WEAK APPROXIMATION OF STOCHASTIC DIFFERENTIAL EQUATIONS AND APPLICATION TO DERIVATIVE PRICING

SYOITI NINOMIYA[1] AND NICOLAS VICTOIR[2]

Abstract. The authors present a new simple algorithm to approximate weakly stochastic differential equations in the spirit of [9][13]. They apply it to the problem of pricing Asian options under the Heston stochastic volatility model, and compare it with other known methods. It is shown that the combination of the suggested algorithm and quasi-Monte Carlo methods makes computations extremely fast.

1. Introduction

1.1. **The Problem and its Motivation.** We consider a stochastic differential equation written in the Stratonovich form

$$Y(t,x) = x + \int_0^t V_0\left(Y(s,x)\right)\, ds + \sum_{i=1}^{d} \int_0^t V_i\left(Y(s,x)\right) \circ dB_s^i, \tag{1}$$
$$V_j \in C_b^\infty\left(\mathbb{R}^N;\mathbb{R}^N\right),$$

where $B = \left(B^1, \cdots, B^d\right)$ is a standard Brownian motion, and $C_b^\infty\left(\mathbb{R}^N;\mathbb{R}^N\right)$ 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 $Vf(x) = \sum_{i=1}^N V^i(x)\left(\partial f/\partial x_i\right)(x)$ for $V \in C_b^\infty(\mathbb{R}^N;\mathbb{R}^N)$ and f a differentiable function from $\mathbb{R}^n$ into $\mathbb{R}$. This stochastic differential equation can be written in Itô form:

$$Y(t,x) = x + \int_0^t \tilde{V}_0\left(Y(s,x)\right)\, ds + \sum_{i=1}^{d} \int_0^t V_i\left(Y(s,x)\right)\, dB_s^i,$$

where

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

Now, given a function f with some regularity, how can one approximate efficiently $E\left[f\left(Y(1,x)\right)\right]$? 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

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

2000 *Mathematics Subject Classification.* 65C30, 65C05.

Key words and phrases. Heston model, numerical methods for stochastic differential equations, mathematical finance, quasi-Monte Carlo method.

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

how does one approximate $u(1,x)$ (which is equal to $E[f(Y(1,x))]$ by Feynman-Kac theorem [7]).

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 price of financial derivatives under stochastic financial models such as Black-Scholes [1].

Non-probabilistic methods to solve the PDE (such as finite difference methods) seem to only work well when L is elliptic and in low dimension. We refer to [11] for a more detailed discussion on the subject. We will focus in this paper on probabilistic methods.

1.2. **Notation.** If V is a smooth vector field, i.e. an element of $C_b^\infty\left(\mathbb{R}^N;\mathbb{R}^N\right)$, $\exp(V)(x)$ 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.

1.3. **Probabilistic Methods.**

1.3.1. *Order 1.* The most popular probabilistic method to approximate $E\left[f\left(Y_1^x\right)\right]$ is called the Euler-Maruyama method [8]. We first fix n independent d-dimensional random variables $Z_1, \cdots, Z_n$ such that, if X denotes a standard normal random variables, $E[p(Z_k)] = E[p(X)]$ for all polynomial of degree less than or equal to 3. [1] Then one defines recursively the following random variables:

$$X_0^{(\mathrm{EM}),n} = x,$$
$$X_{(k+1)/n}^{(\mathrm{EM}),n} = X_{k/n}^{(\mathrm{EM}),n} + \frac{1}{n}\tilde{V}\left(X_{k/n}^{(\mathrm{EM}),n}\right) + \frac{1}{\sqrt{n}}\sum_{i=1}^{d} V_i\left(X_{k/n}^{(\mathrm{EM}),n}\right) Z_{k+1}^i.$$

Then, one can show [8][17] that for all nice enough function f

$$\left\| E\left[f\left(X_1^{(\mathrm{EM}),n}\right)\right] - E[f(Y(1,x))] \right\| \leq C_f \frac{1}{n}. \tag{2}$$

[2] Of course, one needs an algorithm to compute $E\left[f\left(X_1^{(\mathrm{EM}),n}\right)\right]$. If the Z_k are constructed from Bernoulli random variables, $E\left[f\left(X_1^{(\mathrm{EM}),n}\right)\right]$ 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 do some Monte-Carlo on a discrete measure). If the N_i are normal random variables, one then is forced to do use some Monte Carlo or quasi-Monte Carlo techniques. When nd is big, quasi-Monte Carlo method become less effective than Monte-Carlo, but if nd is not too high, quasi-Monte Carlo method can be very efficient.

[1] Such random variable 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 [13][16].

[2] Here, we have used the subdivision $(k/n)_{k\in\{0,\cdots,n\}}$ of $[0,1]$. In no way taking equal time steps is optimal. We do not want to address this problem in this paper, and we will always take subdivisions with equal time steps.

Another method with the same rate of convergence appeared in [13], and is called cubature on Wiener space of degree 3. It is defined with the following recursive formula:

$$X_0^{(\mathrm{cub3}),n} = x,$$
$$X_{(k+1)/n}^{(\mathrm{cub3}),n} = \exp\left(\frac{1}{n}V_0 + \frac{1}{\sqrt{n}}\sum_{i=1}^{d} Z_{k+1}^i V_i\right)\left(X_{k/n}^{(\mathrm{cub3}),n}\right)$$

Such algorithm can be seen as a practical application of the Wong-Zakai theorem [7][18], when the Z_k are normal random variables.
If $B_t^n = (B_t^{n,1}, \dots B_t^{n,d}) \quad (n \in \mathbb{N})$ is the piecewise linear approximation of the Brownian motion defined by

$$B_t^n = (\lfloor nt \rfloor + 1 - nt)\, B_{\lfloor nt \rfloor/n}^n + (nt - \lfloor nt \rfloor)\, B_{(\lfloor nt \rfloor+1)/n}^n,$$

and Y^n denotes the solution of the ordinary differential equation

$$Y_t^n = x + \int_0^t V_0(Y_s^n)\, ds + \sum_{i=1}^{d} \int_0^t V_i(Y_s^n)\, dB_s^{n,i},$$

then the Wong-Zakai theorem states that Y^n converges almost surely to Y^x. It is easy to see that $X_1^{(\mathrm{cub3}),n}$ and Y_1^n 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 close form solution!). One possibility is to take its Taylor approximation of order* 1 *for the approximation of* $\exp(V)(x)$ *and we fall back on an Euler scheme. Taking a better approximation (Taylor approximation of order* 2*) will give a scheme sometimes described as 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.*

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 [3] and [8] for example).When the vector fields V_i commute, it is relatively easy to find a scheme of high order, see [8] 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 [9][12], see also [10], and then generalized with the method cubature on Wiener space [13].

1.4. **Romberg Extrapolation.** Consider a nice scheme of order p, that is, a scheme $X_{k/n}^{(\mathrm{ord}\, p),n}$ such that for smooth f, there exists a constant K_f such that

$$\left| E\left[f\left(X_1^{(\mathrm{ord}\, p),n} \right) \right] - E\left[f\left(Y(1,x) \right) \right] - K_f \frac{1}{n^p} \right| \leq C_f \frac{1}{n^{p+1}}.$$

Then,

$$\frac{2^p}{2^p - 1} E\left[f\left(X_1^{(\mathrm{ord}\, p),2n} \right) \right] - \frac{1}{2^p - 1} E\left[f\left(X_1^{(\mathrm{ord}\, p),n} \right) \right] \tag{3}$$

provides a scheme of order $p+1$. We refer once again to [17] for more details and the proof that the Euler-Maruyama scheme and its successive Romberg extrapolations are "nice" schemes.

It is strongly conjectured that the cubature on Wiener space algorithms and the algorithm presented below are "nice" schemes, but this has not been proved yet.

1.5. **An informal remark on the Monte Carlo method.** Let $X_1^{(\mathrm{ord}\, p),n}$ denotes a scheme of order p of the type above. To calculate $X_1^{(\mathrm{ord}\, p),n}$ numerically, one need 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 integrals, and uses M samples, we denote this new approximation by $X_1^{(\mathrm{ord}\, p,\mathrm{mc}\, M),n}$. Then, roughly speaking,

$$\left| E\left[f\left(X_1^{(\mathrm{ord}\, p,\mathrm{mc}\, M),n} \right) \right] - E\left[f\left(Y(1,x) \right) \right] \right| = O\left(n^{-p} \right) + \frac{\Gamma}{\sqrt{M}}$$

where Γ is a random variable with bounded variance. Therefore, one should, asymptotically at least, effectuate $M = Kn^{2p}$ simulations, to keep the precision of the scheme of order p. To compute $E\left[f\left(X_1^{(\mathrm{ord}\, p,\mathrm{mc}\, M),n} \right) \right]$, one actually generates $L = MnC(d)$ points. Therefore, choosing $M = Kn^{2p}$, we obtain $L = KC(d)n^{2p+1}$ to obtain a precision of $C'_d n^{-p} = C''_d L^{-p/(2p+1)}$. Therefore, an algorithm of order p provides, using Monte Carlo, an approximation whose error is of order $L^{-p/(2p+1)}$ where L is the number of generated random variables. In our discussion, we have assumed that the variance of Γ does not depend on the dimension of the integral domain and M, which is a fact which seems to be confirmed by experiments.

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, those results help little for error estimation in practice when we apply the quasi-Monte Carlo method to weak approximation of SDEs (see [14] or [15]).

The quasi-Monte Carlo method is usually more efficient than the Monte Carlo method in dimension not too high. The integral that we have to approximate to obtain $X_1^{(\mathrm{ord}\, p),n}$ is on a space of dimension $nC(d)$. If the numerical method is of high order and $C(d)$ is not too big, 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 the one of the Euler scheme (one), with $C(d)$ remaining comparable to d (i.e. the $C(d)$ of the Euler 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 prove 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.

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

$$
\begin{aligned}
&X^{(\mathrm{New}),n}_0 = x,\\
&X^{(\mathrm{New}),n}_{(k+1)/n} =\\
&\begin{cases}
\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z^1_k V_1}{\sqrt{n}}\right)\cdots\exp\left(\frac{Z^d_k V_d}{\sqrt{n}}\right)\exp\left(\frac{V_0}{2n}\right)\left(X^{(\mathrm{New}),n}_{k/n}\right) & \text{if } \Lambda_k = +1,\\
\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z^d_k V_d}{\sqrt{n}}\right)\cdots\exp\left(\frac{Z^1_k V_1}{\sqrt{n}}\right)\exp\left(\frac{V_0}{2n}\right)\left(X^{(\mathrm{New}),n}_{k/n}\right) & \text{if } \Lambda_k = -1.
\end{cases}
\end{aligned}
\tag{4}
$$

Then, for all $f \in C^\infty_b(\mathbb{R}^N)$,

$$\left|E\left[f\left(X^{(\mathrm{New}),n}_1\right)\right] - E\left[f\left(Y^x_1\right)\right]\right| \leq \frac{C_f}{n^2},$$

that is, our new algorithm is of order 2.

A few remarks before all: To compute

$$\exp\left(\frac{V_0}{2n}\right)\exp\left(\frac{Z^1_k V_1}{\sqrt{n}}\right)\cdots\exp\left(\frac{Z^d_k V_d}{\sqrt{n}}\right)\exp\left(\frac{V_0}{2n}\right)\left(X^{(\mathrm{New}),n}_{k/n}\right),$$

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 $X^{(\mathrm{New}),n}_{k/n}$, then along V_d from $t = 0$ to $t = Z^d_k/\sqrt{n}$ with starting point the solution of the ODE we have just solved, and we repeat similar operations $d + 2$ 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 algorithm.

One of course will have to use an algorithm to approximate $E\left[f\left(X^{(\mathrm{New}),n}_1\right)\right]$, 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 $(\Lambda_i, Z_i)_{i\in\{1,\cdots,n\}}$. One could also simulate the random variables $(\Lambda_i, Z_i)_{i\in\{1,\cdots,n\}}$ 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. 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,

$$\left|E\left[f\left(X^{(\mathrm{New}),n}_{1/n}\right)\right] - E\left[f\left(Y(1/n, x)\right)\right]\right| \leq \frac{C_f}{n^3}.$$

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 of the Feynman-Kac theorem,

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

Developing L^2, that means

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

Now we need to approximate $E\left[f\left(X_{1/n}^{(\mathrm{New}),n}\right)\right]$. 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

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

is bounded by $\tilde{C}_f n^{-3}$. 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

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

is bounded by $\tilde{C}_f n^{-3}$. Adding up and dividing by 2, we obtain that

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

□

Remark 2.1. *Following* [9], *one could show the convergence of the algorithm with f just continuous, under a condition on the vector fields weaker than Hörmander condition.*

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.

3. Numerical Example: Application to Finance

In this section, we numerically compare our new algorithm to the Euler-Maruyama scheme and their Romberg extrapolation. We calculate 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 [6]):

$$\begin{aligned} Y_1(t,x) &= x_1 + \int_0^t \mu Y_1(s,x)\, ds + \int_0^t Y_1(s,x)\sqrt{Y_2(s,x)}\, dB^1(s), \\ Y_2(t,x) &= x_2 + \int_0^t \alpha\left(\theta - Y_2(s,x)\right) ds + \int_0^t \beta\sqrt{Y_2(s,x)}\, dB^2(s), \end{aligned} \tag{5}$$

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 α, θ, μ are some positive coefficients such that $2\alpha\theta - \beta^2 > 0$ to ensure the existence and uniqueness of a solution to our SDE [5]. The payoff of this option is $\max\left(Y_3(T,x)/T - K, 0\right)$, where

$$Y_3(t,x) = \int_0^t Y_1(s,x)\, ds. \tag{6}$$

The price of this option becomes $D \times E\left[\max\left(Y_3(T,x)/T - K, 0\right)\right]$ where D is the appropriate discount factor. 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. Let $Y(t,x) = {}^t(Y_1(t,x), Y_2(t,x), Y_3(t,x))$. We transform the SDEs (5) and (6) into a Stratonovich form SDE:

$$Y(t,x) = \sum_{i=0}^{2} \int_0^t V_i(Y(s,x)) \circ dB^i(s), \tag{7}$$

where

$$\begin{aligned} V_0\left({}^t(y_1,y_2,y_3)\right) &= {}^t\left(y_1\left(\mu - \frac{y_2}{2}\right),\ \alpha(\theta - y_2) - \frac{\beta^2}{4},\ y_1\right) \\ V_1\left({}^t(y_1,y_2,y_3)\right) &= {}^t\left(y_1\sqrt{y_2},\ 0,\ 0\right) \\ V_2\left({}^t(y_1,y_2,y_3)\right) &= {}^t\left(0,\ \beta\sqrt{y_2},\ 0\right). \end{aligned} \tag{8}$$

3.1. **Implementation of the algorithm.**

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

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

$$\begin{aligned} \exp(sV_1)\left({}^t(y_1,y_2,y_3)\right) &= {}^t\left(y_1 e^{s\sqrt{y_2}}, y_2, y_3\right), \\ \exp(sV_2)\left({}^t(y_1,y_2,y_3)\right) &= {}^t\left(y_1, \left(\frac{\beta s}{2} + \sqrt{y_2}\right)^2, y_3\right). \end{aligned} \tag{9}$$

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

$$\exp(sV_0)\left({}^t(y_1,y_2,y_3)\right) = {}^t\left(y_1(t), y_2(t), y_3(t)\right), \tag{10}$$

where

$$
\begin{aligned}
y_1(t) &= y_1 \exp\left(\left(\mu - \frac{J}{2}\right)s + \frac{y_2 - J}{2\alpha}\left(e^{-\alpha s} - 1\right)\right),\\
y_2(t) &= J + (y_2 - J)\, e^{-\alpha s},\\
y_3(t) &= y_3 + \frac{y_1\left(e^{As} - 1\right)}{A} + O\left(s^3\right),\\
J &= \theta - \frac{\beta^2}{4\alpha}, \quad A = \mu - \frac{y_2}{2}, \quad \text{and} \quad B = \frac{\alpha(y_2 - J)}{4}.
\end{aligned}
\tag{11}
$$

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

Here, we see that one of the advantages of this algorithm over the Euler-Maruyama scheme is the one we mentioned in Remark 1.1. When we apply the Euler-Maruyama scheme to this process (5), it may happens that the square volatility process $(Y_2)_k^{(\mathrm{EM}),n}$ 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 (9) and (11) show that our new algorithm does not share this problem. [3]

3.1.2. *A remark on general implementation.* In general, it is not always possible to obtain the closed form solution to $\exp(sV_i)$. 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 [2].

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 straight forward way, once we embed $(\Lambda_i, Z_i)_{i\in\{1,\dots,n\}}$ into $[0,1)^{n(d+1)}$.

3.2. **Comparison to Euler-Maruyama scheme.** We compare numerically our new algorithm to 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 do these approximation using the Monte Carlo method and the Quasi-Monte Carlo method. Here, we consider $E\left[\max\left(Y_3(T,x)/T - K,\, 0\right)\right] = 6.04720626353478 \times 10^{-2}$ which is obtained by our new algorithm with extrapolation, quasi-Monte Carlo, $n = 256 + 128$, and $M = 1.1 \times 10^9$.

[3] There exists a way of avoiding this problem with the Euler-Maruyama scheme [4].

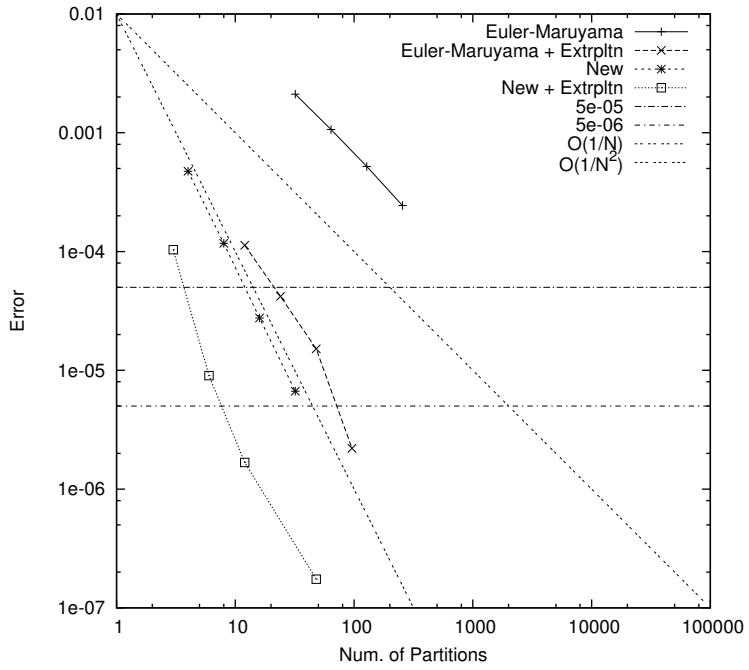

FIGURE 1. Error coming from the discretization

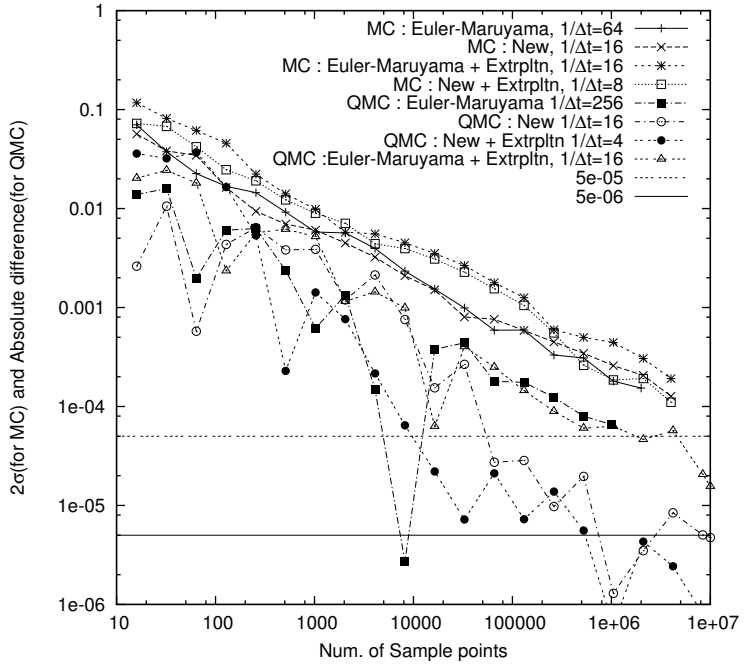

FIGURE 2. Convergence Error from quasi-Monte Carlo and Monte Carlo

Method	#Partition	#Sample	CPU time (sec)
E-M + MC	2000	10^8	1.09×10^5
E-M + Extrpltn + MC	16 + 8	10^8	2.20×10^3
New + MC	16	10^8	3.2×10^3
New + Extrpltn + MC	4 + 2	10^8	1.4×10^3
E-M + Extrpltn + QMC	16 + 8	5×10^6	1.10×10^2
New + QMC	16	5×10^4	1.6
New + Extrpltn + QMC	4 + 2	10^4	1.4×10^{-1}

FIGURE 3. #Partition, #Sample, and CPU time required for 4 digits accuracy.

3.2.1. *Discretization Error.* Figure 1 shows the relation between the number of partitions in our discretization of the interval $[0, 1]$ (n in the description of the algorithm) and the error of the algorithms. We observe that to achieve four digits accuracy, our new method with Romberg extrapolation requires $n = 6$, our new method needs $n = 16$, while the Euler-Maruyama scheme with Romberg extrapolation needs $n = 24$, and the simple Euler-Maruyama scheme needs $n \geq 2000$. In all algorithms, consumed time 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 four digits accuracy with 95% confidence level (2σ) by using Monte Carlo method, we need over 10^8 sample points. We can also see in this figure that the Monte Carlo errors which come from algorithms boosted by the Romberg extrapolation become greater than those of the original algorithms.

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 our new algorithm and that the algorithm needs 5×10^4 sample points for four digits accuracy, the algorithm with extrapolation 10^4 sample points, and Euler-Maruyama with extrapolation 5×10^6 sample points when we use the quasi-Monte Carlo method.

3.2.4. *Performance comparison with respect to consumed time.* The elapsed time of all methods required for four digits accuracy is shown in Figure 3. We find in this figure that our new algorithm with Romberg extrapolation and the quasi-Monte Carlo method provides the fastest calculation. Our new algorithm with Romberg extrapolation and quasi-Monte Carlo is about 800 times faster than Euler-Maruyama scheme with Romberg extrapolation and quasi-Monte Carlo. We also see that even without Romberg extrapolation, our new algorithm is still faster than any boosted Euler-Maruyama method.

REFERENCES

[1] BLACK, F., AND SCHOLES, M. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy 81* (1973), 637–59.

[2] Butcher, J. C. *The Numerical Analysis of Ordinary Differential Equations*. John Wiley & Sons, 1987.
[3] Castell, F. Asymptotic expansion of stochastic flows. *Probability theory and related fields 96(2)* (1993), 225–239.
[4] Diop, A. An efficient discretisation scheme for 1-dimensional sdes with a diffusion coefficient function of the form $|x|^a, a \in [1/2, 1)$. *RR 5396, INRIA* (December 2004).
[5] Feller, W. Two singular diffusion problems. *Annals of Mathematics 54* (1951), 173–182.
[6] Heston, S. L. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies 6* (1993), 327–343.
[7] Ikeda, N., and Watanabe, S. *Stochastic differential equations and diffusion processes*. North Holland/Kodansha, 1981.
[8] Kloeden, P. E., and Platen, E. *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.
[9] Kusuoka, S. Approximation of Expectation of Diffusion Process and Mathematical Finance. In *Advanced Studies in Pure Mathematics, Proceedings of Final Taniguchi Symposium, Nara 1998* (2001), T. Sunada, Ed., vol. 31, pp. 147–165.
[10] Kusuoka, S., and Ninomiya, S. A new simulation method of diffusion processes applied to Finance. In *Stochastic processes and application to mathematical finance, Proceedings of the Ritsumeikan International Symposium* (2004), J. Akahori, S. Ogawa, and S. Watanabe, Eds., World Scientific, pp. 233–253.
[11] Lapeyre, B., Pardoux, E., and Sentis, R. *Méthodes de Monte-Carlo pour les équations de transport et de diffusion* (Mathematics and Applications 29). Springer-Verlag, 1998.
[12] Liu, X. Q., and Li, C. W. Weak approximation and extrapolations of stochastic differential equations with jumps. *SIAM Journal on Numerical Analysis 37* (2000), 1747–1767.
[13] Lyons, T., and Victoir, N. Cubature on Wiener Space. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 460* (2004), 169–198.
[14] Ninomiya, S., and Tezuka, S. Toward real-time pricing of complex financial derivatives. *Applied Mathematical Finance 3* (1996), 1–20.
[15] Paskov, S. H. New methodologies for Valuing Derivatives. In *Mathematics of Derivative Securities (Edited by S .Pliska and M. Dempster)* (1997), S. Pliska and M. Dempster, Eds., Cambridge University Press, pp. 545–582.
[16] Stroud, A. H. *Approximate calculation of multiple integrals*. Prentice-Hall, 1971.
[17] Talay, D., and Tubaro, L. Expansion of the global error for numerical schemes solving Stochastic Differential Equations. *Stochastic Analysis and Applications 8* (1990), 483–509.
[18] Wong, E., and Zakai, M. On the relation between ordinary and stochastic differential equations. *Intern. J. Engng. Sci. 3* (1965), 213–229.

[1]Center for Research in Advanced Financial Technology, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552 Japan
E-mail address: ninomiya@craft.titech.ac.jp

[2]Mathematical Institute, 24-29 St Giles', Oxford, OX1 3LB, UK
E-mail address: victoir@maths.ox.ac.uk