THE UNIVERSITY
of EDINBURGH

THE UNIVERSITY
of EDINBURGH

First-order Numerical Schemes for Stochastic Differential Equations Using Coupling

Yousef Ali S. Alnafisah

Doctor of Philosophy
University of Edinburgh
October 2015

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Yousef Alnafisah*)

Acknowledgments

I would first like to thank my supervisor, Professor Sandy Davie, for his constant encouragement and suggestions throughout my PhD study. He taught me many interesting things, and I greatly appreciate the extra efforts he made to assist me. I would also like to thank my second supervisor Istvan Gyongy for his support and encouragement over the past years.

I would also like to thank my Mather and my family for their support and encouragement right through my study.

Abstract

Title: First-order Numerical Schemes for Stochastic Differential Equations Using Coupling

We study a new method for the strong approximate solution of stochastic differential equations using coupling and we prove order one error bounds for the new scheme in L^p space assuming the invertibility of the diffusion matrix. We introduce and implement two couplings called the exact and approximate coupling for this scheme obtaining good agreement with the theoretical bound. Also we describe a method for non-invertibility case (*Combined method*) and we investigate its convergence order which will give $O(h^{3/4}\sqrt{|\log(h)|})$ under some conditions. Moreover we compare the computational results for the combined method with its theoretical error bound and we have obtained a good agreement between them. In the last part of this thesis we work out the performance of the multilevel Monte Carlo method using the new scheme with the exact coupling and we compare the results with the trivial coupling for the same scheme.

Contents

Declaration **2**

Acknowledgments **3**

Abstract **4**

0.1 Introduction 8

1 Numerical Solution of Stochastic Differential Equation **11**

1.1 Stochastic Differential Equations(SDEs). 11

1.1.1 Definition: 11

1.2 Existence and uniqueness theorems 12

1.3 Strong and weak convergence for SDEs 14

1.3.1 Strong order of convergence 14

1.3.2 Weak order of convergence 15

1.4 Numerical method for approximating the SDEs 15

1.4.1 Euler-Maruyama scheme 15

1.4.2 The Milstein scheme 16

1.5 Two-level approximation 17

1.5.1 Empirical estimation of the error of a numerical method . . . 18

1.6 The implementation of Euler and Milstein schemes 19

1.6.1 Two-dimensional stochastic differential equation 19

1.7 A modification to (1.21) which gives order one 26

2 Two-level bound using the exact coupling of scheme (1.22) 31

2.1 Bounds using two-level coupling 35

2.1.1 Exact coupling in two-dimensional case 37

2.2 The implementation of exact coupling in two-dimensional case with invertible $\left(b_{ik}(\tilde{x}^{(r+1,2j)})\right)$ 51

3 Approximate coupling for general d 54

3.1 The implementation of an approximate coupling in two-dimensional case with invertible $\left(b_{ik}(\tilde{x}^{(r+1,2j)})\right)$ 64

4 Combined Method 67

4.1 Derivation and implementation of the local error of the Combined Method 68

4.1.1 Evaluation of the local error for the scheme (1.22) with exact coupling 68

4.1.2 Evaluation of the local error of scheme (1.22) with trivial coupling 75

4.2 The implementation of exact coupling with the trivial coupling (combined method) in two-dimensional case with non-invertiblity of $b_{ik}(x)$ 83

4.3 The implementation of approximate coupling with the trivial coupling (combined method) in two-dimensional case with non-invertible $b_{ik}(x)$ 86

5 Multilevel method for weak approximation of SDEs 93

5.1 The multilevel Monte Carlo method for scheme (1.22) with the exact coupling 97

5.2 The multilevel Monte Carlo method for scheme (1.22) with the trivial coupling 99

5.3 The multilevel Monte Carlo method for the combined method 100

5.4 Numerical results of the multilevel method for scheme (1.22) with the exact coupling 100

5.4.1 The multilevel implementation of the exact coupling in two-dimensional case with invertible matrix 101
5.4.2 The multilevel implementation of the trivial coupling in two-dimensional case with invertible matrix 104
5.5 Numerical results of the multilevel method for the combined method . 107
5.5.1 The multilevel implementation of combined method in two-dimensional case with non-invertible matrix 107
5.5.2 The multilevel implementation of the trivial coupling in two-dimensional case with invertible matrix 110
5.6 Appendix 113

0.1 Introduction

In this study we investigate the new method developed by Davie [3] which uses coupling and gives order one for the strong convergence for stochastic differential equations (SDEs). There are many numerical methods for solving SDEs. P.E.Kloeden and E.Platen [4] have described a method based on the stochastic Taylor series expansion but the major difficulty with this approach is that the double stochastic integrals cannot be so easily expressed in terms of simpler stochastic integrals when the Wiener process is multi-dimensional. In the multi-dimensional case the Fourier series expansion of Wiener process has been used to represent the double integrals by [4], [9] and [8] but we need to generate many random variables each time therefore it takes a lot of time to compute and also it is hard to extend to higher order.

We will see in this study a modified interpretation for the normal random variables generated in the Taylor expansion. This method will give order one convergence under a non-degeneracy condition for the diffusion term. In standard methods such as Milstein we generate the approximations for the Taylor expansion terms separately. In the coupling method we will generate the approximation for the Taylor expansion as a combination of random variables. The modification is by replacing the iterated integrals by different random variables but with a good approximation in distribution. Then we will obtain a random vector from the linear term which is a good approximation in distribution to the original Taylor expansion. In each chapter of this thesis we will give supporting results for this method using a Matlab implementation.

There have been many studies using coupling for the numerical solution of Stochastic differential equations. In [10] Kanagawa investigate the rate of convergence in terms of two probability metrics between approximate solutions with i.i.d random variables. Rachev and Ruschendorff [6] in volume 2 developed Kanagawa's method

by using the Komlós, Major and Tusnády theorem in [5]. In [11] Fournier uses the quadratic Vaserstein distance for the approximation of the Euler scheme and the results of Rio [12] which gives a very precise rate of convergence for the central limit theorem in Vaserstein distance. Also Rio in [21] continues his research in [12] for the Vaserstein bound to give precise bound estimates. Under uniform ellipticity, Alfonsi, Jourdain and Kohatsu-Higa [1] and [2] have studied the Vaserstein bound for Euler method and they have proved an $O(h^{(\frac{2}{3}-\epsilon)})$ for one-dimensional diffusion process where h is the step-size and then they generalize the result to SDEs of any dimension with $O(h\sqrt{\log(\frac{1}{h})})$ bound when the coefficients are time-homogeneous. Cruzeiro, Malliavin and Thalmaier [13] get an order one method and under the non-degeneracy they construct a modified Milstein scheme which obtains an order one for the strong approximation. Charlbonneau, Svyrydov and Tupper [14] investigate the Vaserstein bound [7] by using the weak convergence and Strassen- Dudley theorem. Convergence of an approximation to a strong solution on a given probability space was established by Gyöngy and Krylov in [15] using coupling. Davie in [22] applied the Vaserstein bound to solutions of vector SDEs and uses the Komlós, Major and Tusnády theorem to get order one approximation under a non-degeneracy assumption. Alhojilan in his thesis [23] is working in higher order numerical scheme using the coupling method of Davie.

In chapter one we give some background material about standard numerical approaches with their implementation results and coupling. In the second chapter we give the proof of an order one convergence for Davie [3] method in L^p space and the proof will be for the two-dimensional SDEs and then we give the supporting results for the exact coupling using Matlab. In chapter 3, we explain the approximating coupling in more details and then show the implementation for the method for general d. We should indicate that in chapter 2 and 3 we assume the nondegenercacy for the diffusion term. After that, in chapter 4 we investigate the order of the *Combined method* which is for the SDEs which are degenerate at some points. We will show

how we could control the degenerate problem and then give computational results. In the final chapter we do some work on the multilevel Monte Carlo method [16], [17] for the two-dimensional SDEs for the exact coupling, combined method and trivial coupling and after that we compare the output results for these methods.

Chapter 1

Numerical Solution of Stochastic Differential Equation

In this chapter we introduce some basic background material and other materials which are used in later chapters to show the convergence of the scheme that we use in this thesis. Furthermore some discrete time approximation schemes will be mentioned which have been done on numerical approximation of stochastic differential equation and results will be shown with the convergence for them.

1.1 Stochastic Differential Equations(SDEs).

1.1.1 Definition:

let $\{W(t)\}_{t\geq 0}$ be a d-dimensional standard Brownian motion on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$, $a = a(t,x)$ be a d-dimensional vector function(called *drift* coefficient) and $b = b(t,x)$ a $d \times d$-matrix function(called *diffusion* coefficient).

Stochastic processes $X = X(t)$, where $t \in [0,T]$, can be described by *stochastic differential equations*

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t), \tag{1.1}$$

Let the initial condition $X(0) = x$ be an $\mathcal{F}_0$-measurable random vector in $\mathbb{R}^d$. An $\mathcal{F}_t$-adapted stochastic process $X = (X(t))_{t\geq 0}$ is called a solution of equation (1.1) if

$$X(t) = X(0) + \int_0^t a(s, X(s))ds + \int_0^t b(s, X(s))dW(s) \tag{1.2}$$

is satisfied.

The conditions that the integral processes

$$\int_0^t a(s, X(s))ds, \qquad \int_0^t b(s, X(s))dW(s)$$

are well-defined are required for(1.2) to hold and for the functions $a(s, X(s))$ and $b(s, X(s))$ we have the following conditions that

$$E\int_0^t b^2(s, X(s))ds < \infty$$

and almost surely for all $t \geq 0$

$$\int_0^t |a(s, X(s))|ds < \infty,$$

One of the most important properties for the stochastic integral is that

$$\begin{aligned}\int_0^t W(s)dW(s) &= \frac{1}{2}\int_0^t d(W^2(s)) - \frac{1}{2}\int_0^t ds \\ &= \frac{1}{2}W^2(t) - \frac{t}{2}\end{aligned}$$

for details of stochastic integral see [4].

1.2 Existence and uniqueness theorems

The following theorem, which will be stated without proof, gives sufficient conditions for existence and uniqueness of a solution of a stochastic differential equation.

(i) **measurability**

let a :$[0,\infty)\times\mathbb{R}^d \to \mathbb{R}^d$ and b:$[0,\infty)\times\mathbb{R}^d \to \mathbb{R}^{d\times d}$ are jointly Borel measurable in $[t_0, T]\times\mathbb{R}^d$.

(ii) **Lipschitz condition**:

There is a constant $A > 0$ such that

$$|a(t,x) - a(t,y)| \le A\,|x-y|$$

and

$$|b(t,x) - b(t,y)| \le A\,|x-y|$$

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

(iii) **Growth condition**:

There is a constant $K > 0$ such that

$$|a(t,x)|^2 \le K^2(1+|x|^2)$$

and

$$|b(t,x)|^2 \le K^2(1+|x|^2)$$

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

Theorem 1.

Under these conditions (i-iii) the stochastic differential equation (1.1) has a unique solution $X(t) \in [t_0, T]$ with

$$\sup_{t_0\le t\le T} E(|X(t)|^2) < \infty$$

.

Proof. see Kloeden and Platen [4], Theorem 4.5.3 □

1.3 Strong and weak convergence for SDEs

1.3.1 Strong order of convergence

Suppose that a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is given. In this probability space Ω is the set of continuous functions with the supremum metric on the interval $[0, T]$, $\mathcal{F}$ is the σ-algebra of Borel sets and $\mathbb{P}$ is the Wiener measure. We consider an approximate solution x_h of (1.1) which uses a subdivision of the interval $[0, T]$ into a finite number N of subintervals which we assume to be of length $h = \frac{T}{N}$. Also we assume the approximate solutions x_h is a random variable on Ω. Now we say that the discrete time approximation x_h with the step-size h converges strongly of order γ at time $T = Nh$ to the solution $X(t)$ if

$$E|x_h - X(T)|^p \leq Ch^{\gamma p}, \qquad h \in (0, 1)$$

where the strong convergence will be in L^p space and $X(T)$ is the solution to the stochastic differential equation. C is a positive constant and C independent of h. Our method will give a strong approximation in the sense of this definition and there will be a further discussion of the strong convergence for our result in the end of chapter 2.

We should mention here without more details that there are several applications of strong approximation and some examples have been mentioned in chapter 13 of [4]. Some of these applications will work with our coupling methods and some will not. For example providing that the metric $b_{ik}(x)$ is invertible then the application to the Duffing -Van der Pol Oscillator which is the simulation of individual trajectories will work. One such application is to the simulation of the stochastic flow defined by an SDE, this method will fail because we try to simulate several starting points in the same time. The filtering application will not work because actually the observation process is given.

1.3.2 Weak order of convergence

We say that a discrete time approximation x_h with the step-size h converges weakly of order γ at time $T = Nh$ to the solution $X(t)$ for any smooth function f

$$|E(f(x_h)) - E(f(X(T)))| \leq Ch^\gamma, \qquad h \in (0, 1)$$

Similarly h is the step-size which divides the interval $[0, T]$ into equal length $h = \frac{T}{N}$ and $X(T)$ is the solution to the stochastic differential equation. C is a positive constant and C independent of h

1.4 Numerical method for approximating the SDEs

There are many numerical methods for solving stochastic differential equation, here we will mention two important schemes.The first one is the Euler-Maruyama scheme which will give strong order $\frac{1}{2}$ and the second one is the Milstein scheme which has an order one for the strong convergence. We will show by numerical example their convergence behaviour.

Suppose we have the stochastic differential equation.

$$dX_i(t) = a_i(t, X(t))dt + \sum_{k=1}^{d} b_{ik}(t, X(t))dW_k(t), \quad X_i(0) = X_i^{(0)} \tag{1.3}$$

where i =1,...,d on an interval $[0, T]$, for a d-dimensional vector $X(t)$, with a d-dimensional Brownian path $W(t)$.

In order to approximate the solution, we assume $[0, T]$ is divided into N equal intervals of length $h = T/N$.

1.4.1 Euler-Maruyama scheme

The simplest numerical method for approximating the solution of stochastic differential equations is the stochastic Euler scheme (also called Euler Maruyama scheme)

which utilizes only the first two terms of the Taylor expansion and it attains the strong convergence $\gamma = \frac{1}{2}$.

Firstly, consider the Euler-Maruyama approximation scheme.

$$x_i^{(j+1)} = x_i^{(j)} + a_i(jh, x^{(j)})h + \sum_{k=1}^{d} b_{ik}(jh, x^{(j)})\Delta W_k^{(j)} \tag{1.4}$$

where $\Delta W_k^{(j)} = W_k((j+1)h) - W_k(jh)$ and our numerical approximation to $X(jh)$ will be denoted $x^{(j)}$.

1.4.2 The Milstein scheme

We shall now introduce the Milstein scheme which gives an order one strong Taylor scheme. We could obtain the Milstein scheme by adding the quadratic terms.

$$\sum_{k,l=1}^{d} \rho_{ikl}(jh, x^{(j)})A_{kl}^{(j)}$$

to Euler scheme which gives the following scheme

$$x_i^{(j+1)} = x_i^{(j)} + a_i(jh, x^{(j)})h + \sum_{k=1}^{d} b_{ik}(jh, x^{(j)})\Delta W_k^{(j)} + \sum_{k,l=1}^{d} \rho_{ikl}(jh, x^{(j)})A_{kl}^{(j)} \tag{1.5}$$

Where

$\Delta W_k^{(j)} = W_k((j+1)h) - W_k(jh)$,
$A_{kl}^{(j)} = \int_{jh}^{(j+1)h} \{W_k(t) - W_k(jh)\} dW_l(t)$
and $\rho_{ikl}(t,x) = \sum_{m=1}^{q} b_{mk}(t,x)\frac{\partial b_{il}}{\partial x_m}(t,x)$.

The implementation of the Euler scheme is easy to do as only needs to generate the normal distribution for the standard Brownian motion $\Delta W_k^{(j)}$ but it is not easy to generate the integral $A_{kl}^{(j)}$ for the Milstein scheme when we have two or more dimensional SDEs. We will show by a numerical example in the next section how

we could generate the integral $A_{kl}^{(j)}$ using the Fourier method when we have two dimensional SDEs.

Before the implementation of Euler and Milstein schemes we need to mention some facts about the two-level approximation.

1.5 Two-level approximation

We need to generate the increments $\Delta W_k^{(j)}$ when we approximate the solution to (1.1) by using Euler or other schemes which we will explain later in this chapter, therefore Levy's construction of the Brownian motion will be used to simulate a sequence of approximations converge to the solution.

That is

$$\Delta W_k^{(r,j)} = \Delta W_k^{(r+1,2j)} + \Delta W_k^{(r+1,2j+1)} \tag{1.6}$$

where $r \in \mathbb{N}$ and $\Delta W_k^{(r,j)} = W_k((j+1)h^{(r)}) - W_k(jh^{(r)})$ with $h^{(r)} = \frac{T}{2^r}$.
We will call the two-level approximation in (1.6) *the trivial coupling*. We could generate the normal distribution in (1.6) for the increments for a given level r by firstly generating the increments in the LHS $\Delta W_k^{(r,j)}$ and then conditionally generating the increments in the RHS. We do the same process for each level $r+2$, $r+3$ and so on. After that we will get the Brownian path $W(t)$.

We will see from the following section that the extension of Milstein to $d \geq 2$ is not easy to do. However we could implement special class of equations for Milstein scheme using only the $\Delta W_k^{(j)}$. This could be done from the observation that $A_{kl}^{(j)} + A_{lk}^{(j)} = 2B_{kl}^{(j)}$ where $B_{kl}^{(j)} = \frac{1}{2}\Delta W_k^{(j)} \Delta W_l^{(j)}$ if $k \neq l$ and $B_{kk}^{(j)} = \frac{1}{2}\{(\Delta W_k^{(j)})^2 - h\}$,

1.5.1 Empirical estimation of the error of a numerical method

Because usually we do not know the solutions of the stochastic differential equation explicitly therefore we could not directly estimate the mean error $E|X(T) - x_h|$ which is the absolute value of the difference between the approximation solution x_h and the solution $X(T)$ of an SDE (1.1). Assume the approximate solution x_h converges to the solution $X(T)$ as we decrease the step-size and go to zero. Then we can estimate the order of convergence for a particular scheme by repeating R different independent simulations of sample paths. We will use the following estimator $\{\epsilon = \frac{1}{R}E(|x_{(r)} - \hat{x}_{(r)}|)\}$ for different approximation solutions $x_{(r)}$ and $\hat{x}_{(r)}$ for different range value of h. So for any numerical method if we have a bound for the error $E|x_h - x_{h/2}| \leq C_1 h^\gamma$ then $E|x_{h/2} - x_{h/4}| \leq C_1(\frac{h}{2})^\gamma$ and then $E|x_{h/4} - x_{h/8}| \leq C_1(\frac{h}{2^2})^\gamma$ and so on. Therefore we will get a geometric series then we will obtain

$$E|X(T) - x_h| \leq \sum_{h=0}^{\infty} C_1\big(\frac{h}{2^k}\big)^\gamma = \frac{C_1 h^\gamma}{1 - 2^{-\gamma}} \tag{1.7}$$

So from (1.7) we could estimate the convergence and the constant.

1.6 The implementation of Euler and Milstein schemes

In this section, we have the two-dimensional stochastic differential equations and we need to test the convergence by using Euler and Milstein schemes. The SDEs that we will choose to implement our methods on is

$$\begin{aligned} dX_1(t) &= X_2(t)dW_1(t) + (X_1(t) + t)dW_2(t), \\ dX_2(t) &= e^{-X_2^2(t)}dW_1(t) + (X_1(t) - X_2(t))dW_2(t), \\ &\text{for } 0 \leq t \leq 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{1.8}$$

where W_1 and W_2 are independent standard Brownian motion.

Now, we need to apply Euler and Milstein methods to (1.8) and show the convergence between the final solutions of these methods.

1.6.1 Two-dimensional stochastic differential equation

For the two-dimensional SDEs (1.8), we could simply implement the Euler method by only generating some normal distributions. Now the Euler approximation on $[0, 1]$ with step-size $h = \frac{1}{N}$ is obtained by the recurrence relations.

$$\begin{aligned} x_1^{(j+1)} &= x_1^{(j)} + x_2^{(j)}\Delta W_1^{(j)} + (x_1^{(j)} + jh)\Delta W_2^{(j)} \\ x_2^{(j+1)} &= x_2^{(j)} + e^{-(x_2^{(j)})^2}\Delta W_1^{(j)} + (x_1^{(j)} - x_2^{(j)})\Delta W_2^{(j)} \end{aligned} \tag{1.9}$$

We now wish to estimate the error for the Euler approximation. The Matlab code in Listing ((5.1)) estimates the absolute error $\epsilon = \frac{1}{R}\sum_{i=1}^{R} |x_h^{(i)} - x_{h/2}^{(i)}|$, for x_h the approximation solution to $X(T)$. Here each simulation is for the same Brownian path. We calculate the error for different step-size(400, 800, 1600, 3200, 6400) over a large number of simulation for example (R=100000).

step-size	error(ϵ)
0.0025	0.1519
0.00125	0.1087
0.00062	0.0762
0.00031	0.0534
0.00015	0.0379

Table 1.1: The error results for the Euler scheme in 2-d case

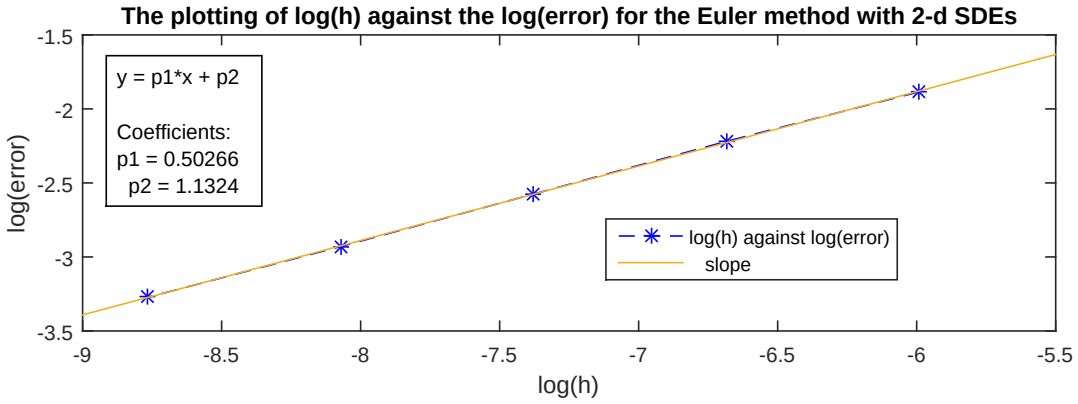


Figure 1.1: Euler method for the two dimension SDEs

In Figure (1.1), we can obtain the coefficients $p1$ and $p2$ using a least square fit to the log of the error against the log of step-size and this will apply of the later figures. Also it is obvious from Table(1.1) and the plot in Figure (1.1), that the convergence seems to occur when we decrease the step-size and we obtain $O(h^{\frac{1}{2}})$ convergence. In addition, as we do not have the explicit solution for the SDE, we could estimate the error and the constant between the solution with step size x_h and explicit solution $x(T)$ by using other approximation solutions for example solution with step-size $x_{h/2}, x_{h/4}$ and $x_{h/8}$ and so on. Therefore from (1.7) we will get that

$$E\left|x_h - X(T)\right| \leq \frac{C}{1-\frac{1}{\sqrt{2}}} h^{1/2} \quad \forall \ h$$

we could see from the result of the table and the slope of the plotting that as the number of step-sizes decreasing we have obtained the $O(\sqrt{h})$ convergence.

We shall now move to find an approximation for the Milstein scheme for two-dimensional SDE.

For the SDEs (1.8), we have the the Milstein scheme

$$x_i^{(j+1)} = x_i^{(j)} + \sum_{k=1}^{2} b_{ik}(x^{(j)})\Delta W_k^{(j)} + \sum_{k=1}^{2} \rho_{ikl}(x^{(j)})A_{kl}^{(j)} \tag{1.10}$$

But the major difficulty here is that the double stochastic integrals,

$$A_{kl}^{(j)} = \int_{jh}^{(j+1)h} (W_l(t) - W_l(jh))dW_k(t) \tag{1.11}$$

for $k \neq l$ cannot be so easily expressed in terms of simpler stochastic integral when the Wiener process is multi-dimensional. Therefore we will use the Fourier series expansion of Wiener process to represent the double integrals.

Before I explain the Fourier method let us start by applying the Milstein scheme (1.10) to (1.8) and then explain which terms that the Fourier method will be represented to.

Before I write the Milstein approximation, we need to find the derivative terms $\rho_{ikl}(x) = \sum_m \frac{\partial b_{ik}}{\partial x_m} b_{ml}(x)$ for the SDEs (1.8).

We have

$$\rho_{111} = e^{-X_2^2}, \quad \rho_{112} = X_1 - X_2 \quad \rho_{121} = X_2 \quad \rho_{211} = -2e^{-2X_2^2}$$
$$\rho_{221} = X_2 - e^{-X_2^2} \quad \rho_{222} = t + X_2 \quad \rho_{122} = X_1 + t \quad \rho_{212} = (-2X_2)(X_1 - X_2)e^{-X_2^2}$$

Then, we have the Milstein approximation for (1.8) which is

$$x_1^{(j+1)} = x_1^{(j)} + x_2^{(j)}\Delta W_1^{(j)} + (x_1^{(j)} + jh)\Delta W_2^{(j)} + e^{-(x_2^{(j)})^2}A_{11} + (x_1^{(j)} - x_2^{(j)})A_{12} + x_2^{(j)}A_{21} + (x_1^{(j)} - jh)A_{22} \tag{1.12}$$

$$x_2^{(j+1)} = x_2^{(j)} + e^{-(x_2^{(j)})^2}\Delta W_1^{(j)} + (x_1^{(j)} - x_2^{(j)})\Delta W_2^{(j)} - 2x_2^{(j)}e^{-2(x_2^{(j)})^2}A_{11} + (x_2^{(j)} - e^{-(x_2^{(j)})^2})A_{21}$$
$$+ (x_2^{(j)} - jh)A_{22} - 2x_2^{(j)}e^{-(x_2^{(j)})^2}(x_1^{(j)} - x_2^{(j)})A_{12} \tag{1.13}$$

Here in this approximation we have the double Wiener integrals A_{11}, A_{12}, A_{21}, and A_{22}. The double Wiener integrals A_{11} and A_{22} in (1.12) and (1.13) are easily computed from the Wiener increments $\Delta W_1^{(j)}$ and $\Delta W_2^{(j)}$ respectively so

$$A_{11} = \int_{jh}^{(j+1)h} (W_1(t) - W_1(jh))dW_1(t) = \frac{1}{2}\left\{(\Delta W_1^{(j)})^2 - h\right\}$$

and

$$A_{22} = \int_{jh}^{(j+1)h} (W_2(t) - W_2(jh))dW_2(t) = \frac{1}{2}\left\{(\Delta W_2^{(j)})^2 - h\right\}$$

On the other hand, the double Wiener integrals

$$A_{12} = \int_{jh}^{(j+1)h} (W_2(t) - W_2(jh))dW_1(t)$$

and

$$A_{21} = \int_{jh}^{(j+1)h} (W_1(t) - W_1(jh))dW_2(t)$$

could not be expressed in term of simpler stochastic integrals when the Wiener process is multi-dimensional. Therefore, for these integrals the Fourier series expansion will be used to approximate them.

Now we will explain the idea of Fourier method as it describe in Kloeden, Platen ([4]) and [18]. The Brownian bridge process

$$W_k(t) - \frac{t}{h}W_k(h) \quad \text{for } 0 \leq \text{t} \leq \text{h}$$

has the Fourier series

$$W_k(t) - \frac{t}{h}W_k(h) = \frac{1}{2}a_{k,0} + \sum_{r=1}^{\infty}\left(a_{k,r}\cos\left(\frac{2rt\pi}{h}\right) + b_{k,r}\sin\left(\frac{2rt\pi}{h}\right)\right) \tag{1.14}$$

where $k = 1, \ldots, d$

Here the coefficients $a_{k,r}$ and $b_{k,r}$ are independent random variables with $N(0, \frac{h}{2r^2\pi^2})$ distributed and we could derive them from the Fourier integrals,

$$a_{k,r} = \frac{2}{h}\int_0^h \left(W_k(s) - \frac{s}{h}W_k(h)\right)\cos\left(\frac{2rt\pi}{h}\right)ds \tag{1.15}$$

and

$$b_{k,r} = \frac{2}{h}\int_0^h \left(W_k(s) - \frac{s}{h}W_k(h)\right)\sin\left(\frac{2rt\pi}{h}\right)ds \tag{1.16}$$

For each $k = 1, \ldots, d$ and $r = 1, \ldots, p$, when we integrate (1.14) over the interval $[0, h]$, we will obtain the approximation of multiple Stratonovich integrals

$$J^p_{(k_1,k_2)} = \frac{1}{2}h\xi_{k_1}\xi_{k_2} - \frac{1}{2}\sqrt{h}(a_{k_2,0}\xi_{k_1} - a_{k_1,0}\xi_{k_2}) + hA^p_{k_1,k_2} \quad k_1, k_2 = 1, \ldots, d \tag{1.17}$$

In the formula (1.17), we have

$$A^p_{k_1,k_2} = \frac{1}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\left(\xi_{k_1,r}\eta_{k_2,r} - \eta_{k_1,r}\xi_{k_2,r}\right); \tag{1.18}$$

and

$$a_{k,0} = -\frac{1}{\pi}\sqrt{2h}\sum_{r=1}^{p}\frac{1}{r}\xi_{k,r} - 2\sqrt{h\rho_p}\mu_{k,p}; \quad \text{where} \quad \rho_{\mathrm{p}} = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{\mathrm{r}=1}^{\mathrm{p}}\frac{1}{\mathrm{r}^2} \tag{1.19}$$

In addition, the $\xi_k, \xi_{k,r}, \eta_{k,r}$ and $\mu_{k,p}$ are independent standard Gaussian random variables.

For the truncation of Fourier series we require a convergence rate of order h for the global error for the Milstein scheme and we will use (1.17) to express the double integral $A^{(j)}_{kl}$ for $k \neq l$. So in order to do this convergence rate we need to compare the mean square error (MSE) of the approximation of the iterated Itô integrals to the discretisation error of the Milstein scheme. As described in Kloeden and Platen [4], Corollary 10.6.5 and equation 10.6.16 we require an MSE bounded by Ch^3 for some positive constant C. The algorithm of Kloeden, Platen and Wright [18] has an MSE of order h^2/p and then we obtain that $Ch^3 = \frac{h^2}{p}$ which gives $h = \frac{1}{Cp}$. Hence we want the number of terms in the truncated sum p to be proportional to h^{-1}.

We know from the symmetry relation that for any double integral we have $A^{(j)}_{12} + A^{(j)}_{21} = 2B^{(j)}_{12}$ where $B^{(j)}_{12} = \frac{1}{2}\Delta W_1{}^{(j)}\Delta W_2{}^{(j)}$

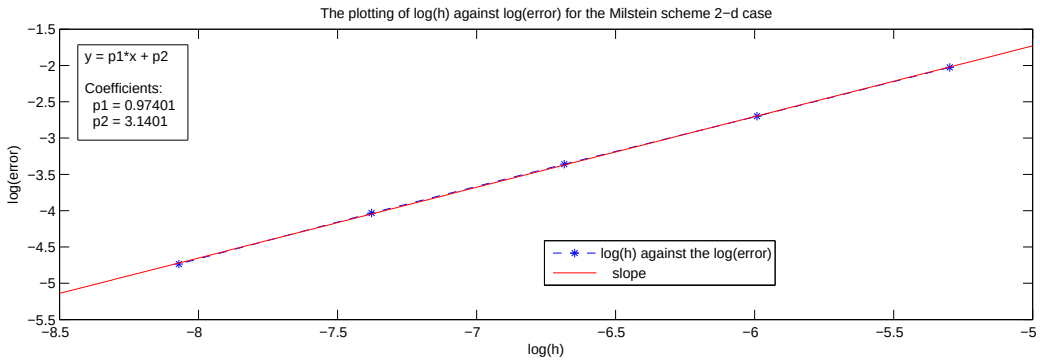
In the M-file in Listing (5.2) in the appendix, I will approximate the value of the double integrals A_{12} and A_{21} and some explanations are shown for the formula (1.17-1.19)

Now, after we represent the approximation of the double integrals A_{11}, A_{22}, A_{12} and A_{21}, we could substitute them in the Milstein approximation in (1.12) and (1.13). After that we need to estimate the error for the Milstein solution in two-dimensional case and test the convergence order.

The Matlab code in Listing (5.3) calculates the Milstein error over the interval $[0, 1]$, with step-size(200, 400, 800, 1600, 3200) with a number of simulation (R=20000).

step-size	error(ϵ)
0.0050	0.1318
0.0025	0.0673
0.00125	0.0347
0.00062	0.0177
0.00031	0.0088

Table 1.2: The error results for the Milstein scheme in 2-d case



It is obvious from Table(1.2) and the plotting in Figure(1.6.1), that the conver-

gence seems to occur when we decrease the step-size and we obtain $O(h)$ convergence. By estimating a range of values of h we could get the estimation of the convergence also the estimation of the constant by using (1.7), so

$$E\left|x_h - X(T)\right| \leq \frac{C}{1-\frac{1}{2}}h \quad \forall \ h$$

We have seen from the previous section that the extension of Milstein to $d \geq 2$ is not easy to do. Moreover if the commutativity condition for

$$\rho_{ikl}(t,x) = \rho_{ilk}(t,x) \tag{1.20}$$

holds for all $x \in \mathbb{R}^d$, $t \in [0,T]$ and all i,k,l then the Milstein scheme (1.5) reduce to

$$x_i^{(j+1)} = x_i^{(j)} + a_i(jh, x^{(j)})h + \sum_{k=1}^{d} b_{ik}(jh, x^{(j)})\Delta W_k^{(j)} + \sum_{k,l=1}^{d} \rho_{ikl}(jh, x^{(j)})B_{kl}^{(j)} \tag{1.21}$$

which only depends on the generation of the Brownian motion $\Delta W_k^{(j)}$. Scheme (1.21) will give an order one if $d = 1$, but if $d > 1$ will have order $\frac{1}{2}$. As it is described in Davie's paper we could do a modification to scheme (1.21)which will give an order one under a non-degeneracy condition.

1.7 A modification to (1.21) which gives order one

As it is described in Davie's paper [3] the interpretation of generating of the normal distribution will be changed in scheme (1.21) which leads to convergence of order one under a non-degeneracy condition.

In the implementation of the Milstein scheme we start by generating the random variables $\Delta W_k^{(j)}$ and $A_{kl}^{(j)}$ separately and then we add these random variables to get the RHS of scheme (1.21). The idea here that we will try to generate the following

$$Y := \sum b_{ik}(jh, x^{(j)})\Delta W_k^{(j)} + \sum \rho_{ikl}(jh, x^{(j)})A_{kl}^{(j)}$$

directly.

If we have a scheme

$$x_i^{(j+1)} = x_i^{(j)} + a_i(jh, x^{(j)})h + \sum b_{ik}(jh, x^{(j)})X_k^{(j)} + \sum \rho_{ikl}(jh, x^{(j)})(X_k^{(j)}X_l^{(j)} - h\delta_{kl}) \tag{1.22}$$

where the increment $X_k^{(j)}$ are independent $N(0,h)$ random variables then it is the same as scheme (1.21) with $\Delta W_k^{(j)}$ replaced by $X_k^{(j)}$ and we do not assume $\Delta W_k^{(j)} = X_k^{(j)}$. Now we need

$$Z_i := \sum b_{ik}(jh, x^{(j)})X_k^{(j)} + \sum \rho_{ikl}(jh, x^{(j)})(X_k^{(j)}X_l^{(j)} - h\delta_{kl})$$

to be a good approximation to Y_i, in other words how we could find a joint distribution of random vectors $(\Delta W_k^{(j)}, A_{kl}^{(j)})$ and $(X_k^{(j)})$ so they have the required marginal distribution, with bound $E(Y_i - Z_i)^2 = O(h^3)$.

We will explain in the following section how we can use a coupling to find the required marginal distribution which will give good bound for the random distribution Y_i and Z_i. After that we will get an order one approximation between the two approximate solutions of the SDEs, $x(jh)$ and $x^{(j)}$ i.e. $E(x(jh) - x^{(j)}) = O(h^2)$.

In chapter 2 we will show the proof of order one convergence using (1.21) with the assumption that $b_{ik}(x)$ is invertible. The proof will be in the two-dimensional case using the coupling method and two different level of approximating solutions of scheme (1.21).

Now we will state some lemmas and theorem which we will used in the later chapters.

Definition 1. (*definition of the Coupling*)

Let $(X_1, \mathbb{F}_1, Q_1)$ and $(X_2, \mathbb{F}_2, Q_2)$ denote two probability spaces. A coupling of the probability measures Q_1 and Q_2 is a probability measure P on $X_1 \times X_2$ whose marginals are Q_1 and Q_2.

Definition 2. (*definition of Vaserstein metrics*)

The p^{th} Vaserstein distance between two probability measures Q_1 and Q_2 on $\mathbb{R}^d$ is defined as the following

$$W_p(Q_1, Q_2) = \inf(E|X - Y|^p)^{1/p} \tag{1.23}$$

Here the infimum is taken over all joint distributions of $\mathbb{R}^d$-value random variables X, Y, where X has distribution Q_1 and Y has distribution Q_2.

The books of Rachev and Ruschendoff [6] and Villani [25], [26] have more information about coupling and Vaserstein distance.

Definition 3.

Let Σ be a positive definite real $q \times q$ matrix and let f be the density function on $\mathbb{R}^q$ of the $N(0, \Sigma)$ normal distribution. Let $\mathcal{P}$ denote the set of polynomials in d variables with real coefficients and let the projection operator P on $\mathcal{P}$ be defined by $(Pp)(x) = p(x) - \bar{p}$ where $\bar{p} = \int_{\mathbb{R}^q} p(x)f(x)dx$. Then $\bar{P}p = 0$. We have the following

Lemma 1.

Let $p \in \mathcal{P}$. Then we can find a vector polynomial $\psi \in \mathcal{P}^q$ such that $\nabla.(f\psi) = fPp$.

Proof. see Lemma 1 in [3] □

Lemma 2.

Let $n \leq N$ and R be positive integers, and for $j = 1, \cdots, N$ let $p_j, r_j \in \mathcal{P}$, all having degree$\leq R$, and such that $p_j = r_j$ for $j \leq n$. Let $\eta > 0$ with $\eta R \leq n$ and let $K > 0$. Then we can find $C > 0$ such that, if $\epsilon > 0$ and we write $\mu_0 = pf\chi_B dx$ and $\nu_0 = rf\chi_B dx$ where $p = 1 + \sum_{j=1}^N \epsilon^j p_j$, $r = 1 + \sum_{j=1}^N \epsilon^j r_j$ and $B = \{x \in \mathbb{R}^q : |x| \leq \epsilon^{-\eta}\}$, and if μ and ν are probability measures on $\mathbb{R}^q$ with $\int_{\mathbb{R}^q}(1 + |x|^2)d|\mu - \mu_0|(x) < K\epsilon^{2n+2}$ and $\int_{\mathbb{R}^q}(1 + |x|^2)d|\nu - \nu_0|(x) < K\epsilon^{2n+2}$, then $\mathbb{W}_2(\mu, \nu) < C\epsilon^{n+1}$.

Proof. see Lemma 2 in [3] □

From the definition shown in [4], we call an equation a *a Stratonovich stochastic differential equation*, writing it in following form

$$dX(t) = A(t, X(t))dt + b(t, X(t)) \circ dW(t), \tag{1.24}$$

or in the equivalent integral equation form

$$X(t) = X(0) + \int_0^t A(s, X(s))ds + \int_0^t b(s, X(s)) \circ dW(s) \tag{1.25}$$

It turns out that the solutions of the Stratonovich SDE (1.24)-(1.25) also satisfy an Ito SDE with the same diffusion coefficient $b(s, X(s))$, but with the modified drift coefficient

$$a(s, x) = A(s, x) + \frac{1}{2} \sum_{j=1}^{d} \sum_{k=1}^{d} b_{kj}(s, x) \frac{\partial b_j}{\partial x^k}(s, x)$$

where b_j is the j^{th} column of the matrix $b(s, x)$.

Definition 4.

The Lie bracket $[U, V]$ of two vector fields U and V on $\mathbf{R}^n$ is the vector field defined by

$$[U, V] = DV(x)U(x) - DU(x)V(x)$$

where we mean by the $DU(x)$ the derivative matrix which given by $(DU(x))_{ij} = \partial_j U_i(x)$

Definition 5.

If we have a stochastic differential equation

$$dX(t) = A(X(t))dt + b(X(t)) \circ dW(t), \tag{1.26}$$

and let $\mathcal{A}_k$ are the collection of vector fields which define by

$$\mathcal{A}_0 = \{b_i : i > 0\}, \qquad \mathcal{A}_{k+1} = \mathcal{A}_k \cup \{[U, b_j] : U \in \mathcal{A}_k \ \& \ j \geq 0\}.$$

where $b_0 = A$ and define the vector spaces by

$$\mathcal{A}_k(x) = span\{V(x) : V \in \mathcal{A}_k\}$$

then we say that (1.26) satisfies the *parabolic Hörmander condition* if $\bigcup_{k \geq 0} \mathcal{A}_k(x) = \mathbf{R}^d$ for every $x \in \mathbf{R}^d$

Now we need to mention the Hörmander Theorem [19] which we will use in chapter 4 .

Theorem 2. *(Hörmander Theorem)*

Suppose we have a stochastic differential equation

$$dX(t) = A(X(t))dt + b(X(t)) \circ dW(t) \tag{1.27}$$

and assume that all vector fields A and b_i's have bounded derivatives of all orders. If (1.27) satisfies the parabolic Hörmander condition, then for positive t the solution $X(t)$ for (1.27) has an infinitely differentiable density with respect to the Lebesgue measure.

Proof. see Theorem 1.3 in [20] □

Chapter 2

Two-level bound using the exact coupling of scheme (1.22)

In this chapter we first prove two lemmas which will be needed for the proof of two-level bound. Then we introduce the exact coupling and then show the proof of a two-level exact coupling bound an explicit versions of scheme (1.22) in L^p space. We get the explicit versions for the coefficients from the Runge-Kutta scheme coefficients(11.1.7) in Kloeden and Platen's book [4] i.e.

$$\beta_{ikl}(x) = \frac{b_{ik}(\Upsilon_n^l) - b_{ik}(x)}{\sqrt{h}}$$

where $\left(\Upsilon_n^l = x + b^l\sqrt{h}\right)$ for $l = 1, 2, \cdots$ and β_{ikl} we will be used an approximation to ρ_{ikl}. In the following section we assume that $b_{ik}(x)$ is twice differentiable with respect to x and $b_{ik}(x)$ and its first and second derivatives are bounded by constants. Moreover we assume the boundedness of the inverse of the $b_{ik}(x)$. Then we will illustrate the computational results which will support the theoretical part. Before we start the proof we want to mention a standard lemma from which we can get the Burkholder's inequality from it and this lemma we will be used in the proof.

Lemma 3.

Suppose X and Y are random variables with $E(Y|X) = 0$. Then for $p \geq 2$ we

have

$$\Big(E|X+Y|^p\Big)^{2/p} \leq \Big(E|X|^p\Big)^{2/p} + C\Big(E|Y|^p\Big)^{2/p} \tag{2.1}$$

where C is a constant depending only on p.

Proof.

We could prove this by using the expansion for $f(X+Y) = |X+Y|^p$ as a Taylor series about $Y = 0$. So by expanding this and take the expectation we will have

$$E|X+Y|^p = E|X|^p + p(E(YX|X|^{p-2})) + E(Y^2 \int_0^1 f''(X+\theta Y)(1-\theta)d\theta) \tag{2.2}$$

where $p(E(YX|X|^{p-2})) = 0$ also $f''(X+\theta Y) = p(p-1)|X+\theta Y|^{p-2}$ which we could bound by $2^{p-2}p(p-1)(|X|^{p-2} + |Y|^{p-2})$, So from (2.2) we will have

$$\begin{aligned}
\Big(E|X+Y|^p\Big)^{2/p} &= \Big(E|X|^p + E(Y^2 \int_0^1 f''(X+\theta Y)(1-\theta)d\theta)\Big)^{2/p} \\
&\leq \Big(E|X|^p + 2^{p-2}p(p-1)[E(Y^2|X|^{p-2}) + E|Y|^p]\Big)^{2/p} \\
&\leq \Big(E|X|^p + 2^{p-2}p(p-1)E(Y^2|X|^{p-2})\Big)^{2/p} \\
&+ \Big(2^{p-2}p(p-1)E|Y|^p\Big)^{2/p} \\
&\leq \Big(E|X|^p + 2^{p-2}p(p-1)[(E|X|^p)^{\frac{p-2}{p}}(E|Y|^p)^{2/p}]\Big)^{2/p} \\
&+ \Big(2^{p-2}p(p-1)E|Y|^p\Big)^{2/p}
\end{aligned} \tag{2.3}$$

In the last step we use the Holder inequality. Now for positive x and y and using the expansion for $(x+y)^{p/2}$ as a Taylor series about $y = 0$, or by the mean value theorem, we obtain

$$(x+y)^{p/2} \geq x^{p/2} + \frac{p}{2}x^{\frac{p-2}{2}}y$$

So

$$\Big(x^{p/2} + \frac{p}{2}x^{\frac{p-2}{2}}y\Big)^{2/p} \leq (x+y).$$

Now from the previous equation and if we take $x = (E|X|^p)^{2/p}$ and $y = 2^{p-1}(p-1)(E|Y|^p)^{2/p}$ then we will obtain the following

$$\begin{aligned}\Big(E|X+Y|^p\Big)^{2/p} &\leq \Big(E|X|^p + 2^{p-2}p(p-1)[(E|X|^p)^{\frac{p-2}{p}}(E|Y|^p)^{2/p}]\Big)^{2/p} \\ &+ \Big(2^{p-2}p(p-1)E|Y|^p\Big)^{2/p} \\ &\leq \big(E|X|^p\big)^{2/p} + 2^{p-1}(p-1)\big(E|Y|^p\big)^{2/p} + \big(2^{p-2}p(p-1)E|Y|^p\big)^{2/p} \\ &= \Big(E|X|^p\Big)^{2/p} + C\Big(E|Y|^p\Big)^{2/p} \qquad (2.4)\end{aligned}$$

□

Also we need the following lemma which will give the bound between the explicit version $\beta_{ikl}(x)$ and the derivatives term $\rho_{ikl}(x)$

Lemma 4.

Suppose we have the Runge-Kutta scheme coefficients(11.1.7) in Kloeden and Platen's book [4], i.e.

$$\beta_{ikl}(x) = \frac{b_{ik}(\Upsilon_n^l) - b_{ik}(x)}{\sqrt{h}}$$

with $b_{ik}(x)$ twice differentiable with respect to x and $\Upsilon_n^l = x + b^l\sqrt{h}$ for $l = 1, 2, \cdots$. Moreover the $b_{ik}(x)$ and its second derivative are bounded by constant. Then the difference approximation between $\beta_{ikl}(x)$ and the derivatives term $\rho_{ikl}(x)$ will be $O(h)$.i.e.

$$\Big(\big|\beta_{ikl}(x) - \rho_{ikl}(x)\big|^p\Big)^{2/p} \leq C_p h \qquad (2.5)$$

where C_p is a constant.

Proof.

We need to use the deterministic Taylor expansion to find $b_{ik}(\Upsilon_n^l)$ where the supporting value is $\Upsilon_n^l = x + b^l\sqrt{h}$ for $l = 1, 2, \cdots$, and for $0 < \theta < 1$

$$b_{ik}(\Upsilon_n^l) = b_{ik}(x) + \sqrt{h}\sum_{n=1}^{d}\frac{\partial b_{ik}(x)}{\partial x^l}b_{l_n}(x) + \frac{1}{2}\sum_{m,n=1}^{d}\frac{\partial^2 b_{ik}(x+\theta b^l\sqrt{h})}{\partial x^l \partial x^m}(b^m b^n\sqrt{h})^2 \tag{2.6}$$

Then we replace (2.6) in $\frac{b_{ik}(\Upsilon_n^l)-b_{ik}(x)}{\sqrt{h}}$ which gives us

$$\begin{aligned}\beta_{ikl}(x) &= \frac{b_{ik}(x)}{\sqrt{h}} + \frac{\sqrt{h}}{\sqrt{h}}\sum_{n=1}^{d}\frac{\partial b_{ik}(x)}{\partial x^l}b_{l_n}(x) \\ &\quad + \frac{\frac{1}{2}\sum_{l,m,n=1}^{d}\frac{\partial^2 b_{ik}(x+\theta b^l\sqrt{h})}{\partial x^l\partial x^m}(b^m b^n\sqrt{h})^2}{\sqrt{h}} - \frac{b_{ik}(x)}{\sqrt{h}} \\ &= \rho_{ikl}(x) + O(h^{1/2})\end{aligned}$$

Thus

$$\left|\left\{\frac{b_{ik}(\Upsilon_n^l)-b_{ik}(x)}{\sqrt{h}}\right\} - \rho_{ikl}(x)\right| \leq C_1 h^{1/2}$$

So

$$\left(\left|\beta_{ikl}(x) - \rho_{ikl}(x)\right|^p\right)^{2/p} \leq C_p h$$

□

2.1 Bounds using two-level coupling

Now, we will proof a two-level bound of scheme (1.22) as described in Davie's paper in section (8), but with the explicit version.

First we consider scheme (1.22) with explicit version and for the simplicity we will let $b_{ik}(x)$ depend only on x and also the drift term equal zero, so

$$x_i^{(j+1)} = x_i^{(j)} + \sum b_{ik}(x^{(j)})X_k^{(j)} + \sum \beta_{ikl}(x^{(j)})(X_k^{(j)}X_l^{(j)} - h\delta_{kl}) \tag{2.7}$$

Now for the step-size $h^{(r)} = \frac{T}{2^r}$ we will have $2^r d$ independent random variables $X_k^{(r,j)}$. Then at two consecutive levels, in other words from level r to level $r+1$, $r \in \mathbb{N}$ we need to find a coupling between $X_k^{(r,j)}$ which is $N(0, h^{(r)})$ and $(X_k^{(r+1,2j)}, X_k^{(r+1,2j+1)})$ so they are independent of each other and they are $N(0, h^{(r+1)})$. If we have that $\tilde{x}_i^{(r,j)}$ is a solution of 2.7 at the level r then for a fix time j we compare $\tilde{x}_k^{(r,j+1)}$ at level r with $\tilde{x}_k^{(r+1,2j+2)}$ in the level $r+1$, we have

$$\tilde{x}_i^{(r,j+1)} = \tilde{x}_i^{(r,j)} + \sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r,j)})X_k^{(r,j)} + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r,j)})(X_k^{(r,j)}X_l^{(r,j)} - h^{(r)}\delta_{kl}) \tag{2.8}$$

and define y as the following which you could see in the diagram in page (43)

$$y = \tilde{x}_i^{(r+1,2j)} + \sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j)})X_k^{(r,j)} + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r,j)}X_l^{(r,j)} - h^{(r)}\delta_{kl}) \tag{2.9}$$

also we have

$$\begin{aligned}\tilde{x}_i^{(r+1,2j+1)} &= \tilde{x}_i^{(r+1,2j)} + \sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j)})X_k^{(r+1,2j)} \\ &\quad + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r+1,2j)}X_l^{(r+1,2j)} - h^{(r+1)}\delta_{kl})\end{aligned} \tag{2.10}$$

$$\begin{aligned}\tilde{x}_i^{(r+1,2j+2)} &= \tilde{x}_i^{(r+1,2j+1)} + \sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j+1)})X_k^{(r+1,2j+1)} \\ &\quad + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j+1)})(X_k^{(r+1,2j+1)}X_l^{(r+1,2j+1)} - h^{(r+1)}\delta_{kl})\end{aligned} \tag{2.11}$$

We should mention that when we write $X = O(M)$ for the random variable X we mean the L^p bound for it i.e. $(E|X|^p)^{1/p} \leq CM$. Now, from lemma 4 we have

$$\begin{aligned} b_{ik}(\tilde{x}^{(r+1,2j+1)}) &= b_{ik}(\tilde{x}^{(r+1,2j)}) + \rho_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r+1,2j)}) + O(h) \\ &= b_{ik}(\tilde{x}^{(r+1,2j)}) + \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r+1,2j)}) + O(h) \end{aligned}$$

and $\beta_{ikl}(\tilde{x}^{(r+1,2j+1)}) = \beta_{ikl}(\tilde{x}^{(r+1,2j)}) + O(h)$

Using these relations in (2.11) and combining it with (2.10) we get.

$$\begin{aligned} \tilde{x_i}^{(r+1,2j+2)} &= \tilde{x_i}^{(r+1,2j)} + \sum_{k=1}^{d} b_{ik}(\tilde{x_i}^{(r+1,2j)})(X_k^{(r+1,2j)} + X_k^{(r+1,2j+1)}) \\ &+ \sum_{l,k=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)}) X_k^{(r+1,2j+1)} X_l^{(r+1,2j)} \\ &+ \frac{1}{2} \sum_{l,k=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r+1,2j)} X_l^{(r+1,2j)} + X_k^{(r+1,2j+1)} X_l^{(r+1,2j+1)} - h^{(r)}\delta_{kl}) \\ &+ \lambda \end{aligned} \tag{2.12}$$

where $\lambda = O((h^{(r)})^{3/2})$

Now, let (c_{ij}) be the matrix inverse of $\left(b_{ik}(\tilde{x}^{(r+1,2j)})\right)$ so that $\sum_j c_{ij} b_{ik}(\tilde{x}^{(r+1,2j)}) = \delta_{ik}$. Then from equation (2.9) and (2.12) if we need the local error $y - \tilde{x}_k^{(r+1,2j+2)} = O((h^{(r)})^{3/2})$

we require the coupling to satisfy

$$\begin{aligned} X_i^{(r,j)} &= X_i^{(r+1,2j)} + X_i^{(r+1,2j+1)} + \sum_{k,l=1}^{d} \tau_{ikl}(X_k^{(r+1,2j+1)} X_l^{(r+1,2j)} - X_l^{(r+1,2j+1)} X_k^{(r+1,2j)}) \\ &+ O((h^{(r)})^{3/2}) \end{aligned} \tag{2.13}$$

where $\tau_{ikl} = \frac{1}{2} \sum_j c_{ij} \beta_{ikl}$

Now we will reformulate (2.13) by a scaling. We fix r write $\epsilon = (h^{(r)})^{1/2}, X_i^{(r,j)} = \epsilon V_i, X_i^{(r+1,2j)} = \epsilon Y_i$ and $X_i^{(r+1,2j+1)} = \epsilon Z_i$. Then $V_1, \cdots, V_d$ are independent and

$N(0,1)$, while $(Y_1, \cdots, Y_d, Z_1, \cdots, Z_d)$ are independent and $N(0, 1/2)$. Now we need to find a coupling between a vector (V_i) and (Y_i, Z_i) so that

$$V_i = Y_i + Z_i + \epsilon \sum_{k,l=1}^{d} \tau_{ikl}(Z_k Y_l - Z_l Y_k) + O(\epsilon^2) \tag{2.14}$$

. We need to write $U_i = Y_i + Z_i$ and $U_i^* = Y_i - Z_i$ that gives U_i and U_i^* are independent and $N(0,1)$. We have $U_l^* U_k - U_k^* U_l = 2(Y_l Z_k - Z_l Y_k)$ so that from equation (2.14) we obtain

$$V_i = U_i + \epsilon \sum_{k,l=1}^{d} \tau_{ikl}(U_l^* U_k - U_k^* U_l) + O(\epsilon^2) \tag{2.15}$$

Therefore, we require a coupling between $(V_1, \cdots, V_d)$ and $(U_1, \cdots, U_d, U_1^*, \cdots, U_d^*)$, here all the random variables are $N(0,1)$, and also $(V_1, \cdots, V_d)$ are mutually independent, $(U_1, \cdots, U_d, U_1^*, \cdots, U_d^*)$ are also mutually independent, and (2.15) holds.

2.1.1 Exact coupling in two-dimensional case

When $d = 2$ from equation (2.15) we have $V_i = U_i + \epsilon a_i(U_2^* U_1 - U_1^* U_2) + O(\epsilon^2)$ where $a_i = \frac{(\tau_{i12} - \tau_{i21})}{2}$, i.e. $a_1 = \frac{(\tau_{112} - \tau_{121})}{2}$ and $a_2 = \frac{(\tau_{212} - \tau_{221})}{2}$. Then we can write $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = R_\theta \begin{pmatrix} a \\ 0 \end{pmatrix}$ Where R_θ is a rotation matrix. i.e. $R_\theta = \begin{pmatrix} a_1/a & -a_2/a \\ a_2/a & a_1/a \end{pmatrix}$ and $a = (a_1^2 + a_2^2)^{1/2}$

Writing $V = R_\theta V'$, $U = R_\theta U'$, and $U^* = R_\theta \tilde{U}$ our required condition becomes

$$V_1' = U_1' + \epsilon a(\tilde{U}_2 U_1' - \tilde{U}_1 U_2') + O(\epsilon^2), \qquad V_2' = U_2' + O(\epsilon^2) \tag{2.16}$$

Lemma 5.

Suppose U and α are independent random variables, where U is $N(0,1)$ and α takes the values ± 1 each with probability $\frac{1}{2}$, and let b and c be fixed real numbers with $|b| < 1$. We define $\mathcal{Y} = U + \alpha(bU + c)$ and $V = \Phi^{-1}(F(\mathcal{Y}))$ where $F(y)$ is the c.d.f.

of $\mathcal{Y}$. *i.e.*

$$\begin{aligned}
F(y) &= P(\mathcal{Y} \le y) \\
&= P(\mathcal{Y} \le y, \alpha = 1) + P(\mathcal{Y} \le y, \alpha = -1) \\
&= \frac{1}{2}P(U + bU + c \le y) + \frac{1}{2}P(U - bU - c \le y) \\
&= \frac{1}{2}P(U \le \frac{y-c}{1+b}) + \frac{1}{2}P(U \le \frac{y+c}{1-b}) \\
&= \frac{1}{2}\left\{\Phi\left(\frac{y-c}{1+b}\right) + \Phi\left(\frac{y+c}{1-b}\right)\right\}
\end{aligned}$$

Here Φ *is the c.d.f. of standard normal distribution; then* V *is* $N(0,1)$. *Otherwise we generate* V *independently to be* $N(0,1)$. *Then*

$$E(V - \mathcal{Y})^p \le K(b^2 + c^2)^p \tag{2.17}$$

where K *is a constant independent of* b *and* c.

Proof.

First, if we have an even integer p then it is obvious that

$$\begin{aligned}
E(V - \mathcal{Y})^p &\le 2^{p-1}E(V^p + \mathcal{Y}^p) = 2^{p-1}E(V^p) + 2^{p-1}E(\mathcal{Y}^p) \\
&= 2^{p-1}(p-1)!! + 2^{p-1}E(U + \alpha(bU + c))^p \\
&\le 2^{p-1}(p-1)!! + 2^{p-1}[2^{p-1}E(U)^p + 2^{p-1}\alpha^p E((bU + c)^p)] \\
&= 2^{p-1}(p-1)!! + 2^{2(p-1)}E(U)^p + 2^{2(p-1)}E(bU)^p + 2^{2(p-1)}E(c)^p \\
&= 2^{p-1}(p-1)!! + 2^{2(p-1)}(p-1)!! + \frac{b^p 2^{2(p-1)}}{2^p}(p-1)!! + \frac{c^p 2^{2(p-1)}}{2^p} \\
&= K_1 + C(b^p + c^p)
\end{aligned} \tag{2.18}$$

Where the constant K_1 and C are depending on p and if either $|b|$ or $|c|$ is greater than $\frac{1}{2}$ and we choose K big enough then the following is true

$$K_1 + C(b^p + c^p) \le K(b^2 + c^2)^p$$

So it suffices to prove the lemma for $|b| \le \frac{1}{2}$, $|c| \le \frac{1}{2}$. Using the expression for F, we find that for $|y| \le \frac{1}{(b^2+c^2)^{1/4}}$ we have

$$\left|F(y) - \Phi(y)\right| = \left|\frac{1}{2}\big(\Phi(\frac{y-c}{1+b}) + \Phi(\frac{y+c}{1-b})\big) - \Phi(y)\right| \tag{2.19}$$

By using the Taylor expansion for $\Phi(\frac{y-c}{1+b})$ and $\Phi(\frac{y+c}{1-b})$ we have

$$\begin{aligned}\Phi(\frac{y-c}{1+b}) &= \Phi(y) + \phi(y)(\frac{y-c}{1+b} - y) + \frac{1}{2}\Phi''(y)(\frac{y-c}{1+b} - y)^2 + O(\frac{y-c}{1+b} - y)^3 \\ &= \Phi(y) + \phi(y)(\frac{-c-yb}{1+b}) + \frac{1}{2}\Phi''(y)((\frac{y-c}{1+b})^2 - 2y\frac{y-c}{1+b} + y^2) \\ &+ O(\frac{y-c}{1+b} - y)^3 \end{aligned} \tag{2.20}$$

and

$$\begin{aligned}\Phi(\frac{y+c}{1-b}) &= \Phi(y) + \phi(y)(\frac{y+c}{1-b} - y) + \frac{1}{2}\Phi''(y)(\frac{y+c}{1-b} - y)^2 \\ &+ O(\frac{y+c}{1-b} - y)^3 \\ &= \Phi(y) + \phi(y)(\frac{c+yb}{1-b}) + \frac{1}{2}\Phi''(y)((\frac{y+c}{1-b})^2 - 2y\frac{y+c}{1-b} + y^2) \\ &+ O(\frac{y+c}{1-b} - y)^3 \end{aligned} \tag{2.21}$$

where for the reminder terms we use the same restrictions that $|y| \leq \frac{1}{(b^2+c^2)^{1/4}}$, $|b| \leq \frac{1}{2}$ and $|c| \leq \frac{1}{2}$

Now,

$$\begin{aligned}\phi(y)(\frac{-c-yb}{1+b}) + \phi(y)(\frac{c+yb}{1-b}) &= \phi(y)(-c + cb - cb^2 - yb + yb^2) \\ &+ \phi(y)(c + cb + cb^2 + yb + yb^2) \\ &= \phi(y)(2cb + 2yb^2) \end{aligned} \tag{2.22}$$

and

$$\begin{aligned}&\Phi''(y)((\frac{y+c}{1-b})^2 - 2y\frac{y+c}{1-b} + y^2) \\ &= \Phi''(y)\big(\frac{y^2 + 2cy + c^2 - 2y^2(1-b) - 2yc(1-b) + y^2(1-b)^2}{(1-b)^2}\big) \\ &= -(y\phi(y))(c^2 + 2bc^2 + 3b^2c^2 + 2byc + 4b^2yc + y^2b^2) \end{aligned} \tag{2.23}$$

$$\begin{aligned}&\Phi''(y)((\frac{y-c}{1+b})^2 - 2y\frac{y-c}{1+b} + y^2) \\ &= \Phi''(y)\big(\frac{y^2 - 2cy + c^2 - 2y^2(1+b) + 2yc(1+b) + y^2(1+b)^2}{(1+b)^2}\big) \\ &= -(y\phi(y))(c^2 - 2bc^2 + 3b^2c^2 + 2byc - 4b^2yc + y^2b^2) \end{aligned} \tag{2.24}$$

So

$$\frac{1}{2}\Phi''(y)(\frac{y-c}{1+b}-y)^2+\frac{1}{2}\Phi''(y)(\frac{y+c}{1-b}-y)^2=-(y\phi(y))(c^2+3b^2c^2+2byc+y^2b^2) \quad (2.25)$$

Then from (2.19) we obtain

$$\begin{aligned}
\big|F(y)-\Phi(y)\big| &= \big|\frac{1}{2}\big(\Phi(\frac{y-c}{1+b})+\Phi(\frac{y+c}{1-b})\big)-\Phi(y)\big| \\
&= \Big|\frac{1}{2}\Big(2\Phi(y)+\phi(y)(2cb+2yb^2)+\frac{1}{2}\Phi''(y)(\frac{y-c}{1+b}-y)^2 \\
&+\frac{1}{2}\Phi''(y)(\frac{y+c}{1-b}-y)^2+O(\frac{y-c}{1+b}-y)^3+O(\frac{y+c}{1-b}-y)^3\Big)-\Phi(y)\Big| \\
&= \big|\phi(y)(cb+yb^2)-(\phi(y))(\frac{1}{2}yc^2+\frac{3}{2}yb^2c^2+by^2c+\frac{1}{2}y^3b^2)) \\
&+O(\frac{y-c}{1+b}-y)^3+O(\frac{y+c}{1-b}-y)^3\big| \\
&= \phi(y)\big|(cb+yb^2)-(\frac{1}{2}yc^2+\frac{3}{2}yb^2c^2+by^2c+\frac{1}{2}y^3b^2))\big| \\
&+O(\frac{y-c}{1+b}-y)^3+O(\frac{y+c}{1-b}-y)^3 \qquad (2.26)
\end{aligned}$$

Thus

$$|F(y)-\Phi(y)|\leq K_1(b^2+c^2)(1+y^2+|y^3|)\phi(y) \quad (2.27)$$

Now we will use the fact that if we have x which is a real number and $0<a<\min(1,|x|^{-1})$, Then for $|z|<a$ we have

$$\begin{aligned}
\phi(x+z) &= \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}-xz-\frac{z^2}{2}} \\
&= \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}e^{-xz-\frac{z^2}{2}} \\
&= \phi(x)e^{-xz-\frac{z^2}{2}}\geq\phi(x)e^{-1-\frac{1}{2}}=\phi(x)e^{-\frac{3}{2}}
\end{aligned}$$

and from this we deduce that

$$\Phi(x+a)\geq\Phi(x)+e^{-\frac{3}{2}}a\phi(x)$$

and similarly for $\Phi(x-a)$ therefore from (2.27) we obtain that

$$|y-\Phi^{-1}(F(y))|\leq K_2(b^2+c^2)(1+y^2+|y^3|) \quad (2.28)$$

where $\phi = \Phi'$ is the $N(0,1)$ density function, and then the lemma follows since $E(V-\mathcal{Y})^p = E(\mathcal{Y} - \Phi^{-1}(F(\mathcal{Y})))^p$ and the contribution from $|\mathcal{Y}| > (b^2+c^2)^{-1/4}$ is negligible if b^2+c^2 is small as we now show.

We have from (2.18) that $E(\mathcal{Y}^M) \leq K$ and M is a big constant $M \geq 4p$ then from Markov's inequality

$$\mathbb{P}(\mathcal{Y} > \frac{1}{(b^2+c^2)^{(1/4)}}) \Rightarrow \mathbb{P}(\mathcal{Y}^M > \frac{1}{(b^2+c^2)^{(1/4)M}}) \leq K(b^2+c^2)^{(1/4)M}$$

From this and the bound in (2.18) and using the Cauchy-Schwartz inequality we obtain that

$$\begin{aligned} E[(V-\mathcal{Y})^p \mathbb{1}_{\mathcal{Y}^M > (b^2+c^2)^{(-1/4)M}}] &\leq (E(V-\mathcal{Y})^{2p})^{(1/2)} (E(\mathbb{1}_{\mathcal{Y}^M > (b^2+c^2)^{(-1/4)M}})^2)^{(1/2)} \\ &= (K_1 + C(b^p+c^p))(K(b^2+c^2)^{(1/4)M}) \end{aligned} \tag{2.29}$$

□

We return to (2.16), and recall that we require to generate the six random variables $V_1', V_2', U_1', U_1', \tilde{U}_1, \tilde{U}_2$ so that each is $N(0,1)$ and that V_1', V_2', are independent and that $U_1', U_1', \tilde{U}_1, \tilde{U}_2$ are also mutually independent. We also require these two sets of random variables are coupled so that (2.16) holds. We start by generating independent $N(0,1)$ variables U_1', U_2', Q, R and α taking the value ± 1 with probability $\frac{1}{2}$ each. Then set $V_2' = U_2'$, $\tilde{U}_1 = \alpha Q$ and $\tilde{U}_2 = \alpha R$. We also define $\mathcal{Y} = U_1' + \alpha(bU_1' + c)$ and $V_1' = \Phi^{-1}(F(\mathcal{Y}))$ where $F(y) = \frac{1}{2}\left\{\Phi(\frac{y-c}{1+b}) + \Phi(\frac{y+c}{1-b})\right\}$ is the cumulative distribution function of $\mathcal{Y}$ (here Φ is the c.d.f of $N(0,1)$, where $b = \epsilon a R$ and $c = -\epsilon a Q U_2'$. This gives $\mathcal{Y} = U_1' + \epsilon a(\tilde{U}_2 U_1' - \tilde{U}_1 U_2')$. Also conditional on Q, R, U_2' we see that V_1' is $N(0,1)$, so V_1' is independent of V_2' and all six variables have $N(0,1)$ distribution.

In the following page, we will give a summary of the exact coupling.

Summary of the exact coupling

The approximations are generated using (2.8), (2.10) and (2.11) the X variables at the j'th step being calculated as follows. We start by generating independent $N(0,1)$ variables U_1', U_2', Q, R and α taking the value ± 1 with probability $\frac{1}{2}$ each. Then set $V_2' = U_2'$, $\tilde{U}_1 = \alpha Q$ and $\tilde{U}_2 = \alpha R$.

If we have $a = (a_1^2 + a_2^2)^{1/2}$, $a_i = (\tau_{i12} - \tau_{i21})/2$ where $\tau_{ik\ell} = \frac{1}{2}\sum_j c_{ij}\left\{\frac{b_{jk}(jh, \Upsilon_n^j) - b_{jk}}{\sqrt{h}}\right\}$ and $\left(\Upsilon_n^l = x^{(r+1,2j)} + b^l\sqrt{h}\right)$ for $l = 1, 2, \cdots, .$

We also define $Y = U_1' + \alpha(bU_1' + c)$ and $V_1' = \Phi^{-1}(F(Y))$ where $F(y) = \frac{1}{2}\left\{\Phi(\frac{y-c}{1+b}) + \Phi(\frac{y+c}{1-b})\right\}$ is the cumulative distribution function of Y (here Φ is the c.d.f of $N(0,1)$, where $b = \epsilon a R$ and $c = -\epsilon a Q U_2'$.

After that, $V = R_\theta V'$ where R_θ is a rotation matrix, i.e.

$$V_1 = \frac{a_1}{a}V_1' - \frac{a_2}{a}V_2' \qquad\qquad V_2 = \frac{a_2}{a}V_1' + \frac{a_1}{a}V_2'$$

and $U = R_\theta U'$ where R_θ is a rotation matrix, i.e.

$$U_1 = \frac{a_1}{a}U_1' - \frac{a_2}{a}U_2' \qquad\qquad U_2 = \frac{a_2}{a}U_1' + \frac{a_1}{a}U_2'$$
$$U_1^* = \frac{a_1}{a}\tilde{U}_1 - \frac{a_2}{a}\tilde{U}_2 \qquad\qquad U_2^* = \frac{a_2}{a}\tilde{U}_1 + \frac{a_1}{a}\tilde{U}_2$$

Then

$$Z_1 = \frac{1}{2}(U_1 - U_1^*), \quad Z_2 = \frac{1}{2}(U_2 - U_2^*), \quad Y_1 = \frac{1}{2}(U_1 + U_1^*), \quad Y_2 = \frac{1}{2}(U_2 + U_2^*)$$

After that

$$X_1^{(r+1,2j+1)} = \sqrt{h}Z_1, \quad X_2^{(r+1,2j+1)} = \sqrt{h}Z_2, \quad X_1^{(r+1,2j)} = \sqrt{h}Y_1, \quad X_2^{(r+1,2j)} = \sqrt{h}Y_2$$

$$X_1^{(r,j)} = \sqrt{h}V_1, \qquad X_2^{(r,j)} = \sqrt{h}V_2$$

.

Finally from the lemma (5) we have

$$\begin{aligned} E(V_1' - \mathcal{Y})^p &\leq K^p E((\epsilon a R)^2 + (-\epsilon a Q U_2')^2)^p \\ &\leq 2^{2p-1} a^{2p} \epsilon^{2p} E(R^{2p} + Q^{2p} (U_2')^{2p}) \\ &\leq 2^{2p-1} a^{2p} \epsilon^{2p} [2(p-1)!!]^2 \\ &= C_p a^{2p} \epsilon^{2p} \end{aligned} \tag{2.30}$$

thus

$$(E(V_1' - \mathcal{Y})^p)^{2/p} \leq C_1 a^4 \epsilon^4$$

Lemma 6.

From the error which we obtain in (2.30), we have that the local error is

$$(E|\tilde{x}_i^{(r+1,2j+2)} - y|^p)^{2/p} \leq C_p a^2 h^3$$

Proof.

It is possible to deduce and derive that from equation (2.16). Firstly, from equation (2.16) we have

$$V_1' = U_1' + \epsilon a(\tilde{U}_2 U_1' - \tilde{U}_1 U_2') + K a^2 \epsilon^2 \tag{2.31}$$

Where K is a random variable and $E(K^p) \leq C_2$ where C_2 is a constant.

Then after we multiply by the term b and rotation matrix R_θ, we obtained

$$bR_\theta V' = bR_\theta U' + \epsilon b R_\theta \begin{pmatrix} a \\ 0 \end{pmatrix} (U_2^* U_1 - U_1^* U_2) + K \epsilon^2 b R_\theta \begin{pmatrix} a^2 \\ 0 \end{pmatrix}$$

Then, this will give us

$$(bV)_i = (bU)_i + \epsilon b_i \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} (U_2^* U_1 - U_1^* U_2) + K\epsilon^2 a b_i \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

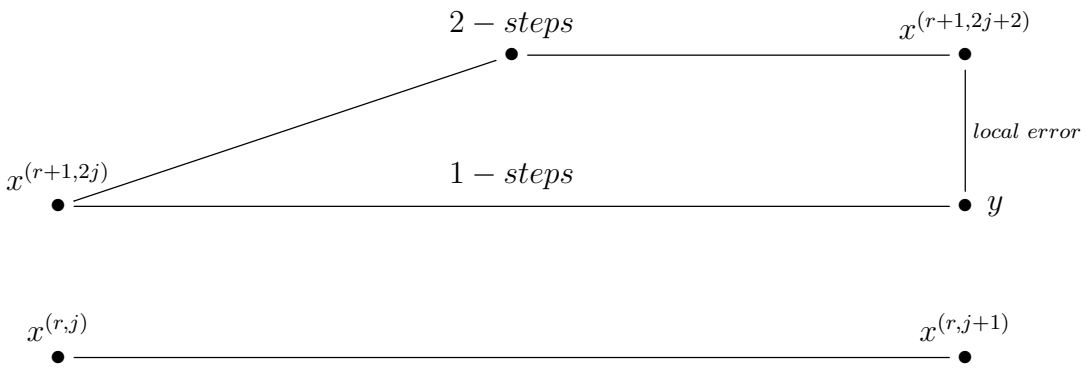
$$\Rightarrow (bV)_i = (bU)_i + \epsilon \sum_{k,l=1}^{2} \beta_{ikl}(x^{(r,j)})(U_l^* U_k - U_k^* U_l) + Ka\epsilon^2(\beta_{i12} - \beta_{i21})$$

$$\begin{aligned}
\epsilon \sum_{k=1}^{2} b_{ik}(x^{(r,j)}) V_k &= \epsilon \sum_{k=1}^{2} b_{ik}(x^{(r,j)})(Y_k + Z_k)) + \epsilon^2 \sum_{k,l=1}^{2} \beta_{ikl}(x^{(r,j)})[Z_k Y_l - Z_l Y_k] \\
&\quad + Ka\epsilon^3(\beta_{i12} - \beta_{i21}) \\
\sum_{k=1}^{2} b_{ik}(x^{(r,j)}) X_k^{(r,j)} &= \sum_{k=1}^{2} b_{ik}(x^{(r,j)})(X_k^{(r+1,2j)} + X_k^{(r+1,2j+1)})) \\
&\quad + \frac{1}{2} \sum_{k,l=1}^{2} \beta_{ikl}(x^{(r,j)})[X_k^{(r+1,2j+1)} X_l^{(r+1,2j)} - X_l^{(r+1,2j+1)} X_k^{(r+1,2j)}] \\
&\quad + Ka(\beta_{i12} - \beta_{i21})h^{3/2} \qquad (2.32)
\end{aligned}$$

Here we need L^p bound for the difference $\tilde{x}_i^{(r+1,2j+2)} - y$. So we combine (2.32) with (2.12), then we get $\tilde{x}_i^{(r+1,2j+2)} - y = \lambda - Ka(\beta_{i12} - \beta_{i21})h^{3/2}$ where λ is the remainder term in (2.12) and then we obtain the bound for the local error $y - \tilde{x}_k^{(r+1,2j+2)}$. □

Now from the Lipschitz condition and the local error bound we need to find the global error.

To have more idea we will illustrate the approximation solutions in the following diagram.



Theorem 3.

Assume the matrix $b_{ik}(x)$ is invertible and twice differentiable with respect to x. Moreover the $b_{ik}(x)$ and its second derivative are bounded by constant. Also we assume the boundedness on the inverse of the matrix $b_{ik}(x)$. Then we obtain the following

$$(E|\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)}|^p)^{2/p} \leq k_2 h^2 e^{TL} \tag{2.33}$$

where $\tilde{x}_i^{(r,j)}$ and $\tilde{x}_i^{(r+1,2j)}$ are define the same in (2.8), (2.10) and (2.11) where the explanation of the generation of the random variables X has been shown in the coupling summary.

Proof.

Let

$$\max_i (E(|\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)}|^p))^{2/p} = e_j,$$

then

$$\begin{aligned}
(E|\tilde{x}_i^{(r+1,2j+2)} - \tilde{x}_i^{(r,j+1)}|^p)^{2/p} &= (E|(y - \tilde{x}_i^{(r,j+1)}) + (\tilde{x}_i^{(r+1,2j+2)} - y)|^p)^{2/p} \\
&= (E|(\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)}) + (y - \tilde{x}_i^{(r+1,2j)}) \\
&\quad - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)}) + (\tilde{x}_i^{(r+1,2j+2)} - y)|^p)^{2/p} \\
&\leq e_j + C_1[|(E(\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)})|\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)}|^{(p-2)} \\
&\quad (y - \tilde{x}_i^{(r+1,2j)}) - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)}) + (\tilde{x}_i^{(r+1,2j+2)} - y))|]^{2/p} \\
&\quad + C_2[(E|(y - \tilde{x}_i^{(r+1,2j)}) - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)}) \\
&\quad + (\tilde{x}_i^{(r+1,2j+2)} - y)|^p)]^{2/p}
\end{aligned}$$

Here we use lemma (3) with $X = (\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)})$ and

$$\begin{aligned} Y &= (y - \tilde{x}_i^{(r+1,2j)}) - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)}) + (\tilde{x}_i^{(r+1,2j+2)} - y) \\ &= (\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j)}) X_k^{(r,j)} + \frac{1}{2} \sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl}))) \\ &- ((\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r,j)}) X_k^{(r,j)} + \frac{1}{2} \sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r,j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl})) + (\tilde{x}_i^{(r+1,2j+2)} - y) \end{aligned} \tag{2.34}$$

and notice that

$$\begin{aligned} E(Y|X) = E[&(\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j)}) X_k^{(r,j)} + \frac{1}{2} \sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl}))) \\ &- ((\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r,j)}) X_k^{(r,j)} + \frac{1}{2} \sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r,j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl})) \\ &+ (\tilde{x}_i^{(r+1,2j+2)} - y) | (\tilde{x}_i^{(r+1,2j)} - \tilde{x}_i^{(r,j)})] = 0 \end{aligned} \tag{2.35}$$

Therefore

$$
\begin{aligned}
(E|\tilde{x}_i^{(r+1,2j+2)} - \tilde{x}_i^{(r,j+1)}|^p)^{2/p} &\leq e_j + C_2[(E|(y - \tilde{x}_i^{(r+1,2j)}) - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)}) \\
&\quad + (\tilde{x}_i^{(r+1,2j+2)} - y)|^p)]^{2/p} \\
&\leq e_j + C_3[(E|(y - \tilde{x}_i^{(r+1,2j)}) - (\tilde{x}_i^{(r,j+1)} - \tilde{x}_i^{(r,j)})|^p]^{2/p} \\
&\quad + C_4 E[|(\tilde{x}_i^{(r+1,2j+2)} - y)|^p)]^{2/p} \\
&= e_j + C_3[(E|(\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,2j)}) X_k^{(r,j)} \\
&\quad + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl})) \\
&\quad - (\sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r,j)}) X_k^{(r,j)} \\
&\quad + \frac{1}{2}\sum_{k,l=1}^{d} \beta_{ikl}(\tilde{x}^{(r,j)})(X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl}))|^p]^{2/p} \\
&\quad + C_4 E[|(\tilde{x}_i^{(r+1,2j+2)} - y)|^p)]^{2/p} \\
&\leq e_j + C_5[E|\sum_{k=1}^{d}(b_{ik}(\tilde{x}^{(r,j)}) - b_{ik}(\tilde{x}^{(r+1,2j)}))X_k^{(r,j)}|^p]^{2/p} \\
&\quad + C_6[E|\frac{1}{2}\sum_{k,l=1}^{d}(\beta_{ikl}(\tilde{x}^{(r,j)}) - \beta_{ikl}(\tilde{x}^{(r+1,2j)})) \\
&\quad (X_k^{(r,j)} X_l^{(r,j)} - h^{(r)}\delta_{kl})|^p]^{2/p} + C_4 E[|(\tilde{x}_i^{(r+1,2j+2)} - y)|^p)]^{2/p}
\end{aligned}
\tag{2.36}
$$

Where C_1, C_2, C_3 and C_4 are constants depending only on p.

Now from the assumption that $b_{ik}(x)$ is twice differentiable with respect to x and its second derivative are bounded by a constant then the Lipschitz condition holds and there is a constant $A > 0$ such that

$$
|b_{ik}(x) - b_{ik}(y)| \leq A\,|x - y|
$$

$$\left| b_{ik}(x)\frac{\partial b_{ik}(x)}{\partial x} - b_{ik}(y)\frac{\partial b_{ik}(y)}{\partial y} \right| \leq A\,|x-y|$$

for all $t \in [t_0, T]$ and $x, y \in \Re$. Therefore we will have

$$C_3[E|\sum_{k=1}^{d}(b_{ik}(\tilde{x}^{(r,j)}) - b_{ik}(\tilde{x}^{(r+1,2j)}))X_k^{(r,j)}|^p]^{2/p} \leq L^2 h e_j$$

and

$$C_3[E|\frac{1}{2}\sum_{k,l=1}^{d}(\beta_{ikl}(\tilde{x}^{(r,j)}) - \beta_{ikl}(\tilde{x}^{(r+1,2j)}))(X_k^{(r,j)}X_l^{(r,j)} - h^{(r)}\delta_{kl})|^p]^{2/p} \leq L_1^2 h^2 e_j$$

Where from equation (2.32) we have that the local error

$$(E|\tilde{x}_i^{(r+1,2j+2)} - y|^p)^{2/p} \leq C_p a^4 h^3$$

From the hypotheses we have $|a|^4$ is bounded by a constant c_1, i.e $|a|^4 \leq c_2$. Then we have

$$(E|\tilde{x}_i^{(r+1,2j+2)} - \tilde{x}_i^{(r,j+1)}|^p)^{2/p} \leq e_j + hL^2 e_j + L_1^2 h^2 e_j + C_p a^4 h^3$$

and as the estimation is taken over all i then it includes the maximum i therefore putting the estimates together we obtain the recurrence inequality

$$\begin{aligned} e_{j+1} &\leq e_j + hL^2 e_j + L_1^2 h^2 e_j + C_p a^4 h^3 \\ &\leq e_j + hL^2 e_j + L_1^2 h e_j + C_p a^4 h^3 \\ &\leq e_j + hL e_j + K_1 h^3 \\ &\leq (1 + hL)e_j + R \end{aligned}$$

Where $R = K_1 h^3$

Noting that $(j+1)h \leq T$ for $j < N$ and $e_0 = 0$

We have

$$\begin{aligned} e_j &\leq R\sum_{k=0}^{j-1}(1+hL)^k \\ &\leq R\sum_{k=0}^{N-1}(1+hL)^k \\ &= R\frac{(1+hL)^N-1}{hL} \\ &= \big(K_1h^3\big)\big(\frac{(1+hL)^N-1}{hL}\big) \\ &\leq K_2h^2e^{TL} \end{aligned}$$

□

Interpretation of the strong convergence of the coupling method

We have shown in the previous theorem how we could achieve $\tilde{x}_i^{(r,j)} - \tilde{x}_i^{(r+1,2j)} = O(h^{(r)})$ for the global bounds from the local bounds. So as described in Davie's paper [3] there is a second method to generate the random variables for the coupling method which is essentially the same as what we use but the finer level is generated conditionally on the coarser level. So by an induction using the exact coupling conditioned on the first level we get an infinite sequence of couplings between successive levels then we can set them together on one probability space. This sequence of approximations will converge with order one. Also from (2.13) we have $X^{(r,j)} = X^{(r+1,2j)} + X^{(r+1,2j+1)} + O(h^{(r)})$, and then we add these sequences up to get the approximate Brownian path i.e. $\sum_{i=0}^{j2^n-1} X_k^{(n+r,i)}$ as $n \to \infty$ which converges to a limit and that limit is $W_k(jh^{(r)})$ for a limiting Brownian path $W(t)$. Then we will obtain the approximate solutions $\tilde{x}_i^{(r,j)}$ will converge with order one to the solution $X(T)$ of (1.3) for this path $W(t)$.

In theory and from the optimal transport theory one can realize the approximation solutions $\tilde{x}^{(r,2^r)}$ on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$ as in section (1.3.1) which is defined by the Brownian path but we do not express the solutions explicitly as functions of the Brownian path. In other words, we can find a sequence of random variables $\mathcal{X}_r$ which will be functions of the exact solution $X(T)$ and have the same distribution as the approximate solution $\tilde{x}^{(r,2^r)}$ and they will give the same error bound on that probability space. When we implement the scheme and generate $\tilde{x}^{(r,2^r)}$, we can regard the process as generating an $\mathbb{R}^{dN}$-valued random variable, which approximates the solution in strong sense.

2.2 The implementation of exact coupling in two-dimensional case with invertible $\left(b_{ik}(\tilde{x}^{(r+1,2j)})\right)$

First of all, we have the 2-dimensional SDE, which is invertible.

$$\begin{aligned} dX_1(t) &= (\sin(X_2(t)))^2 dW_1(t) - \frac{1}{1+X_1^2(t)} dW_2(t), \\ dX_2(t) &= \frac{1}{1+X_2^4(t)} dW_1(t) + (\cos(X_1(t)))^2 dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{2.37}$$

where $W_1(t)$ and $W_2(t)$ are independent standard Brownian motion.

To apply a numerical method to this SDE we need to simulate solutions (for the same Brownian path) simultaneously using two different step sizes (h and $h/2$).

The Matlab implementation for this SDE using the exact coupling is given in Listing (5.6), which will show us the result of the absolute value of the difference between two solutions with step size h and $h/2$.

To construct this experiment, we will decrease the step size (h) every time when we calculate the error and examine the convergence order of the exact coupling method. We will repeat this with different step size using (for example, $R = 2000$) independent simulations. Then the order of convergence of this method between two approximate solutions should be 1.

For the SDE (2.37), the Matlab code in Listing (5.6) estimates the absolute error $\epsilon = \frac{1}{R}\sum_{i=1}^{R} |x_h^{(i)} - x_{h/2}^{(i)}|$, for the approximation solution x_h where each simulation is for the same Brownian path. We will run the Matlab code in Listing (5.6) with different number of steps (200, 400, 800, 1600, 3200, 6400) over a very large number of path.

The table (2.1) and the plotting in Figure (2.1) show the implementation of the approximation solutions of the previous 2-dimensional SDEs with different number of steps (200, 400, 800, 1600, 3200 and 6400). Running the code (exatcouplingin-

step-size	error(ϵ)
0.005	0.0056
0.0025	0.0028
0.00125	0.0014
0.00062	0.00069
0.00031	0.00036
0.00015	0.00017

Table 2.1: The error results for the Exact coupling with the invertible matrix

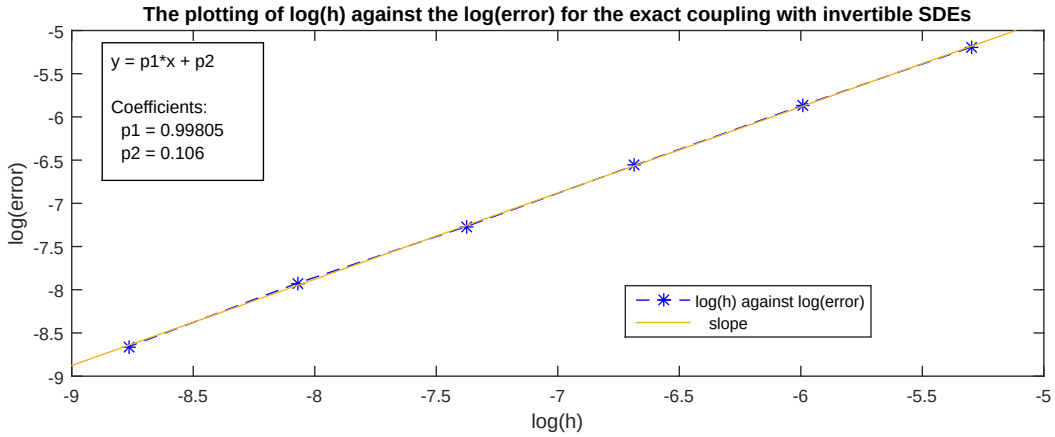


Figure 2.1: plotting for the convergence of the exact coupling

vertible) for 2000 simulations gives a value for its estimator ϵ equal to 0.0056 with the step-size 0.005 i.e.

$$\epsilon = \frac{1}{2000}\sum_{i=1}^{2000} |x_h^{(i)} - x_{h/2}^{(i)}| = 0.0056$$

and 0.0028 with step-size 0.0025 and so on. This means when we increase the number of steps which each time gives a smaller step-size then the estimate error ϵ will give $O(h)$ as it appears in the results in table (2.1). Also the Figure (2.1) is a plot of the log of the estimator ϵ i.e. $\log \epsilon$ against the log of step-size h i.e. $\log(h)$ which has a slope of 0.99805 which again indicates a strong convergence of $O(h)$ for the stochastic differential equation (2.37).

Therefore from these computational results we could see that we have obtained good agreement between the theoretical bound in (2.33) and the implementation results.

Chapter 3

Approximate coupling for general d

In this section we will describe another method for the coupling which satisfies (2.15) with U, U^* having the required distribution but the random variable V has only approximately a standard normal distribution. Here the error bounds are somewhat less precise but the estimates can easily be made rigorous. First of all, we will start with a lemma.

Lemma 7.

Let $U = (U_1, \cdots, U_d)$ be a random vector with $N(0, I)$ distribution and let A be a fixed $d \times d$ matrix. Let $Y = U + \epsilon AU$. Then the density function of Y satisfies

$$f_Y(y) = (2\pi)^{-d/2} e^{-|y|^2/2} \{1 + \epsilon(y^t Ay - trA) + \epsilon^2 \Omega\} + O(\epsilon^3) \tag{3.1}$$

where $\Omega = -(trA)y^t Ay - y^t A^2 y - \frac{1}{2}|Ay|^2 + \frac{1}{2}(y^t Ay)^2 + \frac{1}{2}(trA)^2 + \frac{1}{2}tr(A^2)$.

Proof.

see Davie [3] □

We need now to apply the lemma to (2.15). We will use the same definition for

U and U^* as in (2.15) and define

$$Y_i = U_i + \frac{\epsilon}{2} \sum_{k,l=1}^{d} \tau_{jkl}(U_l^* U_k - U_k^* U_l) \tag{3.2}$$

If we define $\sigma_{ikl} = \frac{1}{2}(\tau_{ikl} - \tau_{ilk})$ then we could rewrite (3.2) in the following way

$$Y_i = U_i + \epsilon \sum_{k,l=1}^{d} \sigma_{ikl} U_k U_l^* \tag{3.3}$$

Now we need to find V which is close to $N(0, I)$ such that $V - U = O(\epsilon^2)$. To do this we first apply the previous lemma to approximate the density function of Y.

We write $Y = U + \epsilon AU$ where the matrix $A = (a_{ik})$ is given by $a_{ik} = \sum_{l=1}^{d} \sigma_{ikl} U_l^*$. Then the density of Y, conditional on U^*, is given by (3.12) from the previous lemma. Now we need to find the unconditional density of Y by substituting for A in (3.12) and taking the expectation with respect to U^*. We will do this for every term separately. Firstly let

$$\delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

If we have the normal distributions $U_1^*, \cdots, U_n^*$ and let

$$N_1 = \sum_{l=1}^{n} a_l U_l^*, \qquad N_2 = \sum_{k=1}^{n} b_k U_k^*$$

Then

$$\begin{aligned} E(N_1 N_2) &= E \sum_{k,l=1}^{n} a_l b_k U_l^* U_k^* \\ &= \sum_{k,l=1}^{n} a_l b_k E(U_l^* U_k^*) \\ &= \sum_{k,l=1}^{n} a_l b_k \delta_{kl} \\ &= \sum_{k=1}^{n} a_k b_k \end{aligned}$$

So from equation (3.12) and taking the expectation w.r.t U^* , we have.

$$\begin{aligned} E((\mathrm{tr}A)y^tAy) &= E(\sum_{i,k=1}^{d}\Big(\sum_{m=1}^{d}a_{mm}\Big)a_{ik}y_ky_i) \\ &= \sum_{i,k=1}^{d}\sum_{m,l,j=1}^{d}\sigma_{mml}\sigma_{ikj}E(U_l^*U_j^*)y_ky_i \\ &= \sum_{i,k=1}^{d}\sum_{m,l=1}^{d}\sigma_{mml}\sigma_{ikl}y_ky_i \end{aligned} \tag{3.4}$$

$$\begin{aligned} E(y^tA^2y) &= E\Big(\sum_{i,k=1}^{d}\Big(\sum_{m=1}^{d}a_{im}a_{mk}y_ky_i\Big)\Big) \\ &= \sum_{i,k=1}^{d}\sum_{m,l,j=1}^{d}\sigma_{iml}\sigma_{mkj}E(U_l^*U_j^*)y_ky_i \\ &= \sum_{i,k=1}^{d}\sum_{m,l=1}^{d}\sigma_{iml}\sigma_{mkl}y_ky_i \end{aligned} \tag{3.5}$$

$$\begin{aligned} E|Ay|^2 = E(y^tA^tAy) &= E\Big(\sum_{i,k=1}^{d}\Big(\sum_{m=1}^{d}a_{mi}a_{mk}y_ky_i\Big)\Big) \\ &= \sum_{i,k=1}^{d}\sum_{m,l,j=1}^{d}\sigma_{mil}\sigma_{mkj}E(U_l^*U_j^*)y_ky_i \\ &= \sum_{i,k=1}^{d}\sum_{m,l=1}^{d}\sigma_{mil}\sigma_{mkl}y_ky_i \end{aligned} \tag{3.6}$$

$$\begin{aligned} E(y^tAy)^2 &= E\Big(\sum_{i,j,k,m=1}^{d}a_{ik}a_{jm}y_ky_iy_my_j\Big) \\ &= \sum_{i,j,k,m=1}^{d}\sum_{l,r=1}^{d}\sigma_{ikl}\sigma_{jmr}E(U_l^*U_r^*)y_ky_iy_my_j \\ &= \sum_{i,j,k,m=1}^{d}\sum_{l=1}^{d}\sigma_{ikl}\sigma_{jml}y_ky_iy_my_j \end{aligned} \tag{3.7}$$

$$E((\mathrm{tr}A)^2) = E\Big(\sum_{i,k=1}^{d} a_{ii}a_{kk}\Big)$$
$$= \sum_{i,k=1}^{d}\sum_{l,j=1}^{d} \sigma_{iil}\sigma_{kkj}E(U_l^*U_j^*)$$
$$= \sum_{i,k,l=1}^{d} \sigma_{iil}\sigma_{kkl} \tag{3.8}$$

Finally

$$E(\mathrm{tr}A^2) = E(\mathrm{tr}AA)$$
$$= E\Big(\sum_{i,k=1}^{d} a_{ik}a_{ki}\Big)$$
$$= \sum_{i,k=1}^{d}\sum_{l,j=1}^{d} \sigma_{ikl}\sigma_{kij}E(U_l^*U_j^*)$$
$$= \sum_{i,k,l=1}^{d} \sigma_{ikl}\sigma_{kil} \tag{3.9}$$

Now from (3.4) to (3.9) the unconditional density of Y satisfies

$$(2\pi)^{-d/2}e^{-|y|^2/2}\Big\{1+\epsilon^2\Big(\mathcal{K}-\sum_{i,k=1}^{d}\theta_{ik}y_iy_k+\sum_{i,j,k,m=1}^{d}\Psi_{ijkm}y_iy_jy_ky_m\Big)\Big\}+O(\epsilon^4) \tag{3.10}$$

Where $\mathcal{K} = \frac{1}{2}\sum_{i,k,l=1}^{d}(\sigma_{ikl}\sigma_{kil}+\sigma_{iil}\sigma_{kkl})$, $\theta_{ik} = \sum_{l,m=1}^{d}(\sigma_{iml}\sigma_{mkl}+\sigma_{ikl}\sigma_{mml}+\frac{1}{2}\sigma_{mil}\sigma_{mkl})$ and $\Psi_{ijkm} = \frac{1}{2}\sum_{l=1}^{d}\sigma_{ikl}\sigma_{jml}$.

Here there is no ϵ^3 term because the density is invariant under $\epsilon \to -\epsilon$.

A correction term needs to be added to the distribution Y to make it close to the standard normal distribution. We consider an $\mathbb{R}^d$-valued random variable V given by

$$V_i = U_i + \epsilon\sum_{k,l=1}^{d}\sigma_{ikl}U_kU_l^* + \epsilon^2 p_i(U) \tag{3.11}$$

Here we need to choose the p which is an $\mathbb{R}^d$-valued polynomial on $\mathbb{R}^d$. After we add the correction term to the V then we need to find its density function f_V and we will use the following Lemma.

Lemma 8.

Let $U = (U_1, \cdots, U_d)$ be a random vector with $N(0, I)$ distribution and let A be a fixed $d \times d$ matrix. Let $Y = U + \epsilon AU + \epsilon^2 p(U)$ where p is an $\mathbb{R}^d$-valued polynomial on $\mathbb{R}^d$. Then the density function of Y satisfies

$$f_Y(y) = (2\pi)^{-d/2} e^{-|y|^2/2} \{1 + \epsilon(y^t Ay - trA) + \epsilon^2 \Omega\} + O(\epsilon^3) \tag{3.12}$$

Where $\Omega = -(trA)y^t Ay - y^t A^2 y - \frac{1}{2}|Ay|^2 + \frac{1}{2}(y^t Ay)^2 + \frac{1}{2}(trA)^2 + \frac{1}{2}tr(A^2) + y.p(y) - \nabla.p(y)$.

Proof. Write $y = (I + \epsilon A)u + \epsilon^2 p(u)$.
We have $u = (I + \epsilon A)^{-1}(y - \epsilon^2 p(y)) + O(\epsilon^3) = (I - \epsilon A + \epsilon^2 A^2)(y - \epsilon^2 p(y)) + O(\epsilon^3)$, which gives that

$$\begin{aligned} |u|^2 &= (y - \epsilon^2 p(y))^t (I - \epsilon A + \epsilon^2 A^2)^2 (y - \epsilon^2 p(y)) + O(\epsilon^3) \\ &= |y|^2 - 2\epsilon y^t Ay + \epsilon^2(2y^t A^2 y + |Ay|^2) - 2\epsilon^2 y.p(y) + O(\epsilon^3) \end{aligned} \tag{3.13}$$

After we divided equation (3.13) by 2 and taking the exponential, we will obtain

$$\begin{aligned} e^{-|u|^2/2} &= e^{-|y|^2/2 + \epsilon y^t Ay - \epsilon^2(y^t A^2 y + \frac{1}{2}|Ay|^2) + \epsilon^2 y.p(y)} \\ &= e^{-|y|^2/2} e^{\epsilon y^t Ay - \epsilon^2(y^t A^2 y + \frac{1}{2}|Ay|^2) + \epsilon^2 y.p(y)} \\ &= e^{-|y|^2/2} \left\{ 1 + \epsilon y^t Ay - \epsilon^2 \big(y^t A^2 y + \frac{1}{2}|Ay|^2\big) + \frac{\epsilon^2}{2}(y^t Ay)^2 + \epsilon^2 y.p(y) \right\} + O(\epsilon^3) \end{aligned} \tag{3.14}$$

Here we should mention that there are some small errors in the following process and because of the rapid decay of the normal density, this will not cause a problem. Now we will have the following density function of Y.

$$\begin{aligned} f_Y(y) &= \det(I + \epsilon A + \epsilon^2 \nabla p(y))^{-1} f_U(u) \\ &= (2\pi)^{-d/2} \det(I + \epsilon A + \epsilon^2 \nabla p(y))^{-1} e^{-|u|^2/2} \\ &= (2\pi)^{-d/2} \det(I + \epsilon A + \epsilon^2 \nabla p(y))^{-1} e^{-|y|^2/2} \left\{ 1 + \epsilon y^t Ay - \epsilon^2 \big(y^t A^2 y + \frac{1}{2}|Ay|^2\big) \right. \\ &\left. + \frac{\epsilon^2}{2}(y^t Ay)^2 + \epsilon^2 y.p(y) \right\} + O(\epsilon^3) \end{aligned} \tag{3.15}$$

Now from the expansion of log and the following property $e^{\mathrm{tr}(\log(I+\epsilon A+\epsilon^2\nabla p(y)))} = \det(I+\epsilon A+\epsilon^2\nabla p(y))$ we will have

$$\begin{aligned}\mathrm{tr}\log(I+\epsilon A+\epsilon^2\nabla p(y)) &= \epsilon \mathrm{tr}A - \frac{\epsilon^2}{2}\mathrm{tr}(A^2) + \epsilon^2\mathrm{tr}(\nabla p(y)) + O(\epsilon^3)\\ &= \epsilon \mathrm{tr}A - \frac{\epsilon^2}{2}\mathrm{tr}(A^2) + \epsilon^2\nabla.p(y) + O(\epsilon^3) \qquad (3.16)\end{aligned}$$

therefore

$$\begin{aligned}\det(I+\epsilon A+\epsilon^2\nabla p(y))^{-1} &= \exp\Big(-\epsilon \mathrm{tr}A + \frac{\epsilon^2}{2}\mathrm{tr}(A^2) - \epsilon^2\nabla.p(y)\Big) + O(\epsilon^3)\\ &= 1+\epsilon\mathrm{tr}A + \frac{\epsilon^2}{2}\{(\mathrm{tr}A)^2 + \mathrm{tr}(A^2)\} - \epsilon^2\nabla.p(y) + O(\epsilon^3)\end{aligned} \qquad (3.17)$$

substituting the result (3.17) in (3.15), then we get the result in (3.12). □

So the density function f_V satisfies

$$f_V(y) = (2\pi)^{-d/2}e^{-|y|^2/2}\big\{1+\epsilon^2\big(\mathcal{K}-\theta(y)+\Psi(y)+y.p(y)-\nabla.p(y)\big)\big\}+O(\epsilon^3) \qquad (3.18)$$

where $\theta(y) = \sum_{i,k=1}^{d}\theta_{ik}y_iy_k$ and $\Psi(y) = \sum_{i,j,k,m=1}^{d}\Psi_{ijkm}y_iy_jy_ky_m$.

From lemma 1 in chapter 1, the polynomial p could be chosen such that

$$\nabla.(fp(y)) = f(\Psi(y)-\theta(y)+\mu) \Rightarrow f\nabla.p(y) - y.p(y)f = f(\Psi(y)-\theta(y)+\mu)$$

This gives

$$\nabla.p(y) - y.p(y) = \Psi(y)-\theta(y)+\mu \qquad (3.19)$$

μ here is a constant. Because f_V is a density and its integral over entire space equals to one then we should have $\mu = \mathcal{K}$. Now we will verify that if we have $p = \frac{1}{2}\nabla(\frac{\nabla^2-\Psi}{4}+\theta-\frac{\Psi}{2})$ then it will satisfy (3.19). We need to find every term separately. If we have that

$$F' = \begin{bmatrix} \frac{\partial y_1}{\partial y_1} & \cdots & \frac{\partial y_d}{\partial y_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial y_d} & \cdots & \frac{\partial y_d}{\partial y_d} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1d} \\ \vdots & \ddots & \vdots \\ \theta_{d1} & \cdots & \theta_{dd} \end{bmatrix} \qquad \text{and} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_d \end{bmatrix}$$

then

$$\begin{aligned} \nabla(\theta(y)) &= \nabla(\sum_{i,k=1}^{d} \theta_{ik} y_i y_k) \\ &= \sum_{i,k=1}^{d} \theta_{ik} \nabla y_i y_k + \sum_{i,k=1}^{d} \theta_{ik} y_i \nabla y_k \\ &= F'\Theta y + F'\Theta^t y \\ &= I_d \Theta y + I_d \Theta^t y \\ &= \Theta y + \Theta^t y \end{aligned} \tag{3.20}$$

On the other hand we have

$$\begin{aligned} \Psi(y) &= \sum_{i,j,k,m=1}^{d} \Psi_{ijkm} y_i y_j y_k y_m \\ &= \sum_{i,j,k,m=1}^{d} \sum_{l}^{d} \sigma_{ikl} \sigma_{jml} y_i y_j y_k y_m \\ &= \big(\sum_{i,k=1}^{d} \sigma_{ik1} y_i y_k\big)\big(\sum_{j,m=1}^{d} \sigma_{jm1} y_j y_m\big) + \cdots + \big(\sum_{i,k=1}^{d} \sigma_{ikd} y_i y_k\big)\big(\sum_{j,m=1}^{d} \sigma_{jmd} y_j y_m\big) \\ &= \big|\sum_{i,k=1}^{d} \sigma_{ik1} y_i y_k\big|^2 + \cdots + \big|\sum_{i,k=1}^{d} \sigma_{ikd} y_i y_k\big|^2 \end{aligned}$$

and

$$\nabla(\Psi(y)) = \nabla\big(\big|\sum_{i,k=1}^{d} \sigma_{ik1} y_i y_k\big|^2 + \cdots + \big|\sum_{i,k=1}^{d} \sigma_{ikd} y_i y_k\big|^2\big)$$

Now let $\sigma_1(y) = \sum_{i,k=1}^{d} \sigma_{ik1} y_i y_k$, $\cdots$, $\sigma_d(y) = \sum_{i,k=1}^{d} \sigma_{ikd} y_i y_k$

and $A_1 = \sigma_{ik1}, \cdots, A_d = \sigma_{ikd}$ Then

$$\begin{aligned}
\nabla(\Psi(y)) &= \nabla(\sigma_1(y)^2 + \cdots + \sigma_d(y)^2) \\
&= \nabla\sigma_1(y)^2 + \cdots + \nabla\sigma_d(y)^2 \\
&= 2\sigma_1(y)\nabla\sigma_1(y) + \cdots + 2\sigma_d(y)\nabla\sigma_d(y) \\
&= 2\sigma_1(y)(A_1y + A_1^ty) + \cdots + 2\sigma_d(y)(A_dy + A_d^ty) \\
&= 2\sigma_1(y)B_1 + \cdots + 2\sigma_d(y)B_d
\end{aligned}$$

Where $B_1 = (A_1y + A_1^ty), \cdots, B_d = (A_dy + A_d^ty)$

$$\begin{aligned}
\nabla^2(\Psi(y)) &= \nabla(2\sigma_1(y)B_1 + \cdots + 2\sigma_d(y)B_d) \\
&= 2\nabla\sigma_1(y)B_1 + 2\sigma_1(y)\nabla B_1 + \cdots + 2\nabla\sigma_d(y)B_d + 2\sigma_d(y)\nabla B_d \\
&= 2(A_1y + A_1^ty)^tB_1 + 4\sigma_1(y)(tr(A_1)) + \cdots + 2(A_dy + A_d^ty)^tB_d \\
&+ 4\sigma_d(y)(tr(A_d)) \\
&= 2y^t\big[A_1^tA_1 + A_1^2 + 2A_1(tr(A_1)) + (A_1^t)^2 + A_1A_1^t\big]y + \cdots \\
&+ 2y^t\big[A_d^tA_d + A_d^2 + 2A_d(tr(A_d)) + (A_d^t)^2 + A_dA_d^t\big]y \\
&= 2y^tE_1y + \cdots + 2y^tE_dy \\
&= 2y^t(E_1 + \cdots + E_d)y \\
&= 2y^tEy
\end{aligned}$$

where $E_1 = A_1^tA_1 + A_1^2 + 2A_1(tr(A_1)) + (A_1^t)^2 + A_1A_1^t$,
$E_d = A_d^tA_d + A_d^2 + 2A_d(tr(A_d)) + (A_d^t)^2 + A_dA_d^t$
and $E = E_1 + \cdots + E_d$

Finally we need to find $\nabla[\nabla^2(\Psi(y))]$

$$\begin{aligned}
\nabla[\nabla^2(\Psi(y))] &= \nabla\big(2y^tEy\big) \\
&= 2(Ey + E^ty)
\end{aligned}$$

So from (3.19) we have

$$\begin{aligned}\nabla.p(y) - y.p(y) &= -E + \frac{(\Theta + \Theta^t)}{2} - y^tEy - y^t[-Ey + \frac{(\Theta + \Theta^t)y}{2} \\ &\quad - (\sigma_1(y)B_1 + \cdots + \sigma_d(y)B_d)] \\ &= -E + \frac{(\Theta + \Theta^t)}{2} - y^tEy + y^tEy - \frac{y^t(\Theta + \Theta^t)y}{2} \\ &\quad + (\sigma_1(y)^2 + \cdots + \sigma_d(y)^2) \\ &= \Psi(y) - \theta(y) + \mu \end{aligned} \tag{3.21}$$

After we find the derivation of p we could see the ϵ^2 term will equal zero, also f_V is an even function of ϵ, therefore we will not have an ϵ^3 term in its expansion, so $f_V(y) = (2\pi)^{-d/2}e^{-|y|^2/2} + O(\epsilon^4)$. We also see that this Vsatisfies (2.15).
Now we need to expand the density f_V further as

$$f_V(y) = \phi(y)(1 + q(y)) + O(\epsilon^8)$$

where $\phi(y)$ is the density of standard normal distribution in $\mathbb{R}^d$. i.e. $\phi(y) = (2\pi)^{-d/2}e^{-|y|^2/2}$ and q is a polynomial in ϵ and y and its expansion has ϵ^4 and ϵ^6 terms but the dominant term will be of order ϵ^4, i.e. it has $O(\epsilon^4)$. Then from lemma 2 we can deduce that the distance ($\mathbb{W}_2$) between the random variables V and $\tilde{V} = N(0, I)$ will be of $O(\epsilon^4)$. This means $\tilde{V}$ will be coupled to V so that

$$E|V - \tilde{V}|^2 = O(\epsilon^8) \tag{3.22}$$

The purpose of the following discussion is to show how we could use the empirical estimate as in section 1.5.1 with the approximate coupling to get an estimate for the error using the exact coupling $\tilde{V}$ and hence we get an empirical upper bound for the Vaserstein distance between the approximate solutions at two levels.
So if we need to generate coupled approximate solutions $\tilde{x}^{(r,j)}$ and $\tilde{x}^{(r+1,2j)}$ at two different levels r and $r+1$ then we could use the above definitions of V, U, U^*. Because V does not have the exact normal distribution $N(0, I)$, therefore we will not get the true implementation for $\tilde{x}^{(r,j)}$ in (2.7). We could get the true implementation of

(2.7) which we will call $\bar{x}^{(r,j)}$ by substituting V by $\tilde{V}$, but we do not have a means of generating it jointly with the level $r+1$ solution, therefore we use $\tilde{x}^{(r,j)}$ as an approximation. As we have done before, we have that the bound between $\tilde{x}^{(r,j)}$ and $\tilde{x}^{(r+1,2j)}$ is $O(h)$, and from (3.22) we obtain the bound $\bar{x}^{(r,j)} - \tilde{x}^{(r,j)} = O(h^2)$.

Now we need to estimate the error as we describe in section 5 in chapter one. If we have that N is the total number of steps at level r, we need to estimate the following $E|\bar{x}^{(r,N)} - \bar{x}^{(r+1,2N)}|$ where $\bar{x}^{(r,N)}$ is the true implementation of (2.7). But as we mentioned before that we could estimate empirically $E|\tilde{x}^{(r,N)} - \tilde{x}^{(r+1,2N)}|$ by using the approximate coupling method. After that we could get $O(h^2)$ bound between $E|\bar{x}^{(r,N)} - \bar{x}^{(r+1,2N)}|$ and $E|\tilde{x}^{(r,N)} - \tilde{x}^{(r+1,2N)}|$. As we expect $E|\bar{x}^{(r,N)} - \bar{x}^{(r+1,2N)}|$ to be of order h, so the error h^2 between them should be negligible for small h, hence the approximate method is effective for empirical estimation

In following section we will show the numerical results of the approximate coupling for the scheme (1.22) which support the theoretical results. I would like to mention that the following implementation will be for 2-dimensional stochastic differential equation but the codes could be applied to d-dimensional SDEs.

3.1 The implementation of an approximate coupling in two-dimensional case with invertible $\left(b_{ik}(\tilde{x}^{(r+1,2j)})\right)$

First of all, we have the 2-dimensional SDE, which is invertible.

$$\begin{aligned} dX_1(t) &= (\sin(X_2(t)))^2 dW_1(t) - \frac{1}{1+X_1^2(t)} dW_2(t), \\ dX_2(t) &= \frac{1}{1+X_2^4(t)} dW_1(t) + (\cos(X_1(t)))^2 dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{3.23}$$

where $W_1(t)$ and $W_2(t)$ are independent standard Brownian motion.

To apply a numerical method to this SDE we need to simulate solutions (for the same Brownian path) simultaneously using two different step sizes (h and $h/2$).

The Matlab implementation for this SDE using the approximate coupling is given in Listing (5.9), which will give us the absolute value of the difference between two solutions for the SDEs (3.23) with step size h and $h/2$.

To construct this experiment, we will decrease the step size (h) every time when we calculate the error and examine the convergence properties of the exact coupling method. We will repeat this with a different step size using (for example, $R = 400$) independent simulations. So the order of convergence of this method between two solution should be 1.

Now we will run the Matlab code (AproxcouplingInverend) in Listing (5.9) with different step sizes over large number of paths R as described in the table below and see the result of the error ϵ, where each simulation is for the same Brownian path and $\epsilon = \frac{1}{R}\sum_{i=1}^{R} |x_h^{(i)} - x_{h/2}^{(i)}|$ will be our estimator.

step-size	error(ϵ)
0.005	0.0058
0.0025	0.0031
0.00125	0.0016
0.00062	0.00076
0.00031	0.00036
0.00015	0.00018
0.00007	0.000097

Table 3.1: The error results for the approximate coupling with invertible matrix

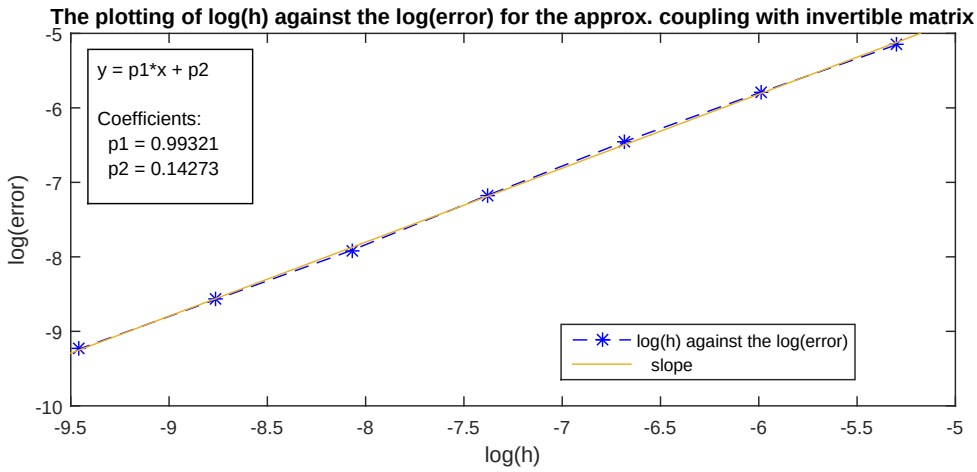


Figure 3.1: log Error against log of step sizes for approx. coupling with invertible matrix

The table (3.1) and the plotting in Figure (3.1) show the implementation of the approximate solutions of the previous 2-dimensional SDEs for the approximate coupling with different number of steps (200, 400, 800, 1600, 3200, 6400, 12800). Running the code (AproxcouplingInverend) for 400 simulations gives a value for its estimator ϵ equal to 0.0058 with the step-size 0.005 i.e.

$$\epsilon = \frac{1}{400}\sum_{i=1}^{400} |x_h^{(i)} - x_{h/2}^{(i)}| = 0.0058$$

and 0.0031 with step-size 0.0025 and so on. This means when we increase the number of steps which each time gives a smaller step-size then the estimate error ϵ will give $O(h)$ as it appears in the results in table (3.1). Also the Figure (3.1) is the plot of the log of the estimator ϵ i.e. $\log(\epsilon)$ against the *log* of step-size (h) i.e. $\log(h)$ which has a slope of 0.99321 which again indicates a strong convergence of $O(h)$ for the stochastic differential equation (3.23).

Therefore from this computational results we could see that we have obtained good agreement between the theoretical bound in (2.33) with the implementation results.

Chapter 4

Combined Method

In the previous chapters we have assumed that the matrix $\big(b_{ik}(x)\big)$ is invertible for all x, but in this chapter we will show how we could control the matrix which is non-invertible for some x using the combined method. For the combined method we will use a two-level coupling method for 2-dimensional SDEs using scheme (1.22). In the little box below, we explain how the combined method will work.

Explanation of the combined method

At the j^{th} step we need to calculate the value of a which is a function of $x^{(r,j)}$ and also we need to calculate K_1 and K_2 in (4.30) which are also functions of $x^{(r,j)}$, then in the same stage we have two choices of approximate solutions. The first one is the approximate solution using scheme (1.22) with the exact coupling which will give local error $E|x^{(r,1)} - x^{(r+1,2)}|^2 \leq K_2 a^2 h^3$
or the second approximate solution that using scheme (1.22) with the trivial coupling which will give local error $E|x^{(r,1)} - x^{(r+1,2)}|^2 \leq K_1 h^2$. So from the value of a and using the following condition that if $K_2 a^2 h^3 > K_1 h^2$ then we choose the solution which has scheme (1.22) with the trivial coupling and if not we use the other solution which has scheme (1.22) with the exact coupling.

We remark that for the implementation of exact coupling with a non-invertible

matrix could not apply to scheme (1.22) directly because we have the matrix $\big(b_{ik}(x)\big)$ which is singular or has determinant near to zero which will effect the convergence order. In the other words that means we will not get the inverse matrix c_{ij} in the following term $\tau_{ik\ell} = \frac{1}{2}\sum_j c_{ij}\left\{\frac{b_{jk}(\tau_n,\Upsilon_n^j)-b_{jk}}{\sqrt{h}}\right\}$ at some points. Therefore we could control this problem by using the condition which has mentioned in the previous box.

We now indicate how will the local error for the combined method behave and what the local error will be achieved. We will show this theoretically and then numerically with examples of implementation for a specific non-invertible stochastic differential equation.

Now we want to show the derivation of the local error for the combined method.

4.1 Derivation and implementation of the local error of the Combined Method

In the combined method we will use the local error for the scheme (1.22) with the exact coupling and the local error for scheme (1.22) with the trivial coupling. Therefore before we start the derivation of the local error for the combined method we want to find the local error for the exact coupling and the trivial coupling.

4.1.1 Evaluation of the local error for the scheme (1.22) with exact coupling

We need to find the explicit value for the local error for $E|x_i^{(r,1)} - x_i^{(r+1,2)}|^2$ from the error which we obtain from $E(V_1' - Y)^2 \leq 10a^4\epsilon^4$. It is possible to deduce that from equation (2.16). Firstly, from equation (2.16) we have

$$V_1' = U_1' + \epsilon a(\tilde{U}_2 U_1' - \tilde{U}_1 U_2') + Ra^2\epsilon^2 \tag{4.1}$$

Where R is a random variable and $E(R^2) = \frac{E(V_1'-Y)^2}{a^4\epsilon^4} \approx 10$.

Then after we multiply by the term b and rotation matrix R_θ, we obtain

$$bR_\theta V' = bR_\theta U' + \epsilon bR_\theta \begin{pmatrix} a \\ 0 \end{pmatrix}(U_2^* U_1 - U_1^* U_2) + R\epsilon^2 bR_\theta \begin{pmatrix} a^2 \\ 0 \end{pmatrix}$$

Then, this will give us

$$\begin{aligned}
(bV)_i &= (bU)_i + \epsilon b_i \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}(U_2^* U_1 - U_1^* U_2) + R\epsilon^2 a b_i \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \\
\Rightarrow (bV)_i &= (bU)_i + \epsilon \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})(U_l^* U_k - U_k^* U_l) + Ra\epsilon^2(\rho_{i12} - \rho_{i21}) \\
\epsilon \sum_{k=1}^{2} b_{ik}(x^{(0)}) V_k &= \epsilon \sum_{k=1}^{2} b_{ik}(x^{(0)})(Y_k + Z_k)) + \epsilon^2 \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})[Z_k Y_l - Z_l Y_k] \\
&\quad + Ra\epsilon^3(\rho_{i12} - \rho_{i21}) \\
\sum_{k=1}^{2} b_{ik}(x^{(0)}) X_k^{(r,0)} &= \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)})) \\
&\quad + \frac{1}{2} \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})[X_k^{(r+1,1)} X_l^{(r+1,0)} - X_l^{(r+1,1)} X_k^{(r+1,0)}] \\
&\quad + Ra(\rho_{i12} - \rho_{i21})h^{3/2}
\end{aligned} \tag{4.2}$$

After we have obtained the coupling in (4.2) we could use it in the approximate solution $\tilde{x}_i^{(r,1)}$ in (2.16) and find the reminder term. So the coupling will be

$$\begin{aligned}
X_i^{(r,0)} &= X_i^{(r+1,0)} + X_i^{(r+1,1)} + (\tau_{i12} - \tau_{i21})(X_1^{(r+1,1)} X_2^{(r+1,0)} - X_2^{(r+1,1)} X_1^{(r+1,0)}) + Raa_i\epsilon^3 \\
&= X_i^{(r+1,0)} + X_i^{(r+1,1)} + Ua_i + Raa_i\epsilon^3
\end{aligned} \tag{4.3}$$

Where $U = (X_1^{(r+1,1)} X_2^{(r+1,0)} - X_2^{(r+1,1)} X_1^{(r+1,0)})$, so from the approximate solution

$$\tilde{x}_i^{(r,1)} = x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)}) X_k^{(r,0)} + \frac{1}{2} \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})(X_k^{(r,0)} X_l^{(r,0)} - h^{(r)}\delta_{kl}) \tag{4.4}$$

we will have

$$\tilde{x}_i^{(r,1)} = x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)} + Ua_k) + Ra(\rho_{i12} - \rho_{i21})h^{3/2}$$
$$+ \frac{1}{2}\sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})\big[\big(X_k^{(r+1,0)} + X_k^{(r+1,1)} + Ua_k\big)\big(X_l^{(r+1,0)} + X_l^{(r+1,1)} + Ua_l\big) - h^{(r)}\delta_{kl}\big] \tag{4.5}$$

From this equation and after we find the multiplication we will have the following bound

$$E\bigg(\sum_{k,l=1}^{2} \rho_{ikl}\big(\big[(X_k^{(r+1,0)} + X_k^{(r+1,1)})Ua_l\big] + \big[(X_l^{(r+1,0)} + X_l^{(r+1,1)})Ua_k\big]\big)\bigg)^2$$
$$= E\bigg(\sum_{k,l=1}^{2} \big(\rho_{ikl} + \rho_{ilk}\big)\big(X_k^{(r+1,0)} + X_k^{(r+1,1)}\big)a_l U\bigg)^2$$
$$= E\bigg(2\rho_{i11}(X_1^{(r+1,0)} + X_1^{(r+1,1)})a_1 U + (\rho_{i12} + \rho_{i21})(X_1^{(r+1,0)} + X_1^{(r+1,1)})a_2 U$$
$$+ (\rho_{i21} + \rho_{i12})(X_2^{(r+1,0)} + X_2^{(r+1,1)})a_1 U + 2\rho_{i22}(X_2^{(r+1,0)} + X_2^{(r+1,1)})a_2 U\bigg)^2$$
$$= 32\rho_{i11}^2 a_1^2 h^3 + 8(\rho_{i12} + \rho_{i21})^2 a_2^2 h^3 + 8(\rho_{i21} + \rho_{i12})^2 a_1^2 h^3 + 32\rho_{i22} a_2^2 h^3$$
$$+ 32\rho_{i11}(\rho_{i12} + \rho_{i21})a_1 a_2 h^3 + 32\rho_{i22}(\rho_{i12} + \rho_{i21})a_1 a_2 h^3$$
$$= 32\sum_{k=1}^{2} \rho_{ikk}^2 a_k^2 h^3 + 8\sum_{k\neq m}^{2} (\rho_{ikm} + \rho_{imk})^2 a_m^2 h^3 + 32\sum_{k\neq m}^{2} \rho_{ikk}(\rho_{ikm} + \rho_{imk})a_m a_k h^3$$
$$= \sum_{k\neq m}^{2} \big[\sqrt{32}\rho_{ikk} a_k + \big(\sqrt{8}(\rho_{ikm} + \rho_{imk})a_m\big)\big]^2 h^3 \tag{4.6}$$

Thus from equation (4.3) and (4.4) we will have

$$\begin{aligned}
\tilde{x}_i^{(r,1)} &= x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)} + Ua_k) + Ra(\rho_{i12} - \rho_{i21})h^{3/2} \\
&+ \frac{1}{2}\sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})\big[X_k^{(r+1,0)}X_l^{(r+1,0)} + X_k^{(r+1,0)}X_l^{(r+1,1)} + X_k^{(r+1,1)}X_l^{(r+1,0)} \\
&+ X_k^{(r+1,1)}X_l^{(r+1,1)} - h^{(r)}\delta_{kl}\big] + \sum_{k,l=1}^{2}\big(\rho_{ikl} + \rho_{ilk}\big)\big(X_k^{(r+1,0)} + X_k^{(r+1,1)}\big)a_l U \\
&= x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) + \frac{1}{2}\sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})[X_k^{(r+1,1)}X_l^{(r+1,0)} \\
&- X_l^{(r+1,1)}X_k^{(r+1,0)}] + Ra(\rho_{i12} - \rho_{i21})h^{3/2} + \frac{1}{2}\sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})\big[X_k^{(r+1,0)}X_l^{(r+1,0)} \\
&+ X_k^{(r+1,0)}X_l^{(r+1,1)} + X_k^{(r+1,1)}X_l^{(r+1,0)} + X_k^{(r+1,1)}X_l^{(r+1,1)} - h^{(r)}\delta_{kl}\big] \\
&+ \sum_{k,l=1}^{2}\big(\rho_{ikl} + \rho_{ilk}\big)\big(X_k^{(r+1,0)} + X_k^{(r+1,1)}\big)a_l U \\
&= x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) + \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})[X_k^{(r+1,1)}X_l^{(r+1,0)}] \\
&+ Ra(\rho_{i12} - \rho_{i21})h^{3/2} \\
&+ \frac{1}{2}\sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})\big[X_k^{(r+1,0)}X_l^{(r+1,0)} + X_k^{(r+1,1)}X_l^{(r+1,1)} - h^{(r)}\delta_{kl}\big] \\
&+ \sum_{k,l=1}^{2}\big(\rho_{ikl} + \rho_{ilk}\big)\big(X_k^{(r+1,0)} + X_k^{(r+1,1)}\big)a_l U \qquad (4.7)
\end{aligned}$$

And we have

$$\begin{aligned}
\tilde{x_i}^{(r+1,2)} &= {x_i}^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) \\
&\qquad + \sum_{l,k=1}^{2} \rho_{ikl}(x^{(0)})X_k^{(r+1,1)}X_l^{(r+1,0)} \\
&+ \frac{1}{2}\sum_{l,k=1}^{2} \rho_{ikl}(x^{(0)})(X_k^{(r+1,0)}X_l^{(r+1,0)} + X_k^{(r+1,1)}X_l^{(r+1,1)} - h^{(r)}\delta_{kl}) + O((h^{(r)})^{3/2}) \qquad (4.8)
\end{aligned}$$

Finally we compare $\tilde{x}_i^{(r,1)}$ with $\tilde{x}_i{}^{(r+1,2)}$ to obtain the local error. i.e.

$$
\begin{aligned}
(\tilde{x}_i^{(r,1)} - \tilde{x}_i{}^{(r+1,2)}) &= \Bigg(x_i^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) \\
&\quad + \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)})[X_k^{(r+1,1)} X_l^{(r+1,0)}] + Ra(\rho_{i12} - \rho_{i21})h^{3/2} \\
&\quad + \frac{1}{2} \sum_{k,l=1}^{2} \rho_{ikl}(x^{(0)}) \big[X_k^{(r+1,0)} X_l^{(r+1,0)} + X_k^{(r+1,1)} X_l^{(r+1,1)} - h^{(r)} \delta_{kl} \big] \\
&\quad + \sum_{k,l=1}^{2} \big(\rho_{ikl} + \rho_{ilk} \big) \big(X_k^{(r+1,0)} + X_k^{(r+1,1)} \big) a_l U \Bigg) \\
&\quad - \Bigg(x_i{}^{(0)} + \sum_{k=1}^{2} b_{ik}(x^{(0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) \\
&\quad + \sum_{l,k=1}^{2} \rho_{ikl}(x^{(0)}) X_k^{(r+1,1)} X_l^{(r+1,0)} \\
&\quad + \frac{1}{2} \sum_{l,k=1}^{2} \rho_{ikl}(x^{(0)})(X_k^{(r+1,0)} X_l^{(r+1,0)} + X_k^{(r+1,1)} X_l^{(r+1,1)} - h^{(r)} \delta_{kl}) \\
&\quad + O((h^{(r)})^{3/2}) \Bigg) \\
&= Ra(\rho_{i12} - \rho_{i21})h^{3/2} + \sum_{k,l=1}^{2} \big(\rho_{ikl} + \rho_{ilk} \big) \big(X_k^{(r+1,0)} + X_k^{(r+1,1)} \big) a_l U
\end{aligned}
\tag{4.9}
$$

So

$$
\begin{aligned}
E|x_i^{(r,1)} - x_i^{(r+1,2)}|^2 &= E|Ra(\rho_{i12} - \rho_{i21})h^{3/2} + \sum_{k,l=1}^{2} \big(\rho_{ikl} + \rho_{ilk} \big) \big(X_k^{(r+1,0)} + X_k^{(r+1,1)} \big) a_l U|^2 \\
&\leq \big\{ 10a^2(\rho_{i12} - \rho_{i21})^2 + \sum_{k \neq m}^{2} \big[\sqrt{32} \rho_{ikk} a_k + \big(\sqrt{8}(\rho_{ikm} + \rho_{imk}) a_m \big) \big]^2 \big\} h^3 \\
&\quad + E \Bigg[\big(Ra(\rho_{i12} - \rho_{i21})h^{3/2} \big) \big(\sum_{k,l=1}^{2} \big(\rho_{ikl} + \rho_{ilk} \big) \big(X_k^{(r+1,0)} + X_k^{(r+1,1)} \big) a_l U \big) \Bigg] \\
&= \big\{ 10a^2(\rho_{i12} - \rho_{i21})^2 + \sum_{k \neq m}^{2} \big[\sqrt{32} \rho_{ikk} a_k + \big(\sqrt{8}(\rho_{ikm} + \rho_{imk}) a_m \big) \big]^2 \big\} h^3
\end{aligned}
\tag{4.10}
$$

In the last step we have that the expectation of the crossing term equals zero, i.e.

$$E\Big[\big(Ra(\rho_{i12}-\rho_{i21})h^{3/2}\big)\big(\sum_{k,l=1}^{2}\big(\rho_{ikl}+\rho_{ilk}\big)\big(X_k^{(r+1,0)}+X_k^{(r+1,1)}\big)a_lU\big)\Big]=0 \quad (4.11)$$

Because from equation (4.1) we have that the random variable R equal to

$$\begin{aligned} R &= \frac{V_1'-U_1'-\epsilon a(\tilde{U}_2U_1'-\tilde{U}_1U_2')}{a^2\epsilon^2} \\ &= \frac{V_1'-U_1'-\epsilon a(\tilde{U}_2U_1'-\tilde{U}_1U_2')}{a^2h} \end{aligned} \quad (4.12)$$

and

$$\begin{aligned} &\sum_{k,l=1}^{2}\big(\rho_{ikl}+\rho_{ilk}\big)\big(X_k^{(r+1,0)}+X_k^{(r+1,1)}\big)a_lU \\ &=\sum_{k,l=1}^{2}\rho_{ikl}\big(\big[(X_k^{(r+1,0)}+X_k^{(r+1,1)})Ua_l\big]+\big[(X_l^{(r+1,0)}+X_l^{(r+1,1)})Ua_k\big]\big) \\ &=[2\rho_{i11}U_1'a_1+(\rho_{i12}+\rho_{i21})U_1'a_2+(\rho_{i21}+\rho_{i12})U_2'a_1+2\rho_{i22}(U_2')a_2]\epsilon U \\ &=\big([2\rho_{i11}a_1+(\rho_{i12}+\rho_{i21})a_2]U_1'+[(\rho_{i21}+\rho_{i12})a_1+2\rho_{i22}a_2]U_2'\big)\epsilon U \\ &=\big(L_1U_1'+L_2U_2'\big)\epsilon U \\ &=\big(L_1U_1'+L_2U_2'\big)\epsilon(\tilde{U}_2U_1'-\tilde{U}_1U_2') \end{aligned} \quad (4.13)$$

where $L_1=2\rho_{i11}a_1+(\rho_{i12}+\rho_{i21})a_2$ and $L_2=(\rho_{i21}+\rho_{i12})a_1+2\rho_{i22}a_2$.
Replacing (4.12) and (4.13) in (4.11) we will obtain the following

$$\begin{aligned} &E\big[\big(\big(\frac{V_1'-U_1'-\epsilon a(\tilde{U}_2U_1'-\tilde{U}_1U_2')}{a^2h}\big)a(\rho_{i12}-\rho_{i21})h^{3/2}\big)\big(L_1U_1'+L_2U_2'\big)\epsilon(\tilde{U}_2U_1'-\tilde{U}_1U_2')\big] \\ &=E\big[\frac{2}{a}(\rho_{i12}-\rho_{i21})h\big(V_1'-U_1'-\epsilon a(\tilde{U}_2U_1'-\tilde{U}_1U_2')\big)\big(L_1U_1'\tilde{U}_2U_1'-L_1U_1'\tilde{U}_1U_2' \\ &+L_2U_2'\tilde{U}_2U_1'-L_2U_2'\tilde{U}_1U_2'\big)\big] \\ &=E\big[\frac{2}{a}(\rho_{i12}-\rho_{i21})h\big(L_1U_1'\tilde{U}_2U_1'V_1'-L_1U_1'\tilde{U}_1U_2'V_1'+L_2U_2'\tilde{U}_2U_1'V_1'-L_2U_2'\tilde{U}_1U_2'V_1' \\ &+L_1U_1'\tilde{U}_2U_1'U_1'-L_1U_1'\tilde{U}_1U_2'U_1'+L_2U_2'\tilde{U}_2U_1'U_1'-L_2U_2'\tilde{U}_1U_2'U_1'+L_1U_1'\tilde{U}_2U_1'\tilde{U}_2U_1' \\ &-L_1U_1'\tilde{U}_1U_2'\tilde{U}_2U_1'+L_2U_2'\tilde{U}_2U_1'\tilde{U}_2U_1'-L_2U_2'\tilde{U}_1U_2'\tilde{U}_2U_1'+L_1U_1'\tilde{U}_2U_1'\tilde{U}_1U_2' \\ &-L_1U_1'\tilde{U}_1U_2'\tilde{U}_1U_2'+L_2U_2'\tilde{U}_2U_1'\tilde{U}_1U_2'-L_2U_2'\tilde{U}_1U_2'\tilde{U}_1U_2'\big)\big] \end{aligned} \quad (4.14)$$

Now we need to find the expectation of each term separately. As we mention before in section 2.1.1, we start by generating independent $N(0,1)$ variables U_1', U_2', Q, R and α taking the value ± 1 with probability $\frac{1}{2}$ each. Then set $V_2' = U_2'$, $\tilde{U}_1 = \alpha Q$ and $\tilde{U}_2 = \alpha R$. Also as we defined $\mathcal{Y} = U_1' + \alpha(bU_1' + c)$ and $V_1' = \Phi^{-1}(F(\mathcal{Y}))$ where $F(y)$ is the c.d.f. of $\mathcal{Y}$ (here Φ is the c.d.f of $N(0,1)$, where $b = \epsilon a R$ and $c = -\epsilon a Q U_2'$. This gives $\mathcal{Y} = U_1' + \epsilon a(\tilde{U}_2 U_1' - \tilde{U}_1 U_2')$. So we can change the sign of the random variables U_1', U_2', Q, R by multiplying by -1 of any subset of them without altering the distribution. So if we only change the sign of the random variables U_2', and Q, then b and c will not change and also $\mathcal{Y}$ will not change, so V_1' will not change. Therefore all the following terms will equal zero i.e.

$$E(-U_2'\tilde{U}_2 U_1' V_1') = E(U_2'\tilde{U}_2 U_1' V_1') = 0 \tag{4.15}$$

$$E(-U_2'\tilde{U}_1 U_2' V_1') = E(U_2'\tilde{U}_1 U_2' V_1') = 0 \tag{4.16}$$

On the other hand, if we change the sign of U_1', and U_2', then we will get $-c$ and $-\mathcal{Y}$ and hence we will have the c.d.f. of $-\mathcal{Y}$ which we call $\bar{F}(y)$, so

$$\begin{aligned} \bar{F}(y) &= P(-\mathcal{Y} \leq y) \\ &= P(\mathcal{Y} \geq -y) \\ &= 1 - F(-y) \end{aligned}$$

so we will have

$$\Phi^{-1}(\bar{F}(-\mathcal{Y})) = \Phi^{-1}(1 - F(\mathcal{Y})) = -\Phi^{-1}(F(\mathcal{Y})) = -V_1'$$

therefore the following terms will equal zero i.e.

$$E(-U_1'\tilde{U}_2 U_1' V_1') = E(U_1'\tilde{U}_2 U_1' V_1') = 0 \tag{4.17}$$

$$E(-U_1'\tilde{U}_1 U_2' V_1') = E(U_1'\tilde{U}_1 U_2' V_1') = 0 \tag{4.18}$$

For the rest of the expectations the result will be zero because all the random variables are mutually independent, so

$$E\big[\frac{2}{a}(\rho_{i12}-\rho_{i21})h\big(L_1U_1'\tilde{U}_2U_1'U_1'-L_1U_1'\tilde{U}_1U_2'U_1'+L_2U_2'\tilde{U}_2U_1'U_1'-L_2U_2'\tilde{U}_1U_2'U_1' \\ +L_1U_1'\tilde{U}_2U_1'\tilde{U}_2U_1'-L_1U_1'\tilde{U}_1U_2'\tilde{U}_2U_1'+L_2U_2'\tilde{U}_2U_1'\tilde{U}_2U_1'-L_2U_2'\tilde{U}_1U_2'\tilde{U}_2U_1' \\ +L_1U_1'\tilde{U}_2U_1'\tilde{U}_1U_2'-L_1U_1'\tilde{U}_1U_2'\tilde{U}_1U_2'+L_2U_2'\tilde{U}_2U_1'\tilde{U}_1U_2'-L_2U_2'\tilde{U}_1U_2'\tilde{U}_1U_2'\big)\big]=0 \tag{4.19}$$

Finally, we have shown from (4.17) to (4.19) that the expectation of all terms in the crossing term equal zero and then the result of equation (4.11) will be zero i.e.

$$E\Big[\big(Ra(\rho_{i12}-\rho_{i21})h^{3/2}\big)\big(\sum_{k,l=1}^{2}\big(\rho_{ikl}+\rho_{ilk}\big)\big(X_k^{(r+1,0)}+X_k^{(r+1,1)}\big)a_lU\big)\Big]=0$$

4.1.2 Evaluation of the local error of scheme (1.22) with trivial coupling

First of all, we compare $\tilde{x}_k^{(r,j+1)}$ with $\tilde{x}_k^{(r+1,2j+2)}$, we have

$$\tilde{x}_i^{(r,1)}=\tilde{x}_i^{(r,0)}+\sum_{k=1}^{d}b_{ik}(\tilde{x}^{(r,0)})X_k^{(r,0)}+\frac{1}{2}\sum_{k,l=1}^{d}\rho_{ikl}(\tilde{x}^{(r,0)})(X_k^{(r,0)}X_l^{(r,0)}-h^{(r)}\delta_{kl}) \tag{4.20}$$

and suppose we have another approximate solution, i.e.

$$y=\tilde{x}_i^{(r+1,2j)}+\sum_{k=1}^{d}b_{ik}(\tilde{x}^{(r+1,2j)})X_k^{(r,j)}+\frac{1}{2}\sum_{k,l=1}^{d}\rho_{ikl}(\tilde{x}^{(r+1,2j)})(X_k^{(r,j)}X_l^{(r,j)}-h^{(r)}\delta_{kl}) \tag{4.21}$$

And also

$$\tilde{x}_i^{(r+1,1)}=\tilde{x}_i^{(r+1,0)}+\sum_{k=1}^{d}b_{ik}(\tilde{x}^{(r+1,0)})X_k^{(r+1,0)} \\ +\frac{1}{2}\sum_{k,l=1}^{d}\rho_{ikl}(\tilde{x}^{(r+1,0)})(X_k^{(r+1,0)}X_l^{(r+1,0)}-h^{(r+1)}\delta_{kl}) \tag{4.22}$$

$$\tilde{x}_i^{(r+1,2)} = \tilde{x}_i^{(r+1,1)} + \sum_{k=1}^{d} b_{ik}(\tilde{x}^{(r+1,1)}) X_k^{(r+1,1)} + \frac{1}{2}\sum_{k,l=1}^{d} \rho_{ikl}(\tilde{x}^{(r+1,1)})(X_k^{(r+1,1)} X_l^{(r+1,1)} - h^{(r+1)}\delta_{kl}) \tag{4.23}$$

Now, $b_{ik}(\tilde{x}^{(r+1,1)}) = b_{ik}(\tilde{x}^{(r+1,0)}) + \rho_{ikl}(\tilde{x}^{(r+1,0)})(X_k^{(r+1,0)}) + O(h)$

and $\rho_{ikl}(\tilde{x}^{(r+1,1)}) = \rho_{ikl}(\tilde{x}^{(r+1,0)}) + O(h^{1/2})$

Using these relations in (4.23) and combining it with (4.22) we get

$$\tilde{x_i}^{(r+1,2)} = \tilde{x_i}^{(r+1,0)} + \sum_{k=1}^{d} b_{ik}(\tilde{x_i}^{(r+1,0)})(X_k^{(r+1,0)} + X_k^{(r+1,1)}) + \sum_{l,k=1}^{d} \rho_{ikl}(\tilde{x}^{(r+1,0)}) X_k^{(r+1,1)} X_l^{(r+1,0)} + \frac{1}{2}\sum_{l,k=1}^{d} \rho_{ikl}(\tilde{x}^{(r+1,0)})(X_k^{(r+1,0)} X_l^{(r+1,0)} + X_k^{(r+1,1)} X_l^{(r+1,1)} - h^{(r)}\delta_{kl}) + O((h^{(r)})^{3/2}) \tag{4.24}$$

Then the coupling will satisfy

$$\tilde{x}_k^{(r,1)} - \tilde{x}_k^{(r+1,2)} = \sum_{k,l=1}^{d} \rho_{ikl}(X_k^{(r+1,1)} X_l^{(r+1,0)} - X_l^{(r+1,1)} X_k^{(r+1,0)}) + O((h^{(r)})^{3/2}) \tag{4.25}$$

Now we will reformulate (4.25) by a scaling. We fix r write $\epsilon = (h^{(r)})^{1/2}$, $X_i^{(r+1,0)} = \epsilon Y_i$ and $X_i^{(r+1,1)} = \epsilon Z_i$. Then $(Y_1, \cdots, Y_d, Z_1, \cdots, Z_d)$ are independent and $N(0, 1/2)$. So that

$$\tilde{x}_k^{(r,1)} - \tilde{x}_k^{(r+1,2)} = \epsilon \sum_{k,l=1}^{d} \rho_{ikl}(Z_k Y_l - Z_l Y_k) + O(\epsilon^2) \tag{4.26}$$

and

$$E(Z_1 Y_2 - Z_2 Y_1)^2 = E(Z_1 Y_2)^2 - 2E(Z_1 Y_2)E(Z_2 Y_1) + E(Z_2 Y_1)^2 = 1 + 1 = 2 \tag{4.27}$$

therefore the local error for the trivial coupling will be

$$E|x_i^{(r,1)} - x_i^{(r+1,2)}|^2 = E\Big|\sum_{k,l=1}^{d} \rho_{ikl}(X_k^{(r+1,1)}X_l^{(r+1,0)} - X_l^{(r+1,1)}X_k^{(r+1,0)})\Big|^2$$
$$= 2(\rho_{i12}^2 + \rho_{i21}^2)h^2 + O(h^3) \tag{4.28}$$

Then we follow the same procedure to get the local error for the j^{th} step.

So we need to compare the local error for scheme (1.22) with exact and trivial coupling in the same starting points. As we have mentioned before that the local error will work for the j^{th} step as the initial step and hence we have obtained that the local error for scheme (1.22) with exact coupling

$$E|y - x^{(r+1,2j+2)}|^2$$
$$\leq \big\{10a^2\sum_{i=1}^{2}(\rho_{i12} - \rho_{i21})^2 + \sum_{i=1,k\neq m}^{2}\big[\sqrt{32}\rho_{ikk}a_k + \big(\sqrt{8}(\rho_{ikm} + \rho_{imk})a_m\big)\big]^2\big\}h^3$$

and for the trivial coupling is

$E|y - x^{(r+1,2j+2)}|^2 \leq 2\sum_{i=1}^{2}(\rho_{i12}^2 + \rho_{i21}^2)h^2$

So as we have mentioned in the previous box that from the value of $a^2 = (a_1^2 + a_2^2)$ which defines as a function of $a_i = \frac{1}{2}\sum_j c_{ij}(x^{(r,j)})\rho_{ikl}(x^{(r,j)})$ and if $K_1 = 2\sum_{i=1}^{2}(\rho_{i12}^2 + \rho_{i21}^2)$ and $K_2a^2h^3 = \big\{10a^2\sum_{i=1}^{2}(\rho_{i12} - \rho_{i21})^2 + \sum_{i=1,k\neq m}^{2}\big[\sqrt{32}\rho_{ikk}a_k + \big(\sqrt{8}(\rho_{ikm} + \rho_{imk})a_m\big)\big]^2\big\}h^3$ which gives $K_2 = \frac{10a^2\sum_{i=1}^{2}(\rho_{i12}-\rho_{i21})^2 + \sum_{i=1,k\neq m}^{2}\big[\sqrt{32}\rho_{ikk}a_k + \big(\sqrt{8}(\rho_{ikm}+\rho_{imk})a_m\big)\big]^2}{a^2}$. Then we use the following condition that if $K_2a^2h^3 > K_1h^2$ which gives $a^2 > \frac{K_3}{h}$, then we choose the solution which has scheme (1.22) with the trivial coupling and if not we use the other solution which has scheme (1.22) with the exact coupling, . So the local error of the combined method will be the minimum of their local errors. i.e.

$$E\big(|y - x^{(r+1,2j+2)}|^2|\mathcal{F}_j\big) \leq \min\big[K_1h^2, K_2a^2h^3\big] \tag{4.29}$$

and then

$$E|y - x^{(r+1,2j+2)}|^2 \leq E\bigg(\min\big[K_1h^2, K_2a^2h^3\big]\bigg) \tag{4.30}$$

We will describe two methods of finding the expectation in (4.30) using the following non-invertible SDE to illustrate the results.

$$\begin{aligned} dX_1(t) &= X_2(t)dW_1(t) + (X_1(t) + t)dW_2(t), \\ dX_2(t) &= e^{-X_2^2(t)}dW_1(t) + (X_1(t) - X_2(t))dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{4.31}$$

When the determinant is near from zero for the inverse matrix $b_{ik}(x)$ then that means we will obtain a big value for a^2 and at the same time we will get the big value for the exact coupling error i.e. $(K_2 a^2 h^3)$. For a particular example from the previous SDE we have the value of the inverse matrix will be

$$c_{ij} = \frac{1}{y(x-y) - e^{-y^2}(x+t)} \begin{bmatrix} (x-y) & -(x+t) \\ -e^{-y^2} & y \end{bmatrix}$$

which means when $\big(y(x-y) - e^{-y^2}(x+t)\big)^2$ becomes close to the zero then a^2 will be too large. So in the following discussion we will try to control this problem and see the behavior of the combined method.

For the first method we will try to find the the following expectation directly $E[\min(K_1 h^2, K_2 a^2 h^3)]$ and then see what the convergence result for the behavior of its integral. To evaluate this we will use the *Hörmander Theorem* (Theorem 2) and then we could deduce the expected error for the integration for the function a. For the second method we will find the estimate of the error in (4.30) by doing a number of simulations for the previous SDE with different step sizes.

For the first method we will assume that the *Hörmander* conditions hold and then let f is a density function of $x^{(r,j)}$ and by applying the *Hörmander theorem* we assume that $|f(x,y)|$ bounded by a constant $K(y)$ and for fixed y we also have the following bound

$$\int f(x,y)dx \le K(y)$$

The constant $K(y)$ depends on y and it decreases rapidly when y become very big. So

$$E[\min(K_1h^2, K_2a^2h^3)] = Lh^2 \int\int \min(1, (a_1^2 + a_2^2)h) f(x,y)dxdy$$
$$\leq Lh^2 \left(K(y) \int \min(1, (a_1^2 + a_2^2)h)dx \right) \tag{4.32}$$

Here L is a constant and because the term $K_2a^2h^3$ depends on $a^2 = (a_1^2 + a_2^2)$ which define as a function of $a_i = \frac{1}{2}\sum_j c_{ij}(x^{(r,j)})\rho_{ikl}(x^{(r,j)})$. So we will have

$$a_1^2 + a_2^2 = \frac{C}{(yx - y^2 - e^{-y^2}x - e^{-y^2}t)^2}$$

where the term $\frac{1}{(yx-y^2-e^{-y^2}x-e^{-y^2}t)^2}$ comes from the inverse matrix c_{ij} and we bound other terms by constant C. Thus from (4.32) we obtain

$$E[\min(K_1h^2, K_2a^2h^3)] = C_1h^2 \left(K(y) \int \min(1, \frac{h}{(yx - y^2 - e^{-y^2}x - e^{-y^2}t)^2})dx \right)$$
$$= C_1h^2 \left(K(y) \int \min(1, \frac{h}{[(y - e^{-y^2})x - y^2 - e^{-y^2}t)]^2})dx \right) \tag{4.33}$$

Now let $(y - e^{-y^2})x - y^2 - e^{-y^2}t) = u$ then

$\Rightarrow du = (y - e^{-y^2})dx$, which gives $\Rightarrow \frac{du}{(y-e^{-y^2})} = dx$.

Then the integral in (4.33) will become

$$E[\min(K_1h^2, K_2a^2h^3)] \leq C_1h^2 \frac{K(y)}{|y - e^{-y^2}|} \left(2\int_0^\infty \min(1, \frac{h}{u^2})du \right)$$
$$= L_1h^2 \frac{K(y)}{|y - e^{-y^2}|} \left(\int_0^{\sqrt{h}} du + h\int_{\sqrt{h}}^\infty \frac{1}{u^2}du \right)$$
$$= L_1h^2 \frac{K(y)}{|y - e^{-y^2}|} \left(\sqrt{h} + \sqrt{h} \right)$$
$$= L_2h^{(5/2)} \frac{K(y)}{|y - e^{-y^2}|} \tag{4.34}$$

We have another bound when we fix y which will be as the following

$$\int \min(1, (a_1^2 + a_2^2)h) f(x,y)dx \leq \int f(x,y)dx \leq K(y) \tag{4.35}$$

If we define y_0 as the point where $y - e^{-y^2} = 0$, then for the expectation we will have the following

$$\begin{aligned} E[\min(K_1h^2, K_2a^2h^3)] &\leq L_2h^2 \int_{-\infty}^{\infty} \min(K(y), \frac{h^{\frac{1}{2}}K(y)}{|y-y_0|})dy \\ &= L_2h^2\bigg(h^{\frac{1}{2}} \int_{-\infty}^{y_0-\sqrt{h}} \frac{K(y)}{|y-y_0|}dy + h^{\frac{1}{2}} \int_{y_0+\sqrt{h}}^{\infty} \frac{K(y)}{|y-y_0|}dy \\ &+ \int_{y_0-\sqrt{h}}^{y_0+\sqrt{h}} K(y)dy\bigg) \\ &\leq h^2L[\sqrt{h}|\log(\sqrt{h})| + \sqrt{h}|\log(\sqrt{h})| + 2\sqrt{h}] \end{aligned} \tag{4.36}$$

So we could see from the last step in (4.36) that the dominant term will be of order $h^{5/2}\log(h)$. Therefore the order of the local error for the combined method will be

$$E[\min(K_1h^2, K_2a^4h^3)] = O(|h^{5/2}(\log(h)|) \tag{4.37}$$

Then by using the same arguments as the exact coupling we will obtain the global error for the combined method

$$h^{\frac{5}{4}-\frac{1}{2}}\sqrt{|\log(h)|} = h^{3/4}\sqrt{|\log(h)|} \tag{4.38}$$

The second method we need to show numerically by using the previous SDEs (4.31) that the expectation at the final time of the function $\min\left[K_1h^2, K_2a^2h^3\right]$ with different step-size will give the local error with order $h^{5/2}\log(h)$. That is we compute the previous function over the number of simulation and sum the results together to get the average estimate. After that, we compare and plotting the log of $E\bigg(\min\left[K_1h^2, K_2a^2h^3\right]\bigg)$ against the log of the different step sizes.

Now for the Matlab implementation we want to run the code in Listing (5.11) with different step sizes over a large number of paths $R = 2000$ and we could see in the table below the outcome of

$$\mu = E\bigg(\min\left[K_1h^2, K_2a^2h^3\right]\bigg) \tag{4.39}$$

step-size	μ
0.005	0.00000246
0.0025	0.00000043
0.00125	0.000000073
0.00062	0.0000000113
0.00031	0.0000000023
0.00015	0.0000000003

Table 4.1: Estimating the error of μ against the step size

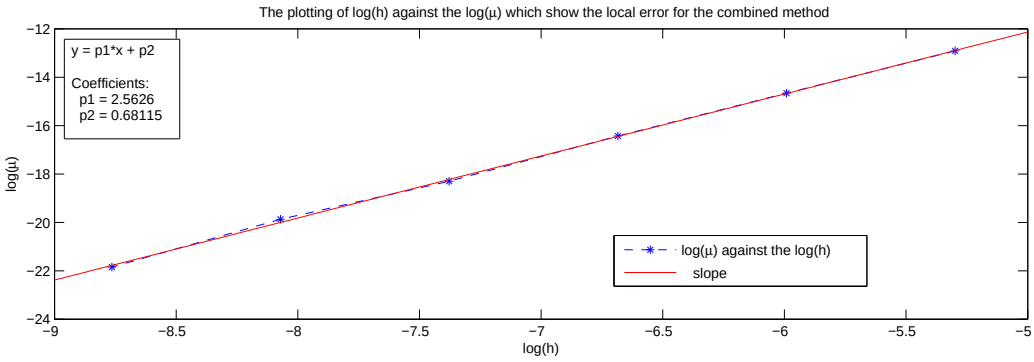


Figure 4.1: The plotting of the local error for the combined method

for the certain number of steps.

Figure (4.1) shows the plotting of the $\log(\mu)$ against the log of the step sizes.

The table (4.1) and the plotting in Figure (4.1) show the implementation of $\mu = E\left(\min\left[K_1 h^2, K_2 a^2 h^3\right]\right)$ for the previous SDEs with different number of steps (200, 400, 800, 1600, 3200 and 6400). Running the code (errorRungcouplingTEST) for 2000 simulations gives a value for its estimator μ equal to 0.00000246 with the step-size 0.005 i.e.

$$\mu = E\left(\min\left[K_1 h^2, K_2 a^2 h^3\right]\right) = 0.00000246$$

and 0.00000043 with step-size 0.0025 and so on. This means when we decrease the step size (h) every time, we calculate the error μ and examine the convergence order

of it where the output results are in the table (4.1). Also the Figure (4.1) is a plot of the log of the estimator μ i.e. $\log(\mu)$ against the log of step-size (h) i.e. $\log(h)$ which has a slope of 2.5626 which is consistent with the local error for the combined method will be $O(h^{\frac{5}{2}}\log(h))$.

Therefore from these computational results we could see that we have obtained good agreement between the theoretical bound for the local error in (4.37) with the implementation results.

In the following section we will show that the order of convergence for the combined method will be $h^{3/4}\sqrt{|\log(h)|}$ by doing a number of simulation for a particular SDE which is singular. Also in this section we will show the combined method for the exact coupling with trivial coupling and the approximate coupling with trivial coupling.

4.2 The implementation of exact coupling with the trivial coupling (combined method) in two-dimensional case with non-invertiblity of $b_{ik}(x)$

Firstly, we have the 2-dimensional SDE, which is not invertible at some points.

$$\begin{aligned} dX_1(t) &= X_2(t)dW_1(t) + (X_1(t)+t)dW_2(t), \\ dX_2(t) &= e^{-X_2^2(t)}dW_1(t) + (X_1(t)-X_2(t))dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{4.40}$$

where $W_1(t)$ and $W_2(t)$ are independent standard Brownian motion.

To apply a numerical method to this SDE we need to simulate solutions (for the same Brownian path) simultaneously using two different step sizes (h and $h/2$).

To construct this experiment, we will decrease the step size (h) every time when we calculate the error and examine the convergence order of the exact coupling method. We will repeat this with different step size using (for example, $R = 2000$) independent simulations. So the order of convergence for the combined method should be $h^{3/4}\sqrt{|\log(h)|}$.

Now we will run the Matlab code in Listing (5.11) with different step sizes over a large number of paths R as described in the table below and see the result of the error ϵ, where each simulation is for the same Brownian path and $\epsilon = \frac{1}{R}\sum_{i=1}^{R}|x_h^{(i)} - x_{h/2}^{(i)}|$ will be our estimator.

In the following table we will show the result of the error by running the Matlab code in Listing (5.11) with different step sizes over a large number of path R.

step-size	ϵ
0.01	0.0672
0.005	0.0387
0.0025	0.0213
0.00125	0.0124
0.00062	0.0069
0.00031	0.0039

Table 4.2: combined method for the exact coupling

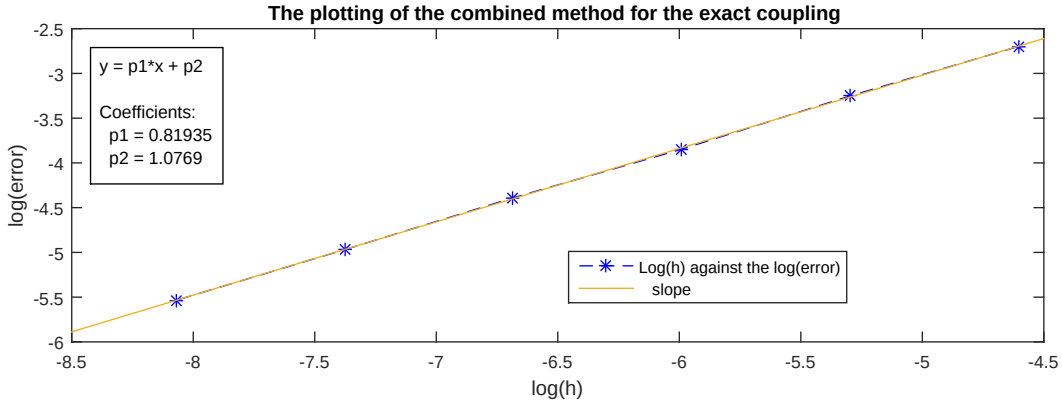


Figure 4.2: The plotting of the combined method for the exact coupling

The table (4.2) and the plotting in Figure (4.2) show the implementation of the combined method for the exact coupling of the previous SDEs with different number of steps (100, 200, 400, 800, 1600 and 3200). Running the code (errorRungcouplingTEST) for 2000 simulations gives a value for its estimator ϵ equal to 0.0672

with the step-size 0.01 i.e.

$$\epsilon = \frac{1}{2000}\sum_{i=1}^{2000} |x_h^{(i)} - x_{h/2}^{(i)}| = 0.0672$$

and 0.0387 with step-size 0.005 and so on. This means when we increase the number of steps which each time gives a smaller step-size then the estimate error ϵ will give $O(h^{\frac{3}{4}}\sqrt{\log(h)})$ as it appears in the results in table (4.2). Also the Figure (4.2) is a plot of the log of the estimator ϵ i.e. $\log(\epsilon)$ against the log of step-size (h) i.e. $\log(h)$ which has a slope of 0.81935 which is consistent with a strong convergence of $O(h^{\frac{3}{4}}\log(h))$ for the stochastic differential equation (4.40).

Therefore from these computational results we could see that we have obtained good agreement between the theoretical bound in (4.38) with the implementation results.

4.3 The implementation of approximate coupling with the trivial coupling (combined method) in two-dimensional case with non-invertible $b_{ik}(x)$

Here we will do the same as in section (4.2) and the same SDEs (4.40) will be used but we will have a different condition for the approximate coupling for the combined method. So for the approximate coupling we will have the following bound

$$E|x^{(r,1)} - x^{(r+1,2)}|^2 \leq CE|(p(U))|^2h^3$$

Here p is an $\mathbb{R}^d$-valued polynomial on $\mathbb{R}^d$ and from chapter 3 we choose $p = \frac{1}{2}\nabla(\frac{\nabla^2-\Psi}{4}+\theta-\frac{\Psi}{2})$. Also we get the constant C here from the norm of the diffusion matrix ,i.e. $||b(x)|| = C$ and we derive this by finding the local error between two approximations $x^{(r,1)} - x^{(r+1,2)}$ from the error which we obtained from $V - Y$.

So for the trivial coupling the local error is

$$E|x^{(r,1)} - x^{(r+1,2)}|^2 \leq K_1h^2$$

Therefore the condition that we will use is that if

$$CE|p(U)|^2h^3 > K_1h^2 \quad \text{so} \quad E|p(U)|^2 > \frac{C_1}{h} \tag{4.41}$$

where $C_1 = \frac{K_1}{C}$, then we will use the scheme (1.22) with trivial coupling instead of the scheme (1.22) with approximate coupling. Firstly, from this condition we need to find $E|p(U)|^2$ after that we will use it in the implementation condition.

For the correction term p we will have $p = \frac{1}{2}\nabla(\nabla^2\frac{-\Psi}{4} + \theta - \frac{\Psi}{2})$, if we have $U = y$ then we need to find

$E|p|^2 = E(\frac{1}{2}\nabla(\nabla^2 \frac{-\Psi}{4} + \theta - \frac{\Psi}{2}))^2$.

$$\begin{aligned}\nabla(\theta(y)) &= \nabla(\sum_{i,k=1}^{2} \theta_{ik} y_i y_k) \\ &= \Theta y + \Theta^t y \end{aligned} \tag{4.42}$$

where $\Theta = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}$

On the other hand we have

$$\nabla(\Psi(y)) = \nabla\big(\big|\sum_{i,k=1}^{2} \sigma_{ik1} y_i y_k\big|^2 + \big|\sum_{i,k=1}^{2} \sigma_{ik2} y_i y_k\big|^2\big)$$

Now let $\sigma_1(y) = \sum_{i,k=1}^{2} \sigma_{ik1} y_i y_k$ and $\sigma_2(y) = \sum_{i,k=1}^{2} \sigma_{ik2} y_i y_k$

Define $A = \sigma_{ik1}$ and $B = \sigma_{ik2}$ Then

$$\begin{aligned}\nabla(\Psi(y)) &= \nabla(\sigma_1(y)^2 + \sigma_2(y)^2) \\ &= 2\sigma_1(y)C + 2\sigma_2(y)D \end{aligned}$$

where $C = (Ay + A^t y)$ and $D = (By + B^t y)$

$$\begin{aligned}\nabla^2(\Psi(y)) &= \nabla(2\sigma_1(y)C + 2\sigma_2(y)D) \\ &= 2y^t My + 2y^t Fy \end{aligned} \tag{4.43}$$

where $M = A^t A + A^2 + 2A(tr(A)) + (A^t)^2 + AA^t$
and $F = B^t B + B^2 + 2B(tr(B)) + (B^t)^2 + BB^t$

Finally we need to find $\nabla[\nabla^2(\Psi(y))]$

$$\begin{aligned}\nabla[\nabla^2(\Psi(y))] &= \nabla\big(2y^t My + 2y^t Fy\big) \\ &= 2(My + M^t y) + 2(Fy + F^t y)\end{aligned}$$

So the expectation of $p(y)$ will be

$$E\big(\frac{1}{2}\nabla(\nabla^2\frac{-\Psi}{4}+\theta-\frac{\Psi}{2})\big)^2 = E\big(\frac{1}{2}(\frac{-2(My+M^ty)-2(Fy+F^ty)}{4}+\Theta y+\Theta^t y - \sigma_1(y)C-\sigma_2(y)D)\big)^2$$
$$= \frac{1}{4}E\big(\frac{-My-M^ty-Fy-F^ty}{2}+\Theta y+\Theta^t y - \sigma_1(y)C-\sigma_2(y)D\big)^2 \quad (4.44)$$

Now let $\frac{-My-M^ty-Fy-F^ty}{2}+\Theta y+\Theta^t y = (\frac{-M-M^t-F-F^t}{2}+\Theta+\Theta^t)y = Gy$

And let $C = Ay + A^ty = (A+A^t)y = Ky$, $D = By + B^ty = (B+B^t)y = Hy$. This will give

$$(\sigma_1(y)C)_n + (\sigma_2(y)D)_n = (\sigma_1(y)Ky + \sigma_2(y)Hy)_n$$
$$= (y^tAyKy + y^tByHy)_n$$
$$= (\sum_{i,k=1}^{2}\sigma_{ik1}y_iy_k)(\sum_{r=1}^{2}\sigma_{nr1}y_r) + (\sum_{i,k=1}^{2}\sigma_{ik2}y_iy_k)(\sum_{r=1}^{2}\sigma_{nr2}y_r)$$
$$= (\sum_{i,k,r=1}^{2}\sigma_{ik1}\sigma_{nr1}y_iy_ky_r) + (\sum_{i,k,r=1}^{2}\sigma_{ik2}\sigma_{nr2}y_iy_ky_r)$$
$$= (\sum_{i,k,r=1}^{2}\lambda_{ikrn}y_iy_ky_r) + (\sum_{i,k,r=1}^{2}\mu_{ikrn}y_iy_ky_r) \quad (4.45)$$

where $\lambda_{ikrn} = \sigma_{ik1}\sigma_{nr1}$ and $\mu_{ikrn} = \sigma_{ik2}\sigma_{nr2}$
so from (4.45) we obtain

$$(\sigma_1(y)C)_n + (\sigma_2(y)D)_n = \sum_{i,k,r=1}^{2}(\lambda_{ikrn}+\mu_{ikrn})y_iy_ky_r$$
$$= \sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r \qquad \text{where } \delta_{ikrn} = \lambda_{ikrn}+\mu_{ikrn} \quad (4.46)$$

Therefore we will have from the expectation in (4.44)

$$\frac{1}{4}E\big(Gy-\sigma_1(y)C-\sigma_2(y)D\big)^2=\frac{1}{4}E\big[Gy-\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)\big]^2$$
$$=\frac{1}{4}E\big[|Gy|^2-2(Gy).\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)$$
$$+\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)^2\big] \tag{4.47}$$

After we expand the square in equation (4.47), we need to find the expectation for every term separately. For the first term we have

$$E|Gy|^2=E[(Gy)^t(Gy)]=E(y^tG^tGy)$$
$$=E(\sum_{i,k,m=1}^{2}g_{ik}g_{im}y_ky_m) \qquad \text{where } Gy=\sum_{i,k=1}^{2}g_{ik}y_k$$
$$=E(\sum_{k,m=1}^{2}B_{km}y_ky_m) \qquad \text{where } B_{km}=g_{ik}g_{im}$$
$$=\sum_{k=1}^{2}B_{kk} \tag{4.48}$$

$$E[(Gy).\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)]=E[\sum_{n=1}^{2}\big((\sum_{m=1}^{2}g_{nm}y_m)\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)\big)]$$
$$=E\big(\sum_{n,m,i,k,r=1}^{2}g_{nm}\delta_{ikrn}y_my_iy_ky_r\big)$$
$$=E\big(\sum_{m,i,k,r=1}^{2}\beta_{mikrn}y_my_iy_ky_r\big)$$

where $\beta_{mikrn}=g_{nm}\delta_{ikrn}$

When we take the expectation after some calculation there will be some terms which will give zero therefore we will omit them in the following equation. So from the last equation will obtain

$$E\big[\sum_{i=1}^{2}\beta_{iiiin}y_i^4+\sum_{m\neq k}^{2}(\beta_{mmkkn}+\beta_{mkkmn}+\beta_{mkmkn})y_m^2y_k^2\big]$$
$$=3\sum_{i=1}^{2}\beta_{iiiin}+\sum_{m\neq k}^{2}(\beta_{mmkkn}+\beta_{mkkmn}+\beta_{mkmkn}) \tag{4.49}$$

Finally

$$E(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r)^2$$
$$=E\big(\sum_{i=1}^{2}\delta_{iiin}^2y_i^6+\sum_{k\neq r}^{2}[(\delta_{kkrn}+\delta_{krkn}+\delta_{rkkn})^2+2\delta_{kkkn}(\delta_{rrkn}+\delta_{rkrn}+\delta_{krrn})]y_k^4y_r^2\big)$$
$$=15\sum_{i=1}^{2}\delta_{iiin}^2+\sum_{k\neq r}^{2}[3(\delta_{kkrn}+\delta_{krkn}+\delta_{rkkn})^2+6\delta_{kkkn}(\delta_{rrkn}+\delta_{rkrn}+\delta_{krrn})] \tag{4.50}$$

So

$$E|p|^2=\frac{1}{4}E\big[|Gy|^2-2(Gy)\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)+\big(\sum_{i,k,r=1}^{2}\delta_{ikrn}y_iy_ky_r\big)^2\big]$$
$$=\frac{1}{4}\big(\sum_{k=1}^{2}B_{kk}-2\times3\sum_{i=1}^{2}\beta_{iiiin}-2\sum_{m\neq k}^{2}(\beta_{mmkkn}+\beta_{mkkmn}+\beta_{mkmkn})$$
$$+15\sum_{i=1}^{2}\delta_{iiin}^2+\sum_{k\neq r}^{2}[3(\delta_{kkrn}+\delta_{krkn}+\delta_{rkkn})^2+6\delta_{kkkn}(\delta_{rrkn}+\delta_{rkrn}+\delta_{krrn})]\big) \tag{4.51}$$

So after we found the $E|p|^2$, then the precise condition in (4.41) will be that if

$$\frac{1}{4}\big(\sum_{k=1}^{2}B_{kk}-2\times3\sum_{i=1}^{2}\beta_{iiiin}-2\sum_{m\neq k}^{2}(\beta_{mmkkn}+\beta_{mkkmn}+\beta_{mkmkn})$$
$$+15\sum_{i=1}^{2}\delta_{iiin}^2+\sum_{k\neq r}^{2}[3(\delta_{kkrn}+\delta_{krkn}+\delta_{rkkn})^2+6\delta_{kkkn}(\delta_{rrkn}+\delta_{rkrn}+\delta_{krrn})]\big)>\frac{C_1}{h} \tag{4.52}$$

then we will use the scheme (1.22) with trivial coupling instead of the scheme (1.22) with approximate coupling .

Now we will run the Matlab code in Listing (5.13) with different step sizes over a large number of paths R as described in the table below and see the result of the error. In the numerical simulation we evaluate 90% confidence interval $(\epsilon - a, \epsilon + a)$ for the mean value μ, where

$$a = Z_{\alpha/2} \times \frac{\sigma}{\sqrt{R}} \tag{4.53}$$

here R is a sample size, $Z_{\alpha/2}$ is the confidence coefficient which equals 1.66 and σ is a standard deviation. Where we calculate the standard deviation as the following, we have R estimates for the quantity $|x_h - x_{h/2}|$, then we could represent their values as a vector, then we use the Matlab function (Std) to calculate their standard deviation σ.

step-size	ϵ	confidence interval of ϵ
0.005	0.0496	(0.0497, 0.0495)
0.0025	0.0289	(0.0294, 0.0284)
0.00125	0.0174	(0.0179, 0.0169)
0.00062	0.0108	(0.0109, 0.0107)
0.00031	0.0069	(0.0073, 0.0066)
0.00015	0.0042	(0.0046 , 0.0037)

Table 4.3: combined method for the approximate coupling

The table (4.3) and the plot in Figure (4.3) show the implementation of the combined method for the approximate coupling for the previous 2-dimensional SDEs with different number of steps (200, 400, 800, 1600, 3200 and 6400). Running the code (AproxcouplingNONinver) for a 2000 number of simulations gives a value for its estimator ϵ equal to 0.0496 with the step-size 0.005 i.e.

$$\epsilon = \frac{1}{2000} \sum_{i=1}^{2000} |x_h^{(i)} - x_{h/2}^{(i)}| = 0.0496$$

and 0.0289 with step-size 0.0025 and so on. This means when we increase the number of steps which each time gives a smaller step-size then the estimate error ϵ will

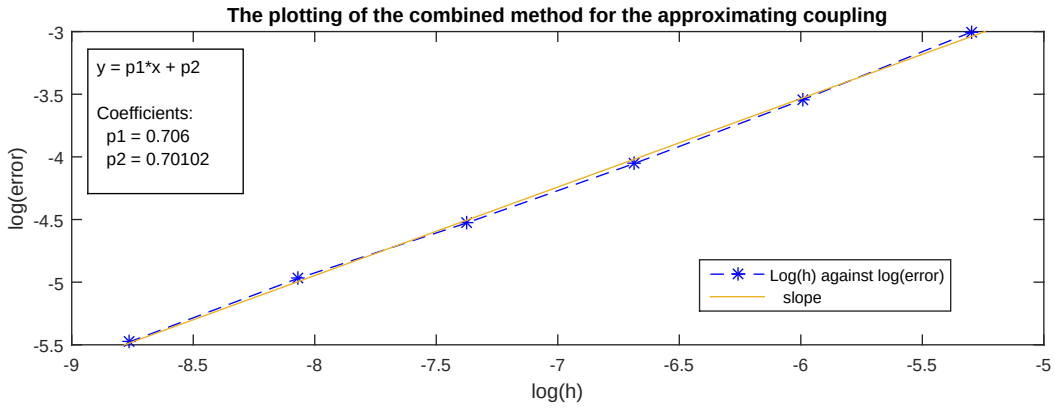


Figure 4.3: The plotting of the combined method for the approximate coupling

give $O(h^{\frac{3}{4}}\log(h))$ as it appears in the results in table (4.3). Also the Figure (4.3) is a plot of the log of the estimator ϵ i.e. $\log(\epsilon)$ against the log of step-size (h) i.e. $\log(h)$ which has a slope of 0.706 which is consistent with a strong convergence of $O(h^{\frac{3}{4}}\sqrt{|\log(h)|})$ for the stochastic differential equation (4.40).

Therefore from these computational results we could see that we have obtained good agreement between the theoretical bound in (4.38) with the implementation results.

Chapter 5

Multilevel method for weak approximation of SDEs

The idea of the multilevel Monte Carlo method which is due to Giles [16], is to apply the method of Monte Carlo several times with different time-steps. First of all, suppose we have the stochastic differential equation

$$dx_i(t) = \sum_{k=1}^{d} b_{ik}(x(t))dW_k(t), \quad x_i(0) = x_i^{(0)} \tag{5.1}$$

where i =1,...,q on an interval $[0, T]$, for a q-dimensional vector $x(t)$, with a d-dimensional Brownian path $W(t)$.

We need to estimate $Ef(x(T))$, where x is the solution of (5.1), and the function f is globally Lipschitz. The estimation can be done by taking the average of N independent simulation i.e. $\frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)})$ where the $x_h^{(i)}$ denotes the approximate solution at time T of the the scheme (1.22) with the exact or the trivial coupling and with step-size h.

The objective of the multilevel method is to reduce the computational load of the estimation. In a standard Monte Carlo simulation, we use a fixed time-step, where as for the multilevel MC simulation we consider MC simulations with different time-steps $h^{(r)} = \frac{T}{2^r}$. Define $\mu_r = Ef(x_{h^{(r)}})$, then we need to estimate μ_n by choosing a

suitable large n.

According to Giles [16] we could write

$$\begin{aligned}\mu_n &= \mu_0 + \mu_1 - \mu_0 + \mu_2 - \mu_1 \pm \cdots + \mu_{n-1} - \mu_{n-2} + \mu_n - \mu_{n-1} \\ &= \mu_0 + \sum_{k=1}^{n} (\mu_k - \mu_{k-1}) \end{aligned} \tag{5.2}$$

The idea of the multilevel Monte Carlo method is to approximate each term on the right hand side of (5.2) independently.
In fact for the first term we have

$$\mu_0 = Ef(x_{h^0}) \tag{5.3}$$

and we could we use the following estimator for (5.3)

$$Y_0 = \frac{1}{N_0} \sum_{i=1}^{N} f(x_{h^0}^{(i)}) \tag{5.4}$$

The other terms are estimated using N_k independent samples s.t.

$$\mu_k - \mu_{k-1} = \frac{1}{N_k} \sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)}) \right) := \hat{Y}_k \tag{5.5}$$

Therefore we could use the following estimator for the multilevel MC method (5.2)

$$\frac{1}{N_0} \sum_{i=1}^{N} f(x_{h^0}^{(i)}) + \sum_{k=1}^{n} \frac{1}{N_k} \sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)}) \right) \tag{5.6}$$

The key point here is that the quantity $\left(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}}) \right)$ comes from two discrete approximations with different time-steps $h^{(k)}$ and $h^{(k-1)}$ respectively but with the same Brownian path. The use of the same Brownian path will reduce the variance, so that smaller N_k can be used.

Now, denoting the variance of a single sample $\left(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}}) \right)$ by V_k, the

variance of the simple estimator $\hat{Y}_k$ is

$$\begin{aligned} Var(\hat{Y}_k) &= Var\big(\frac{1}{N_k}\sum_{i=1}^{N_k}\big(f(x^{(i)}_{h^{(k)}}) - f(x^{(i)}_{h^{(k-1)}})\big)\big) \\ &= \frac{V_k}{N_k} \end{aligned} \tag{5.7}$$

Therefore if we have the variance of the initial level as the following

$$Var(Y_0) = \frac{1}{N_0}\sum_{i=1}^{N} Var(f(x^{(i)}_{h^0})) = \frac{V_0}{N_0}$$

then the variance of the combined estimator $\hat{Y} = \sum_{k=1}^{n}\hat{Y}_k$ is

$$\begin{aligned} Var(\hat{Y}) &= Var(\sum_{k=0}^{n}\hat{Y}_k) \\ &= \sum_{k=0}^{n} Var(\hat{Y}_k) \\ &= \sum_{k=0}^{n}\frac{V_k}{N_k} \end{aligned} \tag{5.8}$$

and the computational load will be proportional to

$$\sum_{k=0}^{n}\frac{N_k}{h_k}$$

Hence the variance is minimized for a fixed computational load by choosing N_k to be proportional to

$$\sqrt{V_k h_k}$$

The following complexity theorem by Giles [16] shows the general application of the multilevel Monte Carlo method. The theorem also does not specify which numerical approximation scheme is used. We will state the theorem and then we will apply it to the exact coupling, the trivial coupling and the combined method to get their computational load results.

Theorem 4. *Let $f(x(T))$ denote a functional of the solution of stochastic differential equation (1) for a given Brownian path $W(t)$, and let $f(x_{h^{(k)}})$ denote the corresponding approximation to $x(T)$ using a numerical discretization with time-step $h_k = \frac{T}{M^k}$.*

If there exist independent estimators $\hat{Y}_k$ using N_k Monte Carlo simulation, and positive constants $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ such that

1. $|E[f(x_{h^{(k)}}) - f(x(T))]| \leq c_1 h_k^{\alpha}$

2. $$E[\hat{Y}_k] = \begin{cases} E[f(x_{h^0})], & k = 0 \\ E[f(x_{h^{(k)}}) - f(x_{h^{(k)-1}})], & k > 0 \end{cases}$$

3. $V[\hat{Y}_k] \leq \frac{c_2 h_k^{\beta}}{N_k}$

4. *C_k, the computational complexity of $\hat{Y}_k$, is bounded by*

$$C_k \leq \frac{c_3 N_k}{h_k},$$

then there exists a positive constant c_4 such that for any $\epsilon < e^{-1}$ there are values n and N_k for which the multi-level estimator

$$\hat{Y} = \sum_{k=0}^{n} \hat{Y}_k,$$

has a mean-square-error with bound

$$MSE = E\left[\left(\hat{Y} - E[f(x(T))]\right)^2\right] < \epsilon^2$$

with a computational complexity C with bound

$$C \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > 1, \\ c_4 \epsilon^{-2} (\log \epsilon)^2, & \beta = 1, \\ c_4 \epsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1, \end{cases}$$

Proof. see Giles [16] □

In the previous theorem we could obtain β which is twice the strong order of the scheme.

In the following applications we will see the order of the variance, i.e. $V_k = O(h_k^{\beta})$, and from that we could apply Theorem 4 and get the computational load.

5.1 The multilevel Monte Carlo method for scheme (1.22) with the exact coupling

We should mention here that Giles and Szpruch in [24] have done some work on the multilevel method and achieved $O(\epsilon^{-2})$ computational load but they use a stronger regularity condition on the function f than Lipschitz. Now as we have proven in chapter 2 under the non-degeneracy condition that the convergence of scheme (1.22) with the exact coupling is $O(h)$ therefore the variance of single sample,

$$\begin{aligned} V_k &= \mathbb{V}\big(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\big) \\ &= O(h_k^2) \end{aligned} \tag{5.9}$$

On other hand it follows from (5.9) and the fact that f is a Lipschitz function and such that the drift and diffusion coefficients satisfy the Lipschitz conditions with uniformly bounded derivatives that there is $O(h)$ weak convergence for scheme (1.22) with the invertibility assumption on the matrix $b_{ik}(t, X(t))$, so that

$$|E\big(f(x_{h^{(k)}}) - f(x(T)\big)| = O(h_k)$$

Hence for the simple estimator (5.5), the single sample variance V_k is $O(h_k^2)$. Furthermore the optimal choice for N_k is asymptotically proportional to $(h_k^{3/2})$.

Consequently, because we have that $\beta = 2 > 1$ for the variance then from Theorem 4 if we have the estimated error of this method is ϵ then the computational load will be $O(\epsilon^{-2})$. Hence because the variance of the multilevel Monte Carlo for

scheme (1.22) with the exact coupling declines fast as a result of this the number of simulations N_k will decrease rapidly therefore the computational load will reduce quickly. So the main order of the convergence of this method will be obtained from the first level of the simulation which will be the dominant term with the computational load $O(\epsilon^{-2})$.

5.2 The multilevel Monte Carlo method for scheme (1.22) with the trivial coupling

The strong convergence of scheme (1.22) with the trivial coupling will be $O(h^{1/2})$. Therefore

$$E[|x_{h^{(k)}} - x(T)|^2] = O(h_k)$$

For scheme (1.22) with trivial coupling and using the property of f being Lipschitz function, we have

$$\begin{aligned} Var[f(x_{h^{(k)}}) - f(x(T))] &\leq E[\big(f(x_{h^{(k)}}) - f(x(T))\big)^2] \\ &= E[|x_{h^{(k)}} - x(T)|^2] \\ &= O(h_k) \end{aligned}$$

Furthermore the variance of a single sample,

$$\begin{aligned} V_k &= \mathbb{V}\big(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\big) \\ &= O(h_k) \end{aligned}$$

Hence for the simple estimator (5.5), the single sample variance V_k is $O(h_k)$. Furthermore the optimal choice for N_k is asymptotically proportional to (h_k).

Consequently, because we have that $\beta = 1$ for the variance then from Theorem 4 the computational load will be $O(\epsilon^{-2}(\log \epsilon)^2)$.

5.3 The multilevel Monte Carlo method for the combined method

We have proven in chapter 4 that the convergence of the combined method under some certain conditions is $O(h^{3/4}\sqrt{|\log(h)|})$ therefore the variance of a single sample,

$$\begin{aligned} V_k &= \mathbb{V}\big(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\big) \\ &= O(h^{3/2}) \end{aligned} \tag{5.10}$$

Hence for the simple estimator (5.5), the single sample variance V_k is $O(h_k^{3/2})$. Furthermore the optimal choice for N_k is asymptotically proportional to $\sqrt{h^{\frac{3}{2}}h} = (h_k^{5/4})$.

Consequently, because we have that $\beta = \frac{3}{2} > 1$ for the variance then from Theorem 4 the computational load will be $O(\epsilon^{-2})$.

5.4 Numerical results of the multilevel method for scheme (1.22) with the exact coupling

In this section we consider two-dimensional stochastic differential equations which are invertible and we will show numerically the performance of the multilevel Monte Carlo method for scheme (1.22) with the exact coupling. Also we will see the effect of decreasing the number of simulation on the computational load. Moreover we will show the implementation result of the multilevel Monte Carlo method for scheme (1.22) with the exact coupling and compare it to the multilevel Monte Carlo method for scheme (1.22) with the trivial coupling.

5.4.1 The multilevel implementation of the exact coupling in two-dimensional case with invertible matrix

We will show the estimation results of the multilevel method for scheme (1.22) with the exact coupling. The following 2-dimensional SDE, which is invertible will be tested to see the results and we will compare the results with the multilevel method for scheme (1.22) with the trivial coupling for the same SDEs.

$$\begin{aligned} dX_1(t) &= (\sin(X_2(t)))^2 dW_1(t) - \frac{1}{1+X_1^2(t)} dW_2(t), \\ dX_2(t) &= \frac{1}{1+X_2^4(t)} dW_1(t) + (\cos(X_1(t)))^2 dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{5.11}$$

where $W_1(t)$ and $W_2(t)$ are independent standard Brownian motion.

In each level of simulation we will have the Lipschitz function $f(x_{h^{(k)}}) = \sin(x_{h^{(k)}})$ to estimate the following quantity $\big(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\big)$, where the $x_{h^{(k)}}$ and $x_{h^{(k-1)}}$ are two discrete approximations with different time-steps $h^{(k)}$ and $h^{(k-1)}$ respectively but they have the same Brownian path. In the following table we will show for each level the number of simulation N_k, step-size h and the result of the simple estimator $\hat{Y}_k$ with its confidence interval.

Now we will run the Matlab code in Listing 5.14 to get the result of each level for the simple estimator $\hat{Y}_k$ in (5.5) with number of levels $n = 9$ and the initial value of the number of simulation N_k for the estimation will be 2^{22}. Also each simulation is for the same Brownian path.

In the following numerical simulations we evaluate 90% confidence interval of $\hat{Y}_k$, $(\mu - a, \mu + a)$ for its mean value μ, where

$$a = Z_{\alpha/2} \times \frac{\sigma}{\sqrt{N_k}} \tag{5.12}$$

here N_k is a sample size, $Z_{\alpha/2}$ is the confidence coefficient which equals 1.66 and σ is a standard deviation. Where we calculate the standard deviation as the following, for each level n we get N_k estimate for the quantity $\big(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\big)$, then we

could represent their values as a vector, then we use the Matlab function (Std) to calculate their standard deviation σ.

Table 5.1: multilevel with exact coupling in 2-dimensional case with invertible matrix

level k	Number of simulation N_k	step-size $h = \frac{T}{2^{(n-1)}}$	Simple estimator $\hat{Y}_k$	The confidence interval of $\hat{Y}_k$
1	4194304	1	0.0971	(0.0975, 0.0968)
2	1482910	1/2	0.0246	(0.0243, 0.248)
3	524288	1/4	0.0076	(0.0074, 0.0079)
4	185364	1/8	0.0035	(0.0033, 0.0038)
5	65536	1/16	0.0016	(0.0014, 0.0018)
6	23170	1/32	0.00076	(0.00059, 0.00094)
7	8192	1/64	0.00053	(0.00038, 0.00068)
8	2896	1/128	0.00016	(0.000039, 0.00028)
9	1024	1/256	0.000035	(-0.00007, 0.00014)

For the multilevel MC method we have two terms, the first one comes from the estimation of the top level μ_0 which has step-size h^0 i.e.

$$\begin{aligned} Ef(x_h) &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) \\ &\approx 0.6846 \end{aligned}$$

The other terms comes from the summation of simple estimator $\hat{Y}_k$ which for this SDE will be

$$\sum_{k=1}^{n} \frac{1}{N_k}\sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right) \approx 0.1360$$

Therefore the estimator for the multilevel MC method with the exact coupling is

$$\begin{aligned} \mu_n &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) + \sum_{k=1}^{n} \frac{1}{N_k}\sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right) \\ &\approx 0.6846 + 0.1360 = 0.8206 \end{aligned} \tag{5.13}$$

After we have obtained the result of the multilevel Monte Carlo method for the scheme (1.22) with exact coupling, we would like to compare it with scheme (1.22) with trivial coupling and see if the outcome results will be approximately the same. We need to do the implementation of the same SDEs (5.11) but for the multilevel for the scheme (1.22) with trivial coupling. Then we will compare these results with the previous results in table 5.1 and see if they will be approximately agreed.

5.4.2 The multilevel implementation of the trivial coupling in two-dimensional case with invertible matrix

Now we want to do the same as in 5.13 and we will run the Matlab code in Listing 5.17 to get the result of each level for the simple estimator $\hat{Y}_k$ in (5.5) with number of levels $n = 9$ and the initial value of the number of simulation N_k for the estimation will be 2^{22}. Also each simulation is for the same Brownian path.

Here the estimate of the top level μ_0 which has step-size h^0 i.e.

$$\begin{aligned} Ef(x_h) &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) \\ &\approx 0.6848 \end{aligned}$$

and the summation of simple estimator $\hat{Y}_k$ will be

$$\sum_{k=1}^{n}\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right) \approx 0.1366$$

Therefore the estimator for the multilevel MC method with the trivial coupling is

$$\begin{aligned} \mu_n &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) + \sum_{k=1}^{n}\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right) \\ &\approx 0.6848 + 0.1366 = 0.8214 \end{aligned} \tag{5.14}$$

Firstly, we have here the implementation of multilevel MC method with the exact and trivial coupling with the invertible matrix. It can be seen from 5.13 that the

Table 5.2: multilevel with trivial coupling in 2-dimensional case with invertible matrix

level k	Number of simulation N_k	step-size $h = \frac{T}{2^{(n-1)}}$	Simple estimator $\hat{Y}_k$	The confidence interval of $\hat{Y}_k$
1	4194304	1	0.0971	(0.0968, 0.0974)
2	2097152	1/2	0.0251	(0.0249, 0.0254)
3	1048576	1/4	0.0080	(0.0078, 0.0082)
4	524288	1/8	0.0035	(0.0033, 0.0036)
5	262144	1/16	0.0016	(0.0014, 0.0017)
6	131072	1/32	0.00068	(0.00057, 0.00080)
7	65536	1/64	0.00035	(0.00024, 0.00046)
8	32768	1/128	0.00016	(0.000067, 0.00026)
9	16384	1/256	0.00018	(0.00009, 0.00028)

value of the final calculation of the multilevel MC method with the exact coupling which is 0.8206 has approximately agreed with the value for the trivial coupling in 5.14 which is 0.8214. In addition, table 5.1 shows the results of the implementation and the computation by using the multilevel MC method with the exact coupling, one could see from the number of simulation N_k that it is proportional to $\sqrt{V_k h_k}$ therefore we need to divide each time by $\sqrt{8}$ for the next level. From table 5.1 we started the number of simulation by 4194304 and for the second level the number of simulation will be divided by $\sqrt{8}$ that is $\frac{4194304}{\sqrt{8}} \approx 1482910$ and so on. Moreover, we could see from both tables 5.1 and (5.2) that when we increase the number of levels then the value of simple estimating $\hat{Y}_k$ decreases.

In the other hand, for the multilevel MC method with the trivial coupling in table 5.2 the number of simulations is divided by two every time when we get to the new level. Here we start the first level with 4194304 simulations then divide by

2 for the next level which will give $\frac{4194304}{2} = 2097152$ and so on for the following level. This means by the observation that we can see a faster decrease in the number of simulations for the multilevel MC method with the exact coupling as compared to the multilevel MC method with the trivial coupling. As a result of this the computational saving has been achieved from calculating the computational load for the multilevel MC method with exact coupling, in contrast to the multilevel MC method with the trivial coupling

5.5 Numerical results of the multilevel method for the combined method

In this section we consider two-dimensional stochastic differential equations which are not invertible in some points and we will show numerically the performance of the multilevel Monte Carlo method for the combined method with the exact coupling. Also we will see the effect of decreasing the number of simulation on the computational load. Moreover we will show the implementation result of the multilevel Monte Carlo method for the combined method with the exact coupling and compare it to the multilevel Monte Carlo method for scheme (1.22) with the trivial coupling.

5.5.1 The multilevel implementation of combined method in two-dimensional case with non-invertible matrix

We will show the estimation results of the multilevel MC method for the combined method. The following SDE will be used to illustrate the results, which is not invertible in some points and we will compare the results with the multilevel method for scheme (1.22) with the trivial coupling for the same SDEs.

$$\begin{aligned} dX_1(t) &= X_2(t)dW_1(t) + (X_1(t)+t)dW_2(t), \\ dX_2(t) &= e^{-X_2^2(t)}dW_1(t) + (X_1(t)-X_2(t))dW_2(t), \\ &\text{for } 0 \le t \le 1, \text{ with } X_1(0) = 2 \quad \text{and} \quad X_2(0) = 0 \end{aligned} \tag{5.15}$$

where $W_1(t)$ and $W_2(t)$ are independent standard Brownian motion.

In each level of simulation we will have the Lipschitz function $f(x_{h^{(k)}}) = \sin(x_{h^{(k)}})$ to estimate the following quantity $\left(f(x_{h^{(k)}}) - f(x_{h^{(k-1)}})\right)$, where the $x_{h^{(k)}}$ and $x_{h^{(k-1)}}$ are two discrete approximations with different time-steps $h^{(k)}$ and $h^{(k-1)}$ respectively but they have the same Brownian path. The difference here is that for the combined

method we will use the same strategy in chapter 4 to control the non-invertible matrix. The initial value of the number of simulation N_k for the estimation will be 2^{22}. In the following table we will show for each level the number of simulation N_k, step-size h and the result of the simple estimator $\hat{Y}_k$ with its confidence interval.

Now we will run the Matlab code in Listing 5.16 to get the result of each level for the simple estimator $\hat{Y}_k$ in (5.5) with number of levels $n = 9$ and the initial value of the number of simulation N_k for the estimation will be 2^{22}. Also each simulation is for the same Brownian path.

Table 5.3: multilevel with combined method in 2-dimensional with non-invertible matrix

level k	Number of simulation N_k	step-size $h = \frac{T}{2^{(n-1)}}$	Simple estimator $\hat{Y}_k$	The confidence interval of $\hat{Y}_k$
1	4194304	1	0.1266	(0.1260, 0.1272)
2	1763488	1/2	0.0621	(0.0613, 0.0630)
3	741455	1/4	0.0150	(0.0140, 0.0159)
4	311744	1/8	0.0049	(0.0039, 0.0059)
5	131072	1/16	0.0017	(0.0007, 0.0027)
6	55109	1/32	0.00079	(-0.0001, 0.0017)
7	23170	1/64	0.00045	(-0.0004, 0.0013)
8	9742	1/128	0.00056	(-0.0002, 0.0013)
9	4096	1/256	0.00084	(0.0002, 0.0015)

For the multilevel MC method with the combined method we have two terms, the first one comes from the estimation of the top level μ_0 which has step-size h^0 i.e.

$$\begin{aligned} Ef(x_h) &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) \\ &\approx -0.0128 \end{aligned}$$

The other terms comes from the summation of simple estimator $\hat{Y}_k$ which for this SDE will be

$$\sum_{k=1}^{n} \frac{1}{N_k} \sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)}) \right) \approx 0.21294$$

Therefore the final value of the estimator for the multilevel MC method with the combined method is

$$\begin{aligned} \mu_n &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) + \sum_{k=1}^{n} \frac{1}{N_k} \sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)}) \right) \\ &\approx -0.0128 + 0.21294 = 0.2002 \end{aligned} \tag{5.16}$$

When we get to the results of the multilevel MC method with trivial coupling we will compare it with table 5.3 and some explanations about the figures will be discussed in more details.

After we have obtained the result of the multilevel Monte Carlo method for the combined method, we would like to compare it with scheme (1.22) with trivial coupling and see if the results will be approximately the same.

Now we need to do the implementation of the same SDEs in the previous section but for the multilevel MC method for the scheme (1.22) with trivial coupling. After that we will compare this results with the previous results in table 5.3 and see if they will be approximately agreed.

5.5.2 The multilevel implementation of the trivial coupling in two-dimensional case with invertible matrix

Now we want to do the same as the previous section and we will run the Matlab code in Listing 5.17 to get the result of each level for the simple estimator $\hat{Y}_k$ in (5.5) with number of levels $n = 9$ and the initial value of the number of simulation N_k for the estimation will be 2^{22}. Also each simulation is for the same Brownian path.

Table 5.4: multilevel with trivial coupling in 2-dimensional case with non-invertible matrix

level k	Number of simulation N_k	step-size $h = \frac{T}{2^{(n-1)}}$	Simple estimator $\hat{Y}_k$	The confidence interval of $\hat{Y}_k$
1	4194304	1	0.1351	(0.1345, 0.1357)
2	2097152	1/2	0.0633	(0.0625, 0.0641)
3	1048576	1/4	0.0146	(0.0137, 0.0155)
4	524288	1/8	0.0041	(0.0031, 0.0051)
5	262144	1/16	0.0019	(0.0009, 0.0029)
6	131072	1/32	0.00056	(-0.0005, 0.0016)
7	65536	1/64	0.00005	(-0.0010, 0.0011)
8	32768	1/128	0.00027	(-0.0008, 0.0013)
9	16384	1/256	-0.00038	(-0.0015, 0.0007)

Here the estimate of the top level μ_0 which has step-size h^0 i.e.

$$\begin{aligned} Ef(x_h) &= \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) \\ &\approx -0.0137 \end{aligned}$$

and the summation of simple estimator $\hat{Y}_k$ will be

$$\sum_{k=1}^{n} \frac{1}{N_k}\sum_{i=1}^{N_k} \left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right) \approx 0.2195$$

Therefore the estimator for the multilevel MC method with the trivial coupling is

$$\mu_n = \frac{1}{N}\sum_{i=1}^{N} f(x_h^{(i)}) + \sum_{k=1}^{n}\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f(x_{h^{(k)}}^{(i)}) - f(x_{h^{(k-1)}}^{(i)})\right)$$
$$\approx -0.0137 + 0.2195 = 0.2057 \tag{5.17}$$

Similar to the previous section, we have here the implementation of multilevel MC method with the combined method and trivial coupling with the non-invertible matrix. it can be seen from 5.16 that the value of the final calculation of the multilevel MC method with combined method which is 0.2002 has approximately agreed with the value for the trivial coupling in 5.17 which is 0.2057. In addition, table 5.3 shows the results of the implementation and the computation by using the multilevel MC method with the combined method, one could see from the number of simulation N_k that it is proportional to $\sqrt{V_k h_k}$ therefore we need to divide each time by $\sqrt{2^{5/2}}$ for the next level. From table 5.3 we started the number of simulation by 4194304 and for the second level the number of simulation will be divided by $\sqrt{2^{5/2}}$ that is $\frac{4194304}{\sqrt{2^{5/2}}} \approx 1763487$ and so on. Moreover, we could see from both tables 5.3 and (5.4) that when we increase the number of levels then the value of simple estimating $\hat{Y}_k$ decreases.

In the other hand, for the multilevel MC method with the trivial coupling for non-invertible matrix in table 5.4 the number of simulations is divided by two every time when we get to the new level. Here we start the first level with 4194304 simulation then divide by 2 for the next level which will give $\frac{4194304}{2} = 2097152$ and so on for the following level. This means by the observation that we can see a faster decrease in the number of simulations for the multilevel MC method with the combined method as compared to the multilevel MC method with the trivial coupling. As a result of this the computational saving has been achieved from calculating the computational load for the multilevel MC method with combined method, in contrast to the multilevel MC method with the trivial coupling.

The following plotting is shows the decreasing of the number of simulations N_k against the number of level for three different schemes the multilevel MC method with the exact coupling, the multilevel MC method with the trivial coupling and the multilevel MC method with combined method.

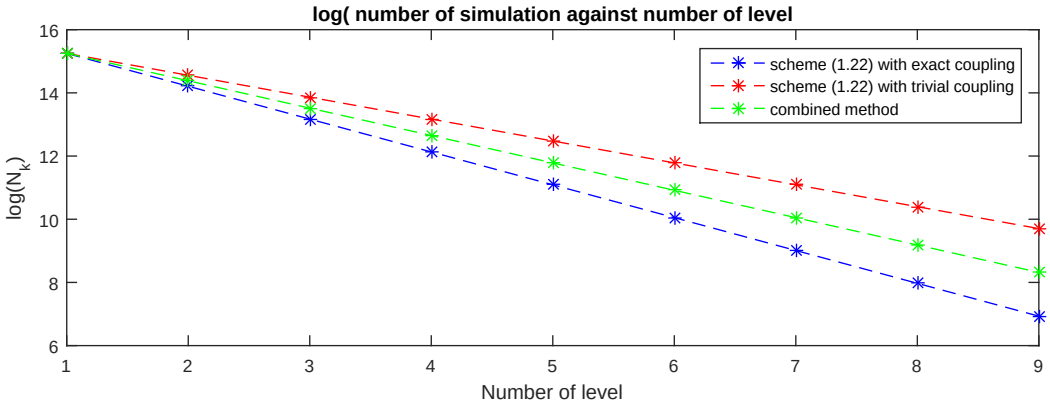


Figure 5.1: Decreasing of the Number of simulations for the schemes

5.6 Appendix

Listing 5.1: code for estimating the error for two-dimensional Euler scheme

```
T=1; N=10; R=10000; h=T/N; s=sqrt(T/(2*N)); q=0;
 for r=1:R, x=2; y=0; xx=2; yy=0;
for m=1:N, wl=s*randn; vl=s*randn; wr=s*randn;
vr=s*randn; w=wl+wr; v=vl+vr;
 u=x+y*wl+(x+(m-1)*h)*vl; y=y+exp(-y^2)*wl+(x-y)*vl; x=u;
 u=x+y*wr+(x+(m-1/2)*h)*vr; y=y+exp(-y^2)*wr+(x-y)*vr; x=u;
 u=xx+yy*w+(xx+(m-1)*h)*v; yy=yy+exp(-yy^2)*w+(xx-yy)*v;
xx=u; end
q=q+abs(x-xx)+abs(y-yy); end
(q/R)
```

Listing 5.2: code to approximate the double integrals A_{12} andA_{21}

```
function [w,v,J1,J2] = F_year_J(N,h)
N=5;  % Here N=p for the truncation of the series in (1.17)

T=1; h=T/N; s=sqrt(h); N1=randn; N2=randn; N7=randn;
w=s*N1; v=s*N2;

 f=0; g1=0; g2=0; z=0; N3=randn(1,N); N4=randn(1,N);
N5=randn(1,N); N6=randn(1,N);
for n=1:N;
f=f+(n.^(-2)); c=(1/12)-((2*(pi).^2).^(-1))*f;
g1=g1+(n.^(-1))*N3(n);
g2=g2+(n.^(-1)).*N4(n);
z=z+(n.^(-1)).*(N3(n).*N6(n)-(N5(n).*N4(n)));
A=1./(2*pi)*z;
% to calculate formula (1.18)
end
```

```
B=(1/2)*(w*v);
d1=(-1/pi)*sqrt(2*h)*g1-(2*sqrt(h*c)*N7);
% to calculate formula (1.19)
d2=(-1/pi)*sqrt(2*h)*g2-(2*sqrt(h*c)*N7);
% to calculate formula (1.19)
J1=(1/2)*h*N1*N2-(1/2)*s*(d2*N1-(d1*N2))+h*A;
% to calculate formula (1.17)
J2=2*B-J1;
end
```

Listing 5.3: code for estimating the error for two-dimensional Milstein scheme

```
T=1; NN=10;  h=T/NN; R=10000; q=0;
 for r=1:R, x=2; y=0; xx=2; yy=0;
for m=1:NN, hh=h/2; N=10;
 [wL,vL,J1L,J2L] = F_year_J(N,hh);
[wr,vr,J1r,J2r] = F_year_J(N,hh);
    w=wL+wr; v=vL+vr; J1=J1L+J1r+wr*vL; J2=J2L+J2r+wL*vr;
    u=x+y*wL+(x+(m-1)*h)*vL+exp(-y.^2)*(1/2)*(wL.^2-hh)
          +(x-y)*J1L+y*J2L+(x+(m-1)*h)*(1/2)*(vL.^2-hh);
y=y+exp(-y.^2)*wL+(x-y)*vL-2*y*exp(-2*y.^2)*(1/2)*(wL.^2-hh)
          +(y-exp(-y.^2))*J2L+(y+(m-1)*h)*(1/2)*(vL.^2-hh)
          -2*y*exp(-y.^2)*(x-y)*J1L; x=u;
    u=x+y*wr+(x+(m-1/2)*h)*vr+exp(-y.^2)*(1/2)*(wr.^2-hh)
          +(x-y)*J1r+y*J2r+(x+(m-1/2)*h)*(1/2)*(vr.^2-hh);
y=y+exp(-y.^2)*wr+(x-y)*vr-2*y*exp(-2*y.^2)*(1/2)*(wr.^2-hh)
          +(y-exp(-y.^2))*J2r+(y+(m-1/2)*h)*(1/2)*(vr.^2-hh)
          -2*y*exp(-y.^2)*(x-y)*J1r; x=u;
    u=xx+yy*w+(xx+(m-1)*h)*v+exp(-yy.^2)*(1/2)*(w.^2-h)
          +(xx-yy)*J1+yy*J2+(xx+(m-1)*h)*(1/2)*(v.^2-h);
yy=yy+exp(-yy.^2)*w+(xx-yy)*v-2*yy*exp(-2*yy.^2)*(1/2)*(w.^2-h)
          +(yy-exp(-yy.^2))*J2+(yy+(m-1)*h)*(1/2)*(v.^2-h)
```

```
            -2*yy*exp(-yy.^2)*(xx-yy)*J1; xx=u; end
q=q+abs(x-xx)+abs(y-yy); end
(q/R)
```

The below code for calculateing $\left\{\frac{b_{jk}(\tau_n,\Upsilon_n^j)-b_{jk}}{\sqrt{h}}\right\}$ and $\tau_{ik\ell}=\frac{1}{2}\sum_j c_{ij}\left\{\frac{b_{jk}(\tau_n,\Upsilon_n^j)-b_{jk}}{\sqrt{h}}\right\}$.

Listing 5.4: calculating the above terms

```
function [UU, XX, GG]=exactlastone(YA,s,m,h,x)
PR=0;

      for r=1:length(YA);
 UU=feval(YA,(m-1)*h,x);  C=inv(UU);
          Y1=x+ UU(:,r)*s;

       C1=1/s*(feval(YA,(m-1)*h,Y1)-UU);
      QQ(:,:,r)=C1;
      K(:,:,r)=(1/2)*C*C1;
  FF(:,r,:)=(1/2)*C*C1;
      end

for j=1:length(YA);
for i=1:length(YA);


      AB=(1/2)*(K-FF);
          VV=squeeze(K(i,:,:));
          dd=VV';
          GG(:,:,i)=(1/2)*(VV-dd);

end


        XX=QQ;
end
```

THE COUPLING CODE

(1) Generating some normal distributions $N(0,1)$ variables U_1', U_2', Q, R

(2) α taking the value ± 1 with probability $\frac{1}{2}$ each.

(3) Then set $V_2' = U_2'$, $\tilde{U}_1 = \alpha Q$ and $\tilde{U}_2 = \alpha R$

(4) $b = \epsilon a R$, $c = -\epsilon a Q U_2'$ and define $Y = U_1' + \alpha(bU_1' + c)$

(5) $F(y) = \frac{1}{2}\left\{\Phi(\frac{y-c}{1+b}) + \Phi(\frac{y+c}{1-b})\right\}$ is the cumulative distribution function of Y (here Φ is the c.d.f of $N(0,1)$

(6) $V_1' = \Phi^{-1}(F(Y))$

(7) $V = R_\theta V'$. (8) $U = R_\theta U'$. (9) $U^* = R_\theta \tilde{U}$.

(10) $U_i = Y_i + Z_i$ and $U_i^* = Y_i - Z_i$

The below code for calculating the exact coupling method

Listing 5.5: calculating multilevel with exact coupling

```
function [Z1, Z2, Y1, Y2, V1, V2] = coupling(aa,a1,a2,s);
UB1=randn; UB2=randn; Q=randn; R=randn;
% generating some normal distribution as in (1)
       u=rand; if u<0.5  zz=1;  else zz=-1;
                % Here we calculate (2)
     end
       UU1=zz*Q;  UU2=zz*R; VB2=UB2;
                % We set some definition in (3)
 Mn=s*aa*R; c=-s*aa*Q*UB2; Y=UB1+zz*(Mn*UB1+c);
% Here we calculate b and c and Y as in (4)

 Er1=erf((1/sqrt(2))*((Y-c)/(1+Mn)));
 A1=1/2*(1+Er1);
% to calculate the c.d.f for the F(y) in (5)
 Er2=erf((1/sqrt(2))*((Y+c)/(1-Mn)));
```

```
  A2=1/2*(1+Er2);
     % to calculate the c.d.f for the F(y) in (5)
        Fy=1/2*(A1+A2);
                    % to calculate F(y) in (5)
     VB1=sqrt(2)*erfinv(2*Fy-1);
          % Here we find V_1^\prime in (6)
     V1=(a1/aa)*VB1-(a2/aa)*VB2;
          % we calculate V in (7)
     V2=(a2/aa)*VB1+(a1/aa)*VB2;
          % we calculate V in (7)
     U1=(a1/aa)*UB1-(a2/aa)*UB2;
          % we calculate U in (8)
     U2=(a2/aa)*UB1+(a1/aa)*UB2;
          % we calculate U in (8)
     US1=(a1/aa)*UU1-(a2/aa)*UU2;
          % we calculate U^* in (9)
     US2=(a2/aa)*UU1+(a1/aa)*UU2;
          % we calculate U^* in (9)
     Z1=(1/2)*(U1-US1); Z2=(1/2)*(U2-US2);
          % we calculate Z in (10)
     Y1=(1/2)*(U1+US1); Y2=(1/2)*(U2+US2);
          % we calculate Y in (10)


end
```

Listing 5.6: calculating the SDEs (2.37)

```
function AAA=exatcouplinginvertible(YA,x0,T,N)
h=T/N; hh=T/(2*N); s=sqrt(T/N); ss=sqrt(T/(2*N)); RR=2000;
q=0;
 for r=1:RR,      x=x0;  y=x0;
       for m=1:N;
      [UU, XX, GG]=exactlastone(YA,ss,m,h,x);
      a1=GG(1,2,1);
      a2=GG(1,2,2);
      aa=(a1^2+a2^2)^(1/2);
     [Z1, Z2, Y1, Y2, V1, V2] = coupling(aa,a1,a2,s);
      wL=s*Y1; wr=s*Z1; w=s*V1; vL=s*Y2; vr=s*Z2;
                v=s*V2; B1=1/2*wL*vL; B2=1/2*wr*vr; B=1/2*w*v;
    x=x+UU*[wL; vL]+XX(:,:,1)*[1/2*(wL.^2-hh); B1]
          +XX(:,:,2)*[B1; (1/2)*(vL.^2-hh)];
    [UU, XX]=exactlastone(YA,ss,m,h,x);

    x=x+UU*[wr; vr]+XX(:,:,1)*[1/2*(wr.^2-hh); B2]
          +XX(:,:,2)*[B2; (1/2)*(vr.^2-hh)];

    [UU, XX]=exactlastone(YA,s,m,h,y);
    y=y+UU*[w; v]+XX(:,:,1)*[1/2*(w.^2-h); B]
          +XX(:,:,2)*[B; (1/2)*(v.^2-h)];
       end
  q=q+abs(x(1)-y(1))+abs(x(2)-y(2));

 end
 AA=q;    AAA=q/RR
end
```

The below code for calculating $\left\{\frac{b_{jk}(\tau_n,\Upsilon_n^j)-b_{jk}}{\sqrt{h}}\right\}$

Listing 5.7: calculating $\left\{\frac{b_{jk}(\tau_n,\Upsilon_n^j)-b_{jk}}{\sqrt{h}}\right\}$

```
************************************************


function [UU, XX]=mfileCCC44(YA,ss,m,h,x)


for r=1:length(YA);
 UU=feval(YA,(m-1)*h,x);
           Y=x+ UU(:,r)*ss;


        C1=1/ss*(feval(YA,(m-1)*h,Y)-UU);
       QQ(:,:,r)=C1;


       end


         XX=QQ;
end
```

Where the below code for calculating the approximation coupling

Listing 5.8: calculating the approx. coupling in equation (3.11)

```
function [UU, XX, w, WW]=ApproxthreeEND(YA,s,m,h,y,U,US)
PR=0;
YY=U';
      for r=1:length(YA);
 UU=feval(YA,(m-1)*h,y);  C=inv(UU);
          Y1=y+ UU(:,r)*s;

       C1=1/s*(feval(YA,(m-1)*h,Y1)-UU);
      QQ(:,:,r)=C1;
      K(:,:,r)=(1/2)*C*C1;
  FF(:,r,:)=(1/2)*C*C1;
      end
for j=1:length(YA);
for i=1:length(YA);

    AB=(1/2)*(K-FF);
          VV=squeeze(K(i,:,:));
          dd=VV';
          GG(:,:,i)=(1/2)*(VV-dd);
 end
EE(:,:,j)=( AB(:,:,j)*YY+AB(:,:,j)'*YY);
 MM(:,:,j)=(2*(AB(:,:,j)'*AB(:,:,j))+2*AB(:,:,j)^2+4*AB(:,:,j)
*trace(AB(:,:,j))+2*(AB(:,:,j)')^2+2*(AB(:,:,j)*AB(:,:,j)'));
 THI=AB(:,:,j)*AB(:,:,j)+AB(:,:,j)*trace(AB(:,:,j))
     +(1/2)*AB(:,:,j)'*AB(:,:,j);
      PR=PR+THI;
end
sum1=0; sum2=0;
```

```
for jj=1:length(YA);
  drpsi=YY'*AB(:,:,jj)*YY*EE(:,:,jj);
    dr3psi=(1/4)*(MM(:,:,jj)*YY+MM(:,:,jj)'*YY);
    sum1=sum1+drpsi;
    sum2=sum2+dr3psi;
    PO(jj)=YY'*GG(:,:,jj)*US';
end


    DrTHI=PR*YY+PR'*YY;
     DrPsi=sum1;
  Dr3Psi=sum2;
 w=s*(YY+s*PO'+s^2*(1/2)*(-Dr3Psi-DrPsi+DrTHI));
      WW=(1/2)*(w*w'-h*eye( length(YA), length(YA)));
        XX=QQ;
end
```

Listing 5.9: calculating the approximate coupling for the SDEs (3.23)

```
function AAA=AproxcouplingInverend(YA,x0,T,N)
f=2;
h=T/N; hh=T/(2*N); s=sqrt(T/N); ss=sqrt(T/(2*N)); RR=400; q=0;
 for r=1:RR,      x=x0;  y=x0;
       for m=1:N;
      U=randn(1,f); US=randn(1,f);
    Z=(1/2)*(U'-US'); Y2=(1/2)*(U'+US');
     [XX, UU]=mfileCCC44(YA,ss,m,h,x);


     WL=s*Y2;  WR=s*Z; BB1=(1/2)*(WL*WL'-hh*eye( f,f));
          BB2=(1/2)*(WR*WR'-hh*eye( f,f));
    x=x+UU*WL+XX(:,:,1)*BB1(1,:)'+XX(:,:,2)*BB1(2,:)';


    [XX, UU]=mfileCCC44(YA,ss,m+1/2,h,x);

     x=x+UU*WR+XX(:,:,1)*BB2(1,:)'+XX(:,:,2)*BB2(2,:)';

    [UU, XX, w, WW]=ApproxthreeEND(YA,s,m,h,y,U,US);
    y=y+UU*w+XX(:,:,1)*WW(1,:)'+XX(:,:,2)*WW(2,:)';
       end
  q=q+abs(x(1)-y(1))+abs(x(2)-y(2));

 end

AAA=q/RR
end
```

Listing 5.10: calculating the combined method for the exact coupling

```
S=[ 400, 800, 1600, 3200, 6400];
Error=zeros(1,length(S));
for i=1:length(S)
    Error(1,i)=log(errorRungcouplingTEST('bk',[2; 0],1,S(1,i)));
end
h=1./S;
fad1=log(h)
plot(log(h), Error)
```

Listing 5.11: calculating the combined method for the exact coupling for the SDEs (4.40)

```
function AAA=errorRungcouplingTEST(bk,x0,T,N)
h=T/N; hh=T/(2*N); s=sqrt(T/(N)); ss=sqrt(T/(2*N));
RR=2000; q=0; ES=0;
 for r=1:RR,      x=x0;  y=x0;
  for m=1:N;
       UB1=randn; UB2=randn; Q=randn; R=randn;
        [UU, XX, GG]=exactlastone(bk,ss,m,h,x);
      a1=GG(1,2,1);
      a2=GG(1,2,2);
      aa=(a1^2+a2^2)^(1/2);

    E0=aa.^2;
     if E0>(1/h)

          wl=ss*UB1; wR=ss*UB2; w1=wl+wR; vl=ss*Q;
                          vR=ss*R; v1=vl+vR; B11=1/2*wl*vl;
                          B22=1/2*wR*vR; BB=1/2*w1*v1;
  x=x+UU*[wl; vl]+XX(:,:,1)*[1/2*(wl.^2-hh); B11]
```

```
     +XX(:,:,2)*[B11; (1/2)*(vl.^2-hh)];
         [UU, XX]=exactlastone(bk,ss,m+1/2,h,x);
    x=x+UU*[wR; vR]+XX(:,:,1)*[1/2*(wR.^2-hh); B22]
          +XX(:,:,2)*[B22; (1/2)*(vR.^2-hh)];
          [UU, XX]=exactlastone(bk,s,m,h,y);
    y=y+UU*[w1; v1]+XX(:,:,1)*[1/2*(w1.^2-h); BB]
          +XX(:,:,2)*[BB; (1/2)*(v1.^2-h)];

     else

    [Z1, Z2, Y1, Y2, V1, V2] = coupling(aa,a1,a2,s);
      wL=s*Y1; wr=s*Z1; w=s*V1; vL=s*Y2; vr=s*Z2;
               v=s*V2; B1=1/2*wL*vL; B2=1/2*wr*vr; B=1/2*w*v;

    x=x+UU*[wL; vL]+XX(:,:,1)*[1/2*(wL.^2-hh); B1]
          +XX(:,:,2)*[B1; (1/2)*(vL.^2-hh)];
    [UU, XX]=exactlastone(bk,ss,m+1/2,h,x);

    x=x+UU*[wr; vr]+XX(:,:,1)*[1/2*(wr.^2-hh); B2]
          +XX(:,:,2)*[B2; (1/2)*(vr.^2-hh)];

    [UU, XX]=exactlastone(bk,s,m,h,y);
    y=y+UU*[w; v]+XX(:,:,1)*[1/2*(w.^2-h); B]
          +XX(:,:,2)*[B; (1/2)*(v.^2-h)];
     end
     EM=h^2*(min(1,10*E0*h));
% calculating minimum function local error combined method
  end

   ES=ES+EM;
```

```
% calculating minimum function local error combined method
  q=q+abs(x(1)-y(1))+abs(x(2)-y(2));

 end
 ERR=ES/RR
% calculating minimum function local error combined method

AAA=q/RR
end
```

Listing 5.12: calculating the $E|p|^2$ in (4.51)

```
function [XX,   UU, PU2]=PUsquare(bk,ss,m,h,x)
 q=0; GR=0;
 ZE = ones(length(bk)); ZE(logical(eye(size(ZE)))) = 0;
      for r=1:length(bk);
 UU=feval(bk,(m-1)*h,x);  C=inv(UU);
          Y1=x+ UU(:,r)*ss;

       C1=1/ss*(feval(bk,(m-1)*h,Y1)-UU);
      QQ(:,:,r)=C1;
      K(:,:,r)=(1/2)*C*C1;
  FF(:,r,:)=(1/2)*C*C1;
      end
sm1=0;  sm4=0;   sm7=0; sm8=0; sm9=0;  DLTA=0;
sm10=0;
for i=1:length(bk);

      AB=(1/2)*(K-FF);

end
for m=1:length(bk);
```

```
    MM(:,:,m)=(2*(AB(:,:,m)'*AB(:,:,m))+2*AB(:,:,m)^2
          +4*AB(:,:,m)*trace(AB(:,:,m))
          +2*(AB(:,:,m)')^2+2*(AB(:,:,m)*AB(:,:,m)'));
   THI=AB(:,:,m)*AB(:,:,m)+AB(:,:,m)*trace(AB(:,:,m))
     +(1/2)*AB(:,:,m)'*AB(:,:,m);
      q=q+THI;
      sumG=- MM(:,:,m)- MM(:,:,m)';
      GR=GR+sumG;
end
G1=(GR/2+q+q');
 BB=trace(G1'*G1);

ONE=ones(2,1);
for n=1:length(bk);
    SG1=AB(n,:,1)+AB(:,n,1)';
SG2=AB(n,:,2)+AB(:,n,2)';

   H1=SG1'*ONE'; H2=SG2'*ONE';
   H3=ONE*SG1; H4=ONE*SG2;
        DLkrk1=AB(:,:,1).*H1+AB(:,:,2).*H2;
        DLrkk1=(AB(:,:,1).*H3+AB(:,:,2).*H4);
        Bmmkk1=(ONE*G1(n,:).*AB(:,:,1).*H3
                    +ONE*G1(n,:).*AB(:,:,2).*H4);
        Bmkmk1=(G1(:,n)*ONE'.*AB(:,:,1).*H1
                    +G1(:,n)*ONE'.*AB(:,:,2).*H2);

    DLiii=((diag(AB(:,:,1))).*SG1'+(diag(AB(:,:,2))).*SG2');
    DLkkr=(diag(AB(:,:,1))*SG1+diag(AB(:,:,2))*SG2);

    Bmkkm=(G1(n,:)'.*diag(AB(:,:,1))*SG1
```

```
          +G1(n,:)'.*diag(AB(:,:,2))*SG2);

    Biiii=3*(G1(n,:)'.*(diag(AB(:,:,1))).*SG1'
          +G1(n,:)'.*(diag(AB(:,:,2))).*SG2');



    sm7=sm7+Biiii;
    sm8=sm8+Bmmkk1.*ZE;
    sm9=sm9+(ZE.*Bmkkm);
    sm4=sm4+ZE.*Bmkmk1;
    sm1=sm1+15*DLiii.^2;
    sm10=sm10+6*DLiii*ONE'.*(ZE.*DLkkr+DLkrk1.*ZE+ZE.*DLrkk1);
 DLTA=DLTA+3*(ZE.*DLkkr+DLkrk1.*ZE+ZE.*DLrkk1).^2;

end
smBiiii=sum(sm7);
smBmmkk=sum(sum(sm8));
smBmkkm=sum(sum(sm9));
smBmkmk=sum(sum(sm4));
smDLiii=sum(sm1);
smDLTA=sum(sum(DLTA));
smDLiiiDLkkr=sum(sum(sm10));



PU2=(BB-2*smBiiii-2*smBmmkk-2*smBmkkm-2*smBmkmk
+smDLiii+smDLTA+smDLiiiDLkkr)/4;


        XX=QQ;
```

Listing 5.13: calculating the combined method for the approximate coupling for the

SDEs (4.40)

```
function AAA=AproxcouplingNONinver(bk,x0,T,N)
f=2;
h=T/N; hh=T/(2*N); s=sqrt(T/N); ss=sqrt(T/(2*N));
RR=2000; q=0; Z=0;
 for r=1:RR,      x=x0;  y=x0;
       for m=1:N;
          U=randn(1,f); US=randn(1,f);
    Z=(1/2)*(U'-US'); Y2=(1/2)*(U'+US');
     [XX,   UU, PU2]=PUsquare(bk,ss,m,h,x);
MX1=XX(:,:,1);
MX2=XX(:,:,2);
EX=((MX1(1,2)-MX2(1,1))^2+(MX1(2,2)-MX2(2,1))^2);

     if PU2>(EX/h)

         wL=ss*randn; wR=ss*randn; w1=wL+wR; vL=ss*randn;
     vR=ss*randn; v1=vL+vR; B11=1/2*wL*vL; B22=1/2*wR*vR;
     BB=1/2*w1*v1;
  x=x+UU*[wL; vL]+XX(:,:,1)*[1/2*(wL.^2-hh); B11]
    +XX(:,:,2)*[B11; (1/2)*(vL.^2-hh)] ;
          [XX, UU]=mfileCCC44(bk,ss,m+1/2,h,x);
    x=x+UU*[wR; vR]+XX(:,:,1)*[1/2*(wR.^2-hh); B22]
          +XX(:,:,2)*[B22; (1/2)*(vR.^2-hh)];
         [XX, UU]=mfileCCC44(bk,s,m,h,y);
    y=y+UU*[w1; v1]+XX(:,:,1)*[1/2*(w1.^2-h); BB]
          +XX(:,:,2)*[BB; (1/2)*(v1.^2-h)];

     else
```

```
     WL=s*Y2;   WR=s*Z; BB1=(1/2)*(WL*WL'-hh*eye( f,f));
          BB2=(1/2)*(WR*WR'-hh*eye( f,f));

    x=x+UU*WL+XX(:,:,1)*BB1(1,:)'+XX(:,:,2)*BB1(2,:)';


    [XX, UU]=mfileCCC44(bk,ss,m+1/2,h,x);

     x=x+UU*WR+XX(:,:,1)*BB2(1,:)'+XX(:,:,2)*BB2(2,:)';


      [UU, XX, w, WW]=ApproxthreeEND(bk,s,m,h,y,U,US);
    y=y+UU*w+XX(:,:,1)*WW(1,:)'+XX(:,:,2)*WW(2,:)';
     end
       end
   q=q+abs(x(1)-y(1))+abs(x(2)-y(2));
 Z(r)=abs(x(1)-y(1))+abs(x(2)-y(2));
 end
 v=std(Z);
   P1=q/RR;
    d=1.66;
    a=d*(v/sqrt(RR));
      p=[P1-a; P1+a];
   Int=p


AAA=q/RR
end
```

Listing 5.14: calculating multilevel with exact coupling

```
function MU=multileveltest(YA,x0,T, NK)
GG=multileveltestmuu(YA,x0,T,NK);
  BB=0; K1=zeros(1,9);
  for n=1:7;
      N=2.^(n-1); q=0; Z=0;
      h=T/N;  hh=T/(2*N); s\
                =sqrt(h); ss=sqrt(hh); RR=NK/sqrt(8^(n-1));
      SIM=int32(RR);
     for r=1:SIM,     x=x0;  y=x0;
       for m=1:N;
              [C1, C2, C, UU]=mfileCCC(YA,ss,m,h,x);
     [to112,to121, to212,to221] = mfilfortaoo(C1,C2,C);
 a1=(to112-to121)/2; a2=(to212-to221)/2; aa=(a1^2+a2^2)^(1/2);
    [Z1, Z2, Y1, Y2, V1, V2] = coupling(aa,a1,a2,s);
    wL=s*Y1; wr=s*Z1; w=s*V1; vL=s*Y2; vr=s*Z2; v=s*V2;
          B1=1/2*wL*vL; B2=1/2*wr*vr; B=1/2*w*v;
    u=x(1)+UU(1,1)*wL+UU(1,2)*vL+1/2*C1(1,1)*(wL.^2-hh)
          +C1(1,2)*B1+C2(1,1)*B1+1/2*C2(1,2)*(vL.^2-hh);
  x(2)=x(2)+UU(2,1)*wL+UU(2,2)*vL+1/2*C1(2,1)*(wL.^2-hh)
 +C1(2,2)*B1+1/2*C2(2,2)*(vL.^2-hh)+C2(2,1)*B1; x(1)=u;
             [C1, C2, C, UU]=mfileCCC(YA,ss,m,h,x);
    u=x(1)+UU(1,1)*wr+UU(1,2)*vr+1/2*C1(1,1)*(wr.^2-hh)
          +C1(1,2)*B2+C2(1,1)*B2+1/2*C2(1,2)*(vr.^2-hh);
  x(2)=x(2)+UU(2,1)*wr+UU(2,2)*vr+1/2*C1(2,1)*(wr.^2-hh)
     +C1(2,2)*B2+1/2*C2(2,2)*(vr.^2-hh)+C2(2,1)*B2; x(1)=u;
             [C1, C2, C, UU]=mfileCCC(YA,s,m,h,y);
    u=y(1)+UU(1,1)*w+UU(1,2)*v+1/2*C1(1,1)*(w.^2-h)
          +C1(1,2)*B+C2(1,1)*B+1/2*C2(1,2)*(v.^2-h);
```

```
    y(2)=y(2)+UU(2,1)*w+UU(2,2)*v+1/2*C1(2,1)*(w.^2-h)
          +C1(2,2)*B+1/2*C2(2,2)*(v.^2-h)+C2(2,1)*B; y(1)=u;
      end

       q=q+(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
       Z(r)=(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
     end
      v=std(Z);  P1=q/RR;
      d=1.66;    a=d*(v/sqrt(RR));
      p=[P1-a; P1+a];     FF=q/RR;
      Int=p;  BB=BB+FF;
  end
  MU=BB+GG
end
```

Listing 5.15: calculating multilevel with exact coupling

```
function GG=multileveltestmuuEULER(YA,y0,T,NK)
h=T;  s=sqrt(T);  RR=NK; q=0; m=0;
 for r=1:RR,        y=y0;

       UB1=randn;  R=randn;
      [C1, C2, UU]=mfileCCMU(YA,s,m,h,y);


    w=s*UB1;    v=s*R;    B=1/2*w*v;


    u=y(1)+UU(1,1)*w+UU(1,2)*v+1/2*C1(1,1)*(w.^2-h)
          +C1(1,2)*B+C2(1,1)*B+1/2*C2(1,2)*(v.^2-h);
    y(2)=y(2)+UU(2,1)*w+UU(2,2)*v+1/2*C1(2,1)*(w.^2-h)
          +C1(2,2)*B+1/2*C2(2,2)*(v.^2-h)+C2(2,1)*B; y(1)=u;


     q=q+(sin(y(1))-sin(y(2)));

 end

  GG=q/RR

end
```

Listing 5.16: calculating multilevel for the combined method

```
function MU=multilevelnoninvertable(b,x0,T)
GG=multileveltestmuu(b,x0,T);
  BB=0;
  for n=1:3;
      N=2.^(n-1); q=0; Z=0;
    h=T/N;
  hh=T/(2*N); s=sqrt(h); ss=sqrt(hh);
     RR=2^(22)/sqrt((2^(5/2))^(n-1));
  SIM=int32(RR);
     for r=1:SIM,      x=x0;  y=x0;
       for m=1:N;
     UB1=randn; UB2=randn; Q=randn; R=randn;
   [C1, C2, C, UU]=mfileCCC(b,ss,m,h,x);
   [to112,to121, to212,to221] = mfilfortaoo(C1,C2,C);
a1=(to112-to121)/2;  a2=(to212-to221)/2; aa=(a1^2+a2^2)^(1/2);
    EO=aa.^2;

     if EO>(1/h)
   wl=ss*UB1; wR=ss*UB2; ww=wl+wR; vl=ss*Q; vR=ss*R;
     vv=vl+vR; B11=1/2*wl*vl; B22=1/2*wR*vR; BC=1/2*ww*vv;
    u=x(1)+UU(1,1)*wl+UU(1,2)*vl+1/2*C1(1,1)*(wl.^2-hh)
          +C1(1,2)*B11+C2(1,1)*B11+1/2*C2(1,2)*(vl.^2-hh);
  x(2)=x(2)+UU(2,1)*wl+UU(2,2)*vl+1/2*C1(2,1)*(wl.^2-hh)
+C1(2,2)*B11+1/2*C2(2,2)*(vl.^2-hh)+C2(2,1)*B11; x(1)=u;
         [C1, C2, C, UU]=mfileCCC(b,ss,m+1/2,h,x);
    u=x(1)+UU(1,1)*wR+UU(1,2)*vR+1/2*C1(1,1)*(wR.^2-hh)
          +C1(1,2)*B22+C2(1,1)*B22+1/2*C2(1,2)*(vR.^2-hh);
 x(2)=x(2)+UU(2,1)*wR+UU(2,2)*vR+1/2*C1(2,1)*(wR.^2-hh)
```

```
+C1(2,2)*B22+1/2*C2(2,2)*(vR.^2-hh)+C2(2,1)*B22; x(1)=u;
        [C1, C2, C, UU]=mfileCCC(b,s,m,h,y);
    u=y(1)+UU(1,1)*ww+UU(1,2)*vv+1/2*C1(1,1)*(ww.^2-h)
          +C1(1,2)*BC+C2(1,1)*BC+1/2*C2(1,2)*(vv.^2-h);
 y(2)=y(2)+UU(2,1)*ww+UU(2,2)*vv+1/2*C1(2,1)*(ww.^2-h)
+C1(2,2)*BC+1/2*C2(2,2)*(vv.^2-h)+C2(2,1)*BC; y(1)=u;
     else

   [Z1, Z2, Y1, Y2, V1, V2] = coupling(aa,a1,a2,s);
   wL=s*Y1; wr=s*Z1; w=s*V1; vL=s*Y2; vr=s*Z2;
v=s*V2; B1=1/2*wL*vL; B2=1/2*wr*vr; B=1/2*w*v;
   u=x(1)+UU(1,1)*wL+UU(1,2)*vL+1/2*C1(1,1)*(wL.^2-hh)
     +C1(1,2)*B1+C2(1,1)*B1+1/2*C2(1,2)*(vL.^2-hh);
  x(2)=x(2)+UU(2,1)*wL+UU(2,2)*vL+1/2*C1(2,1)*(wL.^2-hh)
+C1(2,2)*B1+1/2*C2(2,2)*(vL.^2-hh)+C2(2,1)*B1; x(1)=u;

       [C1, C2, C, UU]=mfileCCC(b,ss,m+1/2,h,x);
   u=x(1)+UU(1,1)*wr+UU(1,2)*vr+1/2*C1(1,1)*(wr.^2-hh)
     +C1(1,2)*B2+C2(1,1)*B2+1/2*C2(1,2)*(vr.^2-hh);
  x(2)=x(2)+UU(2,1)*wr+UU(2,2)*vr+1/2*C1(2,1)*(wr.^2-hh)
     +C1(2,2)*B2+1/2*C2(2,2)*(vr.^2-hh)+C2(2,1)*B2; x(1)=u;

             [C1, C2, C, UU]=mfileCCC(b,s,m,h,y);
    u=y(1)+UU(1,1)*w+UU(1,2)*v+1/2*C1(1,1)*(w.^2-h)
          +C1(1,2)*B+C2(1,1)*B+1/2*C2(1,2)*(v.^2-h);
    y(2)=y(2)+UU(2,1)*w+UU(2,2)*v+1/2*C1(2,1)*(w.^2-h)
          +C1(2,2)*B+1/2*C2(2,2)*(v.^2-h)+C2(2,1)*B; y(1)=u;
     end
       end
       q=q+(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
```

```
  Z(r)=(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
    end
      VA=std(Z);
   P1=q/RR;
    d=1.66;
    a=d*(VA/sqrt(RR));
      p=[P1-a; P1+a];
   FF=q/RR
   Int=p
     BB=BB+FF;
  end
  GG;
  BB;
  MU=BB+GG;

end
```

Listing 5.17: calculating multilevel with trivial coupling

```
function MU=multilevelschemeten(YA,x0,T,NK)
GG=multileveltestmuuEULER(YA,x0,T,NK);
  BB=0;
  for n=1:7;
      N=2.^(n-1);
    h=T/N;
  hh=T/(2*N);  s=sqrt(h); ss=sqrt(hh); RR=NK/((2)^(n-1));
  SIM=int32(RR);
  q=0; Z=0;
     for r=1:SIM,      x=x0;  y=x0;
          for m=1:N;
    [C1, C2, C, UU]=mfileCCC(YA,ss,m,h,x);

 wl=ss*randn; wR=ss*randn; ww=wl+wR; vl=ss*randn;
vR=ss*randn; vv=vl+vR; B11=1/2*wl*vl;
B22=1/2*wR*vR; B=1/2*ww*vv;
    u=x(1)+UU(1,1)*wl+UU(1,2)*vl+1/2*C1(1,1)*(wl.^2-hh)
          +C1(1,2)*B11+C2(1,1)*B11+1/2*C2(1,2)*(vl.^2-hh);
 x(2)=x(2)+UU(2,1)*wl+UU(2,2)*vl+1/2*C1(2,1)*(wl.^2-hh)
+C1(2,2)*B11+1/2*C2(2,2)*(vl.^2-hh)+C2(2,1)*B11; x(1)=u;
          [C1, C2, C, UU]=mfileCCC(YA,ss,m+1/2,h,x);
    u=x(1)+UU(1,1)*wR+UU(1,2)*vR+1/2*C1(1,1)*(wR.^2-hh)
          +C1(1,2)*B22+C2(1,1)*B22+1/2*C2(1,2)*(vR.^2-hh);
 x(2)=x(2)+UU(2,1)*wR+UU(2,2)*vR+1/2*C1(2,1)*(wR.^2-hh)
+C1(2,2)*B22+1/2*C2(2,2)*(vR.^2-hh)+C2(2,1)*B22; x(1)=u;
          [C1, C2, C, UU]=mfileCCC(YA,s,m,h,y);
    u=y(1)+UU(1,1)*ww+UU(1,2)*vv+1/2*C1(1,1)*(ww.^2-h)
          +C1(1,2)*B+C2(1,1)*B+1/2*C2(1,2)*(vv.^2-h);
```

```
 y(2)=y(2)+UU(2,1)*ww+UU(2,2)*vv+1/2*C1(2,1)*(ww.^2-h)
+C1(2,2)*B+1/2*C2(2,2)*(vv.^2-h)+C2(2,1)*B; y(1)=u;


          end
    q=q+(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
  Z(r)=(sin(x(1))-sin(x(2)))-(sin(y(1))-sin(y(2)));
    end
     v=std(Z);
   P1=q/RR;
    d=1.66;
    a=d*(v/sqrt(RR));
      p=[P1-a; P1+a];
   FF=q/RR;
   Int=p;
     BB=BB+FF;
  end


  GG;
  BB;
  MU=BB+GG


end
```

References

[1] A. Alfonsi, B. Jourdain and A. Kohatsu-Higa, Pathwise optimal transport bounds between a one-dimensional diffusion and its Euler scheme, *Ann. Appl. Probab.* **24** (2014), 1049-1080.

[2] A. Alfonsi, B. Jourdain and A. Kohatsu-Higa, Optimal transport bounds between the time-marginals of multidimensional diffusion and its Euler scheme, *arXiv:* 1405.7007.

[3] A. M. Davie, Pathwise approximation of stochastic differential equations using coupling, preprint: www.maths.ed.ac.uk/∼ adavie/coum.pdf

[4] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag 1995.

[5] J. Komlós, P. Major and G. Tusnády, An approximation of partial sums of independent RV's and the sample DF. I, *Z. Wahr. und Wer. Gebiete* **32** (1975), 111-131.

[6] S. T. Rachev and L. Ruschendorff, *Mass Transportation Problems*, Volume 1, Theory; Volume 2, Applications. Springer-Verlag 1998.

[7] L. N. Vaserstein, Markov processes over denumerable products of spaces describing large system of automata (Russian), *Problemy Peredaci Informacii* **5** (1969), 64-72.

[8] M. Wiktorsson, Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions, *Ann. Appl. Probab.* **11** (2001), 470-487.

[9] T. Rydén and M. Wiktrosson, On the simulation of iteraled Itô integrals, *Stochastic Processes Appl.* **91** (2001), 151-168.

[10] S. Kanagawa, The rate of convergence for the approximate solutions of SDEs, *Tokyo J. Math* **12** (1989), 33-48.

[11] N. Fournier, Simulation and approximation of Lévy-driven SDEs, *ESIAM Probab. Stat.* **15** (2011), 233-248.

[12] E. Rio, Upper bounds for minimal distances in the central limit theorem, *Ann. Inst. Henri Poincaré Probab. Stat.* **45** (2009), 802-817.

[13] A. B. Cruzeiro, P. Malliavin and A. Thalmaier, Geometrization of Monte-Carlo numerical analysis of an elliptic operator: strong approximation, C. R. Math. Acad. Sci. Paris **338** (2004), 481-486.

[14] B. Charbonneau, Y. Svyrydov and P. Tupper, Weak convergence in the Prokhorov metric of methods for stochastic differential equations, *IMA J. Numer. Anal.* **30** (2010), 579-594.

[15] I. Gyöngy and N. Krylov, Existence of strong solutions for Itô's stochastic equations via approximations, *Probab. Theory Related Fields* **105** (1996), 143-158.

[16] M. B. Giles, Multilevel Monte Carlo path simulation, *Operations Research* **56** (2008), 607-617.

[17] M. B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, *Monte Carlo and quasi-Monte Carlo methods 2006*, 343-358, Springer, Berlin (2008).

[18] Kloeden, P.E; Platen, E. and Wright, I., The approximation of multiple stochastic integrals. J.Stoch. Anal. Appl. **10** (1992) 431-441.

[19] L.Hörmander. Hypoelliptic second order differential equations *Acta. Math.* **119** (1967) 147-171.

[20] Hairer, M. Malliavin's proof of Hörmander's theorem. Bull Math. Sci. **135** (2011), no 6-7, pp. 650-666.

[21] E. Rio, Asymptotic constants for minimal distances in the central limit theorem, Elect. Comm. in Probab. **16** (2011), 96-103.

[22] A. Davie. KMT theory applied to approximations of SDE. In *Stochastic Analysis and Applications 2014*, pages 185-201. Springer, (2014).

[23] Alhojilan Y. *Higher-order Numerical Scheme for Solving Stochastic Differential Equations*, PhD thesis, University of Edinburgh 2015.

[24] M. B. Giles and L. Szpruch, Antithetic multilevel Monte Carlo estimation for multidimensional SDEs without Lévy area simulation, arXiv:1202.6283.

[25] C. Villani, *Topics in Optimal Transportation*, AMS 2003.

[26] C. Villani, *Optimal Transport Old and New*, Springer-Verlag 2009.