VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
INTERNATIONAL UNIVERSITY
DEPARTMENT OF MATHEMATICS

GRADUATION THESIS

NUMERICAL SIMULATION OF STOCHASTIC DIFFERENTIAL EQUATIONS AND APPLICATIONS IN FINANCIAL OPTION VALUATION

Submitted in partial fulfillment of the requirements for the degree of
BACHELOR OF SCIENCE in
FINANCIAL ENGINEERING AND RISK MANAGEMENT

Student's Name: Kha Kim Bao Han
Student's ID: MAMAIU14065
Thesis Supervisor: Nguyen Minh Quan, PhD

Ho Chi Minh City, Vietnam
June 2018

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
INTERNATIONAL UNIVERSITY

GRADUATION THESIS

NUMERICAL SIMULATION OF STOCHASTIC DIFFERENTIAL EQUATIONS AND APPLICATIONS IN FINANCIAL OPTION VALUATION

Submitted to DEPARTMENT OF MATHEMATICS
In partial fulfillment of the requirements for the degree of
BACHELOR OF SCIENCE in
FINANCIAL ENGINEERING AND RISK MANAGEMENT

Student's Name:	**Kha Kim Bao Han**
Student's ID:	**MAMAIU14065**
Thesis Supervisor:	**Nguyen Minh Quan, PhD**

Ho Chi Minh City, Vietnam
June 2018

NUMERICAL SIMULATION OF STOCHASTIC DIFFERENTIAL EQUATIONS AND APPLICATIONS IN FINANCIAL OPTION VALUATION

By

KHA KIM BAO HAN

Submitted to DEPARTMENT OF MATHEMATICS,
INTERNATIONAL UNIVERSITY, HO CHI MINH CITY
in partial fulfillment of requirements for the degree of
BACHELOR OF SCIENCE in
FINANCIAL ENGINEERING AND RISK MANAGEMENT
June 2018

Signature of Student: ______________________

Kha Kim Bao Han

Certified by: ______________________

Nguyen Minh Quan, Ph.D.
Thesis Supervisor

Approved by: ______________________

Assoc. Prof. Pham Huu Anh Ngoc
Head of Department of Mathematics

NUMERICAL SIMULATION OF STOCHASTIC DIFFERENTIAL EQUATIONS AND APPLICATIONS IN FINANCIAL OPTION VALUATION

By

KHA KIM BAO HAN

Submitted to DEPARTMENT OF MATHEMATICS,
INTERNATIONAL UNIVERSITY, HO CHI MINH CITY
in partial fulfillment of requirements for the degree of
BACHELOR OF SCIENCE in
FINANCIAL ENGINEERING AND RISK MANAGEMENT
June 2018

Abstract

This thesis is aim to study numerical solution methods for stochastic differential equations. We introduce a review of crucial knowledges, a description of our main numerical methods including Euler-Maruyama, Milstein and higher schemes, and the understanding of the order of convergence for SDE solvers. In the remainding parts of this thesis, we present applications of numerical simulation to Monte Carlo sampling for financial options pricing and interest rates dynamics.

Key words: SDE's, Euler-Maruyama, Milstein, Monte Carlo, MATLAB, strong and weak convergence

Acknowledgements

First of all, I would like to express my very profound gratitude to my supervisor, Dr. Nguyen Minh Quan for his conscientious guidance, valuable suggestions and advices during the process. He not only thoroughly provide me the specialized resoures of research but also steered me in the right direction whenever I confront with struggles. At last, I am thankful him from the bottom of my heart for believing and giving me this opportunity.

Additionally, I am sincerely grateful all the lecturers at Department of Mathematics, International University, Vietnam National University, Ho Chi Minh City. Under their significant efforts and enthusiastic teaching, I could gain an understanding of a variety of helpful academic knowledge, which are the fundamental background for me to carry out this thesis.

Last but not least, I would like to denote my deep appreciation for my family and my university friends for their devoted support and encouragement during these memorial four years, especially throughout the progress to accomplish this thesis.

Contents

List of Figures

Listings

Chapter 1

Introduction

A stochastic differential equation (SDE) is a differential equation which involves a random element, resulting in a solution as a random process. SDE models play an essential role in a wide range of areas. More specifically, in physics, SDEs are used to investigate the effects of random excitations on phenomena. In engineering, they are used in filtering and control theory. In biology, they contribute to model the effects of stochastic variability in reproduction and environment on populations. Particularly in finance, they have been used to understand the behavior and the output of stock prices, short-term interest rate and their volatilities. There are a few SDEs such as the geometric Brownian motion for stock prices posessing the theoretical solution whose probability density function can be defined. However, in some cases, analytical solutions of SDEs are not available which requires to utilize numerical methods to approximate the solution. That is the reason why there is a numerous research about numerical simulation of SDEs in recent years [1, 2, 3, 4, 5]. The main content of this thesis is based on the content of the paper by Prof. Desmond Higham (SIAM Review, 2001) [1].

The crucial objective of this thesis is to investigate a few important numerical methods for solving SDEs and their applications. We study the Euler-Maruyama's method, Milstein's method (of order 1), and the higher order accuracy method, particularly the second order scheme. More specifically, we theoretically study the derivation of the numerical schemes and numerically implement them in simulations. Moreover, we use the technique of Monte Carlo simulations to understand the solution of SDEs at the final time. We then apply these schemes to study the models in practical problems such as in finance.

We first start by studying the basic background of stochastic calculus which requires to derive and construct the numerical schemes such as Brownian motion, basic stochastic integral, Itô lemma. Second, we implement these schemes and use the technique of Monte Carlo simulation, which refers to repeated random sampling, in order to simulate the SDEs. The SDE studied in this thesis has the following common form [1]:

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \quad X(0) = X_0, \quad 0 \le t \le T, \tag{1.1}$$

where f and g are scalar functions and $W(t)$ is regarding Brownian motion (Wiener process), which will be discussed in more detail in the thesis. The function f is indicated as the drift of the evolution of $X(t)$, and g is called the diffusion function. In equation (1.1), the first term on the right hand side describes the deterministic rate of change while the second term describes the stochastic process affecting the evolution of $X(t)$. Without the stochastic term, i.e., the second term, equation (1.1) becomes deterministic differential equation: $dX(t) = f(X(t))dt,\ X(0) = X_0$.

One of the most important applications of SDEs in finance is the specification of the stochastic process (model) to examine the behaviour of an asset. Here, we use the term asset to describe any financial object whose value is known at present but is able to change in the future. Typical examples are: shares in a company; commodities such as gold, oil or electricity; currencies, for example, the value of $100 US in VND.

In this thesis, we numerically study the solution of a number of stochastic differential equations intensively used in modeling asset prices, including:

- The linear SDE arises as an asset price model in financial mathematics, which is modelled by [6]

  $$dX(t) = \mu X(t)dt + \sigma X(t)dW(t), \quad X(0) = X_0, \tag{1.2}$$

  where the expected rate of return μ and volatility σ are assumed to be constants. The stochastic process $X(t)$ follows a Geometric Brownian Motion (GBM). Here, $X(t)$ represents the asset price at time t, and we assume that the time-zero asset price, $X(0)$, is known. The former term on the right hand sight of equation (1.2) illustrates deterministic trends while the latter models a set of unpredictable events happening during this motion.

- A common SDE used in econometrics, the mean-reverting square root process [4], also named the Cox-Ingersoll-Ross model was introduced by J. Cox, J. Ingersoll and S. Ross [7]:

  $$dX(t) = \lambda(\mu - X(t))dt + \sigma\sqrt{X(t)}dW(t), \quad 0 \le t \le T, \tag{1.3}$$

where λ, μ and σ are positive constants. The parameter λ corresponds to the speed of adjustment, μ as the mean and σ as volatility. $X(t)$ models short-term interest rate dynamics. The drift factor $\lambda(\mu - X(t))$ ensures the mean reversion of the interest rate towards the long run value μ, and speed of adjustment controlled by the strictly positive parameter λ. The term $\sigma\sqrt{X(t)}dW(t)$ is the standard deviation factor, avoiding the possibility of negative interest rates for all positive values of λ and μ. When the rates $X(t)$ gets close to zero, the term$\sigma\sqrt{X(t)}dW(t)$ becomes very small, which eliminates the effect of the random element, and the interest rate behaviour becomes dominated by the drift factor.

The rest of the thesis is organized as follows. In Chapter 2, we present the literature review about numerical methods of SDEs. In Chapter 3, we study the fundamental knowledge of stochastic calculus required to derive the numerical schemes to solve SDEs, especially the Brownian motion, the Itô and Stratonovich integral. The contents of chapter 4 are respectively the numerical schemes, including the Euler- Maruyama, Milstein scheme with the concept of strong and weak convergence and the higher order scheme. In this chapter, we also examine the applications of EM and Milstein's method in simulating the solution of equations (1.2), (1.3) to evaluate option price and interest rates. In chapter 5, we demonstrate further simulation results and compare these numerical algorithms by studying the code in MATLAB. Finally, chapter 6 is reserved for the conclusions, discussions, and potential idea for future studies.

Chapter 2

Literature Review

Because of crucial applications in a variety of fields including mechanics, chemistry, biology, medicine, population dynamic, economics and finance, etc, as we mentioned before, SDE models have been concerned by more and more scientists all over the world for decades.

Early research on SDEs was conducted by Einstein to describe Brownian motion, which is a model of random behaviour, in The Annus mirabilis papers (1905) [8]. SDEs have been intensively studied, especially followed upon by Langevin with the general form of SDEs in physical science called "Langevin equation" (1908) [9]:

$$m\frac{d^2x}{dt^2} = -\lambda\frac{dx}{dt} + \eta(t).$$

This equation describe Brownian motion with the apparently random movement of a particle in a fluid due to collisions with the molecules of the fluid. The noise term $\eta(t)$ in physical contexts corresponds to stochastic terms in stochastic differential equations. This equation can be written in the form $dx = -rxdt + \sigma dW_t$, [27] where r, σ is positive constants. Until 1927, Brownian motion officially proposed by Robert Brown (1828)[10] while he was looking through a microscope at particles trapped in cavities inside pollen grains in water, he noted that the particles moved through the water.

There are numerous different approaches to the numerical solution of SDEs. A method by Boyce (1978) [11] is very general to investigate general random systems by Monte-Carlo simulations without using the particular structure of SDEs. Kushner (1977) [12] carried out the discretization of both time and space variables, so the approximating processes are then finite state Markov chains which seem to be effective mainly in low dimensional problems with bounded domains.

The simulation of sample paths of the time discrete approximations on the digital computer seems to be applied widely to solve SDEs. Early theoretical and simulation studies by Clements and Anderson (1973) [13] showed that not all heuristic time discrete approximations of equation (1.2) converge in a useful sense.

Another way of classifying numerical methods for SDEs is to compare them with strong and weak Taylor approximations. The increments of such approximations are obtained by shortening the stochastic Taylor formula, which was derived in Wagner and Platen (1978) [14] by a repeated application of the Ito formula. A Stratonovich version of the stochastic Taylor formula was presented in Kloeden and Platen (1991) [15].

There are two main objectives in the simulation of the sample path of a process solution of an SDE: the whole sample path or the expected value of some functions of the process. The later can be achieved by considering weak approximations, which has been presented by Talay (1984) [16], and Milstein (1978) [17].

The difficulty of Taylor approximations is that they require to determine many derivatives. The methods of Runge - Kutta in the strong sense have been proposed, for instance, by Klauder and Petersen (1985) [5], Rümelin (1982) [18]. On the other hand, Talay [16] Milstein [17], and Kloeden and Platen (1992) [3] present Runge - Kutta schemes in the weak sense.

The estimation of SDE's parameters based on the observational data of the process also has great importance in practical applications and substantial attention in the financial econometrics, especially in the last ten years (2007,2010) [19, 20].

SDEs of jump-diffusion type captures the dynamics of the effect of event-driven uncertainty, receive much attention in financial and economic modeling, see Merton (1976) [21] or Cont and Tankov (2004) [22].

A variety of work on the existence of solutions for SDEs is explored by Zanzotto (1997) [23]. Another approach that has been reported to use pseudo-differential operators [24].

Chapter 3

Introduction to Stochastic Calculus

In this chapter, we present the fundamental background of stochastic calculus which requires to derive and construct the numerical schemes studied in this thesis. We first start by describing the Brownian motion, which plays a crucial role in stochastic modelling, and then demonstrate its implementing in simulation.

3.1 Brownian Motion

Definition 3.1.1. *A scalar standard Brownian motion (Wiener process), over* $[0, T]$ *is a random variable* $W(t)$ *and satisfies the following three conditions:*
1. $W(0) = 0$ *(with probability 1).*
2. $W(t) - W(s) \sim \sqrt{t-s}\ N(0, 1)$ *for* $0 \leq s < t \leq T$*, where* $N(0, 1)$ *indicates a standard normal random variable.*
3. For $0 \leq s < t < u < v \leq T$*, the increments* $W(t) - W(s)$ *and* $W(v) - W(u)$ *are independent.*

In order to serve computation purposes, the discretized Brownian motion is considered where $W(t)$ is specified at discrete t values. Then, we set $\Delta t = T/N$, for positive integer N called grid size (or mesh size) and W_j denotes $W(t_j)$ with $t_j = j\Delta t$. From the 3 conditions above:

$$W_j = W_{j-1} + dW_j, \quad j = 1, 2, ..., N, \tag{3.1}$$

where $dW_j \sim \sqrt{\Delta t}N(0, 1)$ and independent.

Listing 3.1: MATLAB code to demonstrate discretized Brownian path

```
randn('state',100)
T=1;N=1000;dt=T/N;
dW=zeros(1,N);
dW=sqrt(dt)*randn(1,N);
W=cumsum(dW);
plot([0:dt:T],[0,W],'r-')
xlabel('t','Fontsize',16)
ylabel('W(t)','Fontsize',16,'Rotation',0)
```

The above MATLAB code illustrates one simulation of discretized Brownian motion over [0,1] with $N = 1000$. The MATLAB function *randn* returns a uniform random number between 0 and 1. The command *randn('state',200)* set the state and hence subsequent runs will produce the same output. By condition 2 of Definition 3.1.1, we compute *dW=sqrt(dt)*randn(1,N)*.

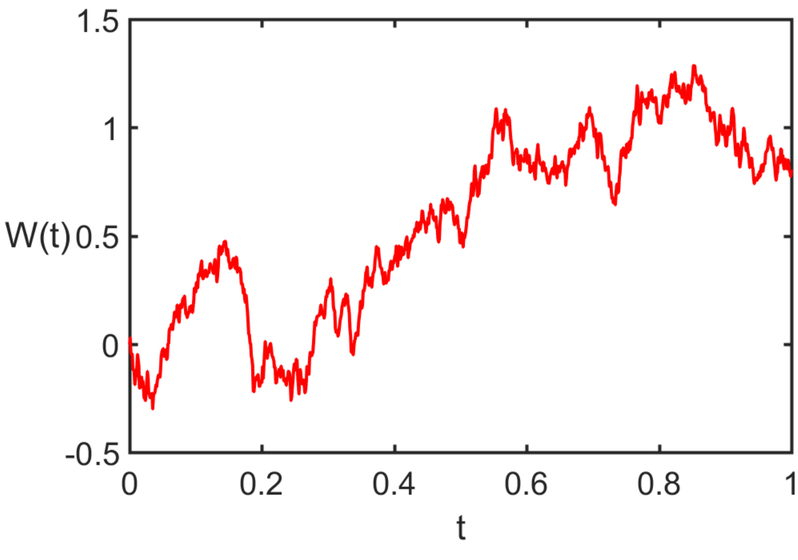

Figure 3.1: Discretized Brownian path.

3.2 Stochastic Calculus

In this section, we represent a number of necessary concepts which support the understanding of the next following sections. We start by defining the limit of a sequence of random variables which can be used to define the stochastic integral.

a) Mean-square convergence

Definition 3.2.1. *A sequence of random variables X_n converges to a random variable X if and only if*

$$\lim_{n\to\infty} \mathrm{E}\left[(X_n - X)^2\right] = 0.$$

Notation:

$$ms - \lim_{n\to\infty} X_n = X.$$

b) Moment generating function

Definition 3.2.2. *Moment generating function of a random variable X is a function $M_X : \mathbb{R} \to [0, \infty)$ given by*

$$M_X(t) = \mathrm{E}(e^{tX})$$

provided that the expectation exists.

Particularly, we have:
-If X is continuous, then

$$M_X(t) = \int_{-\infty}^{\infty} e^{tx} f_X(x)dx.$$

-If X is discrete, then

$$M_X(t) = \sum_{x\in X} e^{tx} R(X = x)dx.$$

For f_X is pdf of normal distribution $N(\mu, \sigma)$

$$M_X(t) = e^{\mu t + \frac{1}{2}\sigma t^2}. \tag{3.2}$$

Theorem 3.2.1. *If X has moment generating function $M_X(t)$, then :*

$$\mathrm{E}(X^n) = M_X^{(n)}(0), \quad \textit{where} \;\; M_X^{(n)}(0) = \frac{d^n}{dt^n} M_X(t)\mid_0 .$$

That is, the n-th moment is equal to the n-th derivative of the moment generating function evaluated at $t = 0$.

Proof Asumming that

$$\begin{aligned}\frac{d}{dt}M_X(t) =& \frac{d}{dt}\int_{-\infty}^{\infty} e^{tx} f_X(x)dx \quad = \int_{-\infty}^{\infty}\left(\frac{d}{dt}e^{tx}\right) f_X(x)dx \\ =& \int_{-\infty}^{\infty}\left(xe^{tx}\right) f_X(x)dx = \mathrm{E}(Xe^{tX}).\end{aligned} \tag{3.3}$$

Hence, evaluating equation (3.3) at t=0, we get

$$\frac{d}{dt}M_X(t)\mid_0 = \mathrm{E}(Xe^{tX})\mid_0 = \mathrm{E}(X).$$

So it is true for $n = 1$.
Suppose it is true for any $n = k \in \mathbb{N}$, it means

$$\frac{d^k}{dt^k}M_X(t)\mid_0 = \mathrm{E}(X^k e^{tX})\mid_0 = \mathrm{E}(X^k).$$

We have to prove it is true for $n = k+1$. Indeed

$$\begin{aligned}\frac{d^{k+1}}{dt^{k+1}}M_X(t) = \frac{d}{dt}\left(\frac{d^k}{dt^k}M_X(t)\right) =& \frac{d}{dt}\left(\mathrm{E}(X^k e^{tX})\right) \\ =& \frac{d}{dt}\int_{-\infty}^{\infty}\left(x^k e^{tx}\right) f_X(x)dx \\ =& \int_{-\infty}^{\infty}\left(\frac{d}{dt}x^k e^{tx}\right) f_X(x)dx \\ =& \int_{-\infty}^{\infty}\left(x^{k+1}e^{tx}\right) f_X(x)dx \\ =& \mathrm{E}(X^{k+1}e^{tX}).\end{aligned}$$

Hence

$$\frac{d^{k+1}}{dt^{k+1k}}M_X(t)\mid_0 = \mathrm{E}(X^{k+1}e^{tX})\mid_0 = \mathrm{E}(X^{k+1}).$$

So the theorem is proved.

c) Riemann sum

The stochastic integral can be defined as a limit, in sense of mean square limit, of a Riemann sum as the mesh size approaching to 0. The Riemann sum is defined as follows. If $[0, T]$ is an inteval, a partition P of [0,T] is a finite collection of points in

[0,T]:

$$\mathrm{P} := \{0 = \mathrm{t}_0 \leq \mathrm{t}_1 \leq \mathrm{t}_2 \leq ... \leq \mathrm{t}_{\mathrm{N}-1} \leq \mathrm{t} = \mathrm{T}\}. \tag{3.4}$$

Let the mesh size of P be N, $t_{j+1} - t_j = \Delta t = T/N$, $j = 0, ..., N-1$.
For fixed $0 \leq \lambda \leq 1$ and P a given partition of $[0, T]$, set

$$\tau_j = \lambda t_j + (1-\lambda)\, t_{j+1}, \quad 0, \quad j = 0, ..., N-1. \tag{3.5}$$

For such a partition P and for $0 \leq \lambda \leq 1$, we define

$$R := R(P, \lambda) := \sum_{j=0}^{N-1} W(\tau_j)(W(t_{j+1}) - W(t_j)).$$

This is the corresponding Riemann sum approximation of $\int_{t_0}^{t} W(s) dW(s)$.

d) Stochastic calculus

Lemma 3.2.1. *Let* $[t_0, t]$ *be an interval in [0,∞) and suppose*

$$P = \{t_0 \leq t_1 \leq t_2 \leq ... \leq t_{N-1} \leq t\}$$

is a partition of $[t_0, t]$*. Then*

$$ms - \lim_{N\to\infty} \left\{ \sum_{j=0}^{N-1} (W(t_{j+1})) - W(t_j))^2 \right\} = t - t_0.$$

Proof By Definition 3.2.1, we have to prove that

$$\lim_{N\to\infty} \mathrm{E}\left[\left(\sum_{j=0}^{N-1} (W_{j+1} - W_j)^2 - (t - t_0) \right)^2 \right] = 0. \tag{3.6}$$

In fact, we first expand $\left(\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2-(t-t_0)\right)^2$:

$$\begin{aligned}
&\left(\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2-(t-t_0)\right)^2\\
&=\left(\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2\right)^2-2(t-t_0)\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2+(t-t_0)^2\\
&=\sum_{j=0}^{N-1}(W_{j+1}-W_j)^4+\sum_{i\neq j}^{N-1}(W_{i+1}-W_i)^2(W_{j+1}-W_j)^2\\
&-2(t-t_0)\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2+(t-t_0)^2,
\end{aligned}\tag{3.7}$$

where $W_{i+1}-W_i$ is a Gaussian random variable and is independent of $W_{j+1}-W_j$. On the other hand, by Theorem 3.2.1 for a normal random variable $X\sim N(\mu,\sigma^2)$, we have the 2^{nd}-moment and the 4^{th}-moment can be evaluated by

$$\mathrm{E}(X^2)=\frac{d^2}{dt^2}M_X(t)\mid_0=\frac{d^2}{dt^2}e^{\mu t+\frac{1}{2}\sigma t^2}\mid_0=\mu^2+\sigma^2,\ \text{and,}$$

$$\mathrm{E}(X^4)=\frac{d^4}{dt^4}M_X(t)\mid_0=\frac{d^4}{dt^4}e^{\mu t+\frac{1}{2}\sigma t^2}\mid_0=\mu^4+6\sigma^2\mu^2+3\sigma^4.$$

Substituting $X=W_{j+1}-W_j\sim N(0,\Delta t)$ into these equations above, it yields

$$\mathrm{E}(W_{j+1}-W_j)^2=\Delta t,\tag{3.8}$$

$$\mathrm{E}(W_{j+1}-W_j)^4=3(\Delta t)^2.\tag{3.9}$$

Since increments are independent, combining with equation (3.8), we get

$$\mathrm{E}(W_{i+1}-W_i)^2(W_{j+1}-W_j)^2=\mathrm{E}(W_{i+1}-W_i)^2\mathrm{E}(W_{j+1}-W_j)^2=(t_{i+1}-t_i)(t_{j+1}-t_j).\tag{3.10}$$

From equations (3.7), (3.8), (3.9), and (3.10), we have

$$\begin{aligned}
&\mathrm{E}\left[\left(\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2-(t-t_0)\right)^2\right]\\
&=3\sum_{j=0}^{N-1}(t_{j+1}-t_j)^2+\sum_{i\neq j}^{N-1}(t_{i+1}-t_i)(t_{j+1}-t_j)-2(t-t_0)\sum_{j=0}^{N-1}(t_{j+1}-t_j)+(t-t_0)^2.
\end{aligned}$$

Therefore,

$$
\begin{aligned}
&\mathrm{E}\left[\left(\sum_{j=0}^{N-1}(W_{j+1}-W_j)^2-(t-t_0)\right)^2\right]\\
&=2\sum_{j=0}^{N-1}(t_{j+1}-t_j)^2+\left(\sum_{j=0}^{N-1}(t_{j+1}-t_j)\right)^2-2(t-t_0)(t-t_0)+(t-t_0)^2\\
&=2\sum_{j=0}^{N-1}(t_{j+1}-t_j)^2=2N(\frac{T}{N})^2=2\frac{T^2}{N}\to 0 \text{ as } N\to\infty.
\end{aligned}
$$

This concludes that $ms-\lim_{N\to\infty}\left\{\sum_{j=0}^{N-1}(W(t_{j+1}))-W(t_j))^2\right\}=t-t_0$.

The following crucial lemma gives the convergence of the Riemann sum, which defines the stochastic integral.

Lemma 3.2.2. *If P denotes a partition of $[0,T]$ and $0\leq\lambda\leq T$ is fixed, we define*

$$R:=\sum_{j=0}^{N-1}W(\tau j)(W(t_{j+1})-W(t_j)).$$

Then

$$ms-\lim_{N\to\infty}R=\frac{W(T)^2}{2}+(\lambda-\frac{1}{2})T.$$

That is,

$$\lim_{N\to\infty}\mathrm{E}\left(\left(R-\frac{W(T)^2}{2}-\left(\lambda-\frac{1}{2}\right)T\right)^2\right)=0.$$

Proof[2]

We note that

$$
\begin{aligned}
R&=\sum_{j=0}^{N-1}W(\tau j)(W(t_{j+1})-W(t_j))\\
&=\frac{W(T)^2}{2}-\frac{1}{2}\sum_{j=0}^{N-1}(W(t_{j+1})-W(t_j))^2\\
&+\sum_{j=0}^{N-1}(W(\tau_j)-W(t_k))^2+\sum_{j=0}^{N-1}(W(t_{j+1})-W(\tau_j))(W(\tau_j)-W(t_j)).
\end{aligned}
$$

Let

$$\begin{aligned}
A &= \frac{1}{2}\sum_{j=0}^{N-1}(W(t_{j+1}) - W(t_j))^2, \\
B &= \sum_{j=0}^{N-1}(W(\tau_j) - W(t_j))^2, \\
C &= \sum_{j=0}^{N-1}(W(t_{j+1}) - W(\tau_j))(W(\tau_j) - W(t_j)).
\end{aligned}$$

Hence, we can expand $\left(R - \frac{W(T)^2}{2} - \left(\lambda - \frac{1}{2}\right)T\right)^2$ in terms of A, B, and C as follows:

$$\begin{aligned}
&\left(R - \frac{W(T)^2}{2} - \left(\lambda - \frac{1}{2}\right)T\right)^2 \\
&= \left(-\left(A - \frac{T}{2}\right) + (B - \lambda T) + C\right)^2 \\
&= \left(A - \frac{T}{2}\right)^2 + (B - \lambda T)^2 + C^2 \\
&+ 2\left(A - \frac{T}{2}\right)(B - \lambda T) + 2(B - \lambda T)C + 2\left(A - \frac{T}{2}\right)C. \qquad (3.11)
\end{aligned}$$

We now calculate the $ms-$limit of each term on the right hand side of Eq. 3.11. By Lemma 3.2.1, we obtain

$$\begin{aligned}
ms - \lim_{N\to\infty} A &= \frac{1}{2}(t - t_0) = \frac{T}{2}, \\
ms - \lim_{N\to\infty} B &= \sum_{j=0}^{N-1}(\tau_j - t_j) = \sum_{j=0}^{N-1}(\lambda t_j + (1-\lambda)\, t_{j+1} - t_j) = \sum_{j=0}^{N-1}\lambda(t_{j+1} - t_j) = \lambda T.
\end{aligned}$$

That is,

$$\lim_{N\to\infty} \mathrm{E}\left(\left(A - \frac{T}{2}\right)^2\right) = 0, \qquad (3.12)$$

$$\lim_{N\to\infty} \mathrm{E}\left((B - \lambda T)^2\right) = 0. \qquad (3.13)$$

Next, we calculate $\mathrm{E}(C^2)$. Since $W(t_{j+1}) - W(\tau_j)$ and $W(\tau_j) - W(t_j)$ are independent, we have

$$\begin{aligned}
\mathrm{E}(C^2) &= \mathrm{E}\left(\sum_{j=0}^{N-1}(W(t_{j+1}) - W(\tau_j))(W(\tau_j) - W(t_j))\right) \\
&= \sum_{j=0}^{N-1} \mathrm{E}([W(t_{j+1}) - W(\tau_j)]^2)\mathrm{E}([W(\tau_j) - W(t_j)]^2).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{E}(C^2) &= \sum_{j=0}^{N-1}(1-\lambda)(t_{j+1} - t_j)\lambda(t_{j+1} - t_j) \\
&= \lambda(1-\lambda)\frac{T^2}{N} \to 0 \text{ as } N \to \infty. \qquad (3.14)
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&\mathrm{E}\left[2\left(A - \frac{T}{2}\right)(B - \lambda T)\right] = 2\mathrm{E}\left(A - \frac{T}{2}\right)\mathrm{E}(B - \lambda T) \\
&= 2\left(\frac{1}{2}\sum_{j=0}^{N-1}\mathrm{E}([W(t_{j+1}) - W(t_j)]^2) - \frac{T}{2}\right)\left(\sum_{j=0}^{N-1}\mathrm{E}([W(\tau_j) - W(t_j)]^2) - \lambda T\right) \\
&= 2\left(\frac{1}{2}\sum_{j=0}^{N-1}(t_{j+1} - t_j) - \frac{T}{2}\right)\left(\sum_{j=0}^{N-1}(\tau_j - t_j) - \lambda T\right) = 2 \times 0 \times 0 = 0. \qquad (3.15)
\end{aligned}$$

Similarly, one can also obtain

$$\mathrm{E}\left[2(B - \lambda T)C\right] = 0, \qquad (3.16)$$

$$\mathrm{E}\left[2\left(A - \frac{T}{2}\right)C\right] = 0. \qquad (3.17)$$

From equations (3.11)-(3.17), we arrive at

$$\lim_{N\to\infty}\mathrm{E}\left(\left(R - \frac{W(T)^2}{2} - \left(\lambda - \frac{1}{2}\right)T\right)^2\right) = 0.$$

This concludes the lemma 3.2.2. This lemma emphasizes that the limit of the Riemann sum is dependent on λ, that is, the stochastic integral is not unique. In the next section, we will introduce two types of stochastic integral: Itô integral and Stratonovich integral.

3.3 Stochastic Integrals

A scalar, autonomous SDE can be written in the form

$$dX(t) = f\left(X(t)\right)dt + g\left(X(t)\right)dW(t), \quad X(0) = X_0,\ 0 \leq t \leq T, \tag{3.18}$$

Here, the scalar functions f and g are called the *drift* and the *diffusion* of the SDE respectively. The solution $X(t)$ is a random variable for each t, $W(t)$ is the Brownian motion.

Equation (3.18) can be re-written as a stochastic integral equation

$$X\left(t\right) = X_0 + \int_0^t f\left(X\left(s\right)\right)ds + \int_0^t g\left(X\left(s\right)\right)dW\left(s\right),\ 0 \leq t \leq T, \tag{3.19}$$

where the first integral is a regular Riemann-Stieltjes integral and the second integral is a stochastic integral which usually interpreted as Itô and Stratonovich integral.

Let us describe the difference between Stratonovich's interpretation and Itô's interpretation. Suppose $h(t)$ is an arbitrary function of time and $W(t)$ is the Wiener process. We define the stochastic integral $\int_{t_0}^t h\left(s\right)dW\left(s\right)$ as a kind of Riemann-Stieltjes integral. We define a partition P of $[0,T]$ with the mesh size N

$$\mathrm{P} := \{0 = \mathrm{t}_0 \leq \mathrm{t}_1 \leq \mathrm{t}_2 \leq ... \leq \mathrm{t}_{\mathrm{N}-1} \leq \mathrm{t} = \mathrm{T}\}. \tag{3.20}$$

The stochastic integral $\int_{t_0}^t h\left(s\right)dW\left(s\right)$ is defined as the limit of the partial sum

$$\sum_{j=1}^{N-1} h\left(\tau_j\right)\left(W\left(\tau_{j+1}\right) - W\left(\tau_j\right)\right), \tag{3.21}$$

where τ_j is the intermediate point of interval $[t_j, t_{j+1}]$, that is, $t_j \leq \tau_j \leq t_{j+1}$. One can write τ_j as

$$\tau_j = \lambda t_j + \left(1 - \lambda\right)t_{j+1}, \quad 0 \leq \lambda \leq 1. \tag{3.22}$$

The stochastic integral $\int_{t_0}^t h(s)dW(s)$ depends on τ_j hence depends on the value of λ. In order to examine this stochastic integral, it's necessary to deal with the mean square limit.

By the definition 3.2.1, we define the stochastic integral in mean square sense

$$\int_0^T h(s)dW(s) = ms - \lim_{N\to\infty}\left\{\sum_{j=0}^{N-1} h(\tau_j)\left(W(t_{j+1}) - W(t_j)\right)\right\}. \tag{3.23}$$

The approximation of $\int_{t_0}^{t} h(s)dW(s)$ converges in the mean square sense to different values, depending on the selection of τ_j.
Consider the stochastic integral $\int_0^T W(s)dW(s)$. By lemma 3.2.2,

$$ms - \lim_{N\to\infty}\left\{\sum_{j=0}^{N-1} W(\tau_j)\left(W(t_{j+1}) - W(t_j)\right)\right\} = \frac{W(T)^2}{2} + (\lambda - \frac{1}{2})T.$$

The choice of τ_j as left-end point ($\lambda = 0$) leads to Itô stochastic integral while the choice as mid-point ($\lambda = \frac{1}{2}$) results in Stratonovich integral.

a) Itô Integral

Choose $\tau_j = t_j$ (left-end point), then we define the Itô stochastic integral

$$\begin{aligned}\int_0^T W(s)dW(s) &= ms - \lim_{N\to\infty}\left\{\sum_{j=1}^{N-1} W(t_j)\left(W(t_{j+1}) - W(t_j)\right)\right\}.\\ &= \frac{W(T)^2}{2} + (0 - \frac{1}{2})T.\end{aligned}$$

As a result, we have

$$\int_0^T W(s)dW(s) = \frac{W(T)^2}{2} - \frac{1}{2}T.$$

b) Stratonovich Integral:

Choose $\tau_j = (t_{j+1} + t_j)/2$ (mid-point) then the resulting stochastic integral is the Stratonovich stochastic integral as

$$\begin{aligned}\int_0^T W(s)dW(s) &= ms - \lim_{N\to\infty}\left\{\sum_{j=1}^{N-1} W\left(\frac{t_j + t_{j+1}}{2}\right)\left(W(t_{j+1}) - W(t_j)\right)\right\}\\ &= \frac{W(T)^2}{2} + (\frac{1}{2} - \frac{1}{2})T.\end{aligned}$$

Thus:

$$\int_0^T W(t)dW(t) = \frac{1}{2}W(T)^2. \tag{3.24}$$

3.4 Change of variables: Itô's formula

Itô Lemma is named after Kiyosi Itô [25], which is occasionally referred to as the Itô–Doeblin theorem.

3.4.1 Itô's Lemma

Suppose random variable X is described by the following Itô process:

$$dX = a(X,t)dt + b(X,t)dW(t), \tag{3.25}$$

where $dW(t)$ is a normal random variable. Suppose the random variable $Y = F(X,t)$. Then Y is described by the following Itô process:

$$dY = \left(a(X,t)F_X + F_t + \frac{1}{2}(b(X,t))^2 F_{XX}\right)dt + b(X,t)F_X dW(t). \tag{3.26}$$

Derivation of Itô Lemma

We have 2nd-order Taylor expansion for a twice-differentiable scalar function $f(x,t)$ has the form as following:

$$df = \frac{\partial f}{\partial t}dt + dX + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}dx^2$$

Substitute X_t for x, $F(X,t)$ for f and $(adt + bdW)$ for dx, it gives

$$dF(X,t) = F_t dt + F_x(adt + bdW) + \frac{1}{2}F_{XX}(a^2dt^2 + 2abdtdW + b^2dW^2).$$

When dt approaches 0, dt^2 and $dtdW$ tend to 0 faster than dB^2. Set dt^2 and $dtdW$ to be 0 and replace dW^2 by dt(due to the 2nd-condition of Brownian Motion), we obtain

$$dF = (aF_X + F_{\frac{1}{2}}b^2F_{XX})dt + bF_X dW.$$

Next, we consider a number of well-known examples using Itô Lemma.

Example 3.4.1 We have SDE for geometric Brownian motion (GBM)

$$dX_t = \mu X_t dt + \sigma X_t dW_t. \tag{3.27}$$

Using Itô's lemma with $Y = lnX_t$

$$\begin{aligned} d\ln X_t &= \left[\mu X_t \left(\frac{1}{X_t}\right) + 0 + \frac{1}{2}(\sigma X_t)^2 \left(-\frac{1}{{X_t}^2}\right)\right] dt + \sigma X_t \left(\frac{1}{X_t}\right) dW_t \\ &= (\mu - \frac{1}{2}\sigma^2)dt + \sigma dW_t. \end{aligned} \tag{3.28}$$

Integrating both sides of (3.28), we get

$$\ln X_{t+\Delta t} - \ln X_t = \int_t^{t+\Delta t} (\mu - \frac{1}{2}\sigma^2)dt + \int_t^{t+\Delta t} \sigma dW_t. \tag{3.29}$$

We have

$$\int_t^{t+\Delta t} (\mu - \frac{1}{2}\sigma^2)dt = (\mu - \frac{1}{2}\sigma^2)\Delta t, \tag{3.30}$$

$$\int_t^{t+\Delta t} \sigma dW_t = \int_{W_t}^{W_{t+\Delta t}} \sigma dx = \sigma(W_{t+\Delta t} - W_t) = \sigma \Delta W_t. \tag{3.31}$$

Then

$$X_{t+\Delta t} = X_t \exp\left[(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t\right]. \tag{3.32}$$

This is the theoretical solution (exact solution) of GBM.

Example 3.4.2. Given the process for stock S is

$$dS = \mu S dt + \sigma S dW.$$

A forward contract is priced as $F_0 = S_0 e^{rT}$. The forward price as time passed

$$F(S,t) = Se^{r(T-t)}.$$

Observe that

$$F_S = e^{r(T-t)}, \quad F_{SS} = 0, \quad F_t = -rSe^{r(T-t)}.$$

Hence,

$$\begin{aligned} dF =& (\mu S e^{r(T-t)} - rSe^{r(T-t)} + \frac{1}{2}\sigma^2 S^2 * 0)dt + \sigma S e^{r(T-t)} dW \\ =& (\mu - r)F dt + \sigma F dz. \end{aligned}$$

3.4.2 Itô's Formula

If f is a function with two continuos derivatives, and W_t is a standard Brownian motion,

$$f(W_t) - f(W_0) = \int_0^t f'(W_s) dW_s + \frac{1}{2} \int_0^t f''(W_s) ds. \tag{3.33}$$

This formula is sometimes written in the differential form

$$df(W_t) = f'(W_t) dW_t + \frac{1}{2} f''(W_t) dt. \tag{3.34}$$

Itô's formula can be derived from Itô Lemma. From Eq.(3.25), let $X = W_t$ and $a(X) = 0,\ b(X) = 1$. Suppose $Y = f(W_t)$, Eq.(3.26) becomes

$$df(W_t) = (0 + \frac{1}{2} f"(W_t))dt + f(W_t) dW_t,$$

which is Eq.(3.34), integrating both sides we obtain Eq.(3.33).

Chapter 4

Numerical Schemes and Applications in finance

4.1 Numerical Methods for Simulation of SDEs

4.1.1 The Euler-Maruyama Method

An SDE can be written in the integral form:

$$X(t) = X_0 + \int_0^t f(X(s))ds + \int_0^t g(X(s))dW(s), \quad 0 \le t \le T. \tag{4.1}$$

It is also written differential form:

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \qquad X(0) = X_0, \qquad 0 \le t \le T. \tag{1.1}$$

Let $\Delta t = T/L$, for positive integer L, $t_j = j\Delta t, X(t_j)$ denoted X_j. We figure out the discretized Brownian paths and use them to create the increment $W(t_j) - W(t_{j-1})$. This method has order of accuracy $O(\Delta t^{\frac{1}{2}})$.

We compute the solution of Eq.(1.1) by the Euler-Maruyama algorithm as following:

Algorithm 1 Euler-Maruyamam Algorithm

$X_0 = X(0)$

for $j = 1 : L$ **do**

$X_j = X_{j-1} + f(X_{j-1})\Delta t_j + g(X_{j-1})\Delta W_j$

end

where

$\Delta t_j = t_j - t_{j-1}$

$\Delta W_j = W(t_j) - W(t_{j-1})$.

In the below section, we present the derivation for the Euler-Maruyama (EM) algorithm.

Derivation for the EM algorithm:

We start from the integral form of the SDE:

$$X_t = X_0 + \int_{t_0}^{t} a(X_s)ds + \int_{t_0}^{t} b(X_s)dW_s. \tag{4.2}$$

Using Itô's formula

$$f(X_s) = f(X_{t_0}) + \int_{t_0}^{s} [a(X_u)f'(X_u) + \frac{1}{2}b^2(X_u)f''(X_u)]du + \int_{t_0}^{s} b(X_u)f'(X_u)dW_u. \tag{4.3}$$

Let $L_0 f := af' + \frac{1}{2}b^2 f'', L_1 f := bf'$, Eq. (4.3) becomes

$$f(X_s) = f(X_{t_0}) + \int_{t_0}^{s} L_0 f(X_u)du + \int_{t_0}^{s} L_1 f(X_u)dW_u. \tag{4.4}$$

Thus

$$a(X_s) = a(X_{t_0}) + \int_{t_0}^{s} L_0 a(X_u)du + \int_{t_0}^{s} L_1 a(X_u)dW_u, \tag{4.5}$$

$$b(X_s) = b(X_{t_0}) + \int_{t_0}^{s} L_0 b(X_u)du + \int_{t_0}^{s} L_1 b(X_u)dW_u. \tag{4.6}$$

Substitute (4.5) and (4.6) into (3.32) at lowest order, Euler formula is obtained.

4.1.2 Milstein Method

The order of accuracy of Milstein method is $O(\Delta t)$. The approximate solution of Eq.(1.1) using this numerical method is calculated by the below algorithm

Algorithm 2 Milstein Algorithm

$X_0 = X(0)$

for $j = 1 : L$ **do**

$$X_j = X_{j-1} + f(X_{j-1})\Delta t_j + g(X_{j-1})\Delta W_j + \tfrac{1}{2}g(X_{j-1})g'(X_{j-1})\left[(\Delta W_j)^2 - \Delta t_j\right]$$

end

where

$\Delta t_j = t_j - t_{j-1}$

$\Delta W_j = W(t_j) - W(t_{j-1})$.

In the following sections, we will respectively represent how to obtain Milstein's method with the specific form of SDE (geometric Brownian motion) and two different derivations of the general form.

Derivation for the Milstein algorithm:

a) The specialized case

Considering the SDE for geometric Brownian motion

$$dX_t = \mu X_t dt + \sigma X_t dW_t. \tag{4.7}$$

From Example 3.4.1, we have

$$X_{t+\Delta t} = X_t \exp\left[(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t\right].$$

By Taylor series expansion

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + O(x^2)$$

and

$$\Delta t \Delta W_t = (\Delta t)^{3/2},$$

We get

$$\begin{aligned}
X_{t+\Delta t} &\approx X_t \left\{ 1 + \left[(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t \right] + \frac{\left[(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t \right]^2}{2} \right\} \\
&= X_t \left[1 + (\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t + \frac{1}{2}\sigma^2 (\Delta W_t)^2 + O(\Delta t) \right] \\
&\approx X_t + \mu X_t \Delta t + \sigma X_t \Delta W_t + \frac{1}{2}\sigma^2 X_t \left[(\Delta W_t)^2 - \Delta t \right],
\end{aligned}$$

where $f(x) = \mu x, g(x) = \sigma x$, we obtain the approximation:

$$X_{t+\Delta t} = X_t + f(X_t)\,\Delta t + g(X_t)\,\Delta W_t + \frac{1}{2} g(X_t)\, g'(X_t) \left[(\Delta W_t)^2 - \Delta t \right]. \tag{4.8}$$

b) The general case

Look at the proof of EM method. When we substitute (4.3) and (4.6) into (3.32) at lowest order, Euler formula is obtained. Whereas Milstein method is derived when collecting terms upto $O(dt) = O(dW^2)$

$$\begin{aligned}
X_t = X_{t_0} &+ \int_{t_0}^{t} \left[a(X_{t_0}) + \int_{t_0}^{s} L_0 a(X_u) du + \int_{t_0}^{s} L_1 a(X_u) dW_u \right] ds \\
&+ \int_{t_0}^{t} \left[b(X_{t_0}) + \int_{t_0}^{s} L_0 b(X_u) du + \int_{t_0}^{s} L_1 b(X_u) dW_u \right] dW_s \qquad (4.9) \\
&= X_{t_0} + a(X_{t_0}) \int_{t_0}^{t} ds + b(X_{t_0}) \int_{t_0}^{t} dW_s + \int_{t_0}^{t} \int_{t_0}^{s} L_1 b(X_u) dW_u dW_s + O(dt). \qquad (4.10)
\end{aligned}$$

Using (4.6) to substitute $b(X_u)$ in (4.10) and dropping terms higher than $O(dt)$. We get

$$\begin{aligned}
X_t &\approx X_{t_0} + a(X_{t_0}) \int_{t_0}^{t} ds + b(X_{t_0}) \int_{t_0}^{t} dW_s \\
&+ \int_{t_0}^{t} \int_{t_0}^{s} L_1 \left[b(X_{t_0}) + \int_{t_0}^{s} L_0 b(X_u) du + \int_{t_0}^{s} L_1 b(X_u) dW_u \right] dW_u dW_s \qquad (4.11) \\
&= X_{t_0} + a(X_{t_0}) \int_{t_0}^{t} ds + b(X_{t_0}) \int_{t_0}^{t} dW_s + L_1 b(X_{t_0}) \int_{t_0}^{t} \int_{t_0}^{s} dW_u dW_s + O(dt). \qquad (4.12)
\end{aligned}$$

Hence,

$$\int_{t_0}^{t}\int_{t_0}^{s} dW_u dW_s = \int_{t_0}^{t} (W_s - W_{t_0})dW_s = \frac{1}{2}[(W_t - W_{t_0})^2 - (t - t_0)]. \qquad (4.13)$$

Therefore,

$$X_t = X_{t_0} + a(X_{t_0})(t - t_0) + b(X_{t_0})(W_t - W_{t_0}) + \frac{1}{2}L_1 b(X_{t_0})[(W_t - W_{t_0})^2 - (t - t_0)].$$

c) Another way to the general case

From SDE

$$X_t = X_0 + \int_0^t f(X_s)ds + \int_0^t g(X_s)dW_s, 0 \le t \le T. \qquad (3.12)$$

By Itô's formula:

$$\begin{aligned} dg(X_t) &= g'(X_t)dX_t + \frac{1}{2}g''(X_t)g^2(X_t)dt \\ &= \left[g'(X_t)f(X_t) + \frac{1}{2}g'(X_t)g^2(X_t)\right]dt + g'(X_t)g(X_t)d(W_t) \\ &= \mu_g(X_t)dt + \sigma_g(X_t)dW_t. \end{aligned}$$

We apply the Euler-Maruyama approximation to the process $g(X_t)$ results in the approximation of $g(X_u), t \le u \le t + h$

$$\begin{aligned} g(X_s) &\approx g(X_t) + \mu_g(X_t)[u - t] + \sigma_g(X_t)(W_u - W_t) \\ &= g(X_t) + \left[g'(X_t)f(X_t) + \frac{1}{2}g'(X_t)g^2(X_t)\right](u - t) + g'(X_t)g(X_t)(W_u - W_t). \end{aligned}$$

Since the drift term is $O(u-t)$, $W_u - W_t$ is $O(\sqrt{u-t})$, we drop the higher-order term:

$$g(X_u) \approx g(X_t) + g'(X_t)g(X_t)(W_u - W_t), \quad u \in [t, t + h]. \qquad (4.14)$$

Thus

$$\int_t^{t+h} g(X_u)dW_u \approx \int_t^{t+h} [g(X_t) + g'(X_t)g(X_t)(W_u - W_t)]dW_u$$
$$= g(X_t)(W_{t+h} - W_t) + g'(X_t)g(X_t)\left(\int_t^{t+h} (W_u - W_t)dW_u\right). \quad (4.15)$$

We compute the integral in the last term on the right hand side of the above equation as

$$\int_t^{t+h} (W_u - W_t)dW_u = \int_t^{t+h} W_u dW_u - W_t \int_t^{t+h} dW_u$$
$$= Y_{t+h} - Y_t - W_t(W_{t+h} - W_t), \quad (4.16)$$

with Y_t is Itô integral, $Y_0 = 0$,

$$Y_t = \int_0^t W_t dW_t = \frac{1}{2}(W_t)^2 - \frac{1}{2}t. \quad (4.17)$$

Substituting (4.17) into (4.16)

$$\int_t^{t+h} (W_u - W_t)dW_u = \frac{1}{2}(W_{t+h} - W_t)^2 - \frac{1}{2}h. \quad (4.18)$$

Using (4.18) and (4.15)

$$\int_t^{t+h} g(X_u)dW_u \approx g(X_t)(W_{t+h} - W_t) + \frac{1}{2}g'(X_t)g(X_t)\left[(W_{t+h} - W_t)^2 - h\right].$$

So, the approximation of X_{t+h} with Euler-Maruyama methods

$$X_{t+h} \approx X_t + f(X_t)h + g(X_t)(W_{t+h} - W_t)$$

to Milstein

$$X_{t+h} \approx X_t + f(X_t)h + g(X_t)(W_{t+h} - W_t) + \frac{1}{2}g'(X_t)g(X_t)\left[(W_{t+h} - W_t)^2 - h\right].$$

So, Milstein's method for the SDEs:

$$X_{j+1} = X_j + f\left(X_j\right)\Delta t + g\left(X_j\right)\Delta W_j + \frac{1}{2}g'(X_j)g(X_j)\left[\left(\Delta W_j\right)^2 - \Delta t\right],$$

where $\Delta W_j = W_{j+1} - W_j$, the initial value $X(t_0) = X_0$

4.2 Strong and weak convergence

4.2.1 Strong order of convergence

Definition 4.2.1. *A method is said to have strong order of convergence equal to γ if there exists a constant C such that*

$$\mathrm{E}\left[X_n - X(\tau)\right] \leq C\Delta t^{\tau} \tag{4.19}$$

for E denotes the expected value, $\tau = n\Delta t \in [0, T]$ and Δt small enough.

a) Euler Method

$$e_{\Delta t}^{strong} := \mathrm{E}\left[X_L - X(T)\right],$$

where $L\Delta t = T$. If the bound (4.19) holds with $\tau = \frac{1}{2}$, then

$$e_{\Delta t}^{strong} \leq C\Delta t^{\frac{1}{2}}. \tag{4.20}$$

Now, we use MATLAB to check whether the strong order of convergence of EM method equals $\frac{1}{2}$ (Listing 4.1).

Listing 4.1: MATLAB code to check strong order convergence of EM method

```
mu=2; sigma=1; Xzero=1;          %problem parameters
T=1; N=2^9; dt=T/N ;
M=1000;                            %M paths simultaneously
Xerr=zeros(M,5);
for s=1:M
    dW=sqrt(dt)*randn(1,N);
    W=cumsum(dW);
```

```
    Xtrue = Xzero*exp((mu-0.5*sigma^2)+sigma*W(end));
    for p=1:5                               %preallocate array
        R=2^(p-1);Dt=R*dt;L=N/R;            %take various Euler timesteps
        Xtemp=Xzero;                        %L Euler steps of size Dt
        for j=1:L
            Winc=sum(dW(R*(j-1)+1:R*j));
            Xtemp=Xtemp+Dt*mu*Xtemp+sigma*Xtemp*Winc;
         end
         Xerr(s,p)=abs(Xtemp-Xtrue);
    end
end
Dtvals=dt*(2.^([0:4]));
loglog(Dtvals,mean(Xerr),'b*-'),hold on
loglog(Dtvals,(Dtvals.^(.5)),'r--'),hold off
axis([1e-3 1e-1 1e-4 1]);
xlabel('\Delta t'),
ylabel('E(X(T))-Sample average of X_L')
%Least square fit of error=C*dt^q
A=[ones(5,1),log(Dtvals)'];rhs=log(mean(Xerr)');
sol=A\rhs; q=sol(2)
resid = norm(A*sol-rhs)
```

We stored in $Xerr(s,p)$ the endpoint error in the s^{th} sample path, for the p^{th} stepsize. The p^{th} element of $mean(Xerr)$ is an approximation to $e^{strong}_{\Delta t}$ for $\Delta t = 2^{p-1}\delta t$. From the inequality (4.20), taking logs, we obtain

$$\log e^{strong}_{\Delta t} \sim \log C + \frac{1}{2}\log \Delta t. \tag{4.21}$$

We use the command $loglog(Dtvals, mean(Xerr),' b * -')$ in order to plot the approximation to $e^{strong}_{\Delta t}$ against Δt on a log-log scale (see Figure 4.1) While the result is the blue dash lines., for reference, a dashed red line of slope $\frac{1}{2}$ is use to compare. It is obvious that the slopes of these curves appear to match each other well. Then, we compute the least squares fit for $logC$ and q, which produce the value 0.5161 for q and the least squares residual is of 0.0654. Thus, we can conclude that our numerical

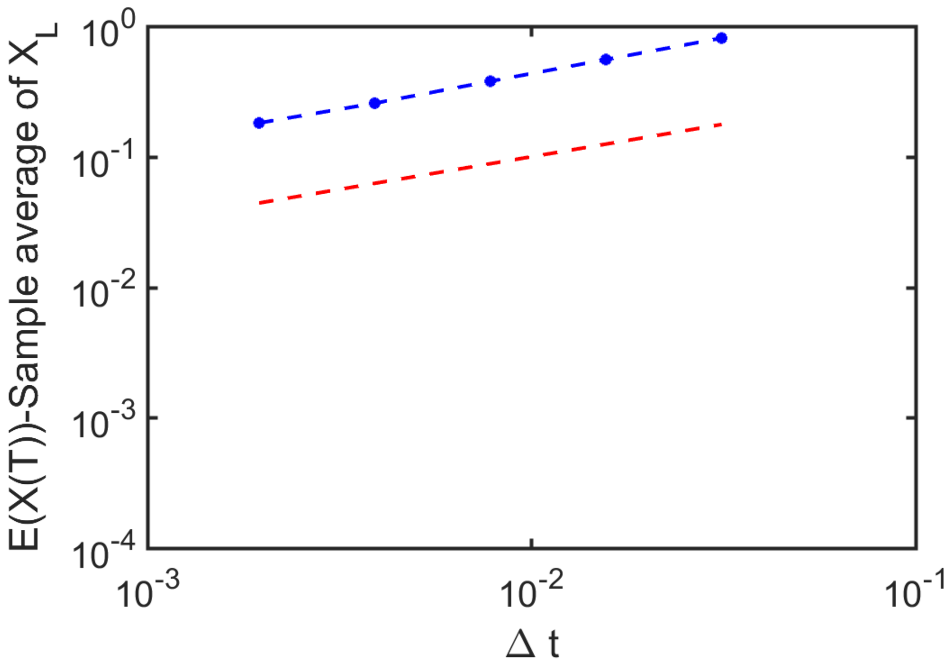

Figure 4.1: Strong error plot and reference slope of 1/2

results are consistent with the strong convergence of EM of order $\frac{1}{2}$.

b) Milstein Method

$$e_{\Delta t}^{strong} \leq C\Delta t. \tag{4.22}$$

The strong order of convergence (4.19) determines the rate at which the "mean of the error" decays as $\Delta t \to 0$.

We consider the following MATLAB code (Listing 4.2).

Listing 4.2: MATLAB code to check strong order convergence of Milstein method

```
r=2; K=2; beta=0.32; Xzero=1;
T=1;N=2^11;dt=T/N;
M=500;
R=[1;16;32;64;128];
dW=sqrt(dt)*randn(M,N);
Xmil=zeros(M,5);
for p=1:5
    Dt=R(p)*dt;L=N/R(p);
```

```
    Xtemp=Xzero*ones(M,1);
    for j=1:L
        Winc=sum(dW(:,R(p)*(j-1)+1:R(p)*j),2);
        Xtemp=Xtemp+Dt*r*Xtemp.*(K-Xtemp)+beta*Xtemp.*Winc+0.5*beta^2*Xtemp.*(Winc.^2-Dt
    end
    Xmil(:,p)=Xtemp;
end
Xref=Xmil(:,1);
Xerr=abs(Xmil(:,2:5)-repmat(Xref,1,4));
mean(Xerr);
Dtvals=dt*R(2:5);
loglog(Dtvals,mean(Xerr),'b*--'),hold on
loglog(Dtvals,Dtvals,'r--'),hold off
xlabel('\Delta t')
ylabel('Sample average of |X(T)-X_L|')
%Least square fit of error=C*Dt^q
A=[ones(4,1),log(Dtvals)];rhs=log(mean(Xerr)');
sol=A\rhs; q=sol(2)
resid = norm(A*sol-rhs)
```

We illustrate the resulting log-log error plot in Figure 3.4 and a reference line which has slope 1. The least-squares fit gives $q = 1.0483$ and resid $= 0.0164$. So is is determined that this method's strong order of convergence is 1.

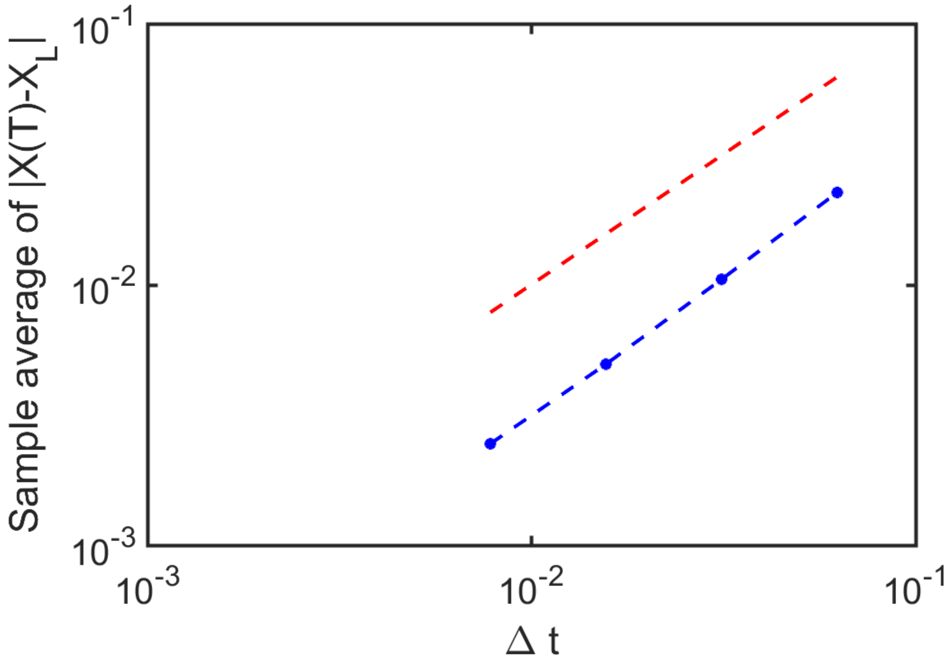

Figure 4.2: Strong error plot and reference slope of 1

Applying EM method and Milstein method with decreasing step size Δt results in successively improved approximation. We use the following code for every value of Δt.

Listing 4.3: MATLAB code to examine strong convergence of EM and Milstein schemes for decreasing Δt

```
mu=2; sigma=1; Xzero=1;
T=1; N=2^7; dt=T/N; M=5000;
err1=zeros(M,1);err2=zeros(M,1);
for s=1:M
    dW=sqrt(dt)*randn(1,N);
    W=cumsum(dW);
    Xtrue=Xzero*exp((mu-0.5*sigma^2)*([dt:dt:T])+sigma*W);
    %plot([0:dt:T],[Xzero,Xtrue],'k-'),hold on
    R=4;Dt=R*dt;L=N/R;
    Xem=zeros(1,L);
    Xmil=zeros(1,L);
    Xtemp1=Xzero;
```

```
    Xtemp2=Xzero;
    for j=1:L
            Winc=sum(dW(R*(j-1)+1:R*j));
            Xtemp1=Xtemp1+Dt*mu*Xtemp1+sigma*Xtemp1*Winc;
            Xtemp2=Xtemp2+Dt*mu*Xtemp2+sigma*Xtemp2*Winc
            +(1/2)*(sigma^2)*Xtemp2*(Winc^2-Dt);
            Xem(j)=Xtemp1;
            Xmil(j)=Xtemp2;
    end
    err1(s)=abs(Xtemp1(end)-Xtrue(end));
    err2(s)=abs(Xtemp2(end)-Xtrue(end));
end
 emerr = mean(err1)
 milerr= mean(err2)
```

The orders 1/2 for Euler–Maruyama and 1 for Milstein are clearly visible in the table 4.1. It is required to cut the step size by a factor of 4 in order to reduce the error by a factor of 2 with the Euler–Maruyama Method. Whereas, for the Milstein Method, cutting the step size by a factor of 2 achieves the same result.

Table 4.1: Strong convergence of Euler-Maruyama and Milstein schemes GBM $\mu = 2,\ \sigma = 1,\ X_0 = 1, M = 5000$.

Δt	Error values by EM	Error values by Milstein
2^{-7}	0.7605	0.4857
2^{-8}	0.5581	0.2721
2^{-9}	0.3786	0.1296
2^{-10}	0.2634	0.0645
2^{-11}	0.1822	0.0833
2^{-12}	0.1312	0.0166
2^{-13}	0.0953	0.0083
2^{-14}	0.0626	0.0038
2^{-15}	0.0478	0.0022

4.2.2 Weak order of convergence

Definition 4.2.2. *A method is said to have weak order of convergence equal to γ if there exists a constant C such that for all function p*

$$|\mathrm{E}p(X_n) - \mathrm{E}p(X(\tau))| \leq C\Delta t^{\gamma}. \tag{4.23}$$

a) Euler Method

We concentrate on the case p is identity function. For appropriate f and g, EM has weak order of convergence $\gamma = 1$. Mimic the strong order of convergence, let:

$$e_{\Delta t}^{weak} := |\mathrm{E}X_L - \mathrm{E}_p(X(T))|, \tag{4.24}$$

where $L\Delta t = T$. From (4.23), for $p(X) \equiv X$ and $\gamma = 1$,

$$e_{\Delta t}^{weak} \leq C\Delta t \tag{4.25}$$

for sufficient Δt.

We examine the weak convergence of EM with the below code (Listing 4.4)

Listing 4.4: MATLAB code to check weak order convergence of EM method

```
mu=2; sigma=0.1; Xzero=1; T=1;       %problem parameters
M=5000;                             %M paths simultaneously
Xem=zeros(5,1);
for p=1:5                             %preallocate array
       Dt=2^(p-10);L=T/Dt;          %take various Euler timesteps
       Xtemp=Xzero*ones(M,1);                     %L Euler steps of size Dt
       for j=1:L
          Winc=sqrt(Dt)*randn(M,1);
          Xtemp=Xtemp+Dt*mu*Xtemp+sigma*Xtemp.*Winc;
        end
        Xem(p)=mean(Xtemp);
end
Xerr=abs(Xem-exp(mu));
```

```
Dtvals=2.^([1:5]-10);
loglog(Dtvals,Xerr,'b*-'),hold on
loglog(Dtvals,Dtvals,'r--'),hold off
xlabel('\Delta t'),
ylabel('E(X(T))-Sample average of X_L')
%Least square fit of error=C*dt^q
A=[ones(5,1),log(Dtvals)'];rhs=log(Xerr);
sol=A\rhs; q=sol(2)
resid = norm(A*sol-rhs)
```

The sample average approximations to $\mathrm{E}X_L$ are stored in Xem. It is shown that

$$\mathrm{E}\left[X(T)\right] = e^{\mu T} \tag{4.26}$$

for the true solution and Xerr stores the corresponding weak endpoint error for each Δt. The Figure (4.3) displays how the weak error varies with Δt on a log-log scale. We add a dashed red reference line of slope 1 and do the least squares fit that gives q = 1.0010 and resid = 0.0240. This figure is confirmed with the weak order convergence of EM equals 1.

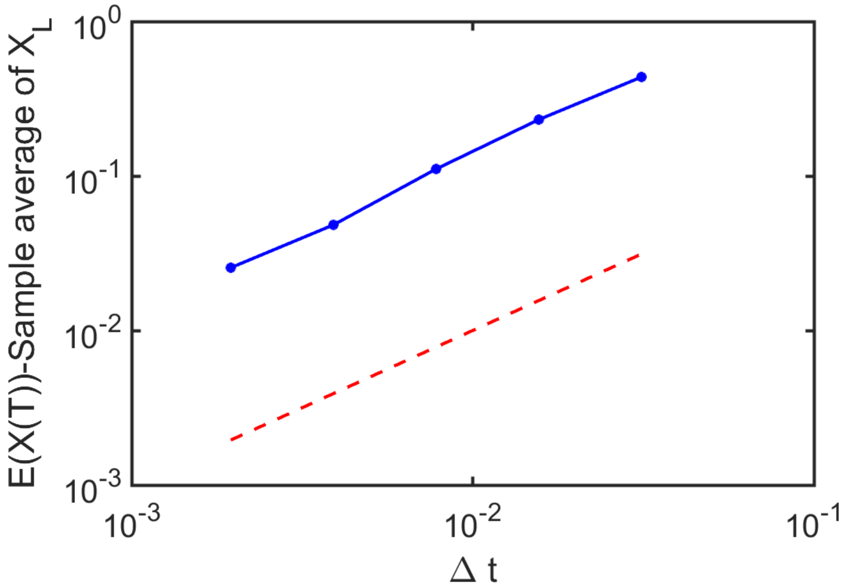

Figure 4.3: Weak error plot and reference slope of 1

Proof Eq.(4.26)

From Example 3.4.1, we will obtain

$$X_T = X_0 \exp\left[(\mu - \frac{1}{2}\sigma^2)T + \sigma W_T\right].$$

Let $X = (\mu - \frac{1}{2}\sigma^2)T + \sigma W_T$, where $W(T) - W(0) \sim \sqrt{T}N(0,1),\ W(0) = 0$. We get

$$X_T = X_0 e^X,$$
$$\mathrm{E(X)} = (\mu - \frac{1}{2}\sigma^2)T,$$
$$Var(X) = Var(\sigma W(T)) = \sigma^2 Var(W(T)) = T\sigma^2.$$

From (3.2),

$$M_X(t) = e^{E(X)t + \frac{1}{2}Var(X)t^2} = \mathrm{E(e^{tX})}.$$

For t=1,

$$\mathrm{E(e^X) = e^{E(X)+\frac{1}{2}Var(X)} = e^{(\mu-\frac{1}{2}\sigma^2)T+\frac{1}{2}T\sigma^2} = e^{\mu T}}.$$

So,

$$\mathrm{E}\left(X_T\right) = X_0 e^{\mu T},$$

where $X_0 = 1$, $T = 1$ (assumed by the code).

b) Milstein Method

$$e_{\Delta t}^{\mathrm{weak}} \leq C\Delta t. \tag{4.27}$$

We consider the following MATLAB code (Listing 4.5).

Listing 4.5: MATLAB code to check weak order convergence of Milstein method

```
r=2; K=2; beta=0.32; Xzero=1;
T=1;N=2^11;dt=T/N;
M=500;
R=[1;16;32;64;128];
dW=sqrt(dt)*randn(M,N);
Xmil=zeros(M,5);
for p=1:5
    Dt=R(p)*dt;L=N/R(p);
```

```
    Xtemp=Xzero*ones(M,1);
    for j=1:L
        Winc=sum(dW(:,R(p)*(j-1)+1:R(p)*j),2);
        Xtemp=Xtemp+Dt*r*Xtemp.*(K-Xtemp)+beta*Xtemp.*Winc+0.5*beta^2*Xtemp.*(Winc.^2-Dt
    end
    Xmil(:,p)=Xtemp;
end
Xref=Xmil(:,1);
Xerr=abs(mean(Xmil(:,2:5))-mean(repmat(Xref,1,4)));
Dtvals=dt*R(2:5);
loglog(Dtvals,Xerr,'b*--'),hold on
loglog(Dtvals,Dtvals,'r--'),hold off
xlabel('\Delta t')
ylabel('Sample average of |X(T)-X_L|')
%Least square fit of error=C*Dt^q
A=[ones(4,1),log(Dtvals)];rhs=log(Xerr');
sol=A\rhs; q=sol(2)
resid = norm(A*sol-rhs)
```

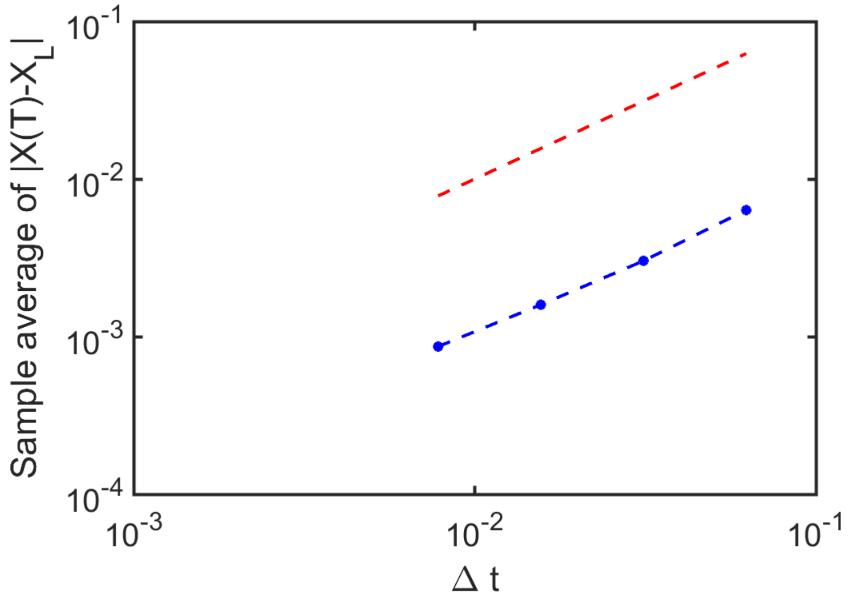

Figure 4.4: Weak error plot and reference slope of 1

The resulting log-log error plot is shown in Figure 4.4 coupled with a reference line of slope 1. The least-squares fit gives q = 1.0267 and resid = 0.0178.

Similar to table 4.1, in order to examine weak convergence of EM and Milstein schemes, we let Δt be decreasing

Table 4.2: Weak convergence of Euler-Maruyama and Milstein schemes for GBM $\mu = 2,\ \sigma = 1,\ X_0 = 1, M = 5000$

Δt	Error values by EM	Error values by Milstein
2^{-7}	0.2638	0.2522
2^{-8}	0.2028	0.1981
2^{-9}	0.1124	0.1113
2^{-10}	0.0907	0.0916
2^{-11}	0,0761	0.0735
2^{-12}	0.0553	0.0534
2^{-13}	0.0207	0.0208
2^{-14}	0.0066	0.0073
2^{-15}	0.0038	0.0036

In table 4.2 we can see that both Euler-Maruyama and Milstein have an experimental weak order of convergence of 1. Indeed, to reduce the error by half we need to cut the time step by a factor of 2.

4.3 Second order method

Reconsider the technique to approach EM and Milstein's method (part b) to solve SDE:

$$X_t = X_0 + \int_{t_0}^{t} a(X_s)ds + \int_{t_0}^{t} b(X_s)dW_s. \tag{3.32}$$

From equation (4.5), which is Itô formula applied to $a(X(s))$

$$a(X_s) = a(X_{t_0}) + \int_{t_0}^{s} L_0 a(X_u)du + \int_{t_0}^{s} L_1 a(X_u)dW_u.$$

Applying Euler approximation to each of the two integrals hence we set $L_0a(X_u) \approx L_0a(X_{t_0})$ and $L_1a(X_u) \approx L_1a(X_{t_0})$ for $u \in [t_0, s]$ to get

$$a(X_s) \approx a(X_{t_0}) + L_0a(X_{t_0}) \int_{t_0}^{s} du + L_1a(X_{t_0}) \int_{t_0}^{s} dW_u, \tag{4.28}$$

Plug (4.28) into the drift term of (3.32), we obtain

$$\begin{aligned}\int_{t_0}^{t} a(X_s)ds &= \int_{t_0}^{t} \left[a(X_{t_0}) + L_0a(X_{t_0}) \int_{t_0}^{s} du + L_1a(X_{t_0}) \int_{t_0}^{s} dW_u\right] ds \\ &= a(X_{t_0})(t-t_0) + L_0a(X_{t_0}) \int_{t_0}^{t}\int_{t_0}^{s} duds + L_1a(X_{t_0}) \int_{t_0}^{t}\int_{t_0}^{s} dW_u ds.\end{aligned}$$

We have

$$\int_{t_0}^{t}\int_{t_0}^{s} duds = \frac{1}{2}(t-t_0)^2,$$

$$\int_{t_0}^{t}\int_{t_0}^{s} dW_u ds = \int_{t_0}^{t} [W_s - W_{t_0}]ds.$$

Similarly,

$$\begin{aligned}b(X_s) &= b(X_{t_0}) + \int_{t_0}^{s} L_0b(X_u)du + \int_{t_0}^{s} L_1b(X_u)dW_u \\ &\approx b(X_{t_0}) + L_0b(X_{t_0}) \int_{t_0}^{s} du + L_1b(X_{t_0}) \int_{t_0}^{s} dW_u.\end{aligned} \tag{4.29}$$

Plug (4.29) into the diffusion term of (3.32)

$$\begin{aligned}\int_{t_0}^{t} b(X_s)dW_s &= \int_{t_0}^{t} \left[b(X_{t_0}) + L_0b(X_{t_0}) \int_{t_0}^{s} du + L_1b(X_{t_0}) \int_{t_0}^{s} dW_u\right] dW_s \\ &= b(X_{t_0})(W_t - W_{t_0}) + L_0b(X_{t_0}) \int_{t_0}^{t}\int_{t_0}^{s} dudW_s + L_1b(X_{t_0}) \int_{t_0}^{t}\int_{t_0}^{s} dW_u dW_s.\end{aligned}$$

From (4.18)

$$\int_{t_0}^{t}\int_{t_0}^{s} dW_u dW_s = \int_{t_0}^{t} [W_s - W_{t_0}]dW_s = \frac{1}{2}[(\Delta W)^2 - (t-t_0)].$$

We have

$$\int_{t_0}^{t}\int_{t_0}^{s} dudW_s = \int_{t_0}^{t} (s-t_0)dW_s.$$

Applying integration by part, we get

$$\begin{aligned}\int_{t_0}^{t}(s-t_0)dW_s =&(t-t_0)W_t - \int_{t_0}^{t} W_s ds \\ =&(t-t_0)(W_t - W_{t_0}) - \int_{t_0}^{t}[W_s - W_{t_0}]ds \\ =&(t-t_0)\Delta W - \int_{t_0}^{t}[W_s - W_{t_0}]ds.\end{aligned}$$

Let

$$\Delta I := \int_{t_0}^{t}[W_s - W_{t_0}]ds.$$

So equation (3.32) gives:

$$\begin{aligned}X_t = X_0 + a(t-t_0) + L_0 a\frac{1}{2}(t-t_0)^2 + L_1 a\Delta I \\ + b\Delta W + L_0 b[(t-t_0)\Delta W - \Delta I] + L_1 b\frac{1}{2}[(\Delta W)^2 - (t-t_0)].\end{aligned}$$

Or

$$\begin{aligned}X_{j+1} = X_j + a\Delta t + b\Delta W + \frac{1}{2}bb'[\Delta W^2 - \Delta t^2] \\ + (aa' + \frac{1}{2}a"b^2)\frac{1}{2}\Delta t^2 + a'b\Delta I + (ab' + \frac{1}{2}b^2b")(\Delta t\Delta W - \Delta I).\end{aligned}$$

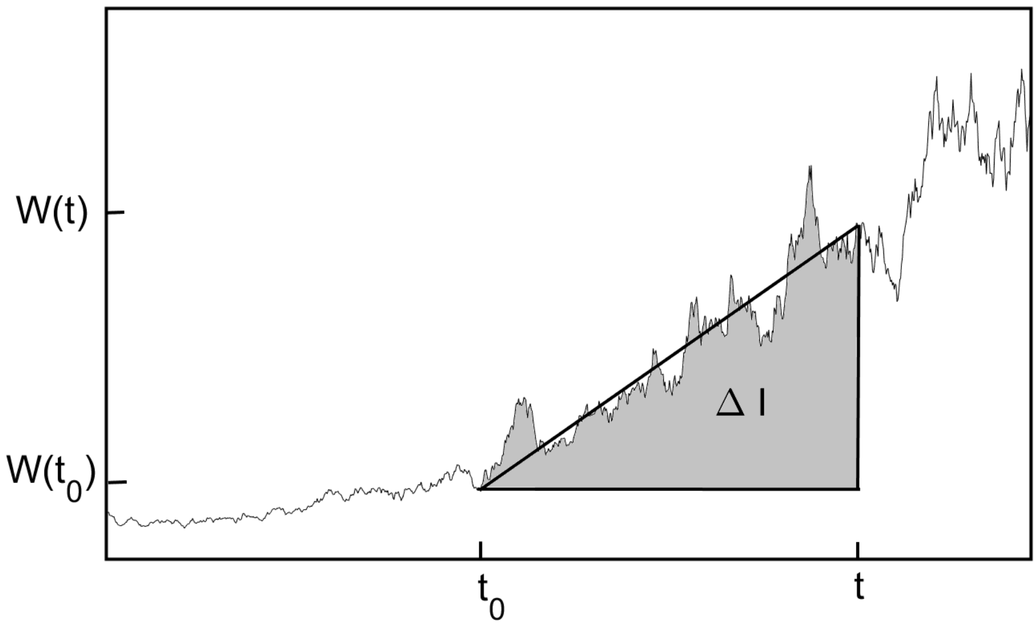

Figure 4.5: The shaded area is ΔI.

Given $W(t_0)$ and $W(t)$, the conditional expectation of W at any intermediate time lies on the straight line connecting these endpoints. The conditional expectation of ΔI is given by the area of the triangle with base $t - t_0$ and height $\Delta W = W(t) - W(t_0)$. Therefore the Δt is replaced by $\Delta W h/2$.

4.4 Application in Financial Option Valuation

4.4.1 Monte Carlo Method

Monte Carlo simulation is a numerical method used for numerically computing integrals, expected values and studying how a model responds to randomly generated inputs. This technique is helpful to control complex stochastic systems.

In Monte Carlo simulation, the entire system is simulated a large number of times. So, a set of sample paths is produced on $[t_0, T]$. Each simulation is equally likely. For each sample, we produce a sample path solution to the SDE on $[t_0, T]$.

Example 4.4.1

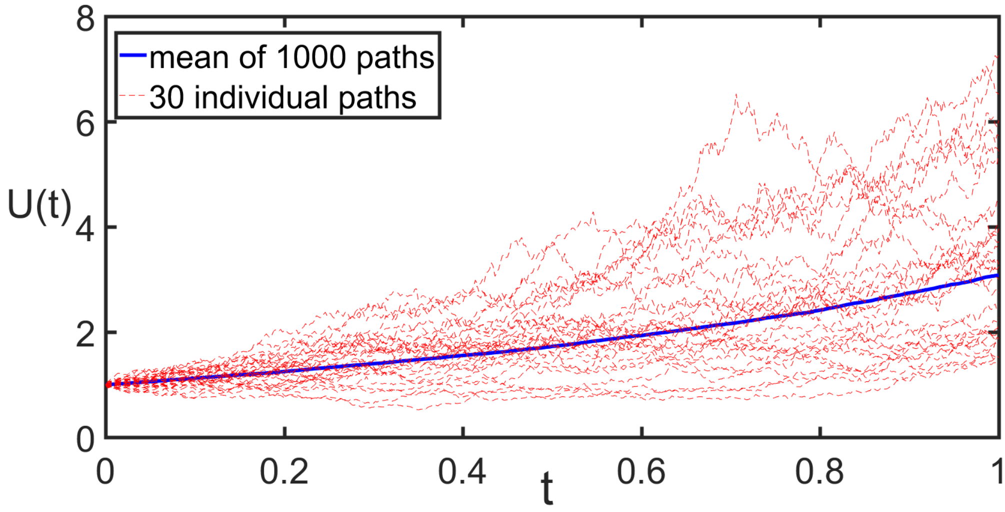

Figure 4.6: The function u(W(t)) averaged over 1000 discretized Brownian paths and along 530 individual paths

Run MATLAB Code: Function along a Brownian path
We evaluate function $U(W(t)) = exp(t + \frac{1}{2}W(t))$, which contains a random variable W(t) thus requires to use Monte Carlo simulation to see the probability distribution at time T. When we construct 1000 discretized Brownian paths, the expected value of $U(W(t))$ is approximated by computing average and is plotted with a solid blue line. We also plot 30 individual paths using the dashed red line for demonstration.

4.4.2 Option Valuation

An option is a contract that gives its holder the right(but not the obligation) to buy (call option) or sell (put option) the underlying asset S at an agreed price, the strike price K, at a specified time or during a specified period of time. The time until the expiry date is called the time-to-maturity.

The most commonly traded options are American and European options. American options can be exercised at any time during the specified period of time that goes from the date of purchase to the expiration of the contract. Whereas, European options are more restrictive regarding the exercise date as they can only be exercised at the maturity. In this thesis, we will concentrate on European options. Let S(T) denotes the asset price at the expiry date.

- **European call option** At $t = T$,
  -If $S(T) > K$, the holder will exercise the call means he will buy the asset for K and sell it in the market for $S(T)$.

-If $S(T) \leq K$, the holder will not exercise the call, so he will gain nothing.
So the value of the European call option at expiry date is given by

$$\max\{S(T) - K, 0\}.$$

- **European put option** At $t = T$,
  -If $S(T) > K$, the holder should do nothing.
  -If $S(T) \leq K$, the holder may buy the asset at S(T) in the market and exercise the call by selling it at K.
  So the value of the European put option at expiry date is given by

$$\max\{K - S(T), 0\}.$$

Our asset price model is

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \quad S(0) = S_0, \tag{1.2}$$

where S(t) represents the stock price at time t, μ and σ are constant parameters. Applying Euler-Maruyama method,

$$\begin{cases} S_{t+1} = S_t + \mu S_t \Delta t + \sigma S_t \Delta t \\ S_0 = S(0) \end{cases}.$$

For a stock price S_0, at time t=0, the value of the call with expiration date t=T is the discounted expected value

$$C(S_0, T) = e^{-rT}\mathrm{E}[\max\{S(T) - K, 0\}], \tag{4.30}$$

where $S(T)$ is given by Eq.1.2. In addition to this insight, one of the most popular options pricing model is Black-Scholes formula which provides

Example 4.4.2
Assume that one share of Apple.Inc, AAPL stock has a price of \$188.18 at 16/05/2018. Consider a European call option with strike price \$170 and the exercise date at 16/11/2018, so that T=0.5 years. Assume that the expected rate of return $\mu = 30.85\%$, a fixed interest rate $r = 0.05$ and the volatility of the stock is $\sigma = 23.05\%$.

Listing 4.6: MATLAB code to compute European call option using EM method.

```
randn('state',100)
mu=0.3085; sigma=0.2305; Szero=188.18; r=0.05;K=170; %problem parameters
T=1/2; N=2^8; dt=T/N;
M=1000;  Xsum=zeros;payoff1=[]; payoff2=[]; Euler=[]; Mil=[];
for k=1:M
    dW=sqrt(dt)*randn(1,N);  %Brownian increments
    W=cumsum(dW);            %discretized Brownian path
    R=1; Dt=R*dt; L=N/R;     %L EM steps of size Dt=R*dt
    Stemp1=Szero;Stemp2=Szero;
    for j=1:L
        Winc=sum(dW(R*(j-1)+1:R*j));
        Stemp1=Stemp1+Dt*mu*Stemp1+sigma*Stemp1*Winc;
        Stemp2=Stemp2+Dt*mu*Stemp2+sigma*Stemp2*Winc+(1/2)*(sigma^2)*Stemp2*(Winc^2-Dt);
        Sem(j)=Stemp1;
        Smil(j)=Stemp2;
    end
    if Sem(end)-K>=0
    payoff1=[payoff1 exp(-r*T)*(Sem(end)-K)];
    else payoff1=[payoff1 0];
    end
    if Smil(end)-K>=0
    payoff2=[payoff2 exp(-r*T)*(Smil(end)-K)];
    else payoff2=[payoff2 0];
    end
end
Euler=mean(payoff1)
Milstein=mean(payoff2)
```

We perform a Monte Carlo simulation with 50000 repetitions to compute the expected value in (4.30). Use the EM and Milstein method to approximate the solution of (1.2) with a step size of $dt = 2^{-8}$ and initial value $S_0 = 188.18$. The value of the call calculated by EM is \$47.1737 and by Milstein is \$47.1707.

In reality, interest rate r and volatility σ are not constant but changing due to the market fluctuation. Hence, in the next part, we will consider a model to examine short-term interest rate dynamics.

4.4.3 Interest rate Estimation

We consider the model

$$dR(t) = \lambda(\mu - R(t))dt + \sigma\sqrt{R(t)}dW(t), \quad 0 \leq t \leq T, \tag{1.3}$$

where λ, μ and σ are positive constants and R(t) representative for short-term interest rate.

Chapter 5

Monte Carlo Simulation

5.1 Euler-Maruyama Method

In this section, we demonstrate the Monte Carlo simulation by using the Euler-Maruyama algorithm. We first simulate the following SDE

$$dX = \mu X dt + \sigma X dW, \; X(0) = X_0,$$

where $\mu = 4, \sigma = 1, X_0 = 2, M = 1000$. Discretized Brownian path has $dt = 2^{-10}$.

Listing 5.1: MATLAB code on EM method recomended by Higham (SIAM Review, 2001)

```
mu = 4; sigma = 1; Xzero = 2; % problem parameters
T = 1; N = 2^10; dt = T/N;
M=1000; Xend=[];
for k=1:M
    dW = sqrt(dt)*randn(1,N);      % Brownian increments
    W = cumsum(dW);                % discretized Brownian path
    R = 4; Dt = R*dt; L = N/R;     % L EM steps of size Dt = R*dt
    Xem = zeros(1,L);              % preallocate for efficiency
    Xtemp = Xzero;
    for j = 1:L
        Winc = sum(dW(R*(j-1)+1:R*j));
```

```
        Xtemp=Xtemp+Dt*mu*Xtemp+sigma*Xtemp.*Winc;
        Xem(j) = Xtemp;
    end
Xend=[Xend Xem(end)];
plot([0:Dt:T],[Xzero,Xem],'r-'), hold on
end
Xmean=mean(Xend)
xlabel('t','Fontsize',12)
ylabel('X','Fontsize',12,'Rotation',0,'HorizontalAlignment','right')
```

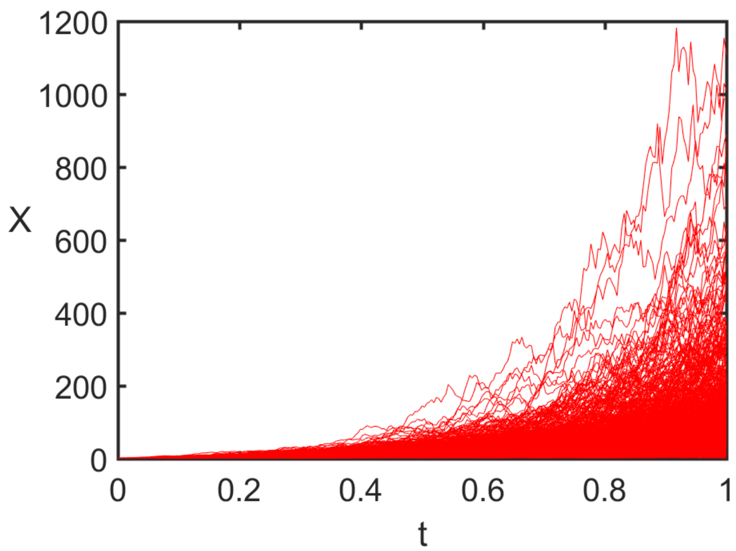

Figure 5.1: Euler-Maruyama approximation under Monte Carlo approach

Figure 5.1 illustrates the Monte Carlo simulation with $M = 1000$ paths, the drift $\mu = 4$, the volatility $\sigma = 1$. The initial condition is $X_0 = 2$ and the maturity time (the final time) $T = 1$. It can be seen that all the output $X(T)$ belong to the range $[0, 1200]$. Moreover, most of the output belong to the interval $[0, 400]$ and the average output is $X(T) = 111.0071$.

5.2 Milstein Method

SDE is

$$dX = rX(K - X)dt + \epsilon X dW, \; X(0) = X_0,$$

where $r = 2, \epsilon = 0.25, K = 1, X_0 = 1, M = 500$, discretized Brownian paths with $\delta t = 10^-2$.

Listing 5.2: MATLAB code on Milstein method

```
r=2; eps=0.25; k=1;Xzero =1; % problem parameters
T =10; N = T*10^3; dt = T/N;
t = zeros(N,1);               %time grid points
t(1) = 0;
M=500;
%Milstein's method
XMilstein = zeros(1,M);
X=zeros(1,N);
a=@(x) r*x*(k-x);
b=@(x) eps*x;
bprime=@(x) eps;
for i=1:M
    Xtemp = Xzero;
    dW = sqrt(dt)*randn(1,N);
    for j = 1:N
        t(j) = j*dt;
        Winc = dW(j); %Delta W_{j}
        Xtemp = Xtemp + a(Xtemp)*dt +
            b(Xtemp)*Winc+.5*b(Xtemp)*bprime(Xtemp)*(Winc^2-dt);
        X(j) = Xtemp;
    end;
Xmilstein(i)=X(N);
plot(t,X,'b-');
xlabel('t','Fontsize',12)
ylabel('X','Fontsize',12,'Rotation',0,'HorizontalAlignment','right')
hold on
end
```

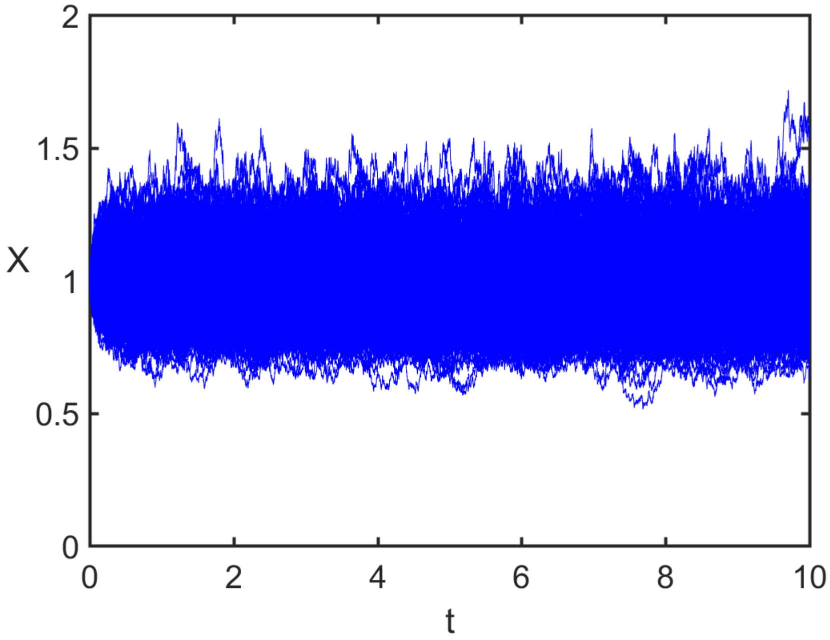

Figure 5.2: Milstein approximation under Monte Carlo approach

In Figure 5.2, similarly, Monte Carlo simulation is demonstrated with T=500 times and above given parameters. The simulation result shows that all the output belong to the interval [0.5,1.7] and seems to have symmetric fluctuation.

5.3 Probability density function

In this section, we implement the code continuing the previous MATLAB code (Listing 5.2, Milstein method) in order to examine the final output $X(T)$ produced by Milstein scheme, even though $X(T)$ is a random variable.

Particularly, we divide the interval of $X(T), [a, b]$, into K subinterval $[X_k, X_{k+1}]$ with length $delta\_int1 = \frac{b-a}{K}$. Then we count the number of $X(T)$ after M repeating paths in each subinterval, stored this count to $N(k)$. The probability density function of $X(T)$ is calculated as the proportion $\dfrac{N(k)}{M * delta\_int1}$ and is plotted using the midpoint of each subinterval.

Listing 5.3: MATLAB code to examine PDF of Milstein simulation's output at time T

```
%We now count number X(i) in each interval [X_int(k), X_int(k+1)]
    K=100;
```

```
    N=zeros(K,1);
    X_int=zeros(K+1,1);
    X_midpoint=zeros(K,1);
    a=0;b=2;
    delta_int1=(b-a)/K;
    for j=1:(K+1)
        X_int(j)=a+delta_int1*(j-1);
    end
    for j=1:K
        X_midpoint(j)=(X_int(j)+X_int(j+1))/2;
    end
    for j=1:K
        for i=1:M
            if (XMilstein(i)>=X_int(j))&(XMilstein(i)<=X_int(j+1))
                N(j)=N(j)+1;
            end
        end
    end
    p=zeros(K,1);
    for k=1:K
        p(k)=N(k)/(M*delta_int1);
    end
area=sum(p(1:K))*delta_int1;
plot(X_midpoint,p,'--.'); hold on
```

From the Figure 5.3, we have explored the probability density function of Milstein's output $S(T)$ obtained the normal distribution with mean 1.

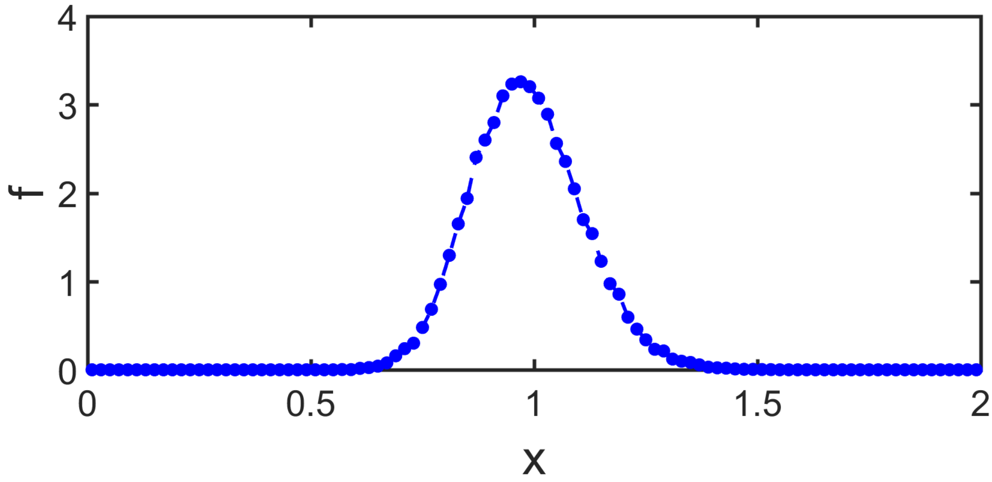

Figure 5.3: PDF of X(T)

Furthermore, to be more clarifying and reliable, we check whether the area under the PDF equal 1. The first simple way is to use Riemann sum. Another way which is more exactly is to use Trapezoidal rule means approximation the region under the graph of the function $f(x)$ as following

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + ... + 2f(x_{n-1}) + f(x_n)\right)$$

Listing 5.4: MATLAB code to examine the region under the PDF of $S(t)$

```
%Check if the area=1 by Riemman sum (a simple and lazy way ^^)
area1=sum(p(1:K))*delta_int1;
%Check if the area=1 by Trapezoidal rule, more exactly
area2=0.5*(p(1)+2*sum(p(2:K-1))+p(K))*delta_int1;

%Find the probability P(0.8<X<1.2)=area(p(40:60)), again a simple way:
%using Riemman sum
area3=sum(p(40:K-40))*delta_int1;
%using Trapezoidal rule, more exactly
area4=0.5*delta_int1*(p(40)+2*sum(p(41:K-41))+p(K-40));
```

Applying these two manners to examine the probability $P(0.8 < X < 1.2)$ gives us the result by Riemann sum is 91.11% and by Trapezoidal rule is 89.26%.

5.4 Comparison the theoretical solution and numerical solution

In this section, we compare the theoretical and numerical solution. As the previous section, we consider the SDE

$$dX = \mu X dt + \sigma X dW, \ X(0) = X_0,$$

where $\mu = 2, \sigma = 1, X_0 = 1$, $dt = 2^{-10}$. Numerical method use timestep $R * dt$. We simulate the above SDE by EM, Milstein and second order method, and compare with the theoretical solution. We illustrate simultaneously the exact solution and approximate solution in the same figure.

Listing 5.5: MATLAB code illustrating the exact solution and approximation using EM and Milstein method

```
mu = 2; sigma = 1; Xzero = 1;  % problem parameters
T = 1; N = 2^10; dt = T/N;
dW = sqrt(dt)*randn(1,N);      % Brownian increments
W = cumsum(dW);                % discretized Brownian path
Xtrue = Xzero*exp((mu-0.5*sigma^2)*([dt:dt:T])+sigma*W);
plot([0:dt:T],[Xzero,Xtrue],'m-'), hold on
R = 4; Dt = R*dt; L = N/R;     % L steps of size Dt = R*dt
Xem = zeros(1,L);              % preallocate for efficiency
Xtemp1 = Xzero;Xtemp2=Xzero;Xtemp3=Xzero;
for j = 1:L
   Winc = sum(dW(R*(j-1)+1:R*j));
   Xtemp1=Xtemp1+Dt*mu*Xtemp1+sigma*Xtemp1*Winc;
   Xtemp2=Xtemp2+Dt*mu*Xtemp2+sigma*Xtemp2*Winc
         +(1/2)*(sigma^2)*Xtemp2*(Winc^2-Dt);
   DI=Winc*Dt/2;
   Xtemp3=Xtemp3+Dt*mu*Xtemp3+sigma*Xtemp3*Winc
         +(1/2)*(sigma^2)*Xtemp3*(Winc^2-Dt)
         +(mu^2*Xtemp3)*1/2*Dt^2+mu*sigma*Xtemp3*DI+(mu*sigma)*(Dt*Winc-DI);
```

```
   Xem(j) = Xtemp1;
   Xmil(j)= Xtemp2;
   Xhigh(j)=Xtemp3;
end
plot([0:Dt:T],[Xzero,Xem],'g-',[0:Dt:T],[Xzero,Xmil],'b-',[0:Dt:T],[Xzero,Xhigh],'r-'),
    off
legend('Exact solution','Solution by EM method','Solution by Milstein
    method','Solution by 2nd-order method')
xlabel('t','Fontsize',12)
ylabel('X','Fontsize',12,'Rotation',0,'HorizontalAlignment','right')
emerr = abs(Xem(end)-Xtrue(end))
milerr= abs(Xmil(end)-Xtrue(end))
higherr=abs(Xhigh(end)-Xtrue(end))
```

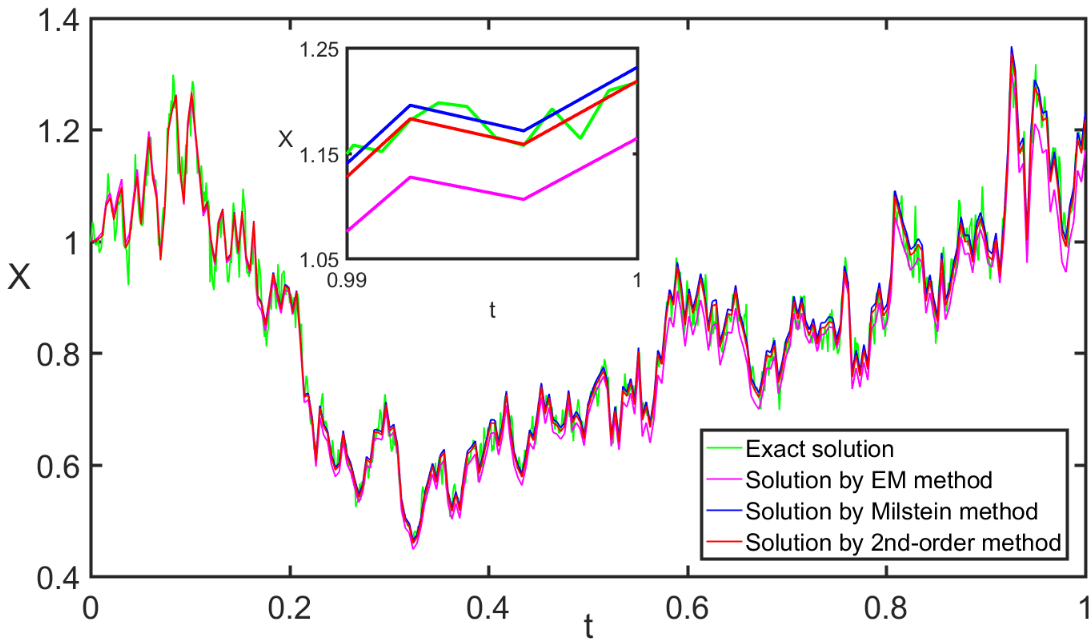

Figure 5.4: Exact solution (in green) and approximations using Euler-Maruyama (in purple), Milstein (in blue) and second order method methods (in red).

In figure 5.4, we demonstrate our theoretical solution by the green solid line, while the purple line stands for EM solution, the blue one is representative for solution using

Milstein method and the red line express the second order solution. At the top right corner of this figure, we insert an inset figure (the magnified one), in order to clearify the position of these solutions. As can be seen from the figure and its inset figure , the result by second order method is more accuracy than Milstein's method and more than EM. Thus the red line (2nd-order) is closer to the blue line (exact solution) than the blue (Milstein) and the purple one (EM). Indeed, we measure $|X_{2nd-order}(T) - X_{exact}(T)| = 0.0013 < |X_{Milstein}(T) - X_{exact}(T)| = 0.0144 < |X_{EM}(T) - X_{exact}(T)| = 0.0530$, for $T = 1$.

Chapter 6

Conclusions and future study

The objective of this thesis is to study the numerical solution of SDEs and implement the algorithms to demonstrate the Monte Carlo simulations for SDEs, including financial models in the stock market. These methods are based on the paper by Prof. Desmond Higham published in SIAM Review in 2001 [1].

More specifically, we study the Euler Maruyama and Milstein schemes for numerically solving the SDEs. The Euler Maruyama scheme has the order of accuracy as of $\Delta t^{1/2}$ while the Milstein scheme has the higher order of Δt which does not require the high resolution when discretizing the time $[0, T]$ compared to the Euler-Maruyama scheme. Furthermore, we have derived a higher order method of second order of accuracy. In order to clarify these methods, we have introduced respectively the basic background of stochastic calculus such as Brownian motion, mean-square convergence and explained the difference between the Itô integral and Stratonovich integral. We derived the Itô lemma for studying the SDEs of changing of variables and proposing the theoretical solution of SDEs and use this lemma to derive the theoretical solution (exact solution) of the geometric Brownian motion. This is a benchmark to figure out the accuracy of our numerical methods by verifying with theoretical solution and the numerical solution of the geometric Brownian motion.

The main part of this thesis is to study the derivation for the Euler-Maruyama, Milstein and the higher order (second order) algorithm. We numerically implemented these algorithms and then verified that the EM method converges with strong order 1/2, i.e., $\Delta t^{1/2}$, and weak order 1, while the Milstein method achieves order 1 in terms of both the strong and weak convergence. On the other hand, we have investigated that as time step Δt decreased, the error of each method correspondingly declined due to their order of strong or weak convergence.

In addition, we also demonstrate the practical applications of these methods in computational finance such as the financial asset pricing model, particularly the European option and short-term interest rates. We figured out that the larger number of repeating paths in Monte Carlo simulation, the more precise simulation result is. This fact is consistent with the theory if Monte Carlo simulation.

Based on using Monte Carlo method, the numerical simulation results were presented in Chapter 5 of this thesis. We investigated the numerical results by taking the average result and/or the probability distribution function. We also distinguished the three numerical methods by illustrating and validating their level of accuracy in simulations.

Due to the limit of time, there are a few issues that we have not studied and discussed in this thesis. These consist of the linear stability, typical existence and uniqueness theorems of the solution for SDEs. We also have not mentioned the application of SDEs in modeling volatility dynamics, which is also one of SDE's major usage in finance, besides stock prices and interest rates. We expect to study them more carefully in the future.

Bibliography

[1] Desmond J. Higham, *An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*, SIAM Review, Vol.43, No. 3,pp. 525–546, 2001

[2] Lawrence C. Evans,*An introduction to stochastic differential equations*, University of California, Berkeley, Berkeley, CA, 2013

[3] P.E.Kloeden,and E.Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, Berlin, 1992.

[4] X. Mao, *Stochastic Differential Equations and Applications*, Horwood, Chichester, 1997

[5] J.R. Klauder, W.P. Petersen,*Numerical integration of multiplicative-noise stochastic differential equations*, SIAM J. Numer. Anal., 1985.

[6] J. C. Hull, *Options, Futures, and Other Derivatives*, 4th ed., Prentice-Hall, Upper Saddle River, NJ, 2000.

[7] J. C. Cox, J. E. Ingersoll, and S. A. Ross, *A theory of the term structure of interest rates*, Econometrica, pp. 385–407, 1985.

[8] Albus Einstein,*Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen*, 1905.

[9] Langevin,*Sur la théorie du mouvement brownien [On the Theory of Brownian Motion]*. C. R. Acad. Sci. Paris, 1908

[10] Brown, Robert *A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies*, Philosophical Magazine, 1928.

[11] W. E. Boyce, *Approximate solution of random ordinary differential equations*,1978.

[12] H. J. Kushner, *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.* Academic Press, New York, 1977.

[13] D. J. Clements, and B. D. O. Anderson, *Well behaved Ito equations with simulations that always misbehave*, IEEE Trans. Automat. Contr., 1973.

[14] W. Wagner, and E. Platen, *Approximation of Ito integral equations*, Preprint ZIMM, Akad. der Wiss. der DDR, Berlin. 1978.

[15] P.E.Kloeden,and E.Platen, *The Stratonovich- and Ito-Taylor expansions*, Math. Nachr. 151, 1991.

[16] D.Talay, *Efficient numerical schemes for the approximation of expectations of functional of the solution of an SDE and applications*. Springer, 1984.

[17] G.N.Milstein, *A method of second-order accuracy integration of stochastic differential equations*, 1978.

[18] W. Rümelin, *Numerical treatment of stochastic differential equations*, SIAM J. Numer. Anal., 1982.

[19] Hurn A.S. Jeisman J.I. and Lindsay K.A.,*Seeing the wood for the tree: A critical evaluation of methods to estimate the parameters of stochastic differential equations*, Journal of Financial Economics, 2007.

[20] De Rossi G.,*Maximum likelihood estimation of the Cox-Ingersoll-Ross model using particle filter*, Computational Economics, 2010.

[21] Merton R.C.,*Option pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, 1976.

[22] Cont R. and Tankov P.,*Financial modelling with jump processes (Financial Mathematics Series)*, Chapman and Hall/CRC, London, UK, 2004.

[23] Zanzotto P.A.,*On solutions of one-dimensional stochastic differential equations driven by stable Levy motion*, Stochastic Processes and their Applications, 1997.

[24] He S.W, Wang J.G. and Yan J.A., *Semimartingale theory and stochastic calculus*, CRC Press, Boca Raton, FL, 1992.

[25] Kiyosi Itô, *Stochastic Integral.*, Proc. Imperial Acad, Tokyo, 1944

[26] Gregory.F.Lawler,*Introduction to Stochastic Processes*, 2nd ed., Chapman and Hall/CRC

[27] Timothy Sauer, *Numerical Analysis*, George Mason University, 2006