

HYBRID NUMERICAL METHODS FOR STOCHASTIC DIFFERENTIAL EQUATIONS

by

IKPE DENNIS CHINEMEREM

Submitted in fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In the subject

APPLIED MATHEMATICS

UNIVERSITY OF SOUTH AFRICA

February 2009



ABSTRACT

In this dissertation we obtain an efficient hybrid numerical method for the solution of stochastic differential equations (SDEs). Specifically, our method chooses between two numerical methods (Euler and Milstein) over a particular discretization interval depending on the value of the simulated Brownian increment driving the stochastic process. This is thus a new¹ adaptive method in the numerical analysis of stochastic differential equation. Mauthner (1998) and Hofmann et al (2000) have developed a general framework for adaptive schemes for the numerical solution to SDEs, [30, 21]. The former presents a Runge-Kutta-type method based on stepsize control while the latter considered a one-step adaptive scheme where the method is also adapted based on step size control. Lamba, Mattingly and Stuart, [28] considered an adaptive Euler scheme based on controlling the drift component of the time-step method. Here we seek to develop a hybrid algorithm that switches between euler and milstein schemes at each time step over the entire discretization interval, depending on the outcome of the simulated Brownian motion increment. The bias of the hybrid scheme as well as its order of convergence is studied. We also do a comparative analysis of the performance of the hybrid scheme relative to the basic numerical schemes of Euler and Milstein.

¹in the sense that the adaptiveness depends on the Brownian motion increment and not on the step size.

ACKNOWLEDGMENTS

I wish to first and foremost acknowledge the efforts of my supervisor, Dr. E Rapoo. Her patience, understanding and support is one of a kind. Eeva, I would wish for a supervisor of your kind in my future academic research works. I also would want to thank the Director of the School of Science, Dr. Gugu Moche and the Executive Dean of the College of Science, Engineering and Technology, Prof. Mamokgethi Setati for believing not only in my academic abilities but also in my extracurricular interests by affording me the opportunity to be part of “MathsEdge”, a flagship community engagement project of the college. For this, I shall remain ever grateful.

This acknowledgment won't be complete if I fail to acknowledge the entities on whose platform this Masters dissertation was conducted. I thank UNISA for not only providing me the opportunity to be identified with it but for also providing me with financial support in the form of tuition fees and field work associated costs. I also thank the South African National Research Foundation, NRF for the financial support through the 2007 Africa Scholarship for non-South African citizens.

Last but not least, I wish to thank all members of my family, my mum in particular for single-handedly raising me with much care, love and support and for her encouragement to me during the period of this research work. My friends from Wits university: Gideon, Okeke, Florence, Bulelwa and Thibaut, my friends from Unisa, Martin, Kennedy, Rendani, Phila and Raymond, your interests and encouragements in this work kept me going all through the period of this dissertation. I also wish to thank Farai from UCT for the valuable comments after reading this manuscript.

To God almighty I give all the glory for giving me the wisdom and understanding that enables me to pursue a career in such a subject area as Applied Mathematics.

DECLARATION

I hereby certify that this project was independently written by me. No material was used other than that referred to. Sources directly quoted and ideas used, including figures, tables, sketches, drawings and photos, have been correctly denoted. Those not otherwise indicated belong to the author.

Contents

Table of Contents	vi
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Outline of the Research	2
2 General Theory of Stochastic Differential Equations	4
2.1 Stochastic process	4
2.2 Brownian Motion	5
2.2.1 Equivalence transformations	5
2.3 Stochastic Integrals	7
2.3.1 Construction of the stochastic integrals	7
2.3.2 Itô integrals	10
2.3.3 Stratonovich Integral	10
2.4 Stochastic differential equations, SDEs	11
2.4.1 Existence and Uniqueness of the Solution of SDEs	12
2.4.2 Itô Process	13
3 Numerical Methods for solutions of SDEs	15
3.1 Introduction	15
3.2 Stochastic Taylor Expansions	16
3.2.1 Strong and Weak Convergence of Stochastic Taylor Approximation	17
3.3 Euler-Maruyama Method	17
3.3.1 Numerical Example	19
3.3.2 Strong and Weak Order of Convergence of the Euler Scheme	20
3.4 Milstein method	23
3.4.1 Strong Convergence of the Milstein method	26
3.5 Order 1.5 stochastic schemes	29

4	Adaptive Numerical Schemes for SDEs	33
4.1	Adaptive schemes for ODEs and R-K schemes	34
4.2	An Adaptive Euler-Maruyama scheme for SDEs	35
4.3	An asymptotically adaptive algorithm	36
4.4	An adaptive algorithm based on embedded Runge-Kutta scheme . . .	37
4.5	A variable Riemannian sum for Itô integrals	38
4.6	Summary: adapted approaches	39
5	Hybrid Numerical Methods for the Solution of SDEs	41
5.1	Introduction	41
5.2	Euler-Milstein hybrid Scheme	42
5.2.1	An intuitive illustration of the order convergence of the Euler-Milstein hybrid scheme	43
5.2.2	Bias of the hybrid scheme	49
5.2.3	Strong order of the error of the hybrid scheme	53
6	Numerical Experiments	65
6.1	The ϵ -Complexity of the Hybrid Scheme	65
6.2	Numerical Comparison	66
6.2.1	Strong Orders of Convergence	67
6.2.2	Efficiency	70
7	Summary	76
A	MATLAB CODES FOR HYBRID METHOD PLOTS	79
A.1	Matlab Codes: Hybrid Bias	79
A.2	Matlab Codes:hybrid order of Convergence	83
A.3	Matlab Codes: Numerical Experiments	86

List of Figures

2.1	A single discretised Brownian motion path	7
2.2	50 discretised Brownian motion paths	8
3.1	Euler Approximation of an SDE	21
3.2	Strong Order of Euler convergence	22
3.3	Milstein Approximation of an SDE	27
3.4	Strong convergence of the Milstein scheme	28
5.1	graph of the functions G_1 and C_1	51
5.2	Bias of the hybrid scheme vs h , $\epsilon = 1, 2, h = 0 : 10$	52
5.3	bias of the hybrid scheme vs h , $\epsilon = 1, 7, h = 0 : 10$	53
5.4	Bias of the hybrid scheme vs h , $\epsilon = 0.1; h = N * \epsilon, N = [2 : 10]$	54
5.5	Bias of the hybrid scheme vs h , $\epsilon = 1, h = N * \epsilon, N = [2, 40]$	55
5.6	Bias of the hybrid scheme vs h , $\epsilon = 0.000001, h = 0.001:0.01$	56
5.7	bias of the hybrid scheme vs h , $\epsilon = 0.1, h = 0 : 011$	57
5.8	Mean square error vs h , $\epsilon = 0.00001$	58
5.9	Numerically calculated global hybrid error vs ϵ , $\epsilon = [3 * 10^{-6}, 5.5 * 10^{-5}, 0.0001], h = 0.001$	59
5.10	Maximum mean square error vs h , $\epsilon = 0.00001$	60
5.11	Mean square error of the extra term vs h , $\epsilon = 0.00001$	60
5.12	Mean square error of the extra term vs h , from left,right to bottom $\epsilon = 0.1, 0.01, 0.00001$	61
5.13	Mean square error of the extra term vs h , $\epsilon = 0.00001$	61
5.14	Order of hybrid method vs h , for different values of ϵ	62
5.15	Order of the hybrid method vs h , for different values of ϵ	63
5.16	Order of convergence for different methods	64
6.1	Comparison of the strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$ 3. SDE(1) with $\mu = 1$ and $\sigma = 2$. In the calculation we use $\epsilon = 0.000054$ and , $h = 0.001$, for Euler and $h = 0.02$ for Milstein	68
6.2	Comparison of the strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(2) 2. SDE(3)	69

6.3	Comparison of strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$. 3. SDE(1) with $\mu = 1$ and $\sigma = 2$. . .	70
6.4	Comparison of strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(2) 2. SDE(3)	71
6.5	Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$. 3.SDE(1) with $\mu = 1$ and $\sigma = 2$. . .	73
6.6	Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost, (SDE(2)).	74
6.7	Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost, (SDE(3)).	75

List of Tables

- 3.1 Average terminal errors of the Euler method 20
- 3.2 Average terminal values and errors of the Milstein method 27

Chapter 1

Introduction

The theory of stochastic differential equations is well developed (See[33]). Applications of SDEs are available in many fields of research reaching from mathematical finance to optimal control, helicopter rotor and satellite orbit stability, blood clotting dynamics and theoretical physics. Analytical solutions of stochastic differential equations can be found for few SDEs, consequently the study of their numerical solutions has become essential. Many numerical methods for solving stochastic differential equations have been developed in recent years (see [26, 27, 32]) but there is still a huge space left to be filled in this area of numerical analysis: There is a need for efficient methods with low complexity. Lehn et al (2003) noted that some of the proposed methods in the recent literature are unrealistic; some of these have never been implemented or tested. For instance, methods of higher order convergence require the simulation of the correlated Itô integrals of the Itô stochastic Taylor expansion¹, a difficult and time consuming task or, as in the case of Monte Carlo methods, need generation of too many Brownian motion paths, a costly exercise in most applications. To this end, the Euler and Milstein scheme methods remain the method of choice for many. This is mainly due to their efficiency and low complexity (see[29]). However, greater efficiency in the strong sense is necessary for higher dimensional applications;

¹See [29] for a comprehensive study of stochastic Taylor expansions

in this case, adaptive schemes may be one solution. But still, in the adaptive scheme arena, Euler-Maruyama and Milstein schemes are the ideal method for many.

The adaptive scheme environment is dominated by step-size adaptiveness. In this dissertation we propose a completely new type of method, a hybrid adaptive method, where instead of step-size the adaptiveness is based on the increment of Brownian motion. More specifically, our approach is a numerical method that chooses a method (Euler or Milstein) based on the outcome of the generated Brownian motion process. We prove that the new method outperforms both Euler and Milstein in giving better accuracy for the same average cost. In what follows, we give a brief outline of the structure of the dissertation.

1.1 Outline of the Research

, **Chapter Two** Introductory topic: A brief explanation of stochastic differential equations and a detailed discussion of their numerical properties will begin this section. Existence and uniqueness theorems for SDEs are also stated.

Chapter Three Numerical Methods for SDEs: The basic numerical methods for stochastic differential equations are derived from the stochastic Taylor expansions of the previous chapter. The convergence properties of these methods are also studied.

Chapter Four Adaptive numerical schemes for SDEs: some adaptive schemes for the solution of SDEs in the literature will be presented.

Chapter Five Hybrid Euler-Milstein Algorithm: The structure of the adaptive hybrid method will be given and the main choice criterion will also be presented. Convergence and analysis of the error of this method will be studied.

Chapter Six: Comparison: We start this chapter by discussing briefly the computational complexity of numerical methods. We compare the performance with respect to accuracy and computational complexity of the Hybrid method with those of ordinary Euler and Milstein methods.

Chapter Seven: Conclusion: We conclude using the results obtained in Chapters five and six.

Chapter 2

General Theory of Stochastic Differential Equations

This chapter is of an informative nature; the necessary definitions are given together with statements of theorems. The intention is to make the work somewhat self-contained and place the rest in proper perspective. We start this introductory chapter with the definition and properties of stochastic processes. A basic knowledge of measure theoretical probability theory is assumed. Please see [11, 23, 9] for details.

2.1 Stochastic process

Definition 2.1.1. *A continuous-time stochastic process is a family of random variables $\{X(t) = X_t, t \in [0, \infty)\}$ defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and indexed by $[0, \infty)$.*

A stochastic process will have the following properties:

- For fixed t it is a random variable, i.e $X_t = X_t(\omega)$
- For fixed ω it is a function of time : $t \rightarrow X_t(\omega) = X(t, \omega)$.

This function is called a sample path (realization or trajectory) of the stochastic process $X = (\{X_t\}_{t \geq 0})$

- It can also be seen as a function of two variables: $X(t, \omega) \rightarrow X_t(\omega)$.

A more specific example and a process which is commonly used in the modelling of real life applications is the Brownian motion process.

2.2 Brownian Motion

Definition 2.2.1. A stochastic process B on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = \{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$ is called a standard Brownian motion w.r.t $\mathbb{F} = \{\mathcal{F}_t\}_{t \geq 0}$, if the following properties hold.

1. $B_0 = 0$ almost surely
2. $B_t - B_s$ is independent of \mathcal{F}_s for all $t \geq s$
3. $B_t - B_s$ has a normal distribution with zero mean and variance $t - s$
i.e $B_t - B_s \sim \mathcal{N}(0, t - s)$, for all $0 \leq t < s < \infty$
- 4.

Zero mean of the Brownian motion reflects that the Brownian particles is as likely to go up as it is to go down. The independent increment property implies that for any integer $k \geq 1$ and real numbers $0 = t_0 < t_1 < t_2 < \dots < t_k < \infty$, the increments

$$B_{t_1} - B_{t_0}, B_{t_2} - B_{t_1}, \dots, B_{t_k} - B_{t_{k-1}}$$

are independent. In other words the displacement is *lack of memory*, the displacement of the Brownian particle during $[t_0, t_{k-1}]$ in no way influences the displacement during $[t_{k-1}, t_k]$. The variance of the Brownian increment grows linearly as the length of the interval increases.

2.2.1 Equivalence transformations

Let B_t be a standard Brownian motion. Then the following processes are also standard Brownian motions.

1.

$$X_1(t) = cB_{t/c^2}, (c > 0 \text{ a fixed constant})$$

2.

$$X_2(t) = tB_{1/t}, (t > 0)$$

3.

$$X_3(t) = B_{t+h} - B_h, (h > 0 \text{ a fixed constant})$$

4.

$$X_4(t) = -B_t$$

Definition 2.2.2 (Arithmetic Brownian motion). *Let $\{B_t, t \geq 0\}$ be a standard Brownian motion and let $x_0 \in \mathbb{R}$ then the process*

$$X_t = x_0 + \mu t + \sigma B_t, t \geq 0$$

is called arithmetic Brownian motion with drift μ and diffusion σ .

Arithmetic Brownian motion has stationary independent increments and has a normal distribution $\mathcal{N}(\mu t, \sqrt{\sigma t})$. It has continuous paths starting at $X_0 = x_0$.

Definition 2.2.3 (Geometric Brownian motion). *Let $\{B_t, t \geq 0\}$ be a standard Brownian motion, then the process*

$$X_t = \exp(x_0 + \mu t + \sigma B_t), t \geq 0$$

is called Geometric Brownian motion with drift μ and diffusion coefficient σ

For computational purposes, we usually consider discretised Brownian motion over a given interval $[0, T]$, with B_t specified at discrete points t . Figure (2.2.1) below shows one simulation of discretised Brownian motion over $[0, 1]$ while Figure(2.2.1) shows 50 paths of the Brownian motion. (See [20] or Appendix A for more on how to generate discretised Brownian Motion using MATLAB)

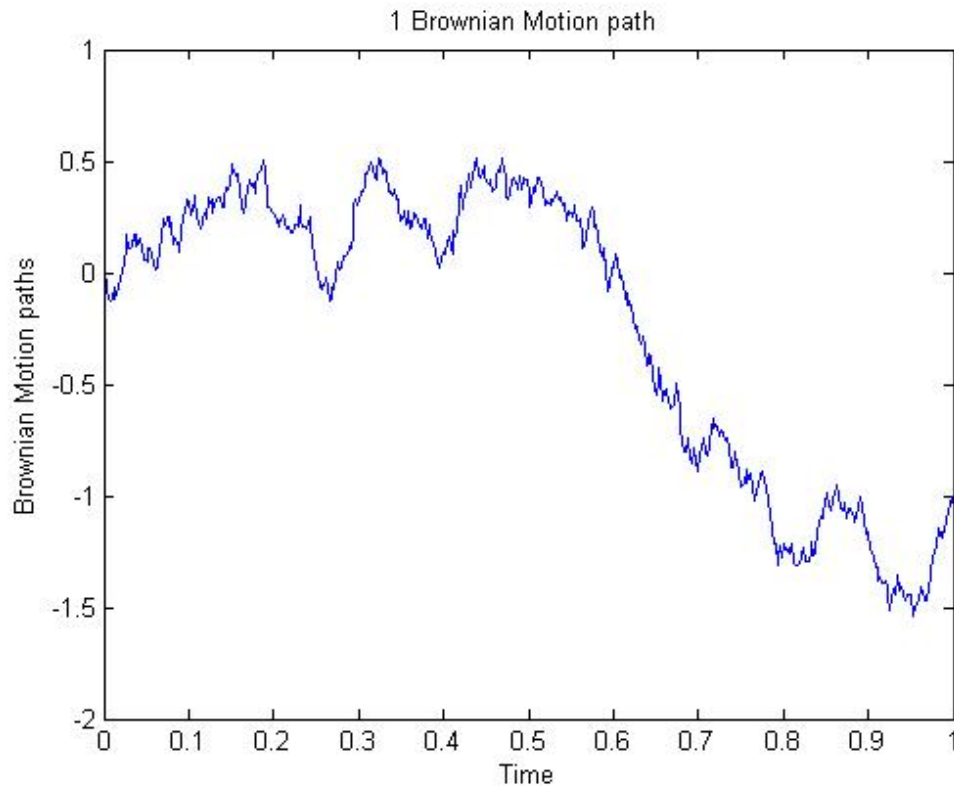


Figure 2.1: A single discretised Brownian motion path

2.3 Stochastic Integrals

In this section, we give a brief discussion of the meaning of a stochastic integral and differentiate between the two most popular forms of stochastic integrals in literature: the Itô and Stratonovich integrals.

2.3.1 Construction of the stochastic integrals

Our goal is to give meaning to the stochastic integral $\int_0^T H(t)dB(t)$ for a certain class of processes adapted to the filtration¹ $(\mathcal{F}_t)_{t \geq 0}$ where B_t is a Brownian motion w.r.t. $\{\mathcal{F}_t\}$. We shall define the integral for *simple processes*, that is processes which

¹See [33, 8, 9, 36] for more on adapted processes, measurability and Filtration

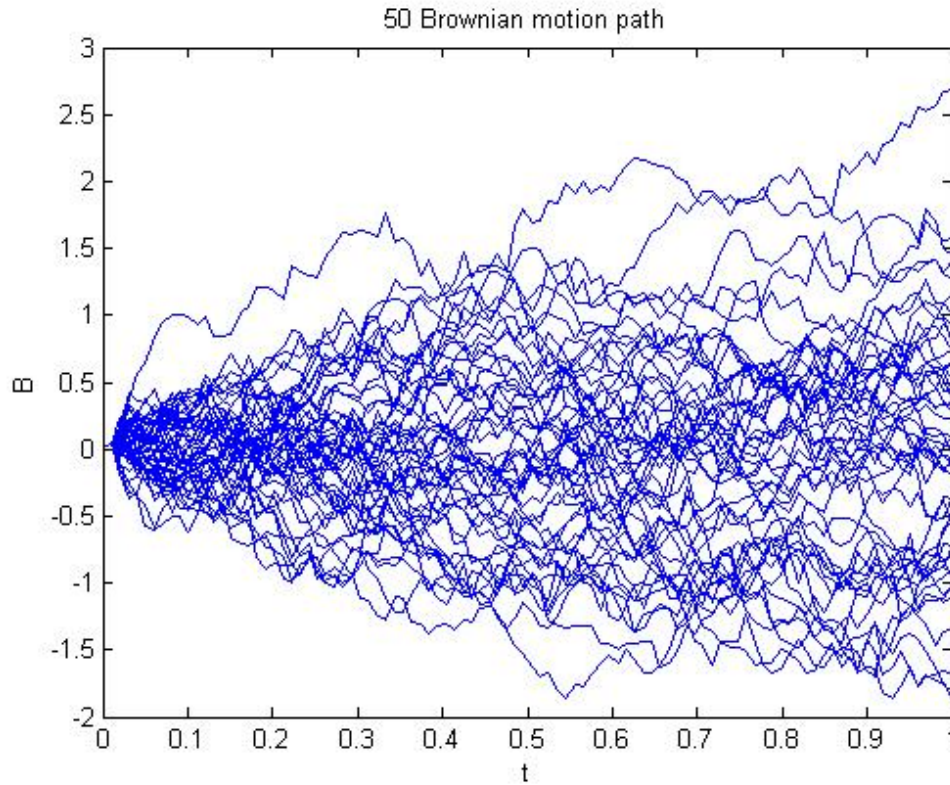


Figure 2.2: 50 discretised Brownian motion paths

are constant on finitely many intervals. By a limiting procedure the integral is then defined for more general processes.²

Definition 2.3.1. $(H_t)_{0 \leq t \leq T}$ is called a simple process if it can be written as

$$H_t(\omega) = \sum_{i=1}^N \phi_i(\omega) 1_{(t_{i-1}, t_i]}(t)$$

where $0 = t_0 < t_1 < \dots < t_N = T$ and ϕ_i is $\mathcal{F}_{t_{i-1}}$ -measurable and bounded.

Then by definition, the stochastic integral of a simple process H is the continuous process $\int_0^T H_s dB_s$ defined for any $t \in (t_k, t_{k+1}]$ almost surely

$$\int_0^T H_s dB_s = \sum_{1 \leq i \leq k} \phi_i(B_{t_i} - B_{t_{i-1}}) + \phi_{k+1}(B_t - B_{t_k}).$$

Stochastic integrals are first defined for simple processes then by extension defined for a general class of regular adapted processes.

² See [5, 26] for the full details of these processes and the proof of their properties

Let $H^n(t)$ be a sequence of simple processes convergent in probability to the process, $H(t)$. Then under some conditions³, the sequence of their integrals $\int_0^T H^n(t)dB(t)$ also converges in probability to a limit J . The random variable J is taken to be the integral $\int_0^T H(t)dB(t)$.

Theorem 2.3.2. *Let $H(t)$ be a regular adapted process such that with probability one $\int_0^T H^2(t)dt < \infty$. Then the stochastic integral $\int_0^T H(t)dB(t)$ is defined and has the following properties (See [5], pg 97-98 for the proof.).*

1. Linearity. *If the stochastic integrals of $H(t)$ and $H'(t)$ are defined and c_1 and c_2 are constants then*

$$\int_0^T (c_1H(t) + c_2H'(t))dB(t) = c_1 \int_0^T H(t)dB(t) + c_2 \int_0^T H'(t)dB(t)$$

2. $\int_0^T H(t)I_{(a,b]}(t)dB(t) = \int_a^b H(t)dB(t)$.

3. Zero mean property. *Let*

$$\int_0^T E(H^2(t))dt < \infty \tag{2.1}$$

If condition(2.1) holds then

$$E \left(\int_0^T H(t)dB(t) \right) = 0 \tag{2.2}$$

4. Isometry property. *If condition (2.1) holds then*

$$E \left(\int_0^T H(t)dB(t) \right)^2 = \int_0^T E(H^2(t))dt. \tag{2.3}$$

In what follows we point out by way of definition the difference between the two most popular types of stochastic integrals.

³See [5] pg. 36-38

2.3.2 Itô integrals

Suppose $H(t)$ is a suitable function,⁴ Then the integral $\int_0^T H(t)dt$ may be approximated by the Riemann sum

$$\sum_{i=0}^{N-1} H(t_i)(t_{i+1} - t_i), \quad (2.4)$$

where $t_i = i\Delta t$ with $\Delta t = T/N$. The exact value of the above integral is obtained by taking the limiting value of (2.4) as $\Delta t \rightarrow 0$. Analogously, we may consider a sum of the form

$$\sum_{i=0}^{N-1} H(t_i)(B(t_{i+1}) - B(t_i)), \quad (2.5)$$

as with (2.4), the above sum,(2.5) may be regarded as the approximation to the stochastic integral, $\int_0^T H(t)dB(t)$. This form of stochastic integral is known as the *Itô integral*. Here, we are integrating H with respect to the Brownian motion $B(t)$ over the interval $[0, T]$ with H evaluated in the approximation (2.5) at the left-end point of each subinterval $[t_i, t_{i+1}]$. Note that H can be a stochastic process depending on the Brownian motion. In particular, we can set $H(t) = B(t)$.

2.3.3 Stratonovich Integral

From deterministic integration (see [13]), the sum given by

$$\sum_{i=0}^{N-1} H\left(\frac{t_i + t_{i+1}}{2}\right)(t_{i+1} - t_i), \quad (2.6)$$

is also a Riemann sum approximation to $\int_0^T H(t)dt$. Hence the corresponding alternative to (2.5) is

$$\sum_{i=0}^{N-1} H\left(\frac{t_i + t_{i+1}}{2}\right)(B(t_{i+1}) - B(t_i)), \quad (2.7)$$

Unlike the deterministic case, (2.5) and (2.7) do not result in the same approximate value to $\int_0^T H(t)dB(t)$ even as $\Delta t \rightarrow 0$. This midpoint Riemann-like sum (2.7)

⁴regular adapted process and $\int_0^T |H_s|^2 ds < \infty$ \mathbb{P} almost surely. See [8] for the definition of regular adapted processes and additional properties of H

produces the Stratonovich integral⁵. Itô and Stratonovich integrals both have their uses in many applications. Throughout this dissertation we shall consider stochastic integrals in the Itô sense. However, it is a simple exercise to transform an Itô integral to a Stratonovich integral and vice versa (see [26]). Having introduced the stochastic integrals, we now define the stochastic differential equations, whose components are stochastic integrals.

2.4 Stochastic differential equations, SDEs

Let $B(t)$, $t \geq 0$, be Brownian motion process. An equation of the form

$$dX(t) = K((t, X_t))dt + H((t, X_t))dB(t) \quad (2.8)$$

where functions $K(t, x)$ and $H(t, x)$ are given and X_t is the unknown process, is called a stochastic differential equation (SDE) driven by the Brownian motion.

The functions $K(t, x)$ and $H(t, x)$ are called the drift and diffusion coefficients, respectively.

This equations are used to model many financial assets, like stocks or interest rate processes.

Definition 2.4.1. *A process $X(t)$ is called a strong solution of the SDE (2.8) if for all $t > 0$ the integrals $\int_0^t K(s, X(s))ds$ and $\int_0^t H(s, X(s))dB(s)$ exist (the second being an Itô integral) and*

$$X(t) = X(0) + \int_0^t K(s, X(s))ds + \int_0^t H(s, X(s))dB(s) \quad (2.9)$$

We now state a few theorems about the solution of stochastic differential equations. We also state and briefly discuss some important properties of the solution to SDEs.

⁵Alternatively, Stratonovich integral is $\sum_{i=0}^{N-1} \frac{1}{2}(H(t_i) + H(t_{i+1}))(B(t_{i+1}) - B(t_i))$

2.4.1 Existence and Uniqueness of the Solution of SDEs

Consider the nonautonomous⁶ stochastic differential equation

$$dX_t = K(t, X_t)dt + H(t, X_t)dB_t, \quad (2.10)$$

which in the integral form is given by

$$X_t = X_{t_0} + \int_{t_0}^t K(s, X_s)ds + \int_{t_0}^t H(s, X_s)dB_s \quad (2.11)$$

where the first integral is a Riemann (or Lebesgue) integral for each sample path and the second integral is an Itô integral. In order to ensure that these integrals are meaningful, some regularity conditions are required of the drift and the diffusion coefficient in (2.10), that is K and H above. This in turn ensures that the solution to (2.10) exists and with additional assumptions is unique. Throughout this dissertation we shall consider SDEs whose drift and diffusion coefficient satisfy the following properties: Suppose that t_0 , $0 \leq t_0 \leq T$ is arbitrary and fixed and that the coefficient functions $K, H: [t_0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ are given and satisfy the following

- C1 (Measurability): $K = K(t, X_t)$ and $H = H(t, X_t)$ are jointly (\mathcal{L}^2) measurable in $(t, X_t) \in [t_0, T] \times \mathbb{R}$;
- C2 (Lipschitz condition): There exist a constant $C > 0$ such that

$$|K(t, x) - K(t, y)| \leq C|x - y| \text{ and } |H(t, x) - H(t, y)| \leq C|x - y|$$

for all $t \in [t_0, T]$ and $x, y \in \mathbb{R}$.

- C3 (Linear growth bound): There exist a constant $C > 0$ such that

$$|K(t, x)|^2 \leq C^2(1 + |x|^2) \text{ and } |H(t, x)|^2 \leq C^2(1 + |x|^2)$$

for all $t \in [t_0, T]$ and $x, y \in \mathbb{R}$.

⁶drift and diffusion coefficients are functions of both time and state

- C4 (Initial Value): X_{t_0} is \mathcal{F}_{t_0} -measurable⁷ with $\mathbb{E}(|X_{t_0}|^2) < \infty$ where \mathcal{F}_{t_0} is the σ -algebra generated by B_{t_0}

Below we state in a more formal way without proof⁸ the theorem that guarantees the existence and uniqueness of a solution to (2.10).

Theorem 2.4.2 (Existence and Uniqueness Theorem [26]). *Under the assumptions C1 – C4, the stochastic differential equation, (2.10) has a pathwise unique strong solution X_t on $[t_0, T]$ with*

$$\sup_{t_0 \leq t \leq T} \mathbb{E}(|X_t|^2) < \infty$$

Yet another important member of the family of stochastic processes is the Itô process.

2.4.2 Itô Process

Definition 2.4.3. *Let $(\Omega, \mathcal{F}, (\mathcal{F}_{t \geq 0}), \mathbb{P})$ be a filtered probability space satisfying the usual conditions⁹ and $B = B_t$ be a standard Brownian motion with respect to \mathcal{F}_t . A real-valued process $X = (X_t)_{0 \leq t \leq T}$ is an Itô process if it can be written as*

$$X_t = X_0 + \int_0^t K_s ds + \int_0^t H_s dB_s \quad (2.12)$$

where $\forall t \leq T$ and

1. X_0 is \mathcal{F}_0 -measurable
2. $\{K_t\}_{0 \leq t \leq T}$ and $\{H_t\}_{0 \leq t \leq T}$ are \mathcal{F}_t -adapted processes
3. $\int_0^T |K_s| ds < \infty$ \mathbb{P} a.s
4. $\int_0^T |H_s|^2 ds < \infty$ \mathbb{P} a.s

⁷See [25] for more information on sigma algebra and measurability

⁸See [26] for the proof.

⁹See [26]

and where $\int_0^t H_s dB_s$ is an Itô integral (which we described in section (2.3)) An Itô process is by convention symbolically expressed in terms of the (stochastic) differential equation¹⁰

$$\begin{aligned} dX_t &= K_t dt + H_t dB_t \\ X_0 &= \text{Initial condition ,} \end{aligned} \tag{2.13}$$

The term K_t is called the drift of the Itô process while H_t is called the diffusion coefficient of the process. Itô processes do not follow the conventional chains rule of ordinary differential equations hence we briefly state the Itô formula of stochastic processes.

Theorem 2.4.4. (One Dimensional Itô Formula) *Let X be an Itô process determined by $dX_t = K(X_t, t)dt + H(X_t, t)dB_t$ and $f \in C^{1,2}([0, \infty] \times \mathbb{R})$, continuous in t and the doubly differetiabile with respect to x . Then the process $f(t, X)$ follows the SDE*

$$\begin{aligned} df(t, X_t) &= \frac{\partial f}{\partial t}(t, X_t)dt + \frac{\partial f}{\partial x}(t, X_t)dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X_t)(dX_t)^2 \\ &= \frac{\partial f}{\partial t}(t, X_t)dt + \frac{\partial f}{\partial x}(t, X_t)[K(t, X_t)dt + H(t, X_t)]dB_t \\ &\quad + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X_t)[K(t, X_t)dt + H(t, X_t)dB_t]^2 \\ &= \left(\frac{\partial f}{\partial t}(t, X_t) + K(t, X_t) \frac{\partial f}{\partial x}(t, X_t) + \frac{1}{2} H^2(t, X_t) \frac{\partial^2 f}{\partial x^2}(t, X_t) \right) dt \\ &\quad + H(t, X_t) \frac{\partial f}{\partial x}(t, X_t)dB_t, \end{aligned} \tag{2.14}$$

and is also an Itô process.

¹⁰Note: This is a mere short hand representation of a stochastic integral(2.12) Stochastic differential equations do not exist except in terms of stochastic integrals.

Chapter 3

Numerical Methods for solutions of SDEs

3.1 Introduction

In this chapter, we present some numerical schemes resulting from the Itô Taylor expansion (3.2) of an Itô process. Numerical methods are time discrete approximations in which the continuous time differential equation is replaced by a discrete-time difference equation generating values $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N$ at discrete times t_1, t_2, \dots, t_N . The study of numerical methods is necessary as it is not possible in general to find an explicit solution of stochastic differential equations that occur in scientific and financial models. Another important reason for the continuing study of numerical methods is the need for these schemes to be considerably more accurate. We shall investigate their strong and weak orders of convergence and analyze their local and global errors. To start with, we briefly discuss the stochastic Taylor expansion from which the standard numerical schemes are derived.

3.2 Stochastic Taylor Expansions

The stochastic Taylor formula allows a function of an Itô process, $f(X_t)$ to be expanded about $f(X_0)$ in terms of multiple stochastic integrals weighted by coefficients which are evaluated at X_0 .

Let $f(X_t)$ be a smooth function of the Itô process X_t , for $t \in [0, T]$. The stochastic Taylor expansion of $f(X_t)$ is given by

$$f(X_t) = f(X_0) + c_1(X_0) \int_0^t ds + c_2(X_0) \int_0^t dB_s + c_3(X_0) \int_0^t \int_0^s dB_u dB_s + R, \quad (3.1)$$

with the coefficients defined as below

$$\begin{aligned} c_1(x) &= K(x)f'(x) + \frac{1}{2}H^2(x)f''(x) \\ c_2(x) &= H(x)f'(x) \\ c_3(x) &= H(x)\{H(x)f''(x) + H'(x)f'(x)\} \end{aligned}$$

The remainder term R in (3.1) consists of higher order multiple stochastic integrals involving the function f , the drift and the diffusion coefficient of the Itô process, X_t and their derivatives. (See [26] for a more comprehensive study of the stochastic Taylor expansion). Basic numerical schemes for the solution of SDEs are obtained by truncating the stochastic Taylor expansion such as (3.1) about successive discretization points. To obtain a numerical scheme with a higher order of strong or weak convergence one must include the appropriate terms from the corresponding stochastic Taylor expansion. It is not enough to only obtain a numerical method approximating the solution of SDEs, such a scheme should at least satisfy the minimum requirement of a good numerical algorithm; This requirement is that the sequence generated by the algorithm converges to the true solution of the SDE. In addition, such convergence should be reasonably fast. This gives rise to the notion of order of convergence of a numerical scheme.

3.2.1 Strong and Weak Convergence of Stochastic Taylor Approximation

Let \hat{X}_t be a strong pathwise approximation¹ of the Itô process X . The absolute error criterion for \hat{X} is given by

$$e = \mathbb{E}(|X_T - \hat{X}_T|), \quad (3.2)$$

This is simply the expectation of the absolute value of the difference between the Itô process and the approximation at finite terminal time T .

An approximation $\hat{X}_0, \hat{X}_{\Delta t}, \hat{X}_{2\Delta t}, \dots, \hat{X}_T$ of a stochastic process X is said to have strong order of convergence γ if, for small enough Δt , there exist a constant C , such that

$$\mathbb{E} \left[|\hat{X}_{i\Delta t} - X_{i\Delta t}| \right] \leq C(\Delta t)^\gamma$$

for every $i \geq 0$; On the other hand the approximation is said to have weak order of convergence γ if for small enough Δt , there exists a constant C such that

$$\mathbb{E} \left[|\hat{X}_{i\Delta t}| \right] - \mathbb{E} [|X_{i\Delta t}|] \leq C(\Delta t)^\gamma$$

for every $i \geq 0$. Having stated the stochastic Taylor expansion, we then rederive some of the numerical schemes for the solutions of SDEs. We proceed in the conventional way, that is, in order of increasing rate of convergence.

3.3 Euler-Maruyama Method

Consider the autonomous stochastic process X defined by

$$dX_t = K(X_t)dt + H(X_t)dB_t, X_0 \text{ a constant} \quad (3.3)$$

where $K, H \in C^2(\mathbb{R})$. X is clearly an Itô process² and the stochastic process is of the type, (3.3) is found in many real life applications for instance, in the modelling stock

¹a numerical scheme on a particular Brownian motion path. See [26]pp. 327-329

²See definition in (2.1) above

price dynamics. Letting $f \in C^2(\mathbb{R})$ and applying Itô formula (in integral form) we have

$$\begin{aligned} f(X_{t+\Delta t}) &= f(X_t) + \int_t^{t+\Delta t} \left(K(X_s) \frac{df}{dx}(X_s) + \frac{1}{2} H^2(X_s) \frac{d^2 f}{dx^2} \right) ds \\ &\quad + \int_t^{t+\Delta t} H(X_s) \frac{df}{dx}(X_s) dB_s \\ &= f(X_t) + \int_t^{t+\Delta t} L^0 f(X_s) ds + \int_t^{t+\Delta t} L^1 f(X_s) dB_s \end{aligned} \quad (3.4)$$

where L^0 and L^1 are operators defined by

$$L^0 : = K \frac{d}{dx} + \frac{1}{2} H^2 \frac{d^2}{dx^2}, \quad (3.5)$$

$$L^1 : = H \frac{d}{dx} \quad (3.6)$$

Now, rewriting (3.3) in integral form and applying (3.4) to K and H we obtain

$$\begin{aligned} X_{t+\Delta t} &= X_t + \int_t^{t+\Delta t} \left(a(X_t) + \int_t^s L^0 K(X_r) dr + \int_t^s L^1 K(X_r) dB_r \right) ds \\ &\quad + \int_t^{t+\Delta t} \left(H(X_t) + \int_t^s L^0 H(X_r) dr + \int_t^s L^1 H(X_r) dB_r \right) dB_s \\ &= X_t + K(X_t) \int_t^{t+\Delta t} ds + H(X_t) \int_t^{t+\Delta t} dB_s + R(t, t + \Delta t) \\ &= X_t + K(X_t) \Delta t + H(X_t) \Delta B_t + R(t, t + \Delta t) \end{aligned} \quad (3.7)$$

$$\approx X_t + K(X_t) \Delta t + H(X_t) \Delta B_t \quad (3.8)$$

where we write

$$\begin{aligned} R(t, t + \Delta t) &:= \int_t^{t+\Delta t} \left(\int_t^s L^0 K(X_r) dr + \int_t^s L^1 K(X_r) dB_r \right) ds \\ &\quad + \int_t^{t+\Delta t} \left(\int_t^s L^0 H(X_r) dr + \int_t^s L^1 H(X_r) dB_r \right) dB_s \\ &= \int_t^{t+\Delta t} \int_t^s L^0 K(X_r) dr ds + \int_t^{t+\Delta t} \int_t^s L^1 K(X_r) dB_r ds \\ &\quad + \int_t^{t+\Delta t} \int_t^s L^0 H(X_r) dr dB_s + \int_t^{t+\Delta t} \int_t^s L^1 H(X_r) dB_r dB_s \end{aligned} \quad (3.9)$$

and $\Delta B = B_{t+\Delta t} - B_t$ is an increment of a Brownian motion which follows a normal distribution with zero mean and variance Δt , that is $\mathcal{N}(0, \Delta t)$. The approximation in (3.8) is the discretization in the Euler scheme. More generally in the

multi-dimensional case with a scalar Brownian motion i.e, $m = 1$ and $d = 1, 2, \dots$ the r th component of the Euler scheme is given by:

$$X_{t+\Delta t}^r = X_t^r + K^r(X_t)\Delta t + H^r(X_t)\Delta B_t \quad (3.10)$$

for $r = 1, 2, \dots$ where $K = (K^1, \dots, K^d)$ and $H = (H^1, \dots, H^d)$.

For the general multi-dimensional case with $d, m = 1, 2, \dots$ the r th component of the Euler scheme has the form:

$$X_{t+\Delta t}^r = X_t^r + K^r(X_t)\Delta t + \sum_{j=1}^m H^{r,j}(X_t)\Delta B_t^j. \quad (3.11)$$

Here $\Delta B^j = B_{t+\Delta t}^j - B_t^j$ is an $\mathcal{N}(0, \Delta t)$ distributed increment of the j th component of the m -dimensional Brownian motion process B on $[t, t + \Delta t]$ and ΔB_t^{j1} and ΔB_t^{j2} are independent for $j1$ and $j2$. Similarly $H = [H^{r,j}]$ is a $d \times m$ - matrix with at least one of its components not equal to zero.

3.3.1 Numerical Example

We now apply the Euler scheme to the popular³ linear SDE

$$dX_t = \mu X_t dt + \sigma X_t dB_t, \quad (3.12)$$

where μ and σ are constants. In analogy with our general SDE, (2.10), $K = \mu X_t$ and $H = \sigma X_t$.

The exact solution (see for example [26]) to (3.12) is given by

$$X_t = X_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B_t\right). \quad (3.13)$$

The following Figure 3.1 plots the Euler approximation of (3.12) over the interval, $[0, 1]$ for different step sizes and compares them with the exact solutions. In Table 3.1, we present the average terminal errors using the absolute error criterion (3.2) at the terminal point, which shows increased convergence as the step size, h tends to zero.

	$h = 2^{-2}$	$h = 2^{-6}$	$h = 2^{-10}$
Error	0.8513	0.4445	0.2443

Table 3.1: Average terminal errors of the Euler method

In what follows we use (3.12) to verify some of the numerical properties of the Euler method and start by looking at its strong and weak orders of convergence.

3.3.2 Strong and Weak Order of Convergence of the Euler Scheme

We now use the result of section, 3.3.1 to illustrate both the strong and weak order of convergence of the Euler method. In the above example, Table 3.1, the Euler approximation for 5000 Brownian motion paths with $\mu = 2$, $\sigma = 1$ and $X_0 = 1$, matches the exact solution of the SDE very closely as h decreases. In other words as the step-size decreases, convergence seems to occur. We shall use the definition of convergence in section 3.2.1 to determine the strong and weak order of convergence of the Euler approximation. If the conditions of Theorem 2.4.2 are satisfied, it can be shown that (see [26]) the Euler approximation has *strong order of convergence of* $\gamma = \frac{1}{2}$. Note that this is different from the deterministic case, $H = 0$, where $\gamma = 1$. To numerically test for this order of convergence, we use the absolute error criterion 3.2.1 at the terminal point, $t = T$. We denote the terminal error in the strong sense by

$$e_h^{str} = \mathbb{E}|\hat{X}_T - X_T|, \quad (3.14)$$

where $L\Delta t = T$. If conditions (C1-C4) in section 2.4.1 hold with $\gamma = \frac{1}{2}$ at any fixed point in the interval $[0, T]$, then it certainly must hold at the end point, hence we

³This SDE is also known as geometric Brownian motion; it is used in the famous Black-Scholes asset pricing model in the mathematics of finance (see [23])

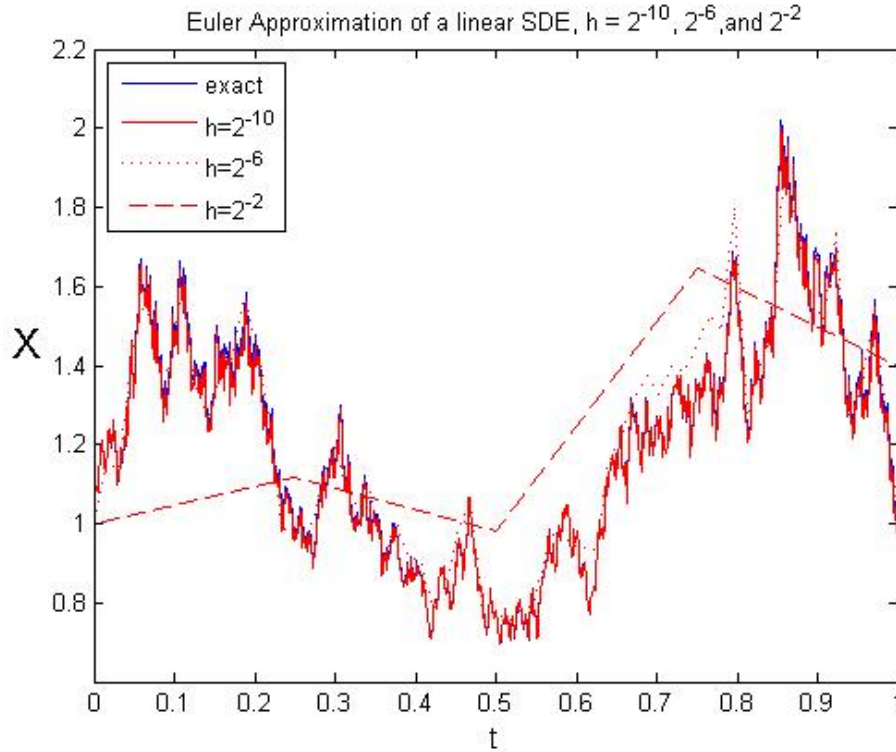


Figure 3.1: Euler Approximation of an SDE

,

have for small enough Δt

$$e_h^{str} \leq C\Delta t^{\frac{1}{2}}. \quad (3.15)$$

If (3.15) holds with actual equality, then by taking logs we have

$$\log e_h^{str} = \log C + \frac{1}{2} \log \Delta t. \quad (3.16)$$

The upper graph on Figure 3.2 plots e_h^{str} against Δt on a log-log scale together with a dashed line of slope $s = \frac{1}{2}$. We see that the slopes of both curves apparently match well and this verifies (3.15). The strong order of convergence measures the rate which the *mean of the error* decays as $\Delta t \rightarrow 0$. On the other hand, the weak order of convergence measures the rate at which *error of the mean*⁴ as $\Delta t \rightarrow 0$. For appropriate K and H it can be shown (see for instance [26]) that the Euler method

⁴Note the difference with the strong of order convergence

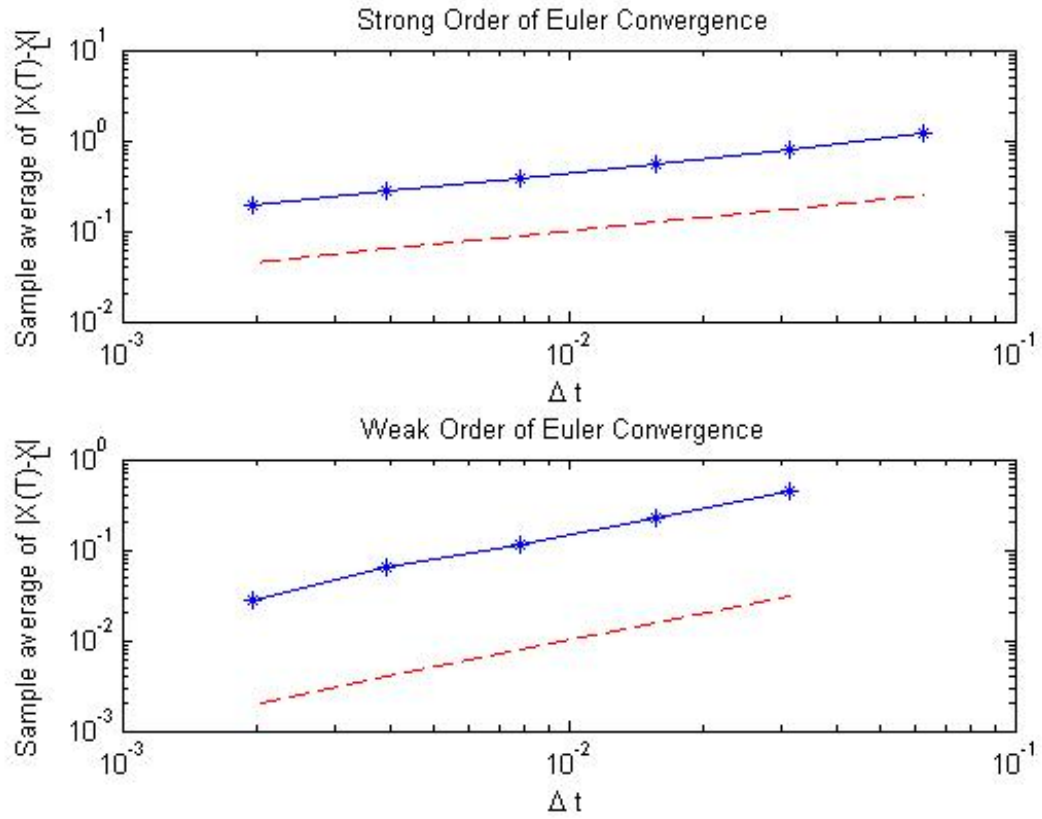


Figure 3.2: Strong Order of Euler convergence

has a weak order of convergence of $\gamma = 1$. Similar to the strong order of convergence, we denote the error of the mean as

$$e_h^{Wk} := |\mathbb{E}X(T) - \mathbb{E}X_L|, \quad (3.17)$$

where $L\Delta t = T$.

Using the SDE(3.12) above with $X(0) = 1$ we have

$$\begin{aligned}
\mathbb{E}X(T) &= \mathbb{E} \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma[B(T) - B(0)]\right) \\
&= \int_{-\infty}^{\infty} \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma x\right) \frac{1}{\sqrt{2\pi T}} \exp\left(\frac{-x^2}{2T}\right) dx \\
&= \exp(\mu T) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi T}} \exp\left(\frac{-x^2}{2T} + \sigma x - \frac{1}{2}\sigma^2 T\right) dx \\
&= \exp(\mu T) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi T}} \exp\left(\frac{-(x - \sigma T)^2}{2T}\right) dx \\
&= \exp(\mu T).
\end{aligned} \tag{3.18}$$

In the same vein, with $\gamma = 1$ we have for small enough h , with $C > 0$

$$e_h^{Wk} \leq C\Delta t, \tag{3.19}$$

Here we examine the weak convergence of the Euler method. We solve (3.12) with $\mu = 2, \sigma = 0.1$ and $X_0 = 1$. Again we sample 5000 discretized Brownian paths and use five step sizes $h = 2^{-s}$ for $s = [-10 : 1 : -5]$ in the Euler method. We use the sample average approximation to $\mathbb{E}X_L$ and (3.18) for $\mathbb{E}X(T)$ to compute e_h^{Wk} . The lower graph in Figure 3.2 shows how the weak error varies with h on a log-log scale, and a reference line of slope 1 shows that (3.19) approximately holds with actual equality. In addition we do a least square power law fit that gives $q = 0.9693$ and residue $r = 0.0471$, verifying the observation in Figure3.2.

3.4 Milstein method

If we use (3.4) to expand the last term in the expression for $\mathcal{R}(t, t + \Delta t)$, (assuming $K, H \in C^4(\mathbb{R})$), we get

$$\begin{aligned}
\int_t^{t+\Delta t} \int_t^s L^1 H(X_r) dB_r dB_s &= \int_t^{t+\Delta t} \int_t^s L^1 H(X_t) dB_r dB_s \\
&+ \int_t^{t+\Delta t} \int_t^s \int_t^r L^0 L^1 H(X_q) dq dB_r dB_s \\
&+ \int_t^{t+\Delta t} \int_t^s \int_t^r L^1 L^1 H(X_q) dB_q dB_r dB_s.
\end{aligned}$$

Evaluating the first term above using Itô formula, we have

$$\begin{aligned}
& \int_t^{t+\Delta t} \int_t^s L^1 H(X_t) dB_r dB_s \\
&= L^1 H(X_t) \int_t^{t+\Delta t} \int_t^s dB_r dB_s \\
&= H(X_t) \frac{dH}{dx}(X_t) \int_t^{t+\Delta t} (B_s - B_t) dB_s \tag{3.20}
\end{aligned}$$

$$\begin{aligned}
&= H(X_t) \frac{dH}{dx}(X_t) \left(\frac{1}{2} (B_{t+\Delta t}^2 - B_t^2 - \Delta t) - B_t \Delta B_t \right) \tag{3.21} \\
&, = \frac{1}{2} H(X_t) \frac{dH}{dx}(X_t) ((B_{t+\Delta t} + B_t) \Delta B_t - 2B_t \Delta B_t - \Delta t) \\
&= \frac{1}{2} H(X_t) \frac{dH}{dx}(X_t) ((\Delta B_t)^2 - \Delta t).
\end{aligned}$$

Notice the use of Itô formula in (3.21). Substituting in (3.8), we get

$$\begin{aligned}
& X_{t+\Delta t} \\
&= X_t + K(X_t) \Delta t + H(X_t) \Delta B_t \\
&\quad + \frac{1}{2} H(X_t) \frac{dH}{dx}(X_t) ((\Delta B_t)^2 - \Delta t) + \bar{\mathcal{R}}(t, t + \Delta t) \\
&\approx X_t + K(X_t) \Delta t + H(X_t) \Delta B_t + \frac{1}{2} H(X_t) \frac{dH}{dx}(X_t) ((\Delta B_t)^2 - \Delta t) \tag{3.22}
\end{aligned}$$

where

$$\begin{aligned}
\bar{\mathcal{R}}(t, t + \Delta t) &:= \int_t^{t+\Delta t} \int_t^s L^0 K(X_r) dr ds + \int_t^{t+\Delta t} \int_t^s L^1 K(X_r) dB_r ds \\
&\quad + \int_t^{t+\Delta t} \int_t^s L^0 H(X_r) dr dB_s \\
&\quad + \int_t^{t+\Delta t} \int_t^s \int_t^r L^0 L^1 H(X_q) dq dB_r dB_s \\
&\quad + \int_t^{t+\Delta t} \int_t^s \int_t^r L^1 L^1 H(X_q) dB_q dB_r dB_s.
\end{aligned}$$

Thus, the Milstein approximation scheme is a sequence of random variables

$\hat{X}_0, \hat{X}_{\Delta t}, \hat{X}_{2\Delta t}, \dots, \hat{X}_T$ generated from (3.22) by setting

$$\begin{aligned}
\hat{X}_{i\Delta t} &:= \hat{X}_{(i-1)\Delta t} + K \left(\hat{X}_{(i-1)\Delta t} \right) \Delta t + H \left(\hat{X}_{(i-1)\Delta t} \right) \Delta B_t \\
&\quad + \frac{1}{2} H \left(\hat{X}_{(i-1)\Delta t} \right) \frac{dH}{dx} \left(\hat{X}_{(i-1)\Delta t} \right) ((\Delta B_t)^2 - \Delta t). \tag{3.23}
\end{aligned}$$

From (3.23) we see that the one-dimensional Milstein scheme is simply an extension of the one-dimensional Euler method obtained by an addition of the extra term

$\frac{1}{2}H\left(\hat{X}_{(i-1)\Delta t}\right)\frac{dH}{dx}\left(\hat{X}_{(i-1)\Delta t}\right)\left((\Delta B_t)^2 - \Delta t\right)$ from the stochastic Taylor expansion⁵.

In the multi-dimensional case with $m = 1$ and $d = 1, 2, \dots$ the r th component of the Milstein scheme has the form:

$$\begin{aligned} \hat{X}_{i\Delta t}^r &:= \hat{X}_{(i-1)\Delta t}^r + K^r\left(\hat{X}_{(i-1)\Delta t}\right)\Delta t + H^r\left(\hat{X}_{(i-1)\Delta t}\right)\Delta B_t \\ &\quad + \frac{1}{2}\left(\sum_{l=1}^d H^l\left(\hat{X}_{(i-1)\Delta t}\right)\frac{\partial H^r}{\partial x^l}\right)\left(\hat{X}_{(i-1)\Delta t}\right)\{(\Delta B_t)^2 - \Delta t\}. \end{aligned} \quad (3.24)$$

The Milstein scheme is a bit complex in the general multi-dimensional case, with $m, d = 1, 2, \dots$. This is because of the presence of the multiple Itô (Stratonovich) integrals in the scheme, which are usually difficult to generate. In the general multi-dimensional case, the k th component of the Milstein scheme is given by

$$\begin{aligned} \hat{X}_{i\Delta t}^r &:= \hat{X}_{(i-1)\Delta t}^r + K^r\left(\hat{X}_{(i-1)\Delta t}\right)\Delta t + \sum_{j=1}^m H^{r,j}\left(\hat{X}_{(i-1)\Delta t}\right)\Delta B_t^j \\ &\quad + \sum_{j^1, j^2=1}^m L^{j^1}H^{r, j^2}\left(\hat{X}_{(i-1)\Delta t}\right)I_{(j^1, j^2)}\left(\hat{X}_{(i-1)\Delta t}\right), \end{aligned} \quad (3.25)$$

where

$$\begin{aligned} L^{j^1} &= \sum_{r=1}^d H^{r, j^1}\frac{\partial}{\partial x^r}, \text{ and} \\ I_{(j^1, j^2)} &= \int_{t_i}^{t_{i+1}} \int_{t_i}^{s_2} dB_{s_1}^{j^1} dB_{s_2}^{j^2} \end{aligned} \quad (3.26)$$

The complexity in the evaluation of the multidimensional Itô integral can be circumvented depending on the nature of the SDE modelling the situation. Platen et al in [26] proposed an approximation to the iterated integral in (3.26) in case it cannot be avoided. (See [26, 27] for more details). If the SDE has time additive noise, i.e $H(t, X) = H(t)$, so that the diffusion coefficient depends at most on time and not on the space variable x , the Milstein scheme reduces to the Euler scheme.

⁵The same scheme can be obtained from the truncated Stratonovich Taylor expansion (see [2, 3, 27])

In the case of diagonal noise, i.e if $d = m$, each component of X^r of the Itô-process is disturbed only by the corresponding component B^r of the Brownian motion B and the diagonal diffusion coefficient $H^{r,r}$ depends only on X^r , i.e for each $(t, x) \in \mathbb{R}^+ \times \mathbb{R}^d$ and $j, r = 1, \dots, m$ with $r \neq j$, $H^{r,j} = 0$ and $\frac{\partial H^{j,j}}{\partial x^r}(t, x) = 0$, the Milstein scheme thus becomes

$$\begin{aligned} \hat{X}_{i\Delta t}^r &:= \hat{X}_{(i-1)\Delta t}^r + K^r \left(\hat{X}_{(i-1)\Delta t} \right) \Delta t + H^{r,r} \left(\hat{X}_{(i-1)\Delta t} \right) \Delta B_t^r \\ &\quad + \frac{1}{2} H^{r,r} \left(\hat{X}_{(i-1)\Delta t} \right) \frac{\partial H^{r,r}}{\partial x^r} \left(\hat{X}_{(i-1)\Delta t} \right) \{(\Delta B_t)^2 - \Delta t\}. \end{aligned} \quad (3.27)$$

There is a more general case, that of commutative noise, in which the diffusion matrix satisfies the commutativity condition, $\underline{L}^{j1} H^{r,j2} = \underline{L}^{j2} H^{r,j1}$ where $\underline{L}^{ji} = L^{ji}$ as in (3.26). The Milstein scheme here becomes

$$\begin{aligned} \hat{X}_{i\Delta t}^r &:= \hat{X}_{(i-1)\Delta t}^r + K^r \left(\hat{X}_{(i-1)\Delta t} \right) \Delta t + \sum_{j=1}^m H^{r,j} \left(\hat{X}_{(i-1)\Delta t} \right) \Delta B_t^j \\ &\quad + \frac{1}{2} \sum_{j1, j2=1}^m \underline{L}^{j1} H^{r,j2} \left(\hat{X}_{(i-1)\Delta t} \right) \Delta B^{j1} \Delta B^{j2}, \end{aligned} \quad (3.28)$$

and again the iterated integrals are not needed.

3.4.1 Strong Convergence of the Milstein method

It can be shown that under appropriate conditions on the drift and the diffusion coefficients⁶, the Milstein scheme has the order of strong convergence $\gamma = 1$ (See [26] for the proof). To numerically test for this order of strong convergence, we solve the one-dimensional SDE,

$$dX_t = aX_t(b - X_t)dt + \beta X_t dB, X_0 = \text{initial condition} \quad (3.29)$$

over the time interval $[0, 1]$ with $X_0 = 0.5$, $a = 3$, $b = 1.5$ and $\beta = 0.25$. We generate 5000 simulations of terminal values of the reference solutions⁷ $X(T)$ as a proxy to the

⁶That is the conditions C1 – C4 in section 2.4.1

⁷where we have used the Milstein solution with step size $h = 2^{-11}$ as a proxy to the exact solution

exact solution and of the values \hat{X}_T of the corresponding Milstein approximation. In Table 3.2, we present the average terminal values and errors of the Milstein scheme using the absolute error criterion. The table shows increasing convergence of the milstein scheme as the step size decreases. Figure 3.3 plots the exact solution

	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$	$h = 2^{-7}$
Error	0.0151	0.0072	0.0035	0.0016

Table 3.2: Average terminal values and errors of the Milstein method

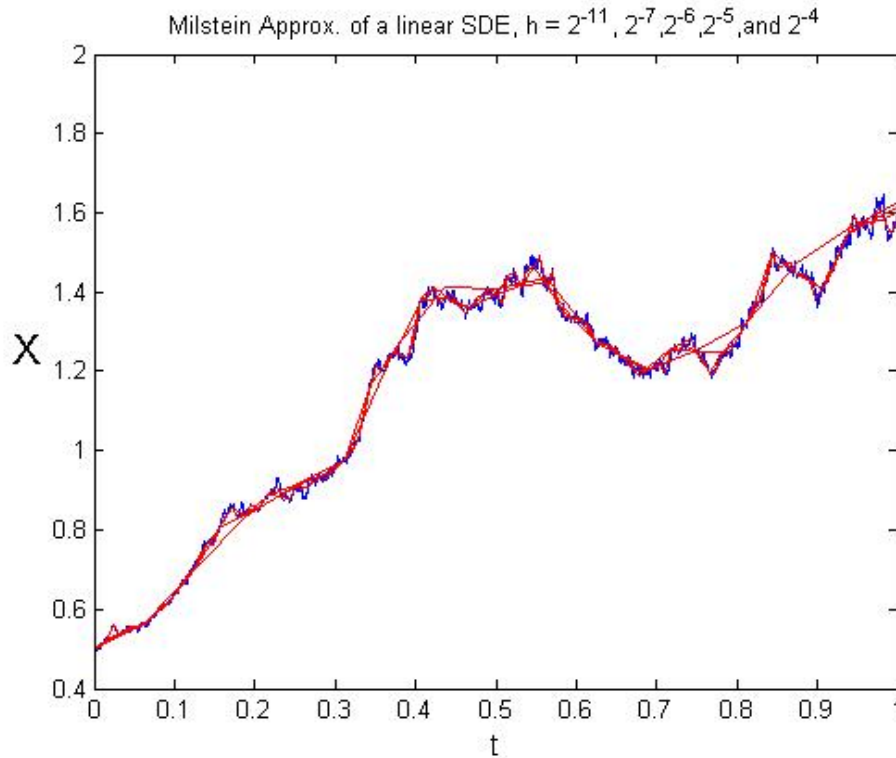


Figure 3.3: Milstein Approximation of an SDE

and the Milstein approximation for different step sizes, $h = 2^k$ for $k = [4, 5, \dots, 7]$ respectively.

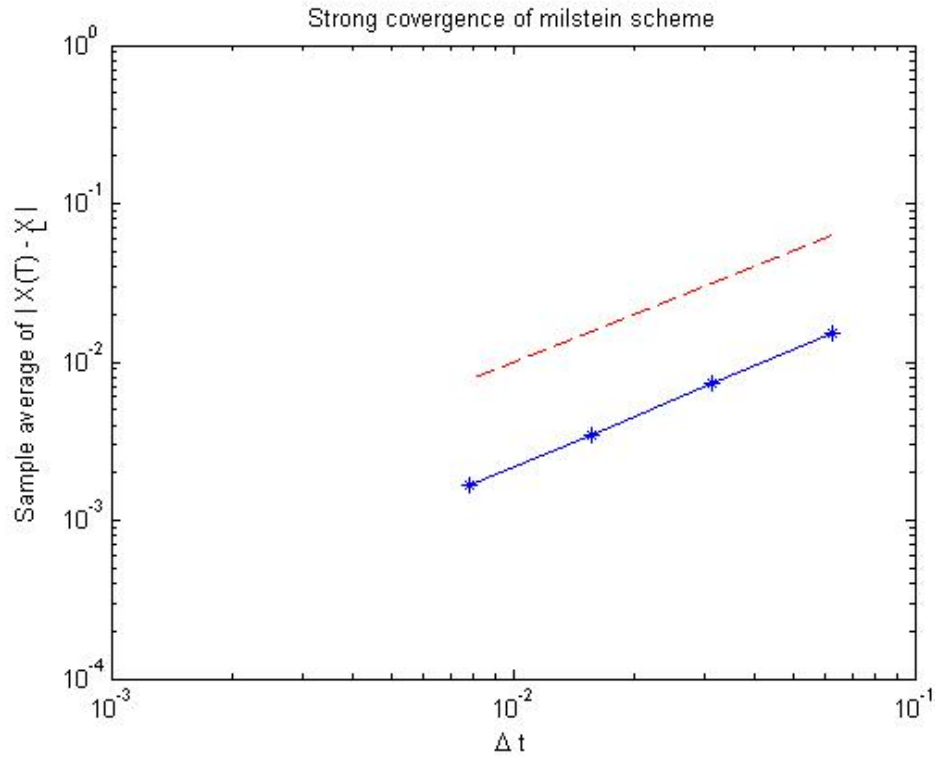


Figure 3.4: Strong convergence of the Milstein scheme

The log-log plot of the absolute error (solid line) along with a reference line (dotted) as shown in Figure 3.4 confirms the assertion of strong order of convergence, $\gamma = 1$ of the Milstein scheme. In addition, the least-square power law fit gives $q = 0.9705$ and $residue = 0.0366$ which further verifies the order of convergence of the Milstein scheme.

To illustrate the complexity, notwithstanding the benefits of the higher order schemes and to further emphasize the need to optimize the Euler and the Milstein schemes, we present below the order 1.5 stochastic scheme.

3.5 Order 1.5 stochastic schemes

The order 1.5 Itô-Taylor scheme is obtained by including additional multiple integrals from the Itô-Taylor expansion in the Milstein scheme. These multiple stochastic integrals contain additional information about the sample path of the Brownian motion.

In the autonomous 1-dimensional case, $d = m = 1$, the order 1.5 strong Itô scheme according to Platen and Wagner [35] is given by

$$\begin{aligned}
\hat{X}_{i\Delta t} &= \hat{X}_{(i-1)\Delta t} + K(\hat{X}_{(i-1)\Delta t})\Delta t + H(\hat{X}_{(i-1)\Delta t})\Delta B_t \\
&+ \frac{1}{2}H(\hat{X}_{(i-1)\Delta t})\frac{dH}{dx}(\hat{X}_{(i-1)\Delta t})[(\Delta B_t)^2 - \Delta t] \\
&+ \frac{dK}{dx}(\hat{X}_{(i-1)\Delta t})H(\hat{X}_{(i-1)\Delta t})\Delta Z_t + \frac{1}{2}(K(\hat{X}_{(i-1)\Delta t})\frac{dK}{dx}(\hat{X}_{(i-1)\Delta t}) \\
&+ \frac{1}{2}[H^2(\hat{X}_{(i-1)\Delta t})\frac{d^2K}{dx^2}(\hat{X}_{(i-1)\Delta t})]\Delta t^2 + [K(\hat{X}_{(i-1)\Delta t})\frac{dK}{dx}(\hat{X}_{(i-1)\Delta t}) \\
&+ \frac{1}{2}H^2(\hat{X}_{(i-1)\Delta t})\frac{d^2H}{dx}(\hat{X}_{(i-1)\Delta t})](\Delta B_t\Delta t - \Delta Z_t) \\
&+ \frac{1}{2}H(\hat{X}_{(i-1)\Delta t})[H(\hat{X}_{(i-1)\Delta t})\frac{d^2H}{dx}(\hat{X}_{(i-1)\Delta t}) + (\frac{dH}{dx}(\hat{X}_{(i-1)\Delta t}))^2]\dots \\
&[\frac{1}{3}(\Delta B_t)^2 - \Delta t]\Delta B_t
\end{aligned} \tag{3.30}$$

where

$$\Delta Z_t = \int_{(i-1)\Delta t}^{i\Delta t} \int_{(i-1)\Delta t}^{s_2} dB_{s_1} dB_{s_2} \tag{3.31}$$

It can be shown (see [26]) that ΔZ is normally distributed with mean $\mathbb{E}[\Delta Z] = 0$, variance $\mathbb{E}[(\Delta Z)^2] = \frac{1}{2}\Delta t^3$ and covariance $\mathbb{E}[\Delta Z\Delta B] = \frac{1}{2}\Delta t^2$. In particular, the pair of correlated normally distributed random variables can be determined from two independent $\mathcal{N}(0, 1)$ distributed random variables n_1 and n_2 by means of the following transformation;

$$\Delta B = n_1\sqrt{\Delta t}, \Delta Z = \frac{1}{2}\Delta t^{\frac{3}{2}}(n_1 + \frac{1}{\sqrt{3}}n_2) \tag{3.32}$$

Notice that if we set $H(\hat{X}) = 0$ then (3.30) reduces to the deterministic 2nd order truncated Taylor method.

With $d = 1, 2, \dots$ and $m = 1$ i.e scalar noise, the k th component of the 1.5 order strong Taylor scheme is given by

$$\begin{aligned}
\hat{X}_{i\Delta t}^r &= \hat{X}_{(i-1)\Delta t}^r + K^r(\hat{X}_{(i-1)\Delta t})\Delta t + H^r(\hat{X}_{(i-1)\Delta t})\Delta B_t \\
&+ \frac{1}{2}H^r(\hat{X}_{(i-1)\Delta t})\frac{dH^r}{dx}(\hat{X}_{(i-1)\Delta t})[(\Delta B_t)^2 - \Delta t] \\
&+ \frac{dK^r}{dx}(\hat{X}_{(i-1)\Delta t})H^r(\hat{X}_{(i-1)\Delta t})\Delta Z_t + \frac{1}{2}(K^r(\hat{X}_{(i-1)\Delta t})\frac{dK^r}{dx}(\hat{X}_{(i-1)\Delta t}) \\
&+ \frac{1}{2}[(H^r)^2(\hat{X}_{(i-1)\Delta t})\frac{d^2K^r}{dx^2}(\hat{X}_{(i-1)\Delta t})]\Delta t^2 + [K^r(\hat{X}_{(i-1)\Delta t})\frac{dK^r}{dx}(\hat{X}_{(i-1)\Delta t}) \\
&+ \frac{1}{2}(H^r)^2(\hat{X}_{(i-1)\Delta t})\frac{d^2H^r}{dx}(\hat{X}_{(i-1)\Delta t})](\Delta B_t\Delta t - \Delta Z_t) \\
&+ \frac{1}{2}H^r(\hat{X}_{(i-1)\Delta t})[H^r(\hat{X}_{(i-1)\Delta t})\frac{d^2H^r}{dx}(\hat{X}_{(i-1)\Delta t}) \\
&+ (\frac{dH^r}{dx}(\hat{X}_{(i-1)\Delta t}))^2][\frac{1}{3}(\Delta B_t)^2 - \Delta t]\Delta B_t
\end{aligned} \tag{3.33}$$

For consistency with the other schemes and using the notation earlier in this chapter, in the multi-dimensional case, i.e $d, m = 1, 2, 3, \dots$, the k th component of the order 1.5 strong Taylor scheme takes the form

$$\begin{aligned}
\hat{X}_{i\Delta t}^r &= \hat{X}_{(i-1)\Delta t}^r + K^r(\hat{X}_{(i-1)\Delta t})\Delta t + \frac{1}{2}L^0K^r(\hat{X}_{(i-1)\Delta t})\Delta t^2 \\
&+ \sum_{j=1}^m (H^{r,j}(\hat{X}_{(i-1)\Delta t})\Delta B_t^j + L^0H^{r,j}(\hat{X}_{(i-1)\Delta t})I_{(0,j)} + L^jK^r(\hat{X}_{(i-1)\Delta t})I_{(j,0)}) \\
&+ \sum_{j_1, j_2=1}^m L^{j_1j_2}H^{r,j_1j_2}(\hat{X}_{(i-1)\Delta t})I_{(j_1, j_2)} + \sum_{j_1, j_2, j_3=1}^m L^{j_1j_2j_3}H^{r,j_1j_2j_3}(\hat{X}_{(i-1)\Delta t})I_{(j_1, j_2, j_3)}
\end{aligned} \tag{3.34}$$

where we have used the following general notations due to Kloeden and Platen

(see[26]).

$$\begin{aligned}
L^0 &= \frac{\partial}{\partial t} + \sum_{r=1}^d K^r \frac{\partial}{\partial x^r} + \frac{1}{2} \sum_{r,l=1}^d \sum_{j=1}^m H^{r,j} H^{l,j} \frac{\partial^2}{\partial x^r \partial x^l} \\
L^j &= \sum_{r=1}^d H^{r,j} \frac{\partial}{\partial x^r} \\
I_{(j_1, \dots, j_l)} &= \int_{i\Delta t}^{(i+1)\Delta t} \dots \int_{i\Delta t}^{s_2} dB_{s_1}^{j_1} \dots dB_{s_l}^{j_l}
\end{aligned}$$

The above multidimensional scheme is difficult to implement due to the presence of multiple Itô integrals. Kloeden et al introduced a method for approximating the multiple integrals by representing them in terms of multiple Stratonovich integrals. (See[26],pp. 354). However, in many practical situation, the order 1.5 Taylor scheme reduces to a form in which the multiple Itô integrals with respect to the Brownian motion do not appear. One such case is the additive noise, where H is a constant or depends only on time t ; here the 1.5 strong Taylor scheme reduces to the form

$$\begin{aligned}
\hat{X}_{i\Delta t}^r &= \hat{X}_{(i-1)\Delta t}^r + K^r(\hat{X})\Delta t + \sum_{j=1}^m H^{r,j}(\hat{X})\Delta B^j + \frac{1}{2}L^0 K^r(\hat{X})\Delta t^2 \quad (3.35) \\
&+ \sum_{j=1}^m [L^j K^r(\hat{X})\Delta Z^j + \frac{\partial}{\partial t} H^{r,j}(\hat{X})\{\Delta B^j \Delta t - \Delta Z^j\}]
\end{aligned}$$

for $j = 1, \dots, m$.

Another special case is that of *diagonal noise* for which we have

$$\hat{X}_{i\Delta t}^r = \hat{X}_{(i-1)\Delta t}^r + K^r(\hat{X})\Delta t + H^{r,r}\Delta B^r + \frac{1}{2}L^0 K^r(\hat{X})\Delta t^2 \quad (3.36)$$

$$\begin{aligned}
&+ \frac{1}{2}L^r H^{r,r}(\hat{X})\{(\Delta B^r)^2 - \Delta t\} + L^0 H^{r,r}\{\Delta B^r \Delta t - \Delta Z^r\} \quad (3.37) \\
&+ L^r K^r \Delta Z^r + \frac{1}{2}L^r L^r H^{r,r}\left\{\frac{1}{3}(\Delta B^r)^2 - \Delta t\right\}\Delta B^r.
\end{aligned}$$

A more general case is the *commutative noise of the second kind* in which the diffusion matrix satisfies the *second commutativity condition*

$$L^{j_1} L^{j_2} H^{r,j_3}(\hat{X}) = L^{j_2} L^{j_1} H^{r,j_3}(\hat{X}) \quad (3.38)$$

for $r = 1, \dots, d$ and $j_1, j_2, j_3 = 1, \dots, m$ for $x \in \mathcal{R}^d$.

The order 1.5 Taylor scheme for commutative noise of the second kind is given by

$$\begin{aligned}
\hat{X}_{i\Delta t}^r &= \hat{X}_{(i-1)\Delta t}^r + K^r \Delta t + \sum_{j=1}^m H^{r,j}(\hat{X}) \Delta B^j & (3.39) \\
&+ \frac{1}{2} L^0 K^r \Delta t + \frac{1}{2} \sum_{j=1}^m L^j H^{r,j} \{(\Delta B^j)^2 - \Delta t\} \\
&+ \sum_{j_1=1}^m \sum_{j_2=1}^{j_1-1} L^{j_1} H^{r,j_2} \Delta B^{j_1} \Delta B^{j_2} \\
&+ \sum_{j=1}^m (L^0 H^{r,j} \{\Delta B^j \Delta t - \Delta Z^j\}) + L^j K^r \Delta Z^j \\
&+ \frac{1}{2} \sum_{j_1, j_2=1, j_1 \neq j_2}^m L^{j_1} L^{j_2} H^{r,j_3} \Delta B^{j_1} \{(\Delta B^{j_2})^2 - \Delta t\} \\
&+ \sum_{j_1=1}^m \sum_{j_2=1}^{j_1-1} \sum_{j_3=1}^{j_2-1} L^{j_1} L^{j_2} H^{r,j_3} \Delta B^{j_1} \Delta B^{j_2} \Delta B^{j_3} \\
&+ \frac{1}{2} \sum_{j=1}^m L^j L^j H^{r,j} \left\{ \frac{1}{2} (\Delta B^j)^2 - \Delta t \right\} \Delta B^j.
\end{aligned}$$

We notice that it is not necessary to generate the ΔZ^j term when the diffusion coefficients satisfy

$$L^0 H^{r,j} = L^j a^r \quad (3.40)$$

for all $j = 1 \dots m$ with $r = 1, \dots, d$.

The coefficients of the 1.5 order Ito-Taylor scheme need to satisfy the regularity condition for it to actually attain order 1.5 strong convergence(See [26]).

So far in this chapter, we have been looking at the standard⁸ numerical methods. Order 1.5 and other higher order schemes do give better approximations to the solutions of SDEs, however, because of the difficulty in the implementation of these higher order schemes, there is the need to improve the performance of the lower order schemes by other means such as adaptation.

⁸schemes obtained directly from Itô-Taylor expansion of SDEs

Chapter 4

Adaptive Numerical Schemes for SDEs

In this chapter, we take a tour of the usual approaches in the literature to adaptive schemes for stochastic differential equations. As previously mentioned, the classical approach to improving the performance of standard numerical schemes is through step-size control. We shall start with a brief introduction of adaptive schemes to ordinary differential equations as well as adaptive Runke-Kutta schemes. This will be followed by a review of some of the most recent step-size control adaptive schemes for SDEs in the literature. The step-size control is not as simple for SDEs as it is for ODEs; this is because in the former, we are not allowed to look into the future i.e stochastic schemes usually need to be non-anticipating. Specifically, we shall present adaptive schemes due to Lamba et al, Hofmann, Rapoo, and Mauthner respectively. This list is not exhaustive, However, more such schemes can be found in the following references, [4, 10, 39, 44].

4.1 Adaptive schemes for ODEs and R-K schemes

Several approaches have been developed in the literature to improve the efficiency of a standard numerical method approximating the solution to ordinary differential equations. The most popular amongst these approaches are control theory, signal processing and adaptivity (see [15]). These approaches are based on the assumption that the step size should follow some prescribed function of the solution. Control theory seeks to *keep the estimated error equal to the tolerance*. To achieve this, one needs to measure the difference between the error and the target tolerance, then use it in a feedback control system to continually correct the step size. A full control theoretical analysis for explicit as well as implicit Runge-Kutta methods was first introduced in [16, 17]. See [14] for a review of the use of control theory for ODEs.

Signal processing is closely related to control theory, but shifts emphasis towards obtaining a smooth step size sequence. The aim is to create a regular sequence of step sizes from the errors while keeping the error sufficiently close to the tolerance (see [15]). Regularity here means that the sequence has the appearance of a sample drawn from a smooth signal and can be analysed in terms of spectral content. It has been shown using well-known codes that a regular step size sequence has a significant impact on the computational stability of a code, see [18, 19]. While both control theory and signal processing can be termed adaptive, the step size sequence can be adapted to a prescribed function of the solution. See [42].

For the Runge-Kutta type schemes for ODEs, adaptive methods are designed to obtain an estimate of the local truncation error of a single Runge-Kutta step. This is done by embedding two methods in the tableau (see [6]), one with order p and the other with order $p - 1$. The simplest adaptive Runge-Kutta method involves combining the Heun method, which is order 2, with the Euler method, which is order 1. The Runge-Kutta-Fehlberg method has two methods of orders 5 and 4. The error estimate from the lower-order step is used to control the step size. Other adaptive Runge-Kutta methods are the Bogacki-Shampine method (orders 3 and 2), the Cash-Karp method

and the Dormand-Prince Method (both with orders 5 and 4). See [6].

In the following sections, we discuss briefly some of the stochastic adaptive schemes in the literature.

4.2 An Adaptive Euler-Maruyama scheme for SDEs

The adaptive Euler-Maruyama scheme proposed by Lamba et al works for SDEs of the form:

$$dX_t = K(X_t)dt + H(X_t)dB_t, \quad (4.1)$$

where $X_0 = x$, $X_t \in \mathbb{R}^d$ for each t , B_t is an m -dimensional Brownian motion and $K : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $H : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}$. The initial condition X_0 is assumed to be deterministic (independent of B). The adaptive algorithm for [28] is given as follows:

$$\lambda_n = \mathcal{H}(x_n, \lambda_{n-1}), \lambda_{-1} = \Lambda \quad (4.2)$$

$$x_{n+1} = \mathcal{K}(x_n, \Delta_n) + \sqrt{\Delta_n}K(x_n)\eta_{n+1}, x_0 = x \quad (4.3)$$

where $\Delta_n = 2^{-\lambda_n} \Delta_{max}$. Here $\mathcal{K}(x, t) = x + tK(x)$ and

$$\mathcal{H}(x, l) = \min\{\lambda \in \mathbb{Z}^+ : |K(\mathcal{K}(x, 2^{-\lambda} \Delta_{max})) - K(x)| \leq \tau \& \lambda \geq l - 1\}$$

The random variables, η_j form an i.i.d sequence distributed as $\mathcal{N}(0, 1)$. The parameter Λ defines the initial time-step and $\tau > 0$ the tolerance. The whole idea of the adaptive Euler-Maruyama scheme of Lamba et al is that the size of the next step is determined such that the increment based only on the drift term over the suggested step length is within a pre-determined tolerance. That is, take the drift step only and accept or reject the step length based on it and finally take the real stochastic step.

4.3 An asymptotically adaptive algorithm

In [21], Hofmann et al proposed a numerical algorithm that works for SDEs of the form

$$dX_t = K(t, X_t)dt + H(t, X_t)dB_t, \quad (4.4)$$

on $[0, 1]$. The Brownian motion B is 1-dimensional and the drift and diffusion coefficients $K, H : [0, 1] \times \mathbb{R} \rightarrow \mathbb{R}$ have to be scalar respectively. Also, they have to be differentiable with respect to the state variable. Again, K and H together with their derivatives have to satisfy a linear growth bound and Lipschitz conditions. That is, the conditions of Theorem 2.4.2, the initial random variable has to be independent of B and has to admit finite moments of fourth order. It is stated in [29] that the numerical approximation of Hofmann et al works in the pathwise sense and the adaptive discretization reflects the local smoothness properties of each trajectory by approximating the conditional Hölder constant, $|H(t, X_t)|$ along the way and then taking a step size proportional to $\frac{1}{|H(\tau_n, X_{\tau_n})|}$. To approximate X_{τ_n} for an arbitrary discretization $0 = \tau_0 < \dots < \tau_{N^*} = 1$, Hofmann et al proposed the following scheme: $\bar{X}_0 = X_0$

$$\begin{aligned} \bar{X}_{\tau_{n+1}} &= \bar{X}_{\tau_n} + K(\tau_n, \bar{X}_{\tau_n})h_n + H(\tau_n, \bar{X}_{\tau_n})\Delta B_{\tau_n} \\ &\quad + \frac{1}{2}H(\tau_n, \bar{X}_{\tau_n})H^{(0,1)}(\tau_n, \bar{X}_{\tau_n})[(\Delta B_{\tau_n})^2 - h_n], \end{aligned} \quad (4.5)$$

where $H^{(0,1)}$ denotes the partial derivative of H with respect to the second or the state variable, with $h_n = \tau_{n+1} - \tau_n$ and N^* determined by the algorithm. To calculate the adaptive step-size with $\tau_n = 0$, one has to choose a basic step size, $h^* > 0$ and

$$\tau_{n+1} = \tau_n + \min \left\{ \frac{1}{|H(\tau_n, \bar{X}_{\tau_n})|}, (h^*)^{\frac{2}{3}} \right\} \quad (4.6)$$

where \bar{X}_{τ_n} is the result of the Milstein scheme. Again, 4.6 means that the next step length is selected based on the value of the diffusion term at the previous step. It is stated in [29] that Hofmann et al's algorithm is constructed based on the L_2 -error criterion, $\left(\mathbb{E}(\|X - \bar{X}\|_2^2) \right)^{\frac{1}{2}}$. The authors prove that the algorithm is asymptotically optimal with respect to this criterion.

4.4 An adaptive algorithm based on embedded Runge-Kutta scheme

Another step size control approach for the solution of stochastic differential equation is the algorithm proposed by Mauthner in [30]. The algorithm for autonomous SDEs is of the form

$$dX_t = K(X_t)dt + H(X_t)odB_t \quad (4.7)$$

$X_{t_0} = X_0$, in the Stratonovich sense (Itô equations have to be transformed into Stratonovich ones, non-autonomous SDEs have to be made autonomous). See [26] for detailed study on how to transform Itô SDEs to Stratonovich and non-autonomous SDEs to autonomous. Unlike Hoffmann et al's scheme, K and H may be multi-dimensional, however, the Brownian motion B needs to be one-dimensional. The basic step size h^* gives the equidistant discretization $0 = \tau_0^* < \dots < \tau_N^* = T$ with $h^* = \tau_n^* - \tau_{n-1}^*$. The stochastic Runge-Kutta method of Mauthner is of the form:

$$\begin{aligned} \bar{X}_{\tau_{n+1}} &= \bar{X}_{\tau_n} + h_n \sum_{i=1}^s \alpha_i K(Y_i) + \sum_{i=1}^s \left(\gamma_i^{(1)} J_1 + \gamma_i^{(2)} \frac{J_{10}}{h_n} \right) H(Y_i), \\ Y_i &= \bar{X}_{\tau_n} + h_n \sum_{j=1}^{i-1} \alpha_{ij} K(Y_j) + \sum_{i=1}^s \left(\beta_{ij}^{(1)} J_1 + \beta_{ij}^{(2)} \frac{J_{10}}{h_n} \right) H(Y_j), \end{aligned} \quad (4.8)$$

where $\bar{X}_0 = X_0$ and $0 = \tau_0 < \tau_1 < \dots < \tau_N = T$ is an arbitrary discretization. J_1 and J_{10} are defined as follows: $J_1 = \int_{\tau_n}^{\tau_{n+1}} odB_t$, $J_{10} = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau_n}^t odB_{t_1} dt_2$. The embedded stochastic Runge-Kutta method of Mauthner is based on two stochastic Runge-Kutta schemes with strong order 1.0 and a strong order between 1.0 and 1.5 respectively.

The high-order solution y_1 serves as a basis for the strong approximation and is used to estimate the local error of the low-order solution, \hat{y}_1 . For the automatic step-size control one wants to reach $|y_{1_i} - \hat{y}_{1_i}| \leq tol_i$, $1 \leq i \leq d$ for each component, where

$$tol_i = Atol_i + \max\{|y_{0_i}|, |y_{1_i}|\} Rtol_i.$$

$Rtol_i$ and $Atol_i$ describe the limit for the relative and absolute error and one calculates

$err = \sqrt{\frac{1}{d} \sum_{i=1}^d \left(\frac{y_{1_i} - \hat{y}_{1_i}}{tol_i} \right)^2}$ as an estimation of the local error. One step is accepted with the accompanying step-size h if $err \leq 1$ holds.

The step-size h may only be halved or doubled. In a situation where $err > 1$, one calculates the last step again with $\frac{h}{2}$. On the other hand, if $err < 1$, the result is accepted and one fixes the approximation value to $\bar{X}_{\tau_{n+1}}$.

The Runge-Kutta algorithm with step-size control calculates solutions which converge strongly to the exact solution of the examined SDE as the step-size converges to zero [30].

4.5 A variable Riemannian sum for Itô integrals

Using a Riemannian sum with random subintervals, Rapoo, in [40] investigates the problem of approximating the iterated Itô integral, $\int w dw$, hence solving a corresponding stochastic differential equation by the Euler method with variable step-size. The full or half step-size strategy involves a dyadic subdivision. For any given value of n , the interval $[0, 1]$ is divided into $N(n) = 2^n$ intervals of $h = 2^{-n}$. This results to preliminary subdivision points $\tau_n = \{k2^{-n}, k = 0, \dots, 2^n\}$. For each of these intervals of length $h = 2^{-n}$, the step length is either accepted as it is, or is rejected and two steps of length $h = 2^{-(n+1)}$ are taken. The criterion to reject or accept is based on the size of the Brownian increment over the interval: a step of length h is rejected if $\Delta\omega \in A_h$, where, for each $h > 0$, A_h is a predetermined set.

The above condition gives rise to a final random subdivision $\hat{\tau}_n$ with $\tau_n \subseteq \hat{\tau}_n \subseteq \tau_{n+1}$ and,

$$\hat{\tau}_n = \tau_n \cup \left(\bigcup_{k: \Delta\omega_k \in A_2^{-n}} (t_k + 2^{-(n+1)}) \right)$$

Hence, the numerical approximation is

$$X^{(n)}(1) = \sum_{t_i \in \hat{\tau}_n} 2\omega(t_i) \Delta\omega_i \quad (4.9)$$

or alternatively,

$$X^{(n)}(1) = \sum_{t_i \in \tau_n} \Delta \hat{X}_i, \quad (4.10)$$

where $\Delta \hat{X}_i$, the increment of the solution over the i th step is adjusted according to the variable-step scheme: $\Delta X_i = 2\omega(t_i)\Delta\omega_i$ if $\Delta\omega_i \in A_h^C$ (the complement of A_h) on the other hand, if $\Delta\omega_i \in A_h$ then $\Delta \hat{X}_i = 2\omega(t_i)\Delta_1\omega_i + 2\omega(t_i + \frac{h}{2})\Delta_2\omega_i$. Here, $h = 2^{-n}$ and $\Delta_1\omega_i$ and $\Delta_2\omega_i$ are the increments of the Brownian motion over the two half-steps of lengths $\frac{h}{2}$. For different types of sets A_h , there are different types of variable step-size schemes. The main result of Rapoo's work is the theorem that gives an optimal choice of the set A_h . Optimality is defined based on the following criteria:

- which selections of A_h give unbiased methods, i.e for which the expected error is zero for all h .
- which selections give convergent methods
- which methods improve on a fixed-step Euler method by given better accuracy at the same cost, or the same accuracy at less cost.

Theorem 4.5.1. $E[\hat{E}] = 0$ for the following choices of sets, A_h : [40]

(a) $A_h = \mathbb{R}_-$ or $A_h = \mathbb{R}_+$,

(b) $A_h = \{x : |x| \leq K\sqrt{h} \text{ or } |x| \geq L\sqrt{h}\}$, where $L \geq 1$ and

$K = -W(-L^2 \exp(-L^2))$,

(c) $A_h = \{x : K\sqrt{h} < |x| < L\sqrt{h}\}$

Here $W(x)$ is the product logarithm' function: $z = W(x)$ is the (real) solution to $z \exp(z) = x$.

4.6 Summary: adapted approaches

One of the above discussed adaptive stochastic schemes determines next step length based on current value, the other three (Lamba et al [28], Holmann et al [21] and

Rapoo [40]) attempt a step and then reject or accept and one of them (Lamba et al) takes initially a deterministic step. The other two (Hofmann et al and Rapoo) attempt a full random step and then adjust the step length based on the result. It is also important to note that in Mathner's approach the numerical method must be of high enough order to work (that is, it must include iterated path integrals).

In Rapoo [40], the step-size decision is based on the value of ΔB over the step.

The 'classic' adaptive scheme of taking a step and then accepting or rejecting it based on what happens may backfire with stochastic integrals because the construction of stochastic integrals only works if we assume that we do not look into the future (the numerical scheme must be non-anticipating). A try-a-step-then-accept-or-reject approach is anticipating. Mauthner's Runge-Kutta rule gets around this because of a result by T. Lyons & J. Gaines [10] which states that any step discretization, even an adapted (anticipative) one will work if the scheme gets the iterated integrals right- which means it needs to be a 1.5 order scheme, which is difficult to implement. Rapoo's scheme works for a low-order (Euler) scheme but only when one makes an effort to ensure no bias, which is a very important factor to consider in an anticipating low-order scheme. In this dissertation, we develop a hybrid method which continues that idea but rather chooses a method based on Brownian motion increment instead of step-size.

Chapter 5

Hybrid Numerical Methods for the Solution of SDEs

5.1 Introduction

In this chapter, we present our idea of a hybrid numerical scheme for the solution of stochastic differential equations. The global error, convergence as well as the unbiasedness of the hybrid method are investigated. For any given value of n , let the interval $[t, T]$, $t \geq 0$ be divided into $N(n) = 2^n$ intervals of length $h = 2^{-n}(T - t)$. Let X and Y be any two standard numerical schemes, with different orders of convergence, approximating the solution of an SDE over each of these subintervals. At each interval, either of X or Y is used to numerically compute the solution to the SDE. The choice of which scheme to apply is determined based on a function of the Brownian increment over the interval. Let $g(z)$ be a real function of the Brownian increment over the interval $[t, t + h]$ ¹, where, $z = B_{t+h} - B_t, h > 0$. The scheme with the lower order of convergence, which we assume is X , is applied to the solution if $g(z) \in \Lambda$, where Λ is a predetermined set; otherwise the other scheme is used. The actual expression for $g(z)$ is determined based on X or Y . Then at each discretization point t , the hybrid

¹ t in this case is an arbitrary point on $[0, T]$

numerical scheme, $\hat{S}(\Delta B)$ obtained from the standard schemes is given by:

$$\hat{S}(\Delta B) = X1_{\Lambda}(\Delta B) + Y1_{\Lambda^c}(\Delta B), \quad (5.1)$$

where Λ^c is the complement of Λ . Here $1_{\Lambda}(z)$ is the indicator function of the event $\{g(z) \in \Lambda\}$, and z is the increment of the Brownian motion over the interval. If $\Lambda = \emptyset$ then there is no hybrid scheme; the resulting numerical scheme is equivalent to Y . On the other hand if $\Lambda^c = \emptyset$ the hybrid scheme corresponds to X , the lower order standard scheme. More importantly, for $\Lambda \neq \emptyset$ and $\Lambda^c \neq \emptyset$, we have a hybrid algorithm given by (5.1). The question of the optimal choice of the set Λ needs to be answered in the light of improvement in accuracy in comparison to the standard scheme of lower order of convergence. To this end, we compute the error as well as the order of convergence of the hybrid algorithm. In the following sections, we look at a specific hybrid scheme and investigate its global and local error as well as its order of convergence. Because our method is anticipating, it becomes necessary that the unbiasedness of the hybrid scheme be investigated.

In general, one can obtain many hybrid schemes using a combination of the standard stochastic schemes provided the choice criterion leads to an improved performance compared to the parent schemes. However, for brevity and for the purpose of this dissertation, we only look at the Euler-Milstein hybrid scheme based on simulated Brownian increments. Development of other hybrid schemes using higher order stochastic schemes and perhaps the Runge-Kutta methods in line with this idea would be of great research interest in future.

5.2 Euler-Milstein hybrid Scheme

We now turn to the Euler-Milstein hybrid scheme. Throughout this chapter, we shall consider the one dimensional SDE of the form

$$dX_t = K(X_t)dt + H(X_t)dB_t, \quad (5.2)$$

where X_0 is a constant and t is an arbitrary point on the interval $[0, T]$. Note that the multidimensional case follows similarly but component-wise. The Euler scheme for the above SDE is given by (3.8) and the Milstein scheme by (3.23). From (5.1), the Euler-Milstein hybrid scheme at the point $t + \Delta t$ is given by

$$\begin{aligned} \hat{X}_{t+\Delta t} = & \{\hat{X}_t + K(\hat{X}_t)\Delta t + H(\hat{X}_t)\Delta B_t\}1_\Lambda(\Delta B) \\ & + \{\hat{X}_t + K(\hat{X}_t)\Delta t + H(\hat{X}_t)\Delta B_t + \frac{1}{2}H(\hat{X}_t)\frac{dH}{dX}(\hat{X}_t)((\Delta B_t)^2 - \Delta t)\}1_{\Lambda^c}(\Delta B) \end{aligned} \quad (5.3)$$

In our case, we define $g(z) = z^2 - \Delta t$. Then the set Λ is defined as $\{z : |g(z)| \leq \epsilon\}$ for $t \in [0, T]$. where ϵ is a predetermined value. Applying the quadratic variation property of Brownian motion,(see [8]), we have that $\mathbb{E}[g(\Delta B_t)] = 0$ for all t . The choice of ϵ determines how often we apply Euler or Milstein respectively. for instance, setting $\epsilon = 0$; $\Lambda = \phi$ implies that we always use Milstein and never Euler and $\epsilon = \infty$ means the opposite. The idea here is that the hybrid scheme uses the Euler algorithm when a factor of the additional term that differentiates the Euler and Milstein schemes over any particular discretization interval is significantly small and uses the Milstein scheme otherwise. This saves the computational cost of computing the first derivative of the diffusion coefficients needed for the Milstein method. However, other types of criteria can still be used to define the set Λ , provided the choice results in an unbiased² method. In what follows, we investigate the unbiasedness as well as the error and strong order of convergence of the Euler-Milstein hybrid scheme.

5.2.1 An intuitive illustration of the order convergence of the Euler-Milstein hybrid scheme

Since the Euler-Milstein scheme is partly Euler and partly Milstein, it is obvious that the global error of the Euler-Milstein hybrid scheme is partly due to the Euler and partly due to Milstein schemes. The error contribution of each subscheme depends only on the value of the Brownian motion increment at any discretization interval

²The expected value of the global error vanishes

over the interval, $[0, T]$. Because the hybrid scheme is partly Milstein, it is expected to have a higher order of convergence compared to the Euler scheme, and perhaps have better accuracy as well. However, what is not trivial is the improvement (in increased order of convergence and better accuracy) on the Euler scheme as a result of the choice of Λ and not as a result of the part-use of the Milstein scheme in computing the solution at some points on the interval. We now proceed to analyse the local and global error of the Euler-Milstein hybrid scheme. Before we look at the theoretical strong order of the mean square error of the hybrid scheme, we first give an intuitive illustration of why the strong order of the hybrid scheme is bounded below and above by the strong orders of convergence of Euler and Milstein, respectively. It should be noted that this is not a proof of order of convergence of the hybrid scheme, but an illustration that it is in line with the idea of a hybrid scheme. From (5.1) we see that the global error, $E^{hyb} = X - \hat{X}$ can be written as

$$E^{hyb} = \sum_{\Lambda_i \in [\Lambda, \Lambda^c]} e_{\Lambda_i} \quad (5.4)$$

where for $i \in [1, 2]$, $e_{\Lambda} = err(X)1_{\Lambda}$ & $e_{\Lambda^c} = err(Y)1_{\Lambda^c}$

For simplicity, we analyse the local and global error of the hybrid scheme using the geometric Brownian motion,

$$dX_t = \mu X_t dt + \sigma X_t dB_t \quad (5.5)$$

where μ and σ are constants. Recall that the exact solution for this process is given by (3.13), i.e

$$X_t = X_0 \exp\left\{\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B_t\right\}$$

Now if $\Lambda = \mathcal{R}$ then the hybrid algorithm coincides with the Euler scheme and we have

$$\hat{X}_{t_{j+1}} = \hat{X}_{t_j} + \mu \hat{X}_{t_j} \Delta t + \sigma \hat{X}_{t_j} \Delta B_{t_j}$$

i.e

$$\hat{X}_{t_{j+1}} = \hat{X}_{t_j} [1 + \mu\Delta t + \sigma\Delta B_{t_j}],$$

which implies that at the point $t = t_k$

$$\hat{X}_{t_k} = X_0 \prod_{j=0}^{k-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}].$$

For $t_0 = 0, t_1 = \frac{T}{n}, t_2 = \frac{2T}{n}, \dots, t_n = T$ the error in strong convergence is given by

$$\begin{aligned} & \mathbb{E} \left| \hat{X}_T - X_T \right| \\ &= X_0 \mathbb{E} \left| \prod_{j=1}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}] - \exp\left\{(\mu - \frac{1}{2}\sigma^2)T + \sigma B_T\right\} \right| \end{aligned} \quad (5.6)$$

The exponential term in (5.6) evaluated to $\mathcal{O}(\Delta t)^2$ Taylor expansion leads to

$$\begin{aligned} \exp\left\{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma B_{t_j}\right\} &= 1 + \left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\Delta B_{t_j} \right] \\ &+ \frac{1}{2} \left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\Delta B_{t_j} \right]^2 \\ &+ \frac{1}{6} \left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\Delta B_{t_j} \right]^3 + \dots \\ &= 1 + \left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\Delta B_{t_j} \right] \\ &+ (\mu - \frac{1}{2}\sigma^2)\sigma\Delta t\Delta B_{t_j} + \frac{1}{2}\sigma^2[\Delta B_{t_j}]^2 \\ &+ \frac{1}{6}\sigma^3[\Delta B_{t_j}]^3 + \mathcal{O}(\Delta t)^2 \\ &= 1 + \mu\Delta t + \sigma\Delta B_{t_j} + \mu'\sigma\Delta t\Delta B_{t_j} + \frac{1}{6}[\Delta B_{t_j}]^3 + \mathcal{O}(\Delta t)^2 \end{aligned} \quad (5.7)$$

where, $\mu' = (\mu - \frac{1}{2}\sigma^2)$ and assuming that $\Delta B_t \approx \Delta t$. Then,

$$\begin{aligned} & 1 + \mu\Delta t + \sigma\Delta B_{t_j} \\ &= \exp\left\{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma B_{t_j}\right\} - \mu'\sigma\Delta t\Delta B_{t_j} - \frac{1}{6}[\Delta B_{t_j}]^3 - \mathcal{O}(\Delta t)^2 \end{aligned} \quad (5.8)$$

and

$$\begin{aligned} & \prod_{j=0}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}] \\ &= \prod_{j=0}^{n-1} \left[\exp\left\{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma\Delta B_{t_j}\right\} - \mu'\sigma\Delta t\Delta B_{t_j} - \frac{1}{6}[\Delta B_{t_j}]^3 - \mathcal{O}(\Delta t)^2 \right] \end{aligned} \quad (5.9)$$

\therefore

$$\begin{aligned} & \prod_{j=0}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}] \\ &= \left[\exp\left\{(\mu - \frac{1}{2}\sigma^2)T + \sigma B_T\right\} + n\mathcal{O}(\Delta t\Delta B_{t_j}) + n\mathcal{O}([\Delta B_{t_j}]^3) + n\mathcal{O}(\Delta t)^2 \right] \end{aligned} \quad (5.10)$$

Therefore, the error (5.6) is

$$\begin{aligned} \mathbb{E} \left| \hat{X}_T - X_T \right| &= \mathbb{E} \left| n\mathcal{O}(\Delta t\Delta B) + n\mathcal{O}(\Delta B)^3 + n\mathcal{O}(\Delta t)^2 \right| \\ &= \mathbb{E} \left| \frac{T}{\Delta t}\mathcal{O}(\Delta t\Delta B) + \frac{T}{\Delta t}\mathcal{O}(\Delta B)^3 + \frac{T}{\Delta t}\mathcal{O}(\Delta t)^2 \right| \\ &= T\mathbb{E} \left| \frac{1}{\Delta t}\mathcal{O}(\Delta t\Delta B) + \frac{1}{\Delta t}\mathcal{O}(\Delta B)^3 + \frac{1}{\Delta t}\mathcal{O}(\Delta t)^2 \right| \\ &= \mathcal{O}(\Delta t)^{\frac{1}{2}} \end{aligned} \quad (5.11)$$

which is the same as the order of strong convergence of the Euler method.

Similarly, when $\Lambda = \phi$ then the hybrid scheme reverts to the standard Milstein scheme in which case we have

$$\begin{aligned} \hat{X}_{t_{j+1}} &= \hat{X}_{t_j} + \hat{X}_{t_j}\mu\Delta t + \hat{X}_{t_j}\sigma\Delta B_{t_j} + \frac{1}{2}\hat{X}_{t_j}\sigma^2((\Delta B_{t_j})^2 - \Delta t) \\ &= \hat{X}_{t_j} [1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}\sigma^2\{(\Delta B_{t_j})^2 - \Delta t\}] \end{aligned}$$

The exact solution, (3.13) evaluated to $\mathcal{O}(\Delta t)^3$ Taylor expansion gives the following

$$\begin{aligned} & 1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}\sigma^2(\Delta B_{t_j}^2 - \Delta t) \\ &= \exp\left\{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma\Delta B_{t_j}\right\} - \frac{1}{2}\mu'^2\Delta t + \frac{1}{2}\mu'\sigma^2\Delta t^2 + \frac{1}{6}\mu'^3\Delta t^3 \end{aligned} \quad (5.12)$$

where $\mu' = (\mu - \frac{1}{2}\sigma^2)$

Then we get

$$\prod_{j=0}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}\sigma^2(\Delta B_{t_j}^2 - \Delta t)] \quad (5.13)$$

$$\begin{aligned} &= \prod_{j=0}^{n-1} [\exp\{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma\Delta B_{t_j}\} - \frac{1}{2}\mu'^2\Delta t^2 \\ &\quad + \frac{1}{6}\mu'^3\Delta t^3 + \frac{1}{2}\mu'\sigma^2\Delta t\Delta B_{t_j}^2] \quad (5.14) \\ &= [\exp\{(\mu - \frac{1}{2}\sigma^2)T + \sigma\Delta B_T + \frac{1}{2}\sigma^2(\Delta B_T^2 - T)\} \\ &\quad + n\mathcal{O}(\Delta t)^2 + n\mathcal{O}(\Delta t)^3 + n\mathcal{O}(\Delta t\Delta B_{t_j}^2)] \end{aligned}$$

then the error expression becomes.

$$\begin{aligned} &\mathbb{E} \left| \hat{X}_T - X_T \right| \\ &= \mathbb{E} \left| n\mathcal{O}(\Delta t)^2 + n\mathcal{O}(\Delta t\Delta B_t^2) + n\mathcal{O}(\Delta t)^3 \right| \\ &= \mathbb{E} \left| \frac{T}{\Delta t}\mathcal{O}(\Delta t)^2 + \frac{T}{\Delta t}\mathcal{O}(\Delta t\Delta B_t^2) + \frac{T}{\Delta t}\mathcal{O}(\Delta t)^3 \right| \\ &= T\mathbb{E} \left| \underbrace{\frac{1}{\Delta t}\mathcal{O}(\Delta t)^2}_{\mathcal{O}(\Delta t)} + \underbrace{\frac{1}{\Delta t}\mathcal{O}(\Delta t\Delta B_t^2)}_{\mathcal{O}(\Delta t)} + \underbrace{\frac{1}{\Delta t}\mathcal{O}(\Delta t)^3}_{\mathcal{O}(\Delta t)^2} \right| \\ &= \mathcal{O}(\Delta t) \end{aligned}$$

which again coincides with the order of the Milstein scheme.

Of more importance is the case where $\Lambda \neq \emptyset$ and $\Lambda^C \neq \emptyset$. Here the hybrid Euler-

Milstein scheme is given by (5.3) i.e

$$\begin{aligned}
& \hat{X}_{t_{j+1}} \\
&= \{\hat{X}_{t_j}[1 + \mu\Delta t + \sigma\Delta B_{t_j}]\}_{1_\Lambda} + \{\hat{X}_{t_j}[1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}\sigma^2\{(\Delta B_{t_j})^2 - \Delta t\}]\}_{1_{\Lambda^c}} \\
&= \left\{ \hat{X}_{t_j} \left\{ [1 + \mu\Delta t + \sigma\Delta B_{t_j}]\}_{1_\Lambda} + [1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}\sigma^2\{(\Delta B_{t_j})^2 - \Delta t\}]\}_{1_{\Lambda^c}} \right\} \\
&= \hat{X}_{t_j} \left\{ [1 + \mu\Delta t + \sigma\Delta B_{t_j}] + \frac{1}{2}(\Delta B_{t_j}^2 - \Delta t)1_{\Lambda^c} \right\} \\
&= X_0 \prod_{j=1}^{n-1} \left\{ [1 + \mu\Delta t + \sigma\Delta B_{t_j}] + \frac{1}{2}(\Delta B_{t_j}^2 - \Delta t)1_{\Lambda^c} \right\}
\end{aligned}$$

Using the stochastic Taylor expansion as in (5.7) we obtain

$$\begin{aligned}
& 1 + \mu\Delta t + \sigma\Delta B_{t_j} + \frac{1}{2}(\Delta B_{t_j}^2 - \Delta t)1_{\Lambda^c} \\
&= \exp\left\{ \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta B_{t_j} \right\} - [\mu'\sigma\Delta t\Delta B_{t_j}]1_\Lambda - \left[\frac{1}{6}[\Delta B_{t_j}]^3\right]1_\Lambda - [\mathcal{O}(\Delta t)^2]1_\Lambda \\
&\quad - [\mathcal{O}(\Delta t\Delta B_{t_j}^2)]1_{\Lambda^c} - [\mathcal{O}(\Delta t)^{\frac{3}{2}}]1_{\Lambda^c} - [\mathcal{O}(\Delta t)^3]1_{\Lambda^c}
\end{aligned}$$

Letting $n(\Lambda)$ and $n(\Lambda^c)$ be the number of times the hybrid scheme uses Euler and Milstein schemes respectively, we have that $n(\Lambda) = \frac{T}{\Delta t}\rho(\Lambda)$ and similarly $n(\Lambda^c) = \frac{T}{\Delta t}\rho(\Lambda^c)$ where $\rho(\Lambda)$ and $\rho(\Lambda^c)$ is the probability that we use Euler and Milstein respectively.

Hence

$$\prod_{j=0}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}]1_\Lambda = \prod_{j=0}^{n(\Lambda)-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}]$$

similarly,

$$\begin{aligned}
& \prod_{j=0}^{n-1} [1 + \mu\Delta t + \sigma\Delta B_{t_j}] + \frac{1}{2}\sigma^2(\Delta B_{t_j}^2 - \Delta t)1_{\Lambda^c} \\
&= \exp\left\{ \left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma B_T \right\} + n(\Lambda)\mathcal{O}(\Delta t\Delta B_{t_j}) + n(\Lambda)\mathcal{O}([\Delta B_{t_j}]^3) + n(\Lambda)\mathcal{O}(\Delta t)^2 \\
&\quad + n(\Lambda^c)[\mathcal{O}(\Delta t\Delta B_{t_j}^2)] + n(\Lambda^c)[\mathcal{O}(\Delta t)^2\Delta B_{t_j}] + n(\Lambda^c)[\mathcal{O}(\Delta t)^3]
\end{aligned}$$

Therefore, the terminal error of the hybrid scheme becomes

$$\begin{aligned}
& \mathbb{E} \left| \hat{X}_T - X_T \right| \\
&= \mathbb{E} \left| \frac{T}{\Delta t} \rho(\Lambda) \mathcal{O}(\Delta t)^2 + \frac{T}{\Delta t} \rho(\Lambda^C) \mathcal{O}(\Delta t \Delta B_{t_j}^2) + \frac{T}{\Delta t} \rho(\Lambda^C) \mathcal{O}(\Delta t)^3 \right| \\
&= T \mathbb{E} \left| \frac{1}{\Delta t} \rho(\Lambda) \mathcal{O}(\Delta t)^2 + \frac{1}{\Delta t} \rho(\Lambda^C) \mathcal{O}(\Delta t \Delta B_t^2) + \frac{1}{\Delta t} \rho(\Lambda^C) \mathcal{O}(\Delta t)^3 \right| \\
&= \mathcal{O}(\Delta t)^\alpha
\end{aligned}$$

where $\alpha \in [\frac{1}{2}, 1]$. Next we investigate the unbiasedness of the Euler-Milstein algorithm.

5.2.2 Bias of the hybrid scheme

In [40] an unbiased method is one in which for all $h = \Delta t$, the expected value of the global error vanishes. Using the above definition and the notation of the previous sections, we see that the Euler-Milstein hybrid scheme is unbiased if $\mathbb{E}[E^{hyb}] = 0$.

It is sufficient to investigate the unbiasedness of the local (one-step) error E^{hyb} , defined as (see (5.3))

$$\begin{aligned}
\mathbb{E}[E_j^{hyb}] &= \mathbb{E}(Euler_{error_j}) - \mathbb{E}err([1_{\Lambda^C}(\Delta B_{t_j}^2 - \Delta t)]) \\
&= \mathbb{E}(Milstein_{error_j}) + \mathbb{E}err([1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t)])
\end{aligned} \tag{5.15}$$

where $\mathbb{E}err([1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t)])$ is the error of the differentiating term between Euler and Milstein methods for a particular discretization interval, and $Euler_{error_j}$ and $Milstein_{error_j}$ are the one-step error of the Euler and Milstein methods, respectively.

The expectation in (5.15) can be evaluated as follows

$$\mathbb{E}[E_j^{hyb}] = \int_{\Lambda} \frac{(x^2 - h)}{h} \rho(h, x) dx \tag{5.16}$$

where $\rho(h, x)$ is the transition probability density function of the standard Brownian motion. i.e. $\rho(t, x) = \frac{1}{\sqrt{2\pi t}} \exp(\frac{-x^2}{2t})$. Since the standard Euler and Milstein schemes are known to be unbiased (see [24],[10] and [26]) it follows then that the Euler-Milstein

hybrid scheme is locally unbiased if

$$\mathbb{E}err([1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t)]) = 0.$$

Now going back to the bias nature of the Euler-Milstein hybrid scheme, we have that our hybrid scheme is unbiased if, for any, $h = \Delta t$, Λ is such that

$$\mathbb{E}[e_j] = -\frac{1}{2} \int_{\Lambda} \frac{(x^2 - h)}{h} \rho(h, x) dx = 0$$

Again, using the approach and notation in [40], we write

$$\mathbb{E}[e_j] = \int_{\Lambda} G_h(x) dx$$

where

$$G_h(x) = \frac{(x^2 - h)}{h} \frac{1}{\sqrt{2\pi h}} \exp\left(-\frac{x^2}{2h}\right).$$

Scaling the function $G_h(x)$ with respect to h we get

$$G_1 = -\frac{1}{2} \sqrt{h} G_h(x\sqrt{h})$$

The graph of G_1 is shown in Figure 5.1. Note that $G_h = 0$ at $x = \pm\sqrt{h}$.

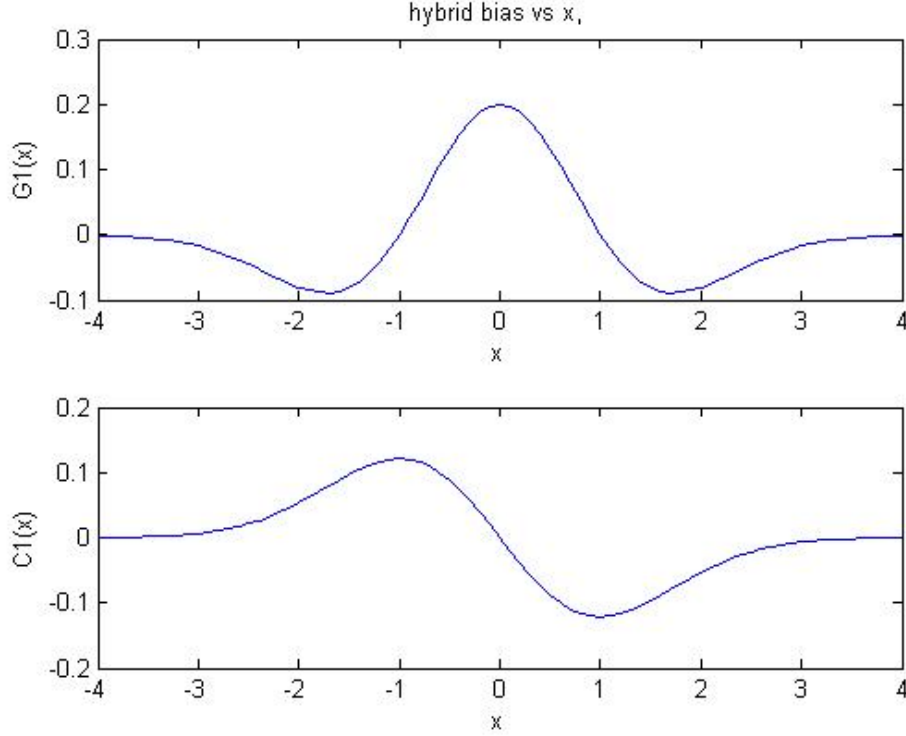
The cumulative integral of $G_h(x)$ denoted by $C_h(x)$, in closed form is calculated as

$$C_h(x) = \int_{-\infty}^x G_h(y) dy = \frac{1}{2\sqrt{2\pi}} \frac{x}{\sqrt{h}} \exp\left(-\frac{x^2}{2h}\right)$$

similarly, C_h scales as $C_1 = C_h(x\sqrt{h})$. Naturally, the value of the cumulative function equals the bias for the hybrid method in which $\Lambda^C = (-\infty, x]$. Note that $x = -\infty$ or $+\infty$ lead to $\Lambda^C = \mathbb{R}$ and $\Lambda^C = \phi$ (corresponding to the Euler and Milstein schemes respectively), which are seen to be unbiased methods.

As evident in the graph of C_1 (see figure 5.1) the absolute maximum and minimum at $x = \pm h$ of C_h for each set, Λ have the same value. That is,

$$\begin{aligned} \max\{C_h(x), x \in \mathbb{R}\} &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\right) \approx 0.1209 \\ \min\{C_h(x), x \in \mathbb{R}\} &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\right) \approx -0.1209 \end{aligned}$$

Figure 5.1: graph of the functions G_1 and C_1

Using the chosen criterion³ of our hybrid scheme, we now analyse the behaviour of the hybrid scheme's bias with respect to ϵ . If $h > \epsilon$, we have that the one-step bias is given by

$$\mathbb{E}e_j = 2 \int_{\sqrt{h-\epsilon}}^{\sqrt{h+\epsilon}} G_h(x) \quad (5.17)$$

On the other for $h < \epsilon$, the hybrid scheme bias becomes

$$\mathbb{E}e_j = \int_{-\sqrt{h+\epsilon}}^{\sqrt{h+\epsilon}} G_h(x) \quad (5.18)$$

The maximum bias is obtained when $h = \epsilon$ and the value is 0.207552 (see Figure 5.2). We can see from Figure 5.2 that there is always a peak at the value $h = \epsilon$ moreover, the size of the peak is a constant for all h and ϵ .

³(i.e. $|(\Delta B^2 - \Delta t)| < \epsilon$)

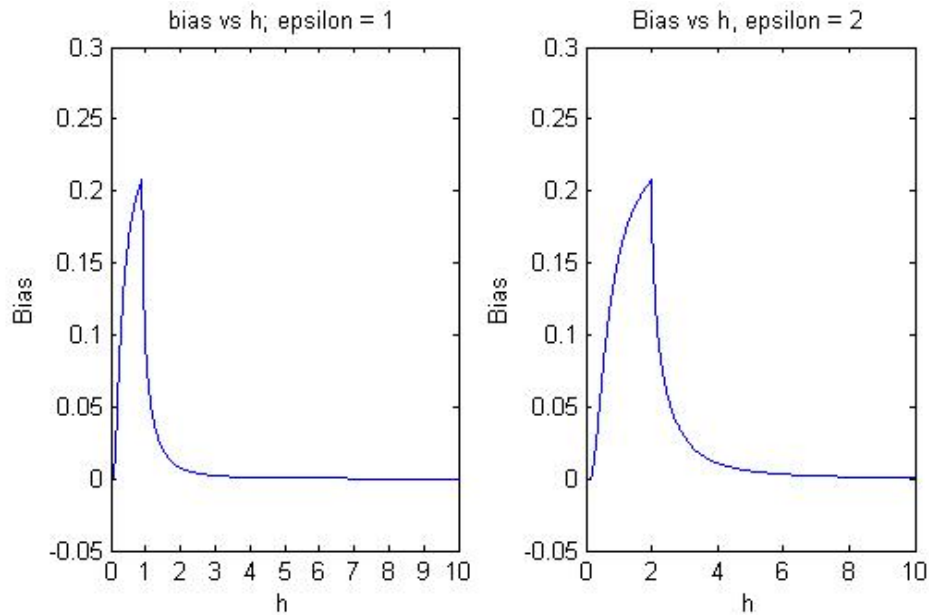


Figure 5.2: Bias of the hybrid scheme vs h , $\epsilon = 1, 2$, $h = 0 : 10$

,

We further establish that the bias of the hybrid scheme scales is as follows

$$bias(h, \epsilon) = bias(N * h, N * \epsilon) \quad (5.19)$$

where N is a positive integer Figure (5.3). An important question to consider would therefore be, what would N need to be so that $h > N * \epsilon$ gives a bias less than δ ? Another important factor is the rate at which the bias goes to zero as N increases; see Figures 5.4 and 5.5

According to Figure (5.5), the bias with fixed ϵ (of whatever value) and h fixed as $h = N * \epsilon$ is of order N^{-3} (this is found by fitting a linear function through the data points in Figure (5.5); the slope is 3.016) for $N > 2$ so that already at $h = \epsilon * 10$, the bias would be $\frac{1}{1000}$. Of course we must look at that in relation to the error of the hybrid method: we would want the bias to be small in relation to the order of the error. With $h = 0.001$, the Milstein error (local) would also be of the order 10^{-3} , so to make the bias negligible, we would just need to take h a hundred times larger than ϵ . In conclusion, by taking h sufficiently smaller than ϵ , the bias is negligible. (See

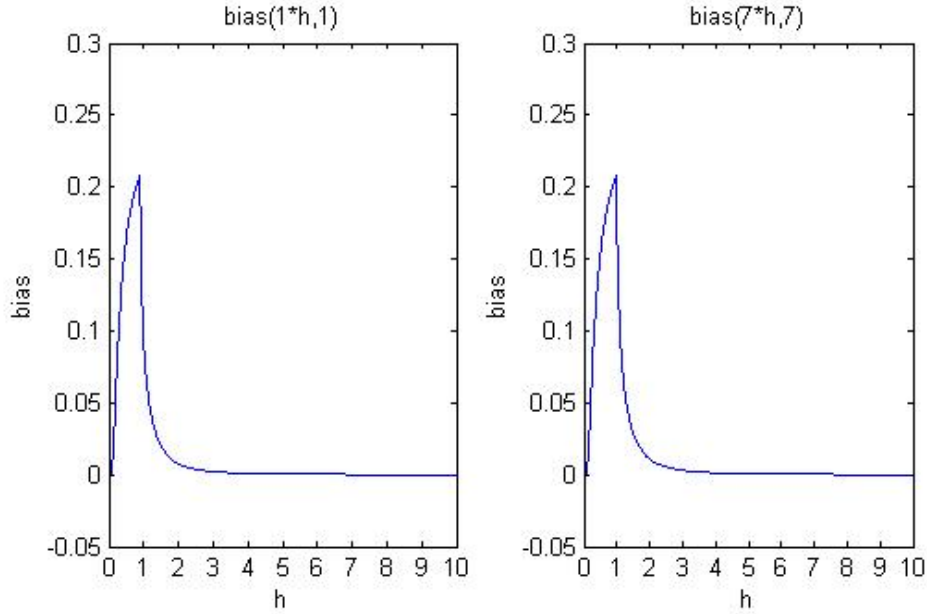


Figure 5.3: bias of the hybrid scheme vs h , $\epsilon = 1, 7$, $h = 0 : 10$

also Figure 5.6).

Figure 5.7 deals with the case, $h \ll \epsilon$. Again we fix ϵ and vary h . We observe that the bias goes to zero as h decreases.

5.2.3 Strong order of the error of the hybrid scheme

We now look more theoretically into the strong order of convergence of the hybrid scheme. We started by analysing the local mean square error of the hybrid scheme, then use the result by Milstein (see [31], pp. 13) to analyse the global mean square error of the hybrid scheme. Let e_i be the local error of the hybrid scheme at the interval $t_i - t_{i+1}$, and p_1 , a real number, be the strong order of the mean square error of the method, if

$$\mathbb{E}[(e_i)^2]^{\frac{1}{2}} \leq \mathcal{O}(h)^{p_1} \quad (5.20)$$

Then, in effect, Milstein's result, [31] stated that

$$\mathbb{E}[(e_{\text{global}})^2]^{\frac{1}{2}} \leq \mathcal{O}(h)^{(p_1 - \frac{1}{2})}. \quad (5.21)$$

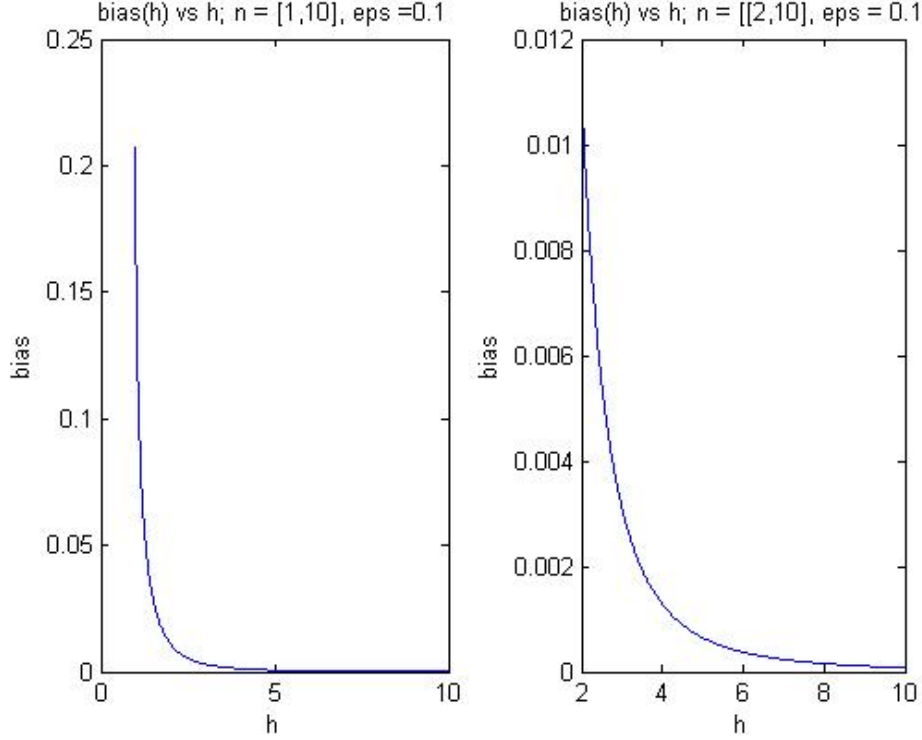


Figure 5.4: Bias of the hybrid scheme vs h , $\epsilon = 0.1$; $h = N * \epsilon$, $N = [2 : 10]$

. The local error of the hybrid scheme is given by

$$E_j^{hyb} = Euler_{error_j} - err(1_{\Lambda^c}(\Delta B_{t_j}^2 - \Delta t)) \quad (5.22)$$

$$= Milstein_{error_j} + err(1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t)) \quad (5.23)$$

Since we know the orders of the mean square error of both Euler and Milstein, it suffices to calculate any of $err(1_{\Lambda^c}(\Delta B_{t_j}^2 - \Delta t))$ or $err(1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t))$ in order to determine the strong order of the mean square error of the hybrid scheme. Using the conditional distribution derived in [40] we have that $err(1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t))$ is given by

$$\mathbb{E}[e_j^2] = \int_{\Lambda} (x^2 - h)^2 \rho(h, x) dx \quad (5.24)$$

where e_j is the local error of the $err(1_{\Lambda}(\Delta B_{t_j}^2 - \Delta t))$ and $\rho(h, x)$ is the transition probability density function of the standard Brownian motion i.e. $\rho(t, x) = \frac{1}{\sqrt{2\pi t}} \exp(-\frac{x^2}{2t})$.

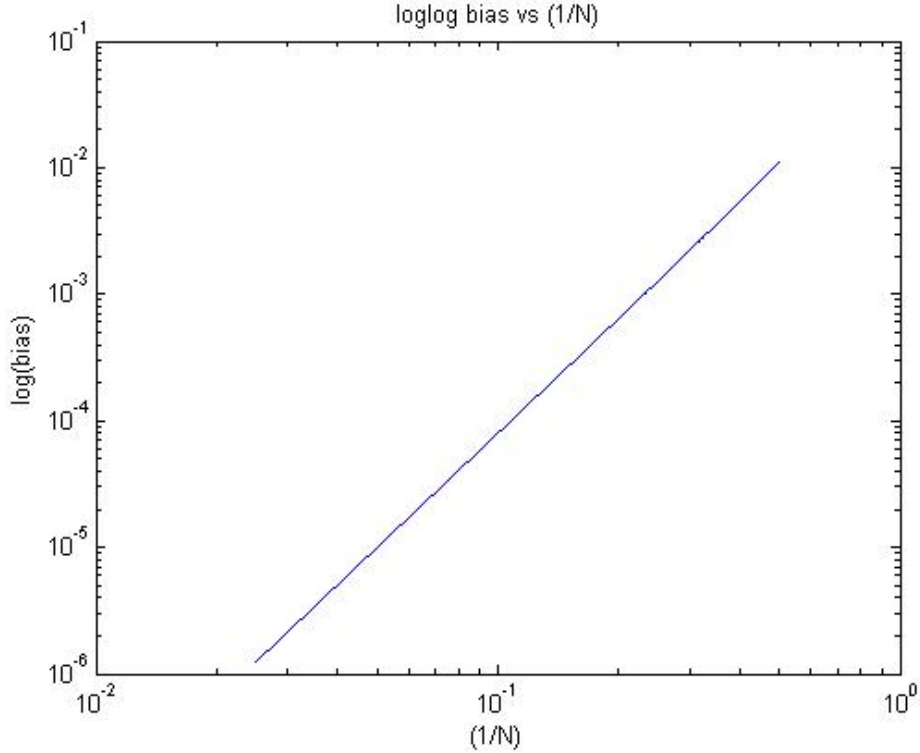


Figure 5.5: Bias of the hybrid scheme vs h , $\epsilon = 1$, $h = N * \epsilon$, $N = [2, 40]$

In particular, for our chosen criterion, if the step length $h > \epsilon$ then 5.24 becomes

$$\mathbb{E}[e_j^2] = 2 \int_{\sqrt{h-\epsilon}}^{\sqrt{h+\epsilon}} (x^2 - h)^2 \rho(h, x) dx \quad (5.25)$$

on the other hand if $\epsilon \leq h$ then we get

$$\mathbb{E}[e_j^2] = \int_{-\sqrt{h+\epsilon}}^{\sqrt{h+\epsilon}} (x^2 - h)^2 \rho(h, x) dx \quad (5.26)$$

The maximum error is obtained at $h = \epsilon$ (see Figure 5.8), the left plot shows how the maximum error is attained at $h = \epsilon$ and how error decreases as $h \gg \epsilon$. The plot at the right of (Figure 5.8) shows how the maximum error increases as ϵ increases. A further investigation specifically shows that the maximum error grows as ϵ^2 (see Figure 5.10). (Figure 5.11) illustrates how the error varies with h only for the case where $h > \epsilon$ with $\epsilon = 0.00001$. It is remarkable here that beyond the initial maximum values, the error actually gets smaller as h increases, in particular the errors for $h > 10 * \epsilon$ are

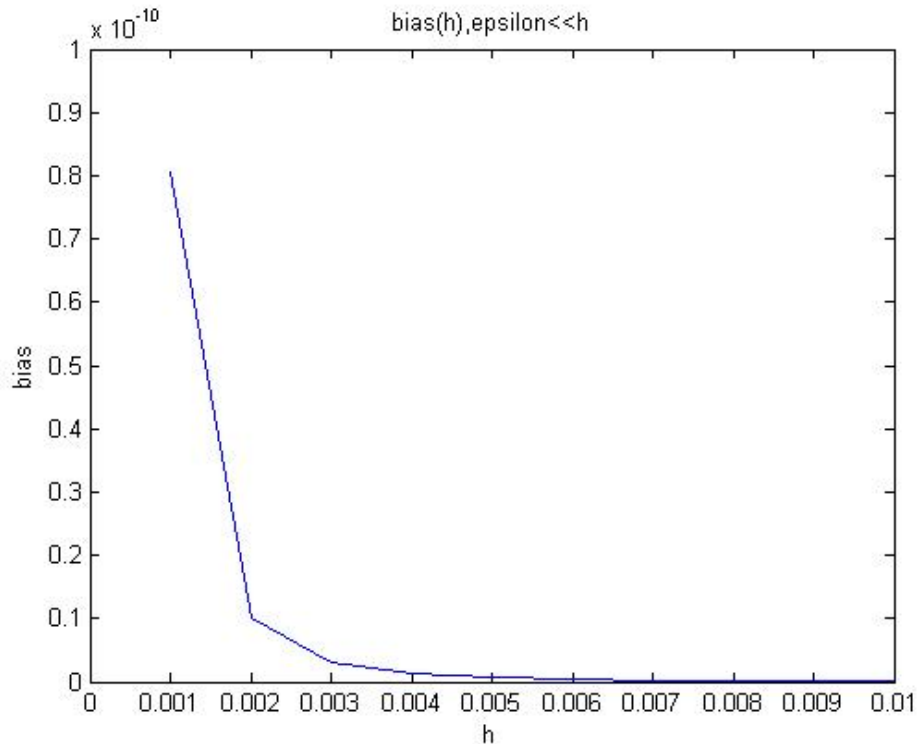


Figure 5.6: Bias of the hybrid scheme vs h , $\epsilon = 0.000001, h = 0.001:0.01$

far below the Milstein errors. This is in agreement with the numerically computed error of the hybrid scheme using a non-linear SDE with polynomial coefficients, (6.4). (See Figure 5.9).

A joint plot of the mean square error of the Euler scheme of order h^2 and Milstein, order h^3 together with the mean square error of the extra term of the hybrid scheme reveals that at some stage, the error of the extra term falls below the Milstein order of mean square error, h^3 (see Figure 5.12). To obtain the actual order of mean square error of the hybrid scheme we use (5.23). The local mean square error in the hybrid method would then be roughly⁴ the calculated mean square error of the extra term plus the mean square error of the Milstein method of order h^3 . The combined error is as shown in (Figure 5.13). (Figures 5.14 and 5.15) show how the order of mean square error of the hybrid scheme decreases as ϵ increases. This is

⁴ Only roughly, since we have not added the correct coefficients

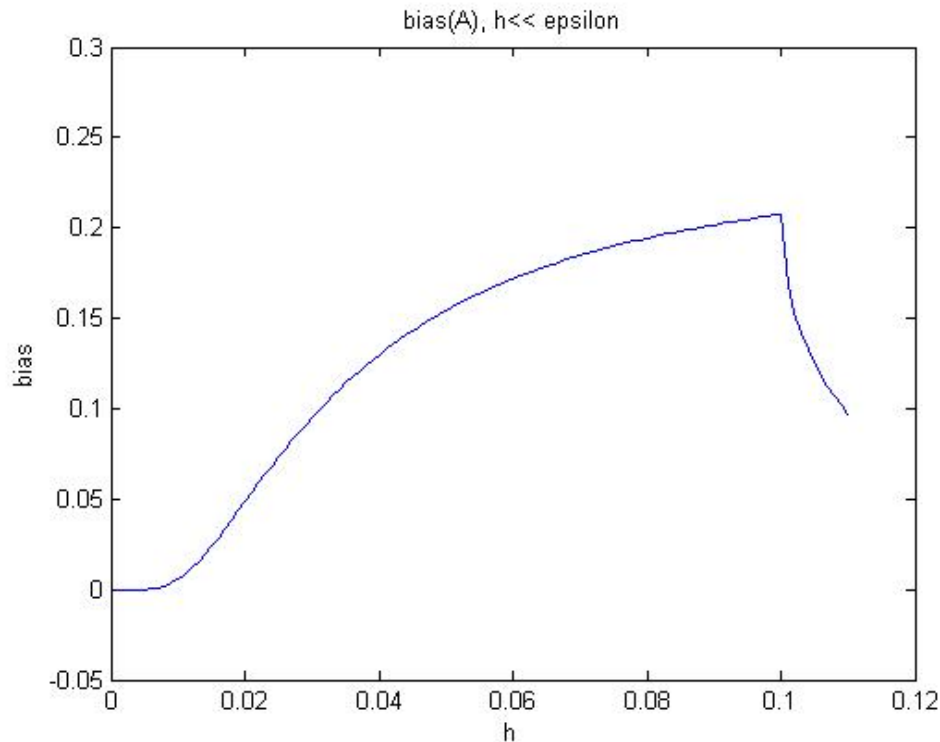
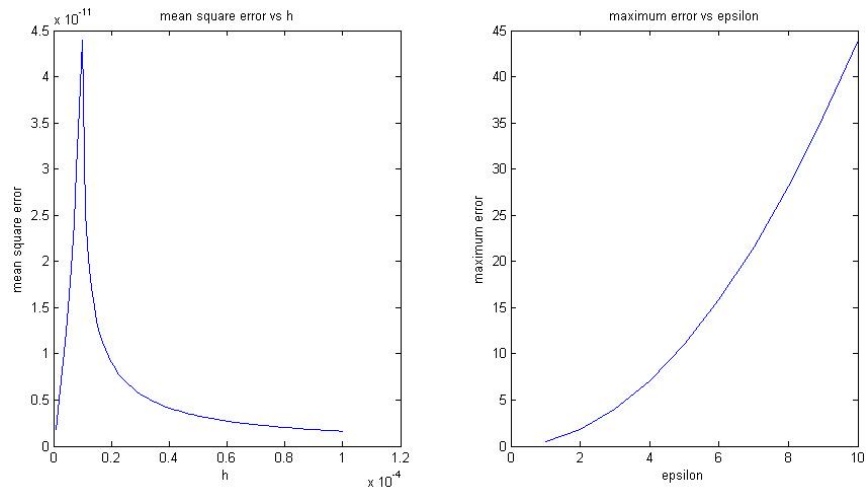


Figure 5.7: bias of the hybrid scheme vs h , $\epsilon = 0.1$, $h = 0 : 0.11$

expected since with our chosen criterion, there is a higher chance of using Milstein instead of Euler as ϵ becomes smaller. We notice an interference at some stage in which case the loglog plot is no longer a straight line; in particular the best result with a nice straight line was obtained at $\epsilon = 10^{-9}$ with h at least 100 times larger than ϵ and mean square order of $h^{2.9404}$. (see Figure 5.14). Using Milstein's result, (5.21), half of 2.9404 is 1.4702, which would give roughly the global order of convergence of the hybrid scheme as 0.97199. Figure 5.16 compares the numerically determined orders of convergence of standard Euler and Milstein schemes with that of the hybrid scheme using the problem in (3.12), where we have used $\epsilon = 0.0000003$. The plot at the bottom of Figure 5.16, a reference line (dotted) of slope 0.8 is used to numerically estimate the order of convergence of the Euler-Milstein hybrid method. That is, in this case we have $\alpha = 0.8$. A least square fit test of the error $C = \Delta t^q$ for the three methods gives; $q = 0.52, 0.96$ and 0.71 for the Euler, Milstein and the Euler-Milstein

Figure 5.8: Mean square error vs h , $\epsilon = 0.00001$

hybrid schemes respectively.

So far in this chapter, we presented the idea of a hybrid numerical scheme for the solution of stochastic differential equations. In particular, we were able to establish roughly the strong order of the mean square error of the Euler-Milstein hybrid scheme. Since the hybrid scheme is anticipating, in the sense that it looks into the future before choosing a method, we investigated the biasedness of the hybrid scheme where we establish the condition under which the hybrid scheme is locally sufficiently unbiased, as long as $h \gg \epsilon$.

In the following chapter, we perform numerical experiments in order to determine, numerically, the payoff of the Euler-Milstein hybrid scheme.

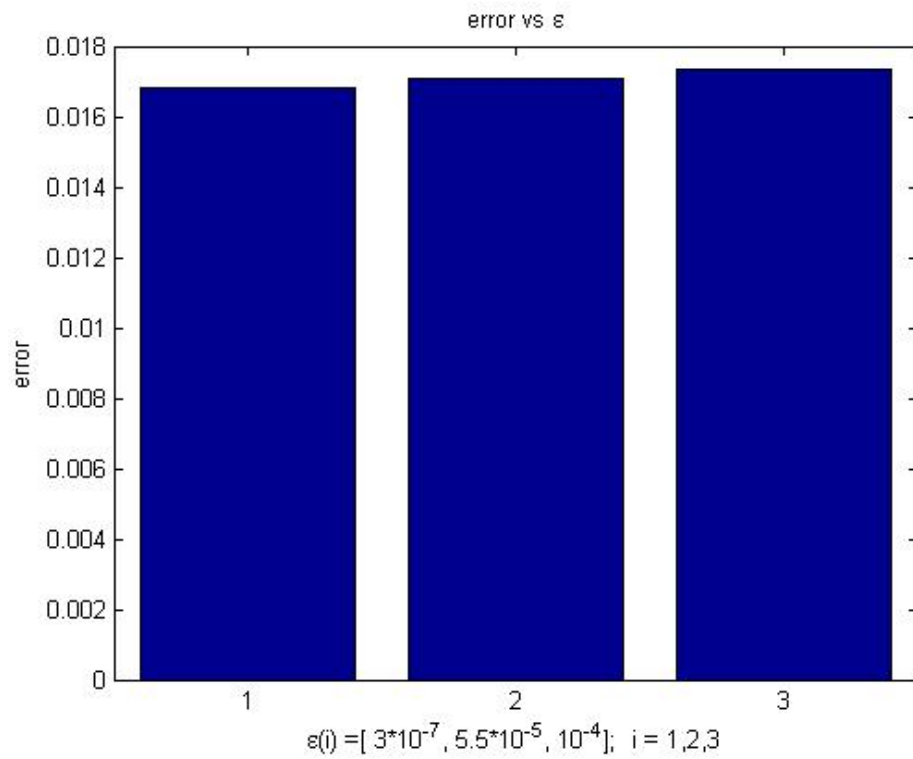


Figure 5.9: Numerically calculated global hybrid error vs ϵ , $\epsilon = [3 \cdot 10^{-6}, 5.5 \cdot 10^{-5}, 0.0001]$, $h = 0.001$

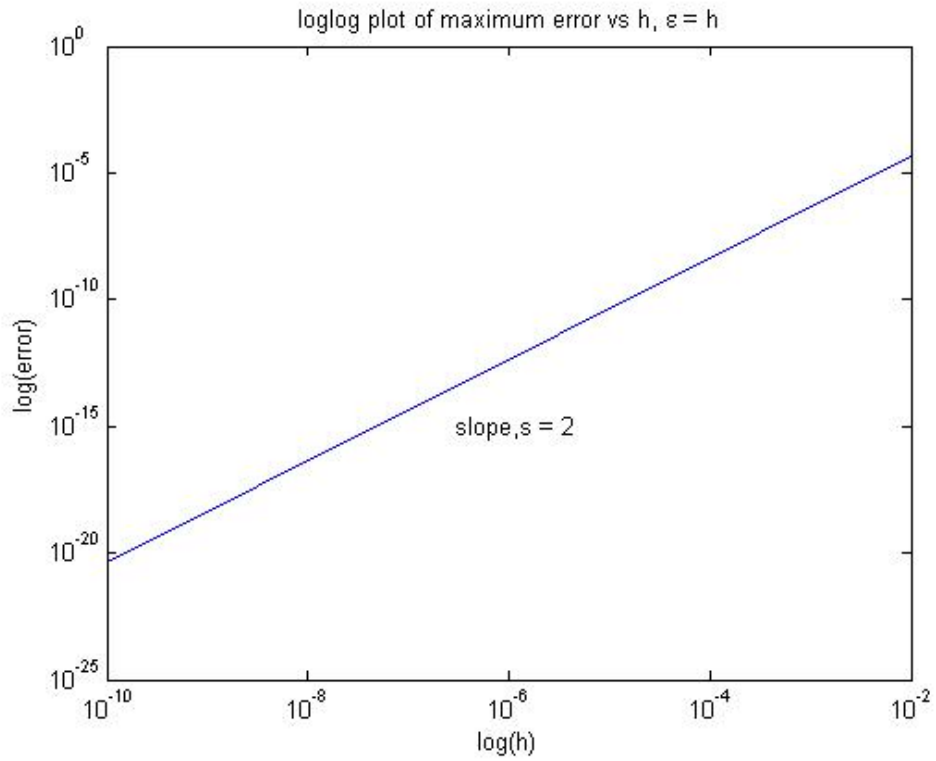


Figure 5.10: Maximum mean square error vs h, $\epsilon = 0.00001$

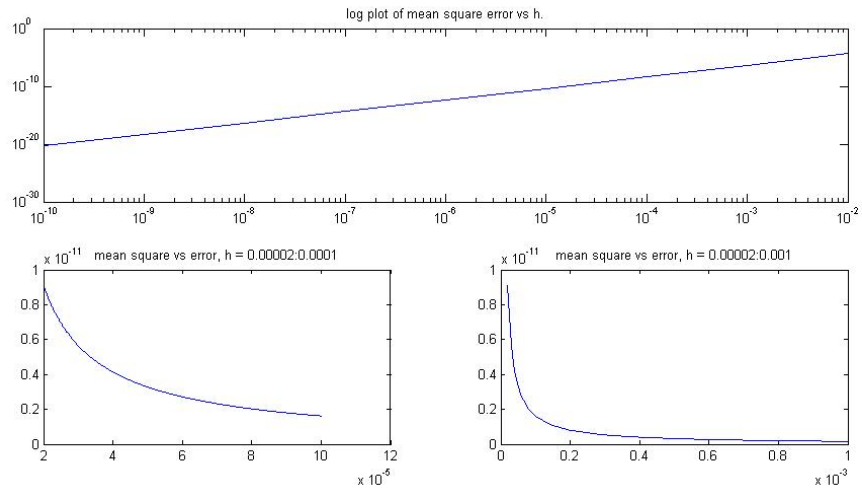


Figure 5.11: Mean square error of the extra term vs h, $\epsilon = 0.00001$

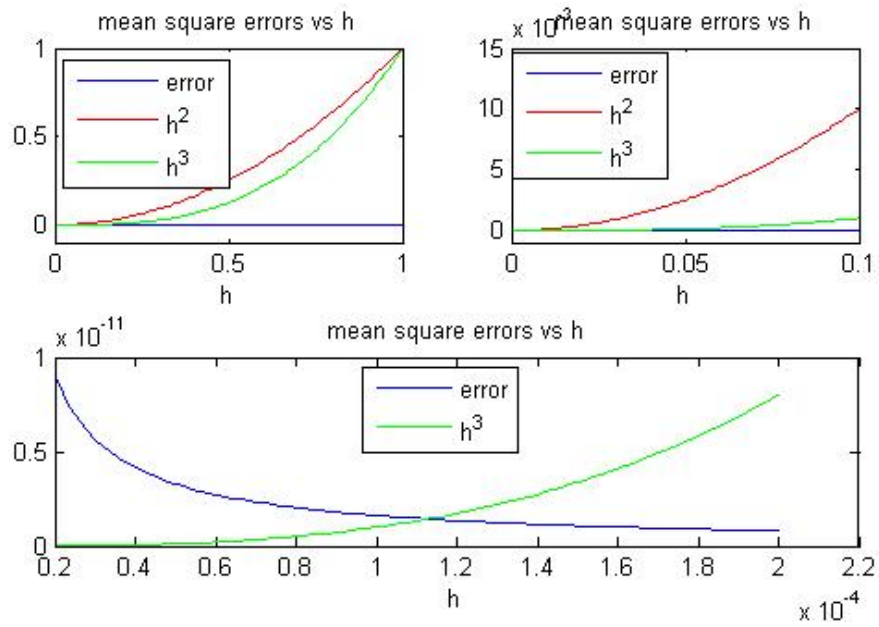


Figure 5.12: Mean square error of the extra term vs h , from left,right to bottom $\epsilon = 0.1, 0.01, 0.00001$

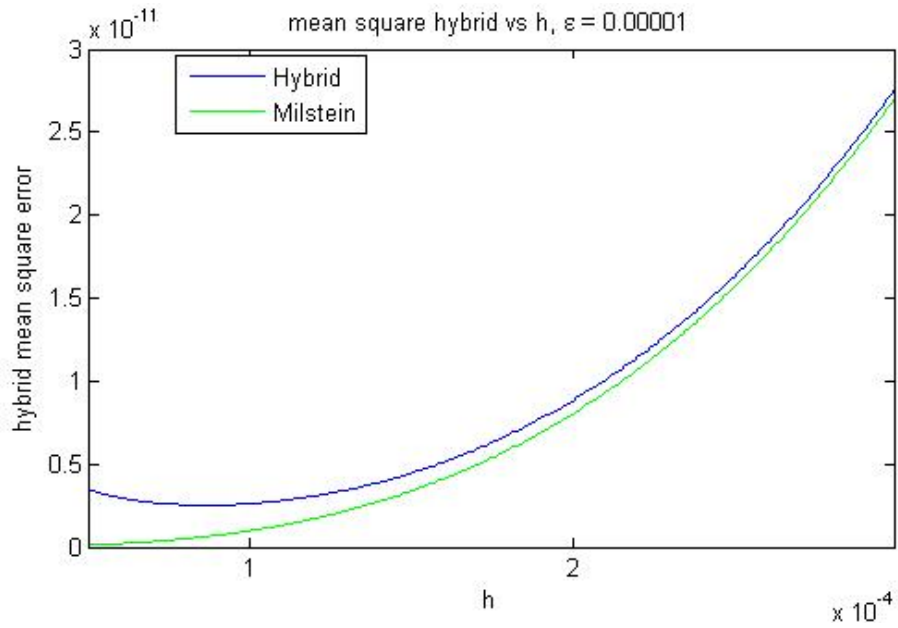


Figure 5.13: Mean square error of the extra term vs h , $\epsilon = 0.00001$

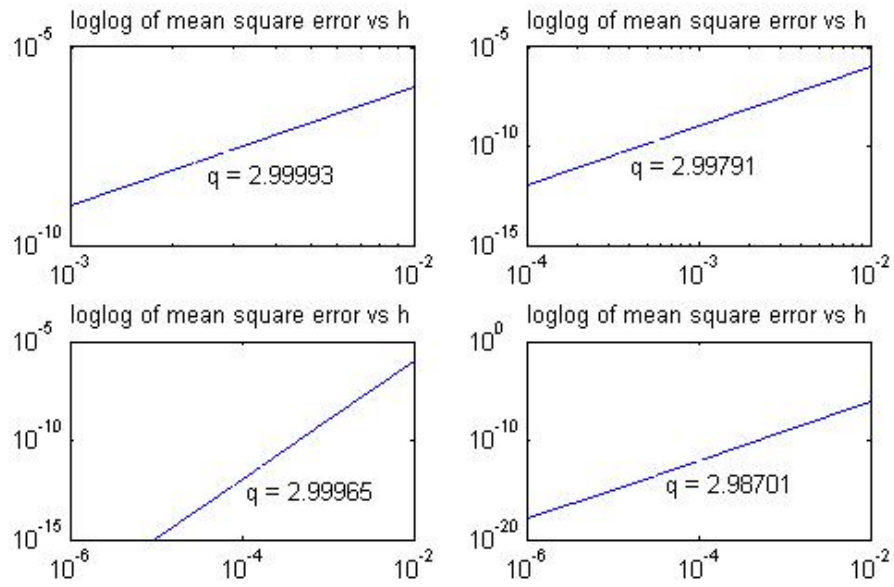
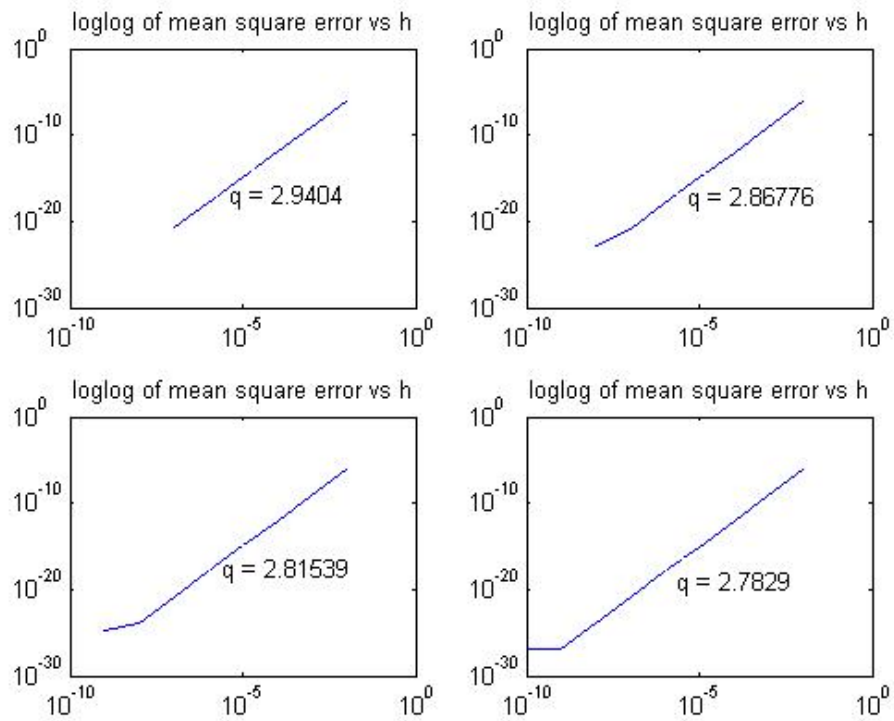


Figure 5.14: Order of hybrid method vs h, for different values of ϵ

Figure 5.15: Order of the hybrid method vs h , for different values of ϵ

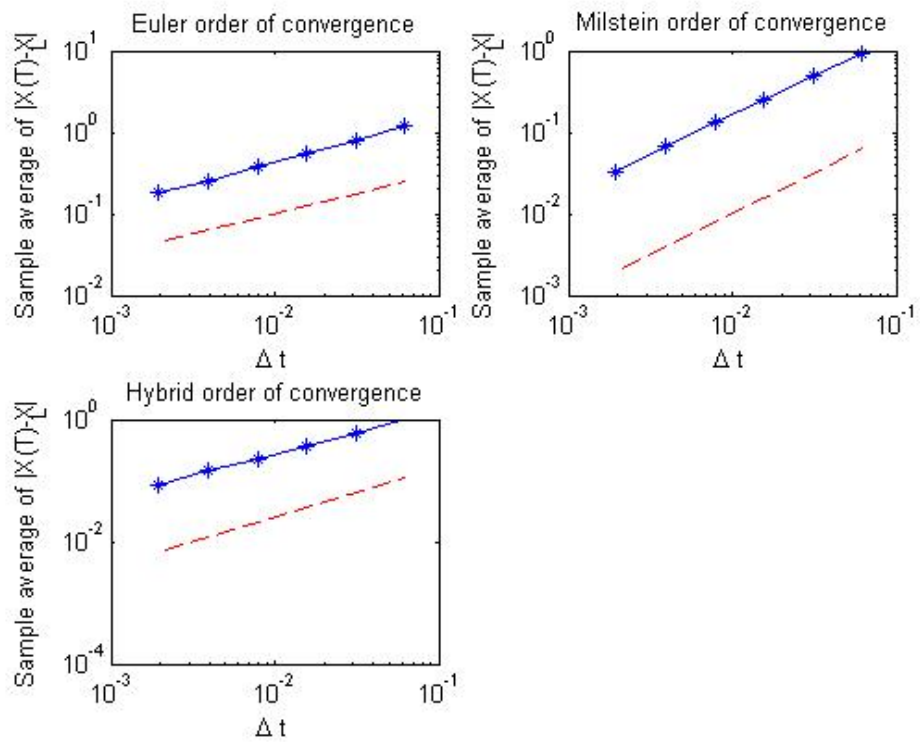


Figure 5.16: Order of convergence for different methods

Chapter 6

Numerical Experiments

6.1 The ϵ -Complexity of the Hybrid Scheme

The ϵ -complexity, $\text{comp}(\epsilon)$, of a numerical problem is the minimal computational cost to solve the problem with error at most ϵ [10]. In this dissertation, the error shall be the mean of the absolute value of the difference between the numerical solution and the exact solution, i.e. $(|X_T - \hat{X}_T|^2)^{\frac{1}{2}}$ and the problem is the pathwise approximation of stochastic differential equations. For a particular method and trajectory the cost of solving a particular problem is determined by the following quantities:

- The number of evaluations of the drift and diffusion coefficients as well as their derivatives.
- The number of Brownian increment ΔB that are generated.

We shall compare the computational cost of solving some SDEs with known explicit solutions using Euler-Maruyama, the Milstein and the hybrid schemes for a particular error estimate.

6.2 Numerical Comparison

We now analyze the performance of the Euler-Milstein hybrid scheme for stochastic differential equations as presented in the previous chapters. To determine the quality of performance of the hybrid scheme, we compute the error at time $T = 1$ referring to a particular path of the Brownian motion and the exact solution. In what follows, we will use the number of calculated steps as a measure of the computational cost. Obviously, the computational cost of the methods is a constant factor for each calculated step for the Euler and Milstein methods, but will depend on the Brownian increment ΔB for the hybrid method.

To determine the payoff of the hybrid scheme, for a fixed cost, we compare the accuracy of each of the schemes where they use the same number of function evaluation. For this, we need to compute the number of steps which leads to a given number of functional evaluation for each method. To be more specific, if we use N steps for the Euler method, we use $\frac{N}{2}$ for the Milstein since the Euler method uses one function evaluation per step while the Milstein method uses two per step. For the hybrid scheme, we find N_{hyb} such that $\mathbb{E}(\text{number of function evaluations}) = N$. Since in the hybrid method we use Euler in each step if $\Delta B \in \Lambda$ and Milstein if $\Delta B \in \Lambda^C$, we should have

$$\begin{aligned}
 N &= N_{hyb}[1 \cdot \mathbb{P}(\Lambda) + 2 \cdot \mathbb{P}(\Lambda^C)] \\
 &= N_{hyb}[(1 - \mathbb{P}(\Lambda^C)) + 2\mathbb{P}(\Lambda^C)] \\
 &= N_{hyb}[1 + \mathbb{P}(\Lambda^C)]
 \end{aligned} \tag{6.1}$$

which implies that $\frac{N}{2} \leq N_{hyb} \leq N$. This gives us the equivalent number of steps used by the Euler-Milstein hybrid scheme, which enables us to make a direct comparison of the hybrid method to the standard Euler and Milstein methods.

To obtain more information about the performance of the hybrid scheme, we study the numerical solution of three stochastic differential equations of which the

exact solutions can be calculated [26, 29]. Specifically, we shall consider the three SDEs used by [29] to compare the performance of the schemes proposed by [30] and [21].

The SDEs are as listed below.

1. A linear homogeneous SDE with constant coefficient

$$dX_t = \mu X_t dt + \sigma X_t dB_t, X_0 = 1, \quad (6.2)$$

i.e (3.12), whose solution is given by the geometric Brownian motion, (3.13)

2. An autonomous SDE with trigonometric drift and diffusion

$$dX_t = -\sin(X_t)\cos^3(X_t)dt + \cos^2(X_t)dB_t, X_0 = 0 \quad (6.3)$$

with the solution

$$X_t = \arctan(B_t + \tan(X_0))$$

3. An autonomous SDE with polynomial drift and diffusion

$$dX_t = -(1 + X_t)(1 - X_t^2)dt + (1 - X_t^2)dB_t, X - 0 = 0 \quad (6.4)$$

with the solution

$$X_t = \frac{(1 + X_0) \exp(-2t + 2B_t) + X_0 - 1}{(1 + X_0) \exp(-2t + 2B_t) + 1 - X_0}$$

6.2.1 Strong Orders of Convergence

We start by studying the comparative strong order of convergence of the hybrid scheme once again using the SDEs described above. We categorize the SDEs into linear and non-linear. We present the results with respect to the strong order of convergence, q for the linear equation in Figure 6.1. and the non-linear equations

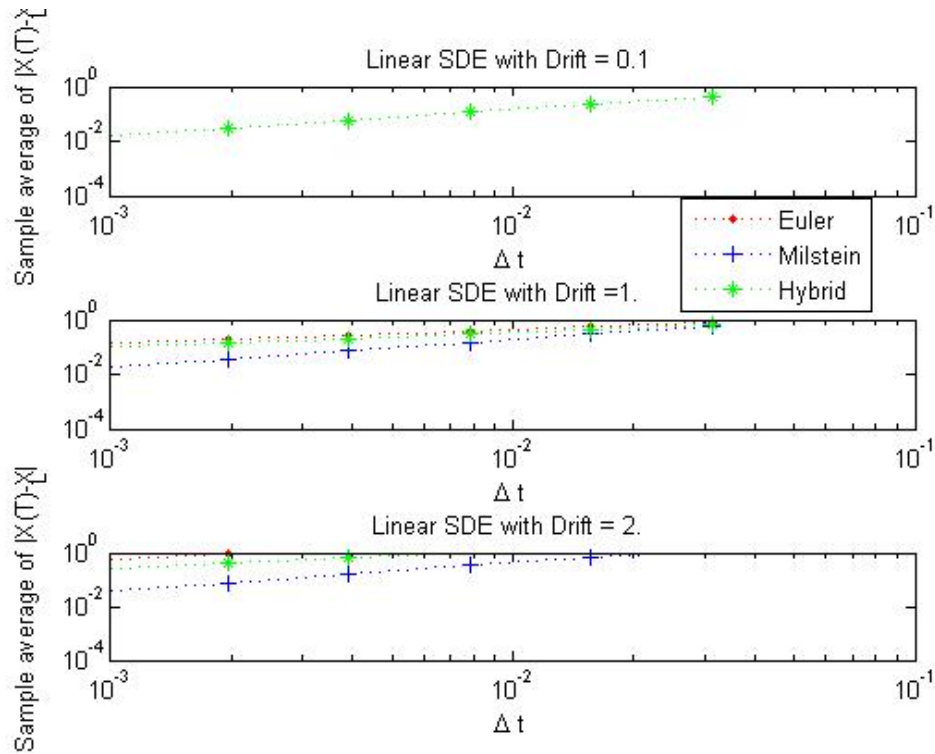


Figure 6.1: Comparison of the strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$ 3. SDE(1) with $\mu = 1$ and $\sigma = 2$. In the calculation we use $\epsilon = 0.000054$ and $h = 0.001$, for Euler and $h = 0.02$ for Milstein

in Figure 6.2. To show explicitly the approximate value of the strong order of convergence of the schemes for the given problems, we carry out a least squares fit that give rise to the plots in Figure 6.3 and Figure 6.4. As can be seen in Figure 6.3, we first look at the nearly deterministic case of SDE(1), with $\mu = 1$ and $\sigma = 0.1$. The three schemes converge with the same order $q \approx 0.9$, which is expected as the Milstein scheme corresponds to the Euler scheme and our hybrid scheme is just an optimal combination of the two schemes.

For the increasing stochastic influence, with $\mu = 1$ and $\sigma = 1$, the Euler and Milstein schemes maintain strong order of convergence of 0.5 and 0.9 respectively while for the hybrid scheme we have a strong convergence order of 0.6. As the

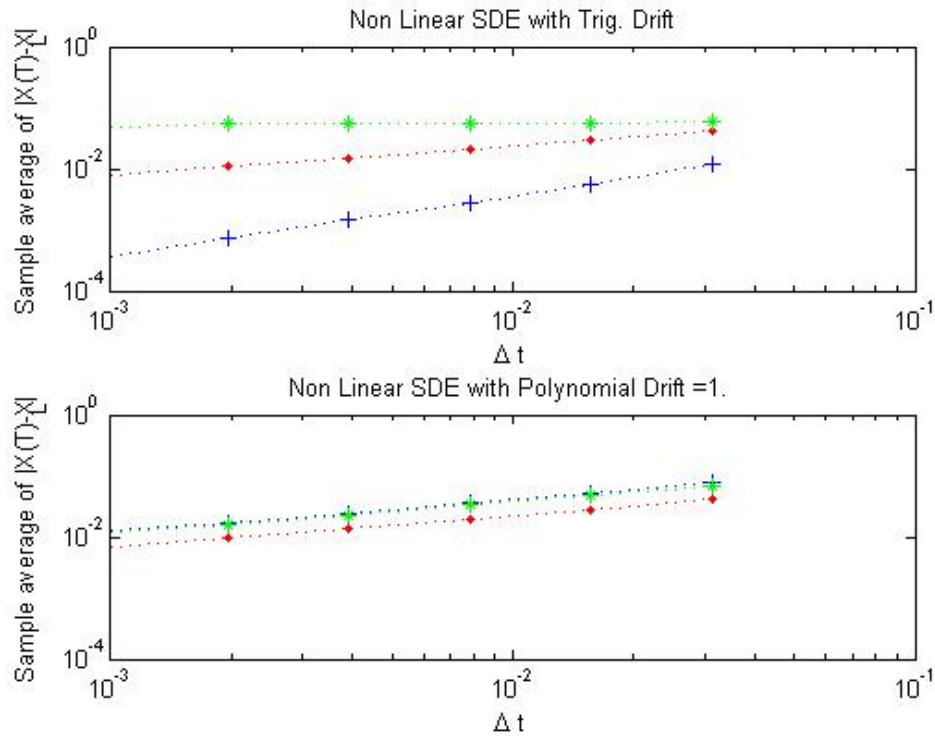


Figure 6.2: Comparison of the strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(2) 2. SDE(3)

stochastic influence dominates SDE(1), with $\mu = 1$ and $\sigma = 2$, the Milstein scheme maintains a strong order of convergence of 0.9 while the Euler and the hybrid scheme converge with 0.42 and 0.48 respectively, as in Figure 6.3.

We conclude by looking at the non-linear SDEs(2) and (3). As shown in Figure (6.4), the Euler-Milstein hybrid scheme converges with strong order 0.58 for the first non-linear SDE with trigonometric coefficients. The Euler and Milstein schemes maintain their respective strong orders of 0.5 and 1. It is important to note that the strong order of convergence of the hybrid scheme is consistent with the order of convergence of the parent schemes. In each case, the strong order of convergence q of the hybrid scheme lies between the orders of convergence of the Euler and the Milstein schemes. The hybrid scheme maintains the same level of consistency even with the second non-linear SDE(3), where the three scheme Euler, hybrid and Milstein have

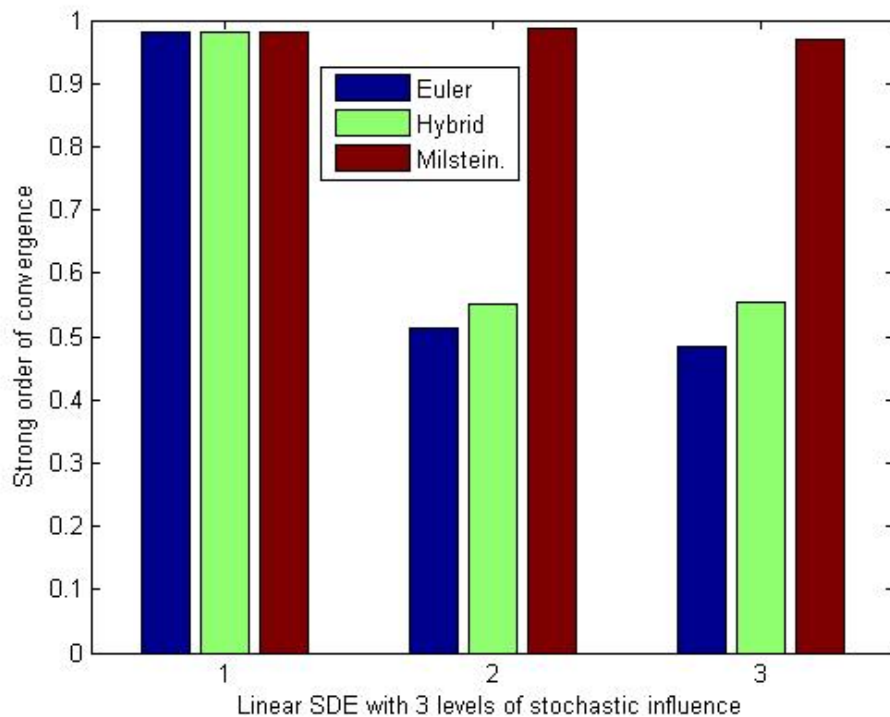


Figure 6.3: Comparison of strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$. 3. SDE(1) with $\mu = 1$ and $\sigma = 2$.

almost the same orders of convergence $q = 0.57$.

6.2.2 Efficiency

Of more importance than the order of convergence is the efficiency (accuracy vs. cost). As in [29], we studied the performance of the Euler-Milstein hybrid scheme with respect to different levels of stochastic influence. Hence, we consider constant drift, $\mu = 1$ with increasing diffusion coefficients $\sigma = 0.1, 1$ and 2 for the linear SDE. We study the second and the third SDEs to get an idea of the performance of the hybrid scheme with non-linear equations. For each of the three SDEs above, we proceed as follows. We specify a number of time steps for the Euler method and use the rela-

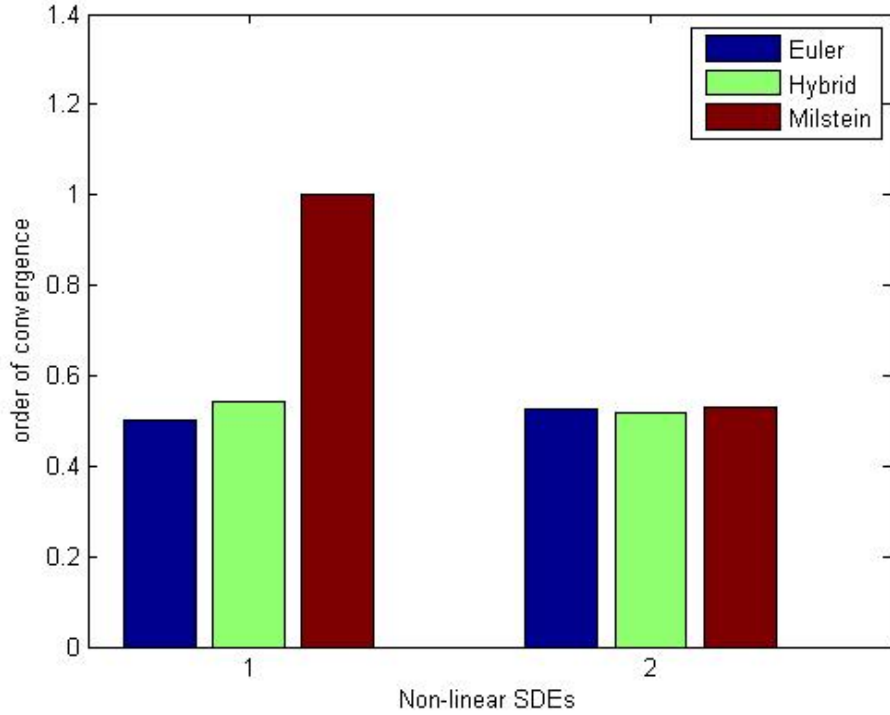


Figure 6.4: Comparison of strong order of convergence of the Euler and Milstein schemes with the hybrid scheme. 1. SDE(2) 2. SDE(3)

tionship in (6.2) to determine the equivalent number of time steps for the hybrid and Milstein schemes. This provides a natural means of comparing the accuracy of the three schemes. We first calculate the Euler approximation, $\bar{X}_T^k, k = 1, \dots, 5000$ using N time steps. Using our chosen criterion we numerically calculate the probability in (6.2) from which we calculate the equivalent number of steps, N_{hyb} to be used for the hybrid scheme. Then we proceed to calculate the approximation, $\bar{X}_T^l, l = 1, \dots, 5000$ using N_{hyb} time steps. We perform similar calculations for the Milstein approximation as well, using $\frac{N}{2}$ steps. With the terminal value of the exact solutions of the examined SDEs we obtain the mean error for each of the schemes as

$$e = \frac{1}{5000} \sum_{i=0}^{5000} (X_T - \bar{X}_T^i) \text{ where } i = k, l, m$$

for Euler, hybrid and Milstein schemes respectively. All the calculations used $\epsilon = 0.000054$, $h = 0.00054$ and $h = 0.001$ for Euler, $h = 0.02$ for Milstein and for the hybrid, h is determined using 6.2.

The results of the above computations are shown in Figures (6.5), (6.6) and (6.7). The mean error of 5000 calculations based on different Brownian motion paths are plotted for the hybrid scheme and for Euler and Milstein schemes which serve as references for every described SDE. Thus we compare the mean error of the hybrid scheme for the cases of nearly equal level of computational effort or cost for each studied SDE. The linear SDE results are shown in Figure 6.5. From the result, we see that there is no significant improvement for the almost deterministic case of the linear SDE. i.e, when $\sigma = 0.1$. However, for increasing levels of stochastic influences, $\sigma = 1$ and $\sigma = 2$, we see that the hybrid scheme has lower mean error compared to the Euler and the Milstein schemes. For the second SDEs, the non-linear one with trigonometric coefficients, the Euler-Milstein hybrid scheme still performs better than the Euler method but only slightly better than the Milstein scheme (see Figure 6.6). Finally for the third SDE with polynomial drift and diffusion coefficients, the hybrid scheme again performs significantly better than both the Euler and Milstein schemes, with very low mean error as shown in Figure 6.7.

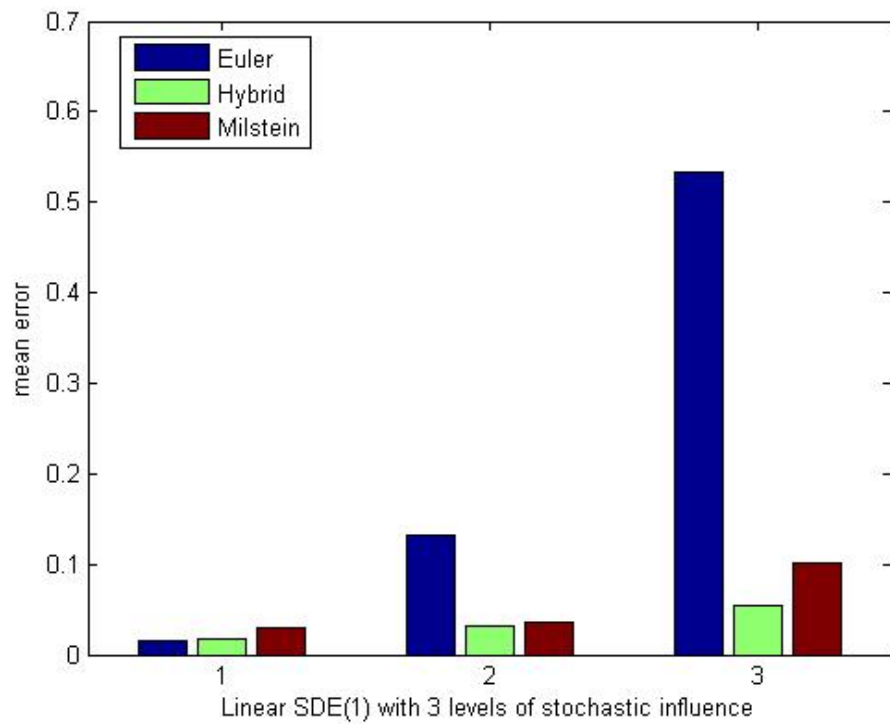


Figure 6.5: Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost. 1. SDE(1) with $\mu = 1$ and $\sigma = 0.1$ 2. SDE(1), with $\mu = 1$ and $\sigma = 1$. 3.SDE(1) with $\mu = 1$ and $\sigma = 2$.

,

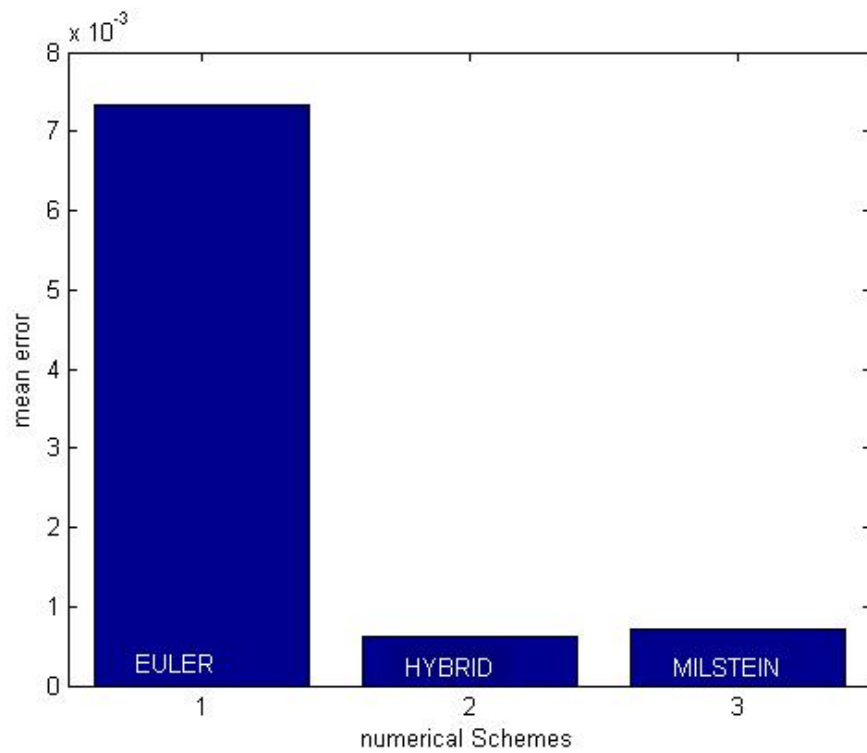


Figure 6.6: Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost, (SDE(2)).

,

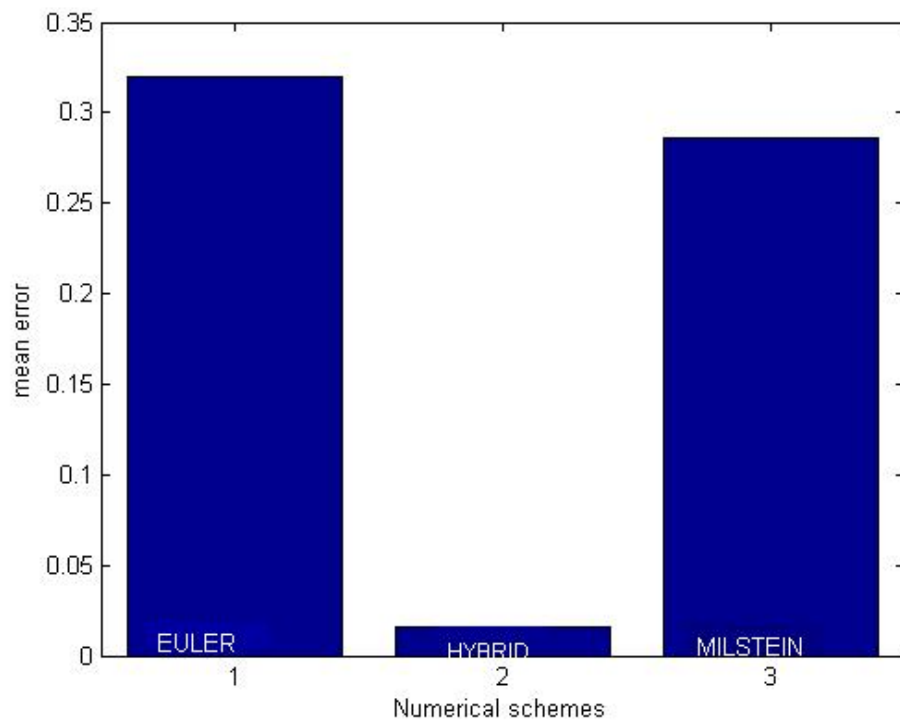


Figure 6.7: Comparison of the Euler and Milstein schemes with the hybrid scheme using the same computational cost, (SDE(3)).

,

Chapter 7

Summary

In Chapter 1, we started by introducing the real life applications of stochastic differential equations. We further emphasized the need for numerical solutions (schemes) for SDEs. We concluded this chapter by giving a brief outline of the structure of the dissertation.

In Chapter 2, we introduced the general theory of stochastic differential equations which is a necessary concept to understand in order for an average reader to have a firm grasp of the entire research work. We defined stochastic processes and more specifically, we stated the main properties of Brownian motion as well as those of Itô processes. We also briefly discussed the Itô and Stratonovich integrals and gave their respective numerical approximations from Riemannian integral point of view. This was followed by the statement of the famous Itô formula, we went on to introduce the concept of SDEs and their strong solutions. We stated without proof, the existence and uniqueness theorem of solutions of stochastic differential equations. This was necessary as it specifies the class of SDEs we are trying to solve by our method.

In Chapter 3, we introduced some numerical methods for the solution of stochastic differential equations by showing how they are derived from truncating the stochastic Itô-Taylor expansion; specifically, we derived the Euler, Milstein and the order 1.5 schemes and investigate numerically their strong order of convergence. This paved the

way for a brief discussion of adaptive schemes, and this was the theme of in chapter 4. We introduce the concept of adaptive schemes for SDEs by first presenting the ODE counterparts. In particular we did a thorough review of the most recent adaptive schemes in the literature. Precisely we reviewed the adaptive schemes of Lamba et al, Hoffman, Rapoo and Mauthner respectively and clearly stated how these adaptive schemes differ from the hybrid scheme of chapter 5.

We started Chapter 5 by introducing the concept of a hybrid scheme. Since our hybrid scheme is based on the simulated Brownian motion increment over an interval. The method is anticipating, as it looks at the value of the Brownian motion over an interval before choosing a method. This necessitated us to introduced the concept of biasedness. More specifically, we developed a hybrid scheme based on Euler and Milstein schemes which we referred to as the ‘Euler-Milstein hybrid scheme’. Both biasedness and strong order of convergence of this scheme was investigated.

Finally, in Chapter 6 we conducted a comparative numerical analysis to investigate the performance of the Euler-Milstein hybrid scheme. An investigation on the order of convergence of the hybrid scheme reveals that the hybrid scheme is consistent with respect to strong order of convergence as the order of convergence across different levels of stochastic influence and for both linear and non-linear SDEs is between those of the Euler and Milstein methods. Also, for equal computational cost, we compared the accuracy, measured by the size of mean absolute errors, of the Euler and Milstein schemes to the accuracy of the hybrid scheme. The comparison shows that for different levels of stochastic influence on a linear SDE, the Euler-Milstein hybrid scheme has a better accuracy than both Euler and the Milstein methods at the same level of computational cost. We have thus proven that the new hybrid scheme is an improvement on the parent (Euler and Milstein) schemes as it gives better accuracy at the same average cost, and is thus more efficient.

In this dissertation, we have developed a hybrid scheme based on the Euler and Milstein schemes which we referred to as the Euler-Milstein hybrid scheme. It is hoped that the general idea of path-dependent hybrid schemes as introduced in this

dessertation can be used to improve the efficiency, if not hte order of convergence, of low=order numerical methods for SDEs. In the future, the extension of the work this dissertation to higher order stochastic numerical schemes,to Runge-Kutta type schemes, also incorporating iterated path integrals, should be of interest. One still also need to embark on a thorough investigation the order of the error for the hybrid method. Questions of stability and perhaps higher dimensional case would be very exciting and hopefully attainable. Finally, we note that the choice of the set Λ in this dessertation is not neccesary an optimal one; further investigations into this are under way.

Appendix A

MATLAB CODES FOR HYBRID METHOD PLOTS

A.1 Matlab Codes: Hybrid Bias

```
%function g = G(x). Plot of G(x) vs epsilon
h = 0.0001;
epsil = 0.0001*h:0.0001*h: 0.2243*h;
N = length(epsil);
bias = zeros(1,N);
for i = 1:N
x = sqrt(epsil(i)-h):(sqrt(epsil(i)+h))/100:sqrt(epsil(i)+h);
g = @(x)-0.5*((x.^2-h)/(h*sqrt(2*pi*h))).*exp(-x.^2/(2*h));
bias(i) = 2*quad(g,sqrt(epsil(i)-h),sqrt(epsil(i)+h));

end

plot(epsil,bias)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

% THIS FUNCTION PLOTS THE THE FUNCTION G_h(x) AND C_1(x)
T =1; N = 2^(11); dt = T/N;
h = dt;
x =[-4:0.1:4];
G1 = ((x.^2-1)/(sqrt(2*pi*1))).*exp(-x.^2/(2*1));
subplot(2,1,1);
C1 = 1/(sqrt(2*pi))*x.*exp(-x.^2/2);
max1 = 1*1/(sqrt(2*pi))*exp(-1/2);

plot(x,G1)
subplot(2,1,2);
plot(x,C1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Fix both h and epsilon
h = 0.01;
epsil =0.000001;
N = length(epsil);
bias = zeros(1,N);
for i = 1:N
x = sqrt(h-epsil(i)):(sqrt(epsil(i)+h))/100:sqrt(epsil(i)+h);
g = @(x)-((x.^2-h)/(h*sqrt(2*pi*h))).*exp(-x.^2/(2*h));

bias(i) = quad(g,sqrt(h-epsil(i)),sqrt(epsil(i)+h));

end
plot(epsil,bias)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fix epsilon, vary h
h = 0:0.01:10;

```

```

epsil =[0.9,2];
N1 = length(h);
N2 = length(epsil);
bias = zeros(N1,N2);
%x = zeros(1,N2);
for j = 1:N2
    for i = 1:N1
        if epsil(j) < h(i)
x = sqrt(h(i)-epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
g = @(x)-((x.^2-h(i))/(h(i)*sqrt(2*pi*h(i))))).*exp(-x.^2/(2*h(i)));
bias(j,i) = quad(g,sqrt(h(i)-epsil(j)),sqrt(epsil(j)+h(i)));

            else
x = -sqrt(h(i)+epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
g = @(x)-0.5*((x.^2-h(i))/(h(i)*sqrt(2*pi*h(i))))).*exp(-x.^2/(2*h(i)));
bias(j,i) = quad(g,-(sqrt(h(i)+epsil(j))),sqrt(epsil(j)+h(i)));
            end
        end
    end
subplot(2,2,[1 3]);plot(h,bias(1,:))
subplot(2,2,[2 4]);plot(h,bias(2,:))

max1 = max(bias(1,:))
Max2 = max(bias(2,:))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fix epsilon vary h
h= 0:0.001:0.01;
epsil =0.000001;
%h = n*epsil;

```



```

h = 0:1:10;
epsil =0:1:10;
N1 = length(h);
N2 = length(epsil);
VI = zeros(N1,N2);
VI_2 = zeros(1,N1);
%x = zeros(1,N2);
for j = 1:N2
    for i = 1:N1
        if epsil(j) < h(i)
x = sqrt(h(i)-epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
g = @(x)2*(((x.^2-h(i))).^2/(sqrt(2*pi*h(i))).*exp(-x.^2/(2*h(i))));
VI(j,i) = quad(g,sqrt(h(i)-epsil(j)),sqrt(epsil(j)+h(i)));

                elseif epsil(j) == h(i)
x = -sqrt(h(i)+epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
g = @(x)((x.^2-h(i))).^2/(sqrt(2*pi*h(i))).*exp(-x.^2/(2*h(i))));
VI(j,i) = quad(g,-(sqrt(h(i)+epsil(j))),sqrt(epsil(j)+h(i)));
VI_2(i) = VI(j,i);
                else
                    x = -sqrt(h(i)+epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
g = @(x)((x.^2-h(i))).^2/(sqrt(2*pi*h(i))).*exp(-x.^2/(2*h(i))));
VI(j,i) = quad(g,-(sqrt(h(i)+epsil(j))),sqrt(epsil(j)+h(i)));
                    end
                end
            end
        end
    end
    %subplot(2,2,[1 3]);plot(h,VI(1,:))
    subplot(2,2,[2 4]);plot(h,VI_2)
    % max1 = max(bias(1,:))

```

```

% Max2 = max(bias(2,:))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fix epsilon, vary h
n = 2:1:10;
h = 10.^(-n);
epsil = 0.000000000001;
N1 = length(h);
N2 = length(epsil);
VI = zeros(N1,N2);
VI_2 = zeros(1,N1);
%x = zeros(1,N2);
for j = 1:N2
    for i = 1:N1
        if epsil(j) < h(i)
            x = sqrt(h(i)-epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
            g = @(x)2*(((x.^2-h(i))).^2/(sqrt(2*pi*h(i))))).*exp(-x.^2/(2*h(i)));
            VI(j,i) = quad(g,sqrt(h(i)-epsil(j)),sqrt(epsil(j)+h(i)));
        else
            x = -sqrt(h(i)+epsil(j)):(sqrt(epsil(j)+h(i)))/100:sqrt(epsil(j)+h(i));
            g = @(x)((x.^2-h(i))).^2/(sqrt(2*pi*h(i))))).*exp(-x.^2/(2*h(i)));
            VI(j,i) = quad(g,-(sqrt(h(i)+epsil(j))),sqrt(epsil(j)+h(i)));
        end
    end
end

%subplot(2,2,1);plot(h,[h.^3 + VI(1,:)], 'b-',h,h.^3, 'g-')
subplot(2,2,4);
loglog((h),[h.^3 + VI(1,:)], 'b-')
%%% Least square fit of error = C *Dt^q%%%
A1 = [ones(length(h),1),log(h)']; rhs = log([h.^3 + VI(1,:)']);

```

```

sol = A1\rhs
q1 = sol(2)
resid = norm(A1*sol - rhs)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55

```

A.3 Matlab Codes: Numerical Experiments

#1.

```

% THIS FUNCTIONS SOLVES THE LINEAR SDE WITH SIGMA = 0.1, USING EULER METHOD
% AND COMPUTES THE ERROR.

```

```

function [L2,ave1] = EulerComp(M)
randn('state',sum(100*clock))

lambda = 2; mu = 0.1; Xzero = 1;
T =1; N = 2^(11); dt = T/N;          % problem parameters
Xerr = zeros(M,1);
count1 = zeros(1,1);
count2 = zeros(1,1);

for s = 1:M,
    dW = sqrt(dt)*randn(1,N);        %Brownian increment
    W = cumsum(dW);                  % discrete Brownian path
    Xtrue = Xzero *exp((lambda-0.5*mu^2) + mu*W(end));
    for p = 2:2
        R = 2^(p-1); Dt = R*dt; L = N./R;
        Xtemp = Xzero;
        Xtemp3 = Xzero;
        test = dW.^2-Dt;
        for j = 1:L
            Winc(j) = sum(dW(R*(j-1)+1:R*j));
            Xtemp = Xtemp + Dt*lambda*Xtemp + mu * Xtemp *Winc(j);

```



```

        if abs(test(j)) < abs(mean(test))
            count1(p-1) = count1(p-1) + 1;
        else
            count2(p-1) = count2(p-1)+1;
        end
    end
    Xerr(s,(p-1)) = abs(Xtemp-Xtrue); % store the error at t= 1
end
end
count1;
count2;
prob_A = count1/(count1 + count2);
N_hyb = L/(1+ prob_A);
L2 = round(N_hyb);
for i = 1:1
ave(i) = mean(Xerr(:,i)); var1(i) = var(Xerr(:,i));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#2.
% THIS FUNCTION SOLVES THE LINEAR SDE USING MISTEIN SCHEME AND OBTAIN THE
% ERROR. AND ALSO CALCULATE THE EQUIVALENT NUMBER OF STEPS TO BE USED IN
% THE HYBRID SCHEME.
function [L2,mave1] = MilstComp(M)
randn('state',sum(100*clock))
lambda = 2; mu = 0.1; Xzero = 1;
T =1; N = 2^10; dt = T/N;          % problem parameters
Xerr = zeros(M,1);
count1 = zeros(1,1);
count2 = zeros(1,1);

```

```

for s = 1:M,
    dW = sqrt(dt)*randn(1,N);          %Brownian increment
    W = cumsum(dW);                    % discrete Brownian path
    Xtrue = Xzero *exp((lambda-0.5*mu^2) + mu*W(end));
    for p = 2:2
        R = 2^(p-1); Dt = R*dt; L = N./R;
        Xtemp2 = Xzero;
        test = dW.^2-Dt;
        for j = 1:L
            Winc(j) = sum(dW(R*(j-1)+1:R*j));
            Xtemp2 = Xtemp2 + Dt*lambda*Xtemp2 + mu*Xtemp2*Winc(j) + ...
                0.5* mu^2* Xtemp2*(Winc(j)^2 - Dt);
            if abs(test(j)) < abs(mean(test))
                count1(p-1) = count1(p-1) + 1;
            else
                count2(p-1) = count2(p-1)+1;
            end
        end
        Xerr(s,(p-1)) = abs(Xtemp2-Xtrue); % store the error at t= 1
    end
end
count1;
count2;
prob_A = count2/(count1 + count2);
N_hyb = L/(1+prob_A);
L2 = round(N_hyb)
for i = 1:L2
    ave(i) = mean(Xerr(:,i)); var1(i) = var(Xerr(:,i));
end

```

```

mave1 = ave;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#3.
% THIS FUNCTION SOLVES THE FIRST NON-LINEAR SDE WITH TRIG. COEFFICIENTS
% USING EULER METHOD.
function [L2,ave1] = EulerComp7(M)
randn('state',sum(100*clock))
Xzero = 0;
T =1; N = 2^(11); dt = T/N;          % problem parameters
Xerr = zeros(M,1);
count1 = zeros(1,1);
count2 = zeros(1,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N);        %Brownian increment
    W = cumsum(dW);                  % discrete Brownian path
    B_T = (W(end)+ tan(Xzero));
    Xtrue =atan(B_T);
    for p = 2:2
        R = 2^(p-1); Dt = R*dt; L = N./R;
        Xtemp = Xzero;
        test = dW.^2-Dt;
        for j = 1:L
            Winc(j) = sum(dW(R*(j-1)+1:R*j));
            Xtemp = Xtemp + Dt*(-sin(Xtemp)*cos(Xtemp)^3) + ...
            cos(Xtemp)^2*Winc(j);
            if abs(test(j)) < abs(mean(test))
                count1(p-1) = count1(p-1) + 1;
            else
                count2(p-1) = count2(p-1)+1;
            end
        end
    end
end

```

```

        end
    end
    Xerr(s,(p-1)) = abs(Xtemp-Xtrue); % store the error at t= 1
end
end
count1;
count2;
prob_A = count2/(count1 + count2);
N_hyb = L/(1+ prob_A);
L2 = round(N_hyb);
for i = 1:L2
ave(i) = mean(Xerr(:,i)); var1(i) = var(Xerr(:,i));
end
ave1 = ave;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
#4
% THIS FUNCTION SOLVES THE FIRST NON-LINEAR SDE USING MILSTEIN METHOD.
function [L2,ave1] = MilstComp7(M)
randn('state',sum(100*clock))
Xzero = 0;
T =1; N = 2^10; dt = T/N; % problem parameters
Xerr = zeros(M,1);
count1 = zeros(1,1);
count2 = zeros(1,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N); %Brownian increment
    W = cumsum(dW); % discrete Brownian path
    B_T = (W(end)+ tan(Xzero));
    Xtrue =atan(B_T);

```



```

#5
% THIS FUNCTION SOLVES THE FIRST SDE USING THE HYBRID METHOD.
function [ave2,ave3,milave1] = EulerComp8(M)
randn('state',sum(100*clock))
Xzero = 0;
[L2, ave1] = EulerComp7(M);
[Lm2,mave1] = MilstComp7(M);
ave2 = ave1;
N = L2; T = 1; dt = T/N;          % problem parameters
Xerr3 = zeros(M,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N);      %Brownian increment
    W = cumsum(dW); % discrete Brownian path
    B_T = (W(end)+ tan(Xzero));
    Xtrue =atan(B_T);
        L = N;
        Xtemp3 = Xzero;
        test = dW.^2-dt;
        for j = 1:L
            Winc(j) = sum(dW(1*(j-1)+1:1*j));
            if abs(test(j)) < abs(mean(test))
                Xtemp3 = Xtemp3 + dt*(-sin(Xtemp3)*cos(Xtemp3)^3) + ...
                    cos(Xtemp3)^2*Winc(j);
            else
                Xtemp3 = Xtemp3 + dt*(-sin(Xtemp3)*cos(Xtemp3)^3) +...
                    cos(Xtemp3)^2*Winc(j)+ 0.5*cos(Xtemp3)^2*...
                    (-2*cos(Xtemp3)*sin(Xtemp3))*(Winc(j)^2 - dt);
            end
        end
    end
end

```

```

        Xerr3(s,1) = abs(Xtemp3-Xtrue); % store the error at t= 1
    end
    for i = 1:1
        ave3(i) = mean(Xerr3(:,i)); %var3(i) = var(Xerr3(:,i));
    end
    err7 =ave2;
% ave2
    err8 = ave3;
    milave1 = mave1;
    err7;
    err8;
    merr4 = milave1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#6
%THIS FUNCTION SOLVES THE NON-LINEAR SDE WITH POLINOMIAL COEFFICIENTS USING
%EULER METHOD.
function [L2,ave1] = EulerComp9(M)
    randn('state',sum(100*clock))
    Xzero = 0;
    T =1; N = 2^(11); dt = 2*T/N;          % problem parameters
    Xerr = zeros(M,1);
    count1 = zeros(1,1);
    count2 = zeros(1,1);
    for s = 1:M,
        dW = sqrt(dt)*randn(1,N);          %Brownian increment
        W = cumsum(dW);                     % discrete Brownian path
        Xtrue =(((1+Xzero)*exp(-2*T + 2*W(end))+Xzero-1)/((1+Xzero)*exp(-2*T + ...
        2*W(end))+1-Xzero));
        for p = 2:2

```

```

R = 2^(p-1); Dt = R*dt; L = N./R;
Xtemp = Xzero;
test = dW.^2-Dt;
for j = 1:L
    Winc(j) = sum(dW(R*(j-1)+1:R*j));
    Xtemp = Xtemp + Dt*(-(1+Xtemp)*(1-Xtemp^2)) + ...
    (1-Xtemp^2)*Winc(j);
    if abs(test(j)) < abs(mean(test))
        count1(p-1) = count1(p-1) + 1;
    else
        count2(p-1) = count2(p-1)+1;
    end
end
Xerr(s,(p-1)) = abs(Xtemp-Xtrue); % store the error at t= 1
end
end
count1;
count2;
prob_A = count2/(count1 + count2);
N_hyb = L/(1+ prob_A);
L2 = round(N_hyb);
for i = 1:L2
ave(i) = mean(Xerr(:,i)); var1(i) = var(Xerr(:,i));
end
ave1 = ave;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#7
%THIS FUNCTION SOLVES THE NON-LINEAR SDE WITH POLINOMIAL COEFFICIENTS USING
%MILSTEIN METHOD.

```



```

function [L2,ave1] = MilstComp9(M)
randn('state',sum(100*clock))
    Xzero = 0;
T =1; N = 2^10; dt = T/N;          % problem parameters
Xerr = zeros(M,1);
count1 = zeros(1,1);
count2 = zeros(1,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N);      %Brownian increment
    W = cumsum(dW);                % discrete Brownian path
    Xtrue =((1+Xzero)*exp(-2*T + 2*W(end))+Xzero-1)/((1+Xzero)*exp(-2*T + ...
    2*W(end))+1-Xzero);
    for p = 2:2
        R = 2^(p-1); Dt = R*dt; L = N./R;
        Xtemp3 = Xzero;
        test = dW.^2-Dt;
        for j = 1:L
            Winc(j) = sum(dW(R*(j-1)+1:R*j));
            Xtemp3 = Xtemp3 + dt*(-(1+Xtemp3)*(1-Xtemp3^2)) ...
            + (1-Xtemp3^2)*Winc(j)+0.5*(-2*Xtemp3)*(Winc(j)^2 - dt);
            if abs(test(j)) < abs(mean(test))
                count1(p-1) = count1(p-1) + 1;
            else
                count2(p-1) = count2(p-1)+1;
            end
        end
    end
    Xerr(s,(p-1)) = abs(Xtemp3-Xtrue); % store the error at t= 1
end
end
end

```

```

count1;
count2;
prob_A = count2/(count1 + count2);
N_hyb = L/(1+prob_A);
L2 = round(N_hyb);
for i = 1:L2
ave(i) = mean(Xerr(:,i)); var1(i) = var(Xerr(:,i));
end
ave1 = ave;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#8
%THIS FUNCTION SOLVES THE SECOND NON-LINEAR SDE USING THE HYBRID
%METHOD.
function [ave2,ave3,milave1] = EulerComp10(M)
randn('state',sum(100*clock))
Xzero = 0;
[L2, ave1] = EulerComp9(M);
[Lm2,mave1] = MilstComp9(M);
ave2 = ave1;
N = L2; T = 1; dt = T/N; % problem parameter
Xerr3 = zeros(M,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N); %Brownian increment
    W = cumsum(dW); % discrete Brownian path
    Xtrue = (((1+Xzero)*exp(-2*T + 2*W(end)))+(Xzero-1))/(((1+Xzero)*exp(-2*T + ...
    2*W(end)))+(1-Xzero));
    L = N;
    Xtemp3 = Xzero;
    test = dW.^2-dt;

```

```

    for j = 1:L
        Winc(j) = sum(dW(1*(j-1)+1:1*j));
        if abs(test(j)) < abs(mean(test))
            Xtemp3 = Xtemp3 + dt*(-(1+Xtemp3)*(1-Xtemp3^2)) + ...
                (1-Xtemp3^2)*Winc(j);
        else
            Xtemp3 = Xtemp3 + dt*(-(1+Xtemp3)*(1-Xtemp3^2)) ...
                + (1-Xtemp3^2)*Winc(j)+0.5*(-2*Xtemp3)*(Winc(j)^2 - dt);
        end
    end
    Xerr3(s,1) = abs(Xtemp3-Xtrue); % store the error at t= 1
end
for i = 1:1
    ave3(i) = mean(Xerr3(:,i)); %var3(i) = var(Xerr3(:,i));
end
err9 = ave2;
milave1 = mave1;
err10 = ave3;
merr5 = milave1;
err9;
err10;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#9
%THIS FUNCTION CALCULATES THE STRONG ORDER OF CONVERGENCE OF THE THREE SDEs FOR
%SDE.
function [q1 q2 q3]= conv1(M)
    randn('state',sum(100*clock))
    lambda = 2; mu = 0.1; Xzero = 1;
    T =1; N = 2^(11); dt = T/N; % problem parameters

```

```

%M = 5000;                                % number of paths sampled
Xerr = zeros(M,5);                        % preallocate array sample over discrete Brownian paths
count1 = zeros(6,1);
count2 = zeros(6,1);
for s = 1:M,
    dW = sqrt(dt)*randn(1,N);             %Brownian increment
    W = cumsum(dW);                       % discrete Brownian path
    Xtrue = Xzero *exp((lambda-0.5*mu^2) + mu*W(end));
    for p = 2:7
        R = 2^(p-1); Dt = R*dt; L = N/R;
        % L Euler steps of size Dt = R*dt
        Xtemp = Xzero;
        Xtemp2 = Xzero;
        Xtemp3 = Xzero;
        test = dW.^2-Dt;
        for j = 1:L
            Winc = sum(dW(R*(j-1)+1:R*j));
            Xtemp = Xtemp + Dt*lambda*Xtemp + mu * Xtemp *Winc;
            Xtemp2 = Xtemp2 + Dt*lambda*Xtemp2 + mu*Xtemp2*Winc + 0.5* mu^2*...
                Xtemp2*(Winc^2 - Dt);
            if abs(test(j)) < abs(mean(test))
                count1(p-1) = count1(p-1) + 1 ;
                Xtemp3 = Xtemp3 + Dt*lambda*Xtemp3 + mu * Xtemp3 *Winc;
            else
                Xtemp3 = Xtemp3 + Dt*lambda*Xtemp3 + mu*Xtemp3*Winc + 0.5* ...
                    mu^2* Xtemp3*(Winc^2 - Dt);
                count2(p-1) = count2(p-1)+1;
            end
        end
    end
end

```



```

% THIS FUNCTION CALCULATES THE STRONG ORDER OF CONVERGENCE OF THE THREE
% SCHEMES FOR THE FIRST NON-LINEAR SDE.
function [q1 q2 q3]= conv4(M)
    randn('state',sum(100*clock))
    lambda = 2; mu = 2; Xzero = 0;
    T =1; N = 2^(11); dt = T/N;          % problem parameters
    %M = 5000;                          % number of paths sampled
    Xerr = zeros(M,5); % preallocate array sample over discrete Brownian paths
    count1 = zeros(6,1);
    count2 = zeros(6,1);
    for s = 1:M,
        dW = sqrt(dt)*randn(1,N);      %Brownian increment
        W = cumsum(dW);                 % discrete Brownian path
        B_T = ((W(end)+ tan(Xzero)));
        Xtrue =atan(B_T);
        for p = 2:7
            R = 2^(p-1); Dt = R*dt; L = N/R; % L Euler steps of size Dt = R*dt
            Xtemp = Xzero;
            Xtemp2 = Xzero;
            Xtemp3 = Xzero;
            test = dW.^2-Dt;
            for j = 1:L
                Winc(j) = sum(dW(R*(j-1)+1:R*j));
                Xtemp = Xtemp + Dt*(-sin(Xtemp)*cos(Xtemp)^3) +
                    cos(Xtemp)^2*Winc(j);
                Xtemp2 = Xtemp2 + Dt*(-sin(Xtemp2)*cos(Xtemp2)^3) +...
                    cos(Xtemp2)^2*Winc(j)+0.5*cos(Xtemp2)^2*(-2*cos(Xtemp2)*...
                    sin(Xtemp2))*(Winc(j)^2 - Dt);
                if abs(test(j)) < abs(mean(test))

```

```

        count1(p-1) = count1(p-1) + 1 ;
        Xtemp3 = Xtemp3 + Dt*(-sin(Xtemp3)*cos(Xtemp3)^3) +...
        cos(Xtemp3)^2*Winc(j);
    else
        Xtemp3 = Xtemp3 + Dt*(-sin(Xtemp3)*cos(Xtemp3)^3) + ...
        cos(Xtemp3)^2*Winc(j)+0.5*cos(Xtemp3)^2*(-2*cos(Xtemp3)*...
        sin(Xtemp3))*(Winc(j)^2 - Dt);
        count2(p-1) = count2(p-1)+1;
    end
end
end
Xerr(s,(p-1)) = abs(Xtemp-Xtrue); % store the error at t= 1
Xerr2(s,(p-1)) = abs(Xtemp2 - Xtrue);
Xerr3(s,(p-1)) = abs(Xtemp3-Xtrue);
end
end
count1;
count2;
for i = 1:6
ave1(i) = mean(Xerr(:,i)); %var1(i) = var(Xerr(:,i));
ave2(i) = mean(Xerr2(:,i)); %var2(i) = var(Xerr2(:,i));
ave3(i) = mean(Xerr3(:,i)); %var3(i) = var(Xerr3(:,i));
end
Dtvals = dt*(2.^([1:6])); % top LH picture
format short
ave1;
ave2;
ave3;
%%% Least square fit of error = C *Dt^q%%%
A1 = [ones(6,1),log(Dtvals)']; rhs = log(ave1');

```

```

A2 = [ones(6,1),log(Dtvals)']; rhs2 = log(ave2');
A3 = [ones(6,1),log(Dtvals)']; rhs3 = log(ave3');
sol = A1\rhs; q1 = sol(2);
sol2 =A2\rhs2; q2 = sol2(2);
sol3 = A3\rhs3; q3 = sol3(2);
resid = norm(A1*sol - rhs);
resid2 = norm(A2*sol2 - rhs2);
resid3 = norm(A3*sol3 - rhs3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#11
% THIS FUNCTION CALCULATES THE STRONG ORDER OF CONVERGENCE OF THE THREE
% SCHEMES FOR THE SECOND NON-LINEAR SDE.
function [q1 q2 q3]= conv5(M)
    randn('state',sum(100*clock))
    lambda = 2; mu = 2; Xzero = 0;
    T =1; N = 2^(11); dt = T/N;          % problem parameters
    Xerr = zeros(M,5);    % preallocate array sample over discrete Brownian paths
    count1 = zeros(6,1);
    count2 = zeros(6,1);
    for s = 1:M,
        dW = sqrt(dt)*randn(1,N);      %Brownian increment
        W = cumsum(dW);                 % discrete Brownian path
        Xtrue =(((1+Xzero)*exp(-2*T + 2*W(end))+Xzero-1)/((1+Xzero)*exp(-2*T + ...
        2*W(end))+1-Xzero);
        for p = 2:7
            R = 2^(p-1); Dt = R*dt; L = N/R; % L Euler steps of size Dt = R*dt
            Xtemp = Xzero;
            Xtemp2 = Xzero;
            Xtemp3 = Xzero;

```



```

test = dW.^2-Dt;
for j = 1:L
    Winc(j) = sum(dW(R*(j-1)+1:R*j));
    Xtemp = Xtemp + Dt*(-(1+Xtemp)*(1-Xtemp^2)) + (1-Xtemp^2)*Winc(j);
    Xtemp2 = Xtemp2 + Dt*(-(1+Xtemp2)*(1-Xtemp2^2)) ..
+ (1-Xtemp2^2)*Winc(j)+0.5*(-2*Xtemp2)*(Winc(j)^2 - Dt);
    if abs(test(j)) < abs(mean(test))
        count1(p-1) = count1(p-1) + 1 ;
        Xtemp3 = Xtemp3 + Dt*(-(1+Xtemp3)*(1-Xtemp3^2)) + ...
(1-Xtemp3^2)*Winc(j);
    else
        Xtemp3 = Xtemp3 + Dt*(-(1+Xtemp3)*(1-Xtemp3^2)) + ...
(1-Xtemp3^2)*Winc(j)+0.5*(-2*Xtemp3)*(Winc(j)^2 - Dt);
        count2(p-1) = count2(p-1)+1;
    end
end
end
Xerr(s,(p-1)) = abs(Xtemp-Xtrue); % store the error at t= 1
Xerr2(s,(p-1)) = abs(Xtemp2 - Xtrue);
Xerr3(s,(p-1)) = abs(Xtemp3-Xtrue);
end
end
count1;
count2;
for i = 1:6
ave1(i) = mean(Xerr(:,i)); %var1(i) = var(Xerr(:,i));
ave2(i) = mean(Xerr2(:,i)); %var2(i) = var(Xerr2(:,i));
ave3(i) = mean(Xerr3(:,i)); %var3(i) = var(Xerr3(:,i));
end
Dtvals = dt*(2.^[1:6]); % top LH picture

```

```

format short
ave1;
ave2;
ave3;
%%% Least square fit of error = C *Dt^q%%%
A1 = [ones(6,1),log(Dtvals)']; rhs = log(ave1');
A2 = [ones(6,1),log(Dtvals)']; rhs2 = log(ave2');
A3 = [ones(6,1),log(Dtvals)']; rhs3 = log(ave3');
sol = A1\rhs; q1 = sol(2);
sol2 =A2\rhs2; q2 = sol2(2)
sol3 = A3\rhs3; q3 = sol3(2);
resid = norm(A1*sol - rhs);
resid2 = norm(A2*sol2 - rhs2)
resid3 = norm(A3*sol3 - rhs3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
#12
% THIS FUNCTION PLOTS ORDER OF CONVERGENCE FOR THE SCHEMES AND FOR THE
% THREE LEVELS OF STOCHASTIC INFLUENCE.
function convplot1(M)
T =1; N = 2^(11); dt = T/N;
Dtvals = dt*(2.^([1:6]));
[ave1,ave2,ave3] = conv_1(M);
[ave4,ave5,ave6] = conv_2(M);
[ave7,ave8,ave9] = conv_3(M);
subplot(3,1,1);
loglog(Dtvals, ave1, 'r:'),hold on
loglog(Dtvals, ave2, 'b:'),hold on
loglog(Dtvals, ave3, 'g:'),hold off
axis([1e-3 1e-1 1e-4 1]);

```

```

ave1;
ave2;
ave3;
    xlabel('\Delta t'), ylabel('Sample average of |X(T)-X_L|')
    title('Linear SDE with Drift = 0.1 ', 'FontSize', 10)
    subplot(3,1,2);
    loglog(Dtvals, ave4, 'r:'), hold on
loglog(Dtvals, ave5, 'b:'), hold on
loglog(Dtvals, ave6, 'g:'), hold off
axis([1e-3 1e-1 1e-4 1]);
    xlabel('\Delta t'), ylabel('Sample average of |X(T)-X_L|')
    title('Linear SDE with Drift = 1. ', 'FontSize', 10)
    subplot(3,1,3);
    loglog(Dtvals, ave7, 'r:'), hold on
loglog(Dtvals, ave8, 'b:'), hold on
loglog(Dtvals, ave9, 'g:'), hold off
axis([1e-3 1e-1 1e-4 1]);
    xlabel('\Delta t'), ylabel('Sample average of |X(T)-X_L|')
    title('Linear SDE with Drift = 2. ', 'FontSize', 10)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#13
% THIS FUNCTION PLOTS THE ORDR OF CONVERGENCE FOR THR THREE SCHEMES AND FOR
% THE TWO NON-LINEAR SDEs
function convplot2(M)
T = 1; N = 2^(11); dt = T/N;
Dtvals = dt*(2.^([1:6]));
[ave1,ave2,ave3] = conv_4(M);
[ave4,ave5,ave6] = conv_5(M);
%[ave7,ave8,ave9] = conv3(M);

```

```

subplot(2,1,1);
loglog(Dtvals, ave1, 'r:'),hold on
loglog(Dtvals, ave2, 'b:'),hold on
loglog(Dtvals, ave3, 'g:'),hold off
axis([1e-3 1e-1 1e-4 1]);
ave1
ave2
ave3
%loglog(Dtvals,(Dtvals.^(.5)), 'r--'), hold off % reference slope of 1/2
xlabel('\Delta t'), ylabel('Sample average of |X(T)-X_L|')
title('Non Linear SDE with Trig. Drift','FontSize',10)
subplot(2,1,2);
loglog(Dtvals, ave4, 'r:'),hold on
loglog(Dtvals, ave5, 'b:'),hold on
loglog(Dtvals, ave6, 'g:'),hold off
axis([1e-3 1e-1 1e-4 1]);
xlabel('\Delta t'), ylabel('Sample average of |X(T)-X_L|')
title('Non Linear SDE with Polynomial Drift =1. ','FontSize',10)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#14

% THIS FUNCTION GIVES THE BAR PLOTS TO ERROR OF THE LINEAR SDE
function barplot3(M)
[err1,err2,merr1] = EulerComp2(M);
[err3,err4,merr2] = EulerComp4(M);
[err5, err6,merr3] = EulerComp6(M);
errMat = [err1,err2,merr1;err3,err4,merr2;err5,err6,merr3];
figure;
bar(errMat,'group')

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
#15
```

```
% THIS FUNCTION GIVES THE BAR PLOTS OF THE ERROR FOR THE TWO NON-LINEAR SDEs
```

```
function nonbarplot3(M)
```

```
[err7,err8,merr4] = EulerComp8(M);
```

```
[err9, err10,merr5] = EulerComp10(M);
```

```
errMat = [err7,err8,merr4;err9, err10,merr5];
```

```
figure;
```

```
bar(errMat,'group')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
#16
```

```
% THIS FUNCTION GIVES THE BAR PLOTS FOR THE TWO NON-LINEAR SDEs
```

```
function nonbarplot3(M)
```

```
[err7,err8,merr4] = EulerComp8(M);
```

```
[err9, err10,merr5] = EulerComp10(M);
```

```
errMat = [err7,err8,merr4;err9, err10,merr5];
```

```
figure;
```

```
bar(errMat,'group')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
#17
```

```
% THIS FUNCTION GIVES THE BAR PLOTS OF THE STRONG ORDER OF CONVERGENCE OF
```

```
% THE THREE SCHEMES FOR THE TWO NON-LINEAR SDEs
```

```
function nonconvbarplot3(M)
```

```
[q10,q11,q12] = conv4(M);
```

```
[q13,q14,q15] = conv5(M);
```

```
convMat = [q10,q12,q11;q13,q15,q14];
```

```
figure;
```

```
bar(convMat,'group')
```

Bibliography

- [1] BRAIN, D. H. (1997). *Essential Matlab: for Scientists and Engineers*, John Wiley and sons Inc, New York.
- [2] BURRAGE, K. BURRAGE, P. AND MITSUI, T. (2000). *Numerical Solutions of Stochastic Differential Equations: implementation and stability issues*, J. Comp.and Appl. Maths, **125**, 171-182.
- [3] BURRAGE, K. AND BURRAGE, P. M., (2000). *Order Conditions of Stochastic Runge-Kutta Methods: by B-Series*, J. Numer. Anal, **38**, (5) 1626-1646.
- [4] BURRAGE, K. AND BURRAGE, P. M. (2004). *Adaptive step size based on control theory for stochastic differential equations*, J. Compt. Appl. Math., **170**, 317-336.
- [5] DAMIEN, L. AND BERNARD, L. (1996). *Introduction to Stochastic Calculus Applied to Finance*, Chapman and Hall , London.
- [6] DORMAND, J., R., PRINCE, P., J. (1980). “*A family of embedded Runge-Kutta Formulae*”, SIAM J. Appl. Maths **6**,(1),19-26.
- [7] FAURE, O. (1990). *Numerical Pathwise Approximation of Stochastic Differential Equations*. L.A.M.M.(CERMA), France.
- [8] FIMA, C. AND KLEBANER (2006). *Introduction to Stochastic Calculus with Application*, 2nd edn. Imperial College Press.

- [9] FRIEDMAN, A. (1975). *Stochastic Differential Equations and Applications*, Academic Press.
- [10] GAINES, J. G. AND LYONS, T. J. (1997). *Variable step-size Control In The Numerical Solutions of Stochastic Differential Equations*, SIAM J. Appl. Maths , 57 1455-1484,
- [11] GARD, T. C. (1987). *Introduction to Stochastic Differential Equation*, chapter 7.,
- [12] GEORGE, L. AND JOHN, P. (1999). *Numerical methods using Matlab*, 2nd edn Prentice-Hall Inc, New Jersey.
- [13] GERALD, C.F. AND PATRICK, O.W. (1999). *Applied Numerical Analysis* , 6th edn. Addison-Wesley Longman Inc.
- [14] GUSTAF, S. (2002). *Automatic control and adaptive time-stepping* , Numer. Algorithms, 31 281-310
- [15] GUSTAF, S. (2006). *Time-step selection algorithm: Adaptivity, control theory and signal processing*, Applied Numerical Mathematics, 56 488-502.
- [16] GUSTAFSSON, K. (1991). *Control theoretical techniques for stepsize selection in explicit Runge-Kutta methods*, Numer. Algorithms, 17 533-554.
- [17] GUSTAFSSON, K. (1991). *Control theoretical techniques for stepsize selection in explicit Runge-Kutta methods*, Numer. Algorithms, 17 533-554.
- [18] GUSTAF, S., WANG, L. (2006). *Evaluating numerical ODE/DAE methods, algorithms and software*, Comp. Appl. Math, 185 244-260.

- [19] GUSTAF, S. AND WANG, L. (2005). *Adaptive time-stepping and computational stability*, Comp. Methods in Sc. ans Eng., 2 3 1-3.
- [20] HIGHAM, D. J. MAO, X. AND STUART, A.M. (2002). *Strong Convergence of Euler-Type methods: For non- linear Stochastic Differential Equations*,SIAM J. Numer. Anal., 40 1041-1063.
- [21] HOFMANN, N. MLLER-GRONBACH, T. AND RITTER, K.(2000). *Optimal Approximation of SDEs: by Adaptive Step-size Control*, maths. Comp. 69 1017-1032.
- [22] HOFMANN, N. MLLER-GRONBACH, T. AND RITTER, K. (2001). *The Optimal Discretization of SDEs*, J. of Complexity, 17 117-153.
- [23] Ikeda, N. and Watanabe, S. (1981). *“Stochastic Differential Equations and Diffusion processes”*, North-Holland, Amsterdam,
- [24] KAMAGAWA, S. (1988). *On the Rate of Convergence for Maruyama’s Approximate of SDEs*, Yokohama mathematical journal vol, 36
- [25] KARATZAS, I., SHREVE, S. (1992). *Brownian motion and stochastic Calculus, 2nd Editon*, Springer , Berlin, New York.
- [26] KLOEDEN, P.E. AND PLATEN, E. (1999). *Numerical Solutions of Stochastic Differential Equations.*, Springer, Berlin.
- [27] KLOEDEN, P.E. AND PLATEN, E. (1991). *Stratonovich and Itô Stochastic Taylor Expansion*, J. maths Nachr, 151 33-50.
- [28] LAMBER, H. MATTINGLY, J.C. AND STUART, A. M. (2006). *An adaptive Euler-Maruyama Scheme for SDEs: Convergence and Stability*, IMA J. Numerical Analysis, 1-28.

- [29] LEHN, J. ROLER, A. AND SCHEIN, O. (2002). *Adaptive Schemes for Numerical Solutions of SDEs: A Comparison*, J. comp. Appl. Maths, 138 297-308.
- [30] MAUTHNER, S. (1998). *Step-size Control in the Approximation of Stochastic Differential Equations*, J. Comp. Appl. Maths 100 93-109.
- [31] MILSTEIN, G.N. (1995). *Numerical Integration of Stochastic Differential Equations*, Kluwer, London.
- [32] NEWTON, N.J. (1991). *Asymptotically efficient Runge-Kutta methods for a class Of It and Stratonovich Equations*, SIAM J. Appl. Math. 2 542-567.
- [33] OKSENDAL, B. (2000). *Stochastic Differential Equations: An introduction with Application.*, Springer, Berlin.
- [34] PETER, L. AND RICHARD, L. C. (2000). *Exploring Numerical Methods: An Introduction to scientific computing Using matlab.*, Jones and Bartlett publishers, Inc. USA.
- [35] PLATEN, E. AND WAGNER, W. (1982). *On a Taylor formular for a class of Itô process*, Prob. Math. statist. (3) 37-51.
- [36] PROTTER, P. (1991). *Stochastic Integration and Differential equations*, springer, New york.
- [37] RICHARD, L.B. AND FAIRES, J.D. (2005). *Numerical Analysis* , 8th edn, Thomson brooks/Cole, USA.
- [38] ROGER, P. (1992). *Stratonovich-Taylor Expansion and numerical Methods*, journal of Stochastic Analysis and applications 10 (5) 603-612.

- [39] ROMISCH, W., WINKLER, R. (2006). *Stepsize control for mean-square numerical methods for stochastic differential equations with small noise.*, SIAM J. Sci. Comput. 28 604-625.
- [40] RAPOO E. (2008). *A Variable step Riemannian Sum for an Itô Integral*, J. Appl. Prob., 45, 551-567.
- [41] RUMELIN, W. (1982). *Numerical treatment of stochastic Differential Equations*, SIAM J. Numer. Anal. 19 604-613.
- [42] STOFFER, G. (1995). *Variable steps for reversible integration methods*, J. of Computing , 55 15-25.
- [43] STUART, A. M. AND HUMPHREIS, A. R. (1995). *The Essential Stability of Local Error Control for Dynamical Systems*, SIAM J. Numer. Anal. 32 1940-1971.
- [44] SZEPESEY, A., TEMPONE R. AND ZOURARIS G. E (2001). *Adaptive weak approximation of stochastic differential equation.*, Commun. Pure and Appl. Math. 54 1169-1214.
- [45] TALAY, D. (1999). *Expansion of the Global error for Numerical Schemes solving stochastic differential Equations*, Stochastic analysis and Applications, 4 483-509.