Ninomiya and Tezuka, 1996; Paskov, 1997). The following observation seems to be widely accepted:
Remark 1.5. In contrast to the Monte Carlo case, the number of sample points needed by the quasi-Monte Carlo method for numerical approximation of \( E[f(Y(1, x))] \) depends heavily on the dimension of integration. The smaller the dimension, the smaller number of samples needed.

The integral that we have to approximate to obtain \( X_{1}^{\text{ord } 1, n} \) is on a space of dimension \( nC(d) \). If the numerical method is of high order and \( nC(d) \) is not too large, one can then use quasi-Monte Carlo with this numerical method to obtain a very fast algorithm.

Therefore, it seems optimal to look for a (simple) scheme of order greater than that of the Euler–Maruyama scheme (one), with \( C(d) \) remaining comparable to \( d \) (i.e. the \( C(d) \) of the Euler-Maruyama scheme). This is the object of this paper, where we suggest a new numerical scheme of order 2, with \( C(d)=d+1 \). We will show its efficiency by numerically pricing an Asian option under the Heston model.

2. Presentation of the New Algorithm

We present our new algorithm, of order 2.

2.1 The Algorithm

**Theorem 2.1.** Let \((\Lambda_i, Z_i) \in \{1, \ldots, n\} \) be \( n \) independent random variables, where each \( \Lambda_i \) is a Bernoulli random variable independent of \( Z_i \), which is a standard \( d \)-dimensional normal random variable. Define \( \{X_{k/n}^{\text{New}, n}\}_{k=0, \ldots, n} \) to be a family of random variables as follows:

\[
X_{0}^{\text{New}, n} = x,
X_{(k+1)/n}^{\text{New}, n} = \begin{cases} 
\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z_k V_1}{\sqrt{n}}\right) \cdots \exp\left(\frac{Z_k V_d}{\sqrt{n}}\right) X_{k/n}^{\text{New}, n} & \text{if } \Lambda_k = +1, \\
\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z_k V_1}{\sqrt{n}}\right) \cdots \exp\left(\frac{Z_k V_d}{\sqrt{n}}\right) X_{k/n}^{\text{New}, n} & \text{if } \Lambda_k = -1.
\end{cases}
\]

Then, for an arbitrary Lipschitz continuous function \( f \),

\[
\left| E\left[f\left(X_{1}^{\text{New}, n}\right)\right] - E[f(Y(1, x))] \right| \leq \frac{C_f}{n^2},
\]

that is, our new algorithm is of order 2.

To compute

\[
\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z_k V_1}{\sqrt{n}}\right) \cdots \exp\left(\frac{Z_k V_d}{\sqrt{n}}\right) \exp\left(\frac{V_0}{2n}\right) X_{k/n}^{\text{New}, n},
\]

one needs to solve \( d+2 \) ordinary differential equations. First along the vector field \( V_0 \) from \( t=0 \) to \( t=1/(2n) \) with starting point \( X_{k/n}^{\text{New}, n} \), then along \( V_d \) from \( t=0 \) to \( t=Z_k V_d / \sqrt{n} \) with starting point the solution of the ODE we have just solved, and we
repeat similar operations \(dt+2\) times. One would need an algorithm to solve this ODE numerically (unless one has a close form solution), and we, once again, strongly suggest that one pays a lot of attention to the quality of such an algorithm.

One of course will have to use an algorithm to approximate \(E[f(X_{1/n}^{(\text{New})}n)]\), but this is just a (difficult but classical, common to Euler algorithm for example) problem of integrating a function on a finite dimensional space. The simplest but quite effective method is to do some basic Monte Carlo simulation of the random variables \((\Lambda_i, Z_i)_{i \in \{1, \ldots, n\}}\). One could also simulate the random variables \((\Lambda_i, Z_i)_{i \in \{1, \ldots, n\}}\) with some quasi-Monte Carlo techniques, or replace the random variables \(Z_i\) with some discrete random variables with the right moment up to order 5. As this is a very classical problem and common to all the other probabilistic solutions to our numerical problem, we do not provide anymore precisions here.

**Proof.** We give a proof only for the case in which \(f\) is smooth. Using the results in Kusuoka (2005, 2001 and 2003), one can show the convergence of the algorithm with \(f\) Lipschitz continuous, under a condition on the vector fields weaker than Hörmander condition. We do not do it here to avoid writing a very technical paper.

The proof is quite classical, so we will not go into details. The reader should be convinced that the algorithm is of order 2 once we show that for \(f\) smooth enough,

\[
\left| E[f(X_{1/n}^{(\text{New})}n)] - E[f(Y(1/n, x))] \right| \leq C_f n^3.
\]

The error over \(n\) steps, from the Markov property of \(Y\), would then be \(n\) times \(n^{-3}\). We consider a smooth function \(f\). First observe that, from the Feynman-Kac theorem,

\[
\left| E[f(Y(1/n, x))] - \left( f(x) + \frac{1}{n} Lf(x) + \frac{1}{2n^2} L^2 f(x) \right) \right| \leq C_f n^{-3}.
\]

Developing \(L^2\), that means

\[
f(x) + \frac{1}{n} Lf(x) + \frac{1}{2n^2} L^2 f(x) = f(x) + \frac{1}{n} \left( V_0 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 \right) f(x)
\]

\[
+ \frac{1}{2n^2} \left( V_0^2 + \frac{1}{2} V_0 \sum_{i=1}^{d} V_i^2 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 V_0 + \frac{1}{4} \sum_{i,j=1}^{d} V_i^2 V_j^2 \right) f(x).
\]

Now we need to approximate \(E[f(X_{1/n}^{(\text{New})}n)]\). Using Taylor approximation of the ODEs involved, we quickly see that the absolute value of

\[
E\left[ f \left( \exp \left( \frac{1}{2n} V_0 \right) \exp \left( \frac{1}{\sqrt{n}} Z_k^1 V_1 \right) \cdots \exp \left( \frac{1}{\sqrt{n}} Z_k^d V_d \right) \exp \left( \frac{1}{2n} V_0 \right) x \right) \right]
\]
minus
\[ f(x) + \frac{1}{n} \left( V_0 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 \right) f(x) \]
\[ + \frac{1}{2n^2} \left( V_0^2 + \frac{1}{2} V_0 \sum_{i=1}^{d} V_i^2 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 V_0 + \frac{1}{4} \sum_{i=1}^{d} V_i^4 + \frac{1}{2} \sum_{i<j} V_i^2 V_j \right) f(x) \]
is bounded by \( C_f n^{-3} \). Inverting the order in which the vector fields are integrated, we obtain that the absolute value of
\[
E \left[ f \left( \exp \left( \frac{1}{2n} V_0 \right) \exp \left( \frac{1}{\sqrt{n}} Z_k^d V_d \right) \cdots \exp \left( \frac{1}{\sqrt{n}} Z_k^1 V_1 \right) \exp \left( \frac{1}{2n} V_0 \right) x \right) \right]
\]
minus
\[ f(x) + \frac{1}{n} \left( V_0 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 \right) f(x) \]
\[ + \frac{1}{2n^2} \left( V_0^2 + \frac{1}{2} V_0 \sum_{i=1}^{d} V_i^2 + \frac{1}{2} \sum_{i=1}^{d} V_i^2 V_0 + \frac{1}{4} \sum_{i=1}^{d} V_i^4 + \frac{1}{2} \sum_{i<j} V_i^2 V_j \right) f(x) \]
is bounded by \( C_f n^{-3} \). Adding up and dividing by 2, we obtain that
\[
\left| E \left[ f \left( X_{1/n}^{\text{New}} \right) \right] - E[f(Y(1/n, x))] \right| \leq \frac{C_f + C'_f}{n^3}.
\]
This algorithm could be seen in a non-trivial way as a particular case of the algorithm cubature on Wiener space of degree 5. One should also notice some common features with splitting methods.

2.2 Advantages of the Proposed Algorithm

There are some studies on higher order schemes (Glasserman, 2004; Kloeden and Platen, 1999). The algorithm which we proposed above is distinguished from existing ones by having the following advantages:

1. The algorithm does not care whether the vector fields \( \{ V_i \}_{i=0,1,\ldots,d} \) commute or not.
2. The algorithm is completely free from symbolical computations.
3. It is often the case that we can find analytic solutions of \( \exp(tV)x \). In such cases, using the algorithm, we can avoid problems caused by numerical approximation procedures.

From a practical point of view, points listed above are all important and sometimes critical.

For 1., many real problems are not commutative. 2. is critical in practice. In 3.1.1, we will see an example of 3.
3. Numerical Example: Application to Finance

In this section, we numerically compare the new algorithm with the Euler–Maruyama scheme and their Romberg extrapolation. We calculate the price of the option which we presented in Example 1.1 by using the proposed algorithms. We set $T=1$, $K=1.05$, $\mu=0.05$, $\alpha=2.0$, $\beta=0.1$, $\theta=0.09$, and $(x_1, x_2)=(1.0, 0.09)$. We ignore $D$ in this experiment.

3.1 Implementation of the Algorithm

We apply the algorithm introduced in Section 2 to this problem.

3.1.1 Solutions of the ODEs. We can easily obtain $\exp(sV_1)$ and $\exp(sV_2)$ ($s \in \mathbb{R}$) as follows:

$$\exp(sV_1)(y_1, y_2, y_3) = \left( y_1 e^{y_2 + \rho y_2^2/4}, \left( \frac{\rho \beta s}{2} + \sqrt{y_2^2} \right)^2, y_3 \right),$$

$$\exp(sV_2)(y_1, y_2, y_3) = \left( y_1, \left( \frac{s\beta \sqrt{1-\mu^2}}{2} + \sqrt{y_2^2} \right)^2, y_3 \right).$$

As there exists no closed form solution to $\exp(sV_0)$, we are forced to use an approximation and we choose:

$$\exp(sV_0)(y_1, y_2, y_3) = (g_1(s), g_2(s), g_3(s)),$$

where

$$g_1(s) = y_1 \exp \left( \left( \mu - \frac{\rho \beta^2}{4} - \frac{J}{2} \right) s + \frac{y_2 - J}{2\alpha} (e^{-\alpha s} - 1) \right),$$

$$g_2(s) = J + (y_2 - J)e^{-\alpha s},$$

$$g_3(s) = y_3 + \frac{y_1 (e^{As} - 1)}{A} + O(s^3),$$

$$J = \theta - \frac{\beta^2}{4\alpha}, \quad \text{and} \quad A = \mu - \frac{\rho \beta^2}{4} - \frac{y_2}{2}.$$

The error compared with the true solution is $O(t^3)$ in small time $t$, creating an additional error of $O(n^{-3})$ at every step of the algorithm, but as the error of our scheme at every step was also $O(n^{-3})$, taking the above approximation of $\exp(sV_0)$ does not alter the convergence rate of the algorithm.

Following the same discussion, it is easy to see that we have to approximate $\exp(tV_0)$ in such a way that the order of the error produced is $O(t^4)$ when we use Romberg extrapolation, which we introduced in 1.4, together with the algorithm. In this experiment, we approximate $g_3(s)$ by the traditional order 4 Runge–Kutta method when we use Romberg extrapolation.
Here, we see that one of the advantages of this algorithm over the Euler–Maruyama scheme is the one mentioned in Remark 1.1. When we apply the Euler–Maruyama scheme to this process (2), it may happen that the square volatility process \( (Y_k)_{k=1}^{EM} \) becomes negative, and the algorithm then fails at the next step (as we will have to take its square root). On the other hand, equations (12) and (14) show that our new algorithm does not share this problem. One can avoid this problem by putting an absolute value under the square root (see Bossy and Diop, 2004).

3.1.2 A remark on general implementation. In general, it is not always possible to obtain the closed form solution to \( \exp(sV) \). Even in such cases, it is not difficult to implement our new algorithm. All we have to do is to find an approximation of \( \exp(sV_0) \) whose error is \( O(s^3) \) and approximations of \( \exp(sV_i), (i \neq 0) \) whose errors are \( O(s^6) \) This can be achieved by Runge–Kutta-like methods and we can find some examples of them in Butcher (1987).

3.1.3 Application of the quasi-Monte Carlo method. Our new algorithm has the virtue that the application of the quasi-Monte Carlo method to this algorithm is possible in a straightforward way, once we embed \((A_i, Z_i)_{i=1, \ldots, n} \) into \([0, 1)^{(d+1)}\). This is an advantage of the algorithm over algorithms proposed in Ninomiya (2003a, 2003b) and Kusuoka and Ninomiya (2004) which also enable one to proceed to a higher order weak approximation.

3.2 Comparison with Euler–Maruyama Scheme

We compare numerically the new algorithm with the Euler–Maruyama scheme with and without Romberg extrapolation. Such methods involve, as we saw, approximation of an integral over a finite dimensional space; we will make these approximations using the Monte Carlo method and the quasi-Monte Carlo method.

There are many studies on acceleration of Monte Carlo methods (Glasserman, 2004; Lapeyre and Temam, 2001) but we choose the crude Euler–Maruyama scheme with and without Romberg extrapolation as the only competitors for the following reasons:

1. The two algorithms are applicable to the same type of problems, both in a very easy way.
2. Almost all variance reduction and acceleration techniques that one can apply to the Euler–Maruyama scheme are also applicable to the new algorithm.

In this experiment, we consider

\[
E[\max(Y_3(T, x)/T - K, 0)] = 6.0473907415 \times 10^{-2}
\]

which is obtained by our new algorithm with extrapolation, quasi-Monte Carlo, \( n=96+48, \) and \( M=8.0 \times 10^9. \)

3.2.1 Discretization error. Figure 1 shows the relation between the number of partitions in the discretization of the interval \([0, 1]\) (\( n \) in the description of the algorithm) and the error of the algorithms. We observe that to achieve \( 10^{-4} \)
accuracy, the new method with Romberg extrapolation requires $n=6$, the new method needs $n=12$, while the Euler–Maruyama scheme with Romberg extrapolation needs $n=24$, and the simple Euler–Maruyama scheme needs $n=2000$. In all algorithms, the time taken is proportional to $n \times M$, where $M$ is the number of sample points.

3.2.2 Convergence error from Monte Carlo. We have already mentioned in 1.5 that the convergence performance of the Monte Carlo method is independent of the number of partitions. We can see in Figure 2 that in this experiment this statement holds. This figure also shows that to achieve $10^{-4}$ accuracy with 95% confidence level (2$sigma$) using Monte Carlo method, we need over $10^8$ sample points.

3.2.3 Convergence error from quasi-Monte Carlo and Monte Carlo. Figure 2 also shows that the performance of the convergence of the quasi-Monte Carlo method depends on the number $n$ of partitions and on the algorithms. Figure 2 seems to show that the quasi-Monte Carlo method outperforms the Monte Carlo method specially when used with the new algorithm and that the algorithm needs $2 \times 10^5$ sample points for $10^{-4}$ accuracy, the algorithm with extrapolation $2 \times 10^5$ sample points, and Euler–Maruyama with extrapolation $5 \times 10^6$ sample points when the quasi-Monte Carlo method is used.
3.2.4 Performance comparison with respect to time taken. The time required for each method to achieve $10^{-4}$ accuracy is shown in Table 1. We find that the new algorithm with Romberg extrapolation and the quasi-Monte Carlo method provides the fastest calculation. The new algorithm with Romberg extrapolation and quasi-Monte Carlo is about 80 times faster than Euler–Maruyama scheme with Romberg extrapolation and quasi-Monte Carlo. We also see that even without Romberg extrapolation, the new algorithm is still faster than any boosted Euler–Maruyama method.

![Figure 2. Convergence error from quasi-Monte Carlo and Monte Carlo.](image)

### Table 1. #Partition, #Sample, and CPU time required for $10^{-4}$ accuracy.

| Method                  | #Partition | #Sample | CPU time (sec) |
|-------------------------|------------|---------|----------------|
| E-M+MC                  | 2000       | $10^8$  | $1.72 \times 10^5$ |
| E-M+Extrpltn+MC         | 16+8       | $10^8$  | $2.06 \times 10^3$ |
| New+MC                  | 4+2        | $10^8$  | $1.24 \times 10^3$ |
| New+Extrpltn+MC         | 4+2        | $10^8$  | $6.2 \times 10^2$  |
| E-M+Extrpltn+QMC        | 16+8       | $5 \times 10^6$ | $1.28 \times 10^2$ |
| New+QMC                 | 12         | $2 \times 10^5$ | 3.3       |
| New+Extrpltn+QMC        | 4+2        | $2 \times 10^5$ | 1.73      |
Finally we would like to mention Remark 1.4 and Remark 1.5 again: having a high order method, as well as being obviously more efficient, allows the efficiency of quasi-Monte Carlo methods to further improve the accuracy of the method.

Acknowledgement

This research was partially supported by the Japanese Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 15540110, 2003.

References

Black, F. and Scholes, M. (1973) The Pricing of Options and Corporate Liabilities, *Journal of Political Economy*, 81, pp. 637–59.

Bossy, Mireille and Diop, Awa. (2004) An efficient discretisation scheme for 1-dimensional sides with a diffusion coefficient function of the form $|x|^a$, $a \in [1/2, 1)$, *RR 5396, INRIA*, December.

Butcher, J. C. (1987) *The Numerical Analysis of Ordinary Differential Equations* (Chichester: John Wiley & Sons).

Castell, F. (1993) Asymptotic expansion of stochastic flows, *Probability theory and related fields*, 96(2), pp. 225–239.

Feller, W. (1951) Two singular diffusion problems, *Annals of Mathematics*, 54, pp. 173–182.

Glasserman, P. (2004) *Monte Carlo Methods in Financial Engineering* (New York: Springer-Verlag).

Heston, S. L. (1993) A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *The Review of Financial Studies*, 6, pp. 327–343.

Ikeda, N. and Watanabe, S. (1981) *Stochastic differential equations and diffusion processes* (North Holland/Kodansha).

Kusuoka, S. (2001) Approximation of Expectation of Diffusion Process and Mathematical Finance, In: T. Sunada (Ed.), *Advanced Studies in Pure Mathematics, Proceedings of Final Taniguchi Symposium, Nara 1998*, 31, pp. 147–165.

Kusuoka, S. (2003) Malliavin Calculus Revisited, *Journal of Mathematical Sciences The University of Tokyo*, 10, pp. 261–277.

Kusuoka, S. (2005) Kusuoka Scheme and Gaussian type approximation, *Presentation at “Mathematical Finance Seminar in Graduate School of Mathematical Sciences The University of Tokyo (11June 2005)”*.

Kusuoka, S. and Ninomiya, S. (2004) A new simulation method of diffusion processes applied to Finance, In: J. Akahori, S. Ogawa & S. Watanabe (Eds.), *Stochastic processes and application to mathematical finance, Proceedings of the Ritsumeikan International Symposium*, pp. 233–253, (Singapore: World Scientific).

Lapeyre, B. and Temam, E. (2001) Competitive Monte Carlo methods for the pricing of Asian options, *Journal of Computational Finance*, 5, pp. 39–59.

Lapeyre, B. Pardoux, E. and Sentis, R. (1998) *Méthodes de Monte-Carlo pour les équations de transport et de diffusion* (Mathematics and Applications 29). (Berlin: Springer-Verlag).

Liu, X. Q. and Li, C. W. (2000) Weak approximation and extrapolations of stochastic differential equations with jumps, *SIAM Journal on Numerical Analysis*, 37, pp. 1747–1767.

Lyons, T. and Victoir, N. (2004) Cubature on Wiener Space, *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 460, pp. 169–198.

Ninomiya, S. (2003a) A new simulation scheme of diffusion processes: Application of the Kusuoka approximation to Finance Problems, *Mathematics and Computers in Simulation*, 623–6, pp. 479–486.

Ninomiya, S. (2003b) A partial sampling method applied to the Kusuoka approximation, *Monte Carlo Methods and Applications*, 9, pp. 27–38.

Ninomiya, S. and Tezuka, S. (1996) Toward real-time pricing of complex financial derivatives, *Applied Mathematical Finance*, 3, pp. 1–20.
Paskov, S. H. (1997) New methodologies for Valuing Derivatives, In: S. Pliska & M. Dempster (Eds.), Mathematics of Derivative Securities, pp. 545–582, (Cambridge University Press).

Stroud, A. H. (1971) Approximate calculation of multiple integrals (Englewood Cliffs, New Jersey: Prentice Hall).

Talay, D. (1990) Second-order discretization schemes of stochastic differential systems for the computation of the invariant law, Stochastics and Stochastics Reports, 29, pp. 13–36.

Talay, D. (1995) Simulation of Stochastic Differential Systems, In: P. Kree & W. Wedig (Eds.), Probabilistic Methods in Applied Physics, LNP 451, pp. 54–96 (Springer-Verlag).

Talay, D. and Tubaro, L. (1990) Expansion of the global error for numerical schemes solving Stochastic Differential Equations, Stochastic Analysis and Applications, 8, pp. 483–509.

Wong, E. and Zakai, M. (1965) On the relation between ordinary and stochastic differential equations, Intern. J. Engng. Sci., 3, pp. 213–229.