University of Strathclyde
Department of Mathematics and Statistics

Mean Exit Times and Multi-level Monte Carlo with Applications in Finance

by

Mikolaj Roj

A thesis presented in fulfilment of the
requirements for the degree of
Doctor of Philosophy

2013

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Date:

Acknowledgments

I would like to thank my supervisors, Prof. Des Higham and Prof. Xuerong Mao, for their support, guidance and encouragement throughout my research and for giving me the opportunity to participate in various conferences, seminars, lectures and research visits. A big thank you goes to Prof. Mike Giles for conversations about the research topic and invitations to conferences. I would also like to thank my friends, Dr Lukasz Szpruch and Dr Stephen Corson, for numerous conversations about the research topic, mathematics and life in general.

I acknowledge the financial support provided to me by Engineering and Physical Sciences Research Council and Numerical Algorithms Group Ltd.

Last, but not least, I would like to express my gratitude to my parents, Ewa and Krzysztof, and my sister, Joanna, for their constant assistance and encouragement throughout the years. A special thanks goes to my partner, Aleksandra Stadler, for her support and patience which helped me survive the most difficult periods during this research.

Abstract

This thesis is focused broadly on the *stopped exit time* problem in the stochastic differential equations setting. It currently appears that one of the most efficient and powerful approaches to solve this problem (especially in higher dimensions for which there are no explicit formulae) is the simulation of sample paths of time discrete approximations. We compute the mean exit time using a Monte Carlo technique, which has the advantage of being straightforward to implement. However, controlling the Monte Carlo sampling error and large biases in the numerical method make this method computationally expensive. In our work we employ a variance reduction technique called multi-level Monte Carlo which dramatically reduces the complexity of the method. Surprisingly, even though we need to compute the quantity in a weak sense (an expected value), the multi-level method also relies on a strong convergence property of the numerical scheme. In order to justify the multi-level method we then establish a rate of strong convergence for exit times. We also present an extension to the basic method which reduces the computational complexity even further. The extended version uses a Brownian bridge technique which is applied on the simplest nontrivial numerical scheme for stochastic differential equations with a strong order of convergence one (the Milstein scheme.) Our results have been derived for multi-dimensional stochastic differential equations and can be applied to nonlinear stochastic differential equations models, including those arising in finance and chemical kinetics. Our theoretical work is complemented by computational tests for a number of practical problems.

Contents

List of Figures

Chapter 1

Introduction

If you can't explain it simply,
you don't understand it well enough.

Albert Einstein, 1879-1955

1.1 Motivation

Even though stochastic differential equations (SDEs) are one of the most popular tools used to model processes in financial mathematics, it is extremely rare that one can solve them in some explicit manner. Therefore, numerical methods have become a very useful tool to solve SDEs. It currently appears that the most efficient and applicable approach to solve SDEs is the simulation of sample paths of time discrete approximations. There are numerous examples in the literature where authors discretise SDEs, typically with Euler-type or Milstein-type schemes. Regarding applications, there are four main motivations for such simulations:

- computing an expected value of a function of some process using a Monte Carlo approach, for example to value a bond or an expected payoff of an option [5, 14, 45],
- generating time series in order to test parameter estimation algorithms [28],
- approximating the likelihood estimator effectively [85],

- estimating a mean exit time using Monte Carlo [3, 13, 71, 73].

In our work we focus our attention on the last problem. The approach of approximating a mean exit time by directly simulating paths of the SDE and applying a Monte Carlo technique, relative to the alternative of solving an associated deterministic partial differential equation, has the advantages of

- being straightforward to implement,
- coping naturally with high dimensions,
- dealing effectively with complicated boundaries.

However, in this context the inherently high cost of controlling the Monte Carlo sampling error is exacerbated by the large biases in the numerical method—exit time samples are less accurate than the corresponding samples of the solution process itself. Our work is thus motivated by a practical problem for which there is a need for efficient algorithms to approximate mean exit times in a more effective way and therefore using less computational effort.

1.2 Contribution

Here we list the major contributions of this thesis.

Strong convergence for exit times (Chapter 4.) We prove that the most basic discretisation, the Euler–Maruyama method, converges strongly to the solution of the multi-dimensional SDE with a rate slightly slower than 1/2 in terms of its ability to approximate exit times. Thanks to this we can justify the use of multi-level Monte Carlo (MLMC).

Non Lipschitz cases (Chapter 4.) The stopped exit time problem restricts the solution to a compact domain and therefore we may redefine f and g outside this domain, if necessary, in order to ensure that they are globally Lipschitz. This allows us to derive results that apply to a wide range of nonlinear SDE models that would otherwise pose analytical difficulties through non-differentiability of the drift at the origin or superlinear growth of the diffusion at infinity, including those arising in finance [1, 2, 16, 17, 57, 70] and stochastic population dynamics [7, 33, 75, 76, 84].

Adaptation of MLMC to mean exit times (Chapters 4 & 5.) We adapt the Euler-based multi-level algorithm developed by Giles [38] in order to estimate exit times using less computation than the standard Monte Carlo method. We also test the algorithm numerically on examples in cases ranging from one-dimensional to five-dimensional.

Strong convergence for exit times using Brownian bridges (Chapter 6.) We present a new numerical method for estimating the expected exit time of a multi-dimensional SDE. The algorithm uses a Brownian bridge technique to compute conditional exit probabilities. We note that some ideas behind the new algorithm were tested numerically in [89]. In order to justify the multi-level Monte Carlo approach we establish a rate of strong convergence for the exit time approximation of fine and coarse paths.

Application of the Brownian bridge algorithm to MLMC (Chapter 6.) We show that the general multi-level Monte Carlo philosophy can be exploited very effectively in the mean exit time context by computing conditional exit probabilities in a manner that allows pairs of coarse and fine grids to be combined. We prove that the variance of the new multi-level algorithm gives significant computational savings of two orders of magnitude, comparing with the crude Euler-Monte Carlo approach, in terms of the required root-mean-square accuracy. We also test the new method numerically in a two-dimensional case.

1.3 Outline

In Chapter 2 we recall numerical methods, concepts of weak and strong convergence, and the mean exit time problem. Chapter 3 introduces multi-level Monte Carlo and its extensions. It also presents financial applications for various processes in various settings. The main body of research is contained in Chapters 4, 5 and 6. In Chapter 4 we present the multi-level Monte Carlo algorithm adapted to the mean exit time problem. In order to justify MLMC we derive what appears to be the first proof of strong convergence of Euler–Maruyama in terms of its ability to approximate exit times. We show that the method performs well on scalar cases of a geometric Brownian motion and a mean reverting square root process.

Chapter 5 provides more numerical examples in the financial and neural network setting and shows that the multi-level algorithm performs well in more than one dimension. We also show that even if we violate the assumption of a non-compact domain, so that the convergence result is no longer applicable, we can still apply the multi-level method successfully. Chapter 6 provides the reader with a new Milstein-Monte Carlo algorithm to estimate expected exit times. The method is based on a novel representation of a mean exit time and a Brownian bridge interpolation technique. We then adapt this algorithm to the multi-level setting. The method yields significant computational savings of two orders of magnitude in terms of the required root-mean-square accuracy. We also test the algorithm numerically in a two-dimensional case. In the final chapter we summarise our findings and suggest some possible extensions for further research.

Chapter 2

Numerical Methods

> Everything should be made as simple as possible, but not simpler.
>
> Albert Einstein, 1879-1955

2.1 Introduction

In this chapter we present the most popular explicit approximation schemes, explain the concepts of weak and strong convergence, introduce the mean exit time problem and emphasise the need for a strong convergence result for exit times. For basic definitions and background of numerical schemes and stochastic processes we mention some key sources here instead of flooding the text with repeated citations [58, 67, 82, 87].

We begin with the system of stochastic differential equations (SDEs)

$$dx(s) = f(x(s))ds + g(x(s))dw(s), \tag{2.1}$$

with deterministic initial condition $x(0) = x_0$, over a finite time interval $[0, T]$. We assume x takes values in $\mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}^d$, and $g : \mathbb{R}^d \to \mathbb{R}^{d\times m}$. Here $w = \{w(t) : t \geq 0\}$ is a standard m-dimensional Brownian motion and we let $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_t)$ be a complete, filtered probability space satisfying the usual conditions. We specify an open set $O \subset \mathbb{R}^d$.

We impose the following condition on the drift and diffusion coefficients of the SDE throughout.

Assumption 2.1.1. *(**C² continuity**) Functions f and g have two continuous bounded derivatives on O.*

The above is imposed on f and g to ensure existence and uniqueness of strong solutions of SDE and to provide extra smoothness needed for (2.9).

2.2 Explicit Euler–Maruyama

The simplest discretisation scheme is the Euler–Maruyama (EM) method. Given any stepsize Δt, we define the partition $\mathcal{P}_{\Delta t} := \{k\Delta t : k = 0, 1, 2, ..., N\}$ of the time interval $[0, T]$, $N\Delta t = T$. The EM approximation $X_k \approx x(k\,\Delta t)$ for equation (2.1) has the form

$$X_{k+1} = X_k + f(X_k)\,\Delta t + g(X_k)\,\Delta w_{k+1}, \tag{2.2}$$

where $\Delta w_{k+1} = w((k+1)\Delta t) - w(k\Delta t)$ and $X_0 = x_0$. A vector form of equation (2.2) has the ith component given by the formula

$$X_{i,k+1} = X_{i,k} + f_i(X_k)\,\Delta t + \sum_{j=1}^{m} g_{ij}(X_k)\,\Delta w_{j,k+1}.$$

A straightforward, continuous-time extension $X(s) \approx x(s)$ may then be defined as the step process

$$X(s) = X_k, \quad \text{for } s \in [k\Delta t, (k+1)\Delta t). \tag{2.3}$$

2.3 Explicit Milstein

If we add a correction to the stochastic increment in the EM method we obtain the Milstein method. The correction arises because the traditional Taylor expansion must be modified in the case of Itô calculus. A so-called Itô-Taylor expansion can be formed by applying Itô's result, which is a fundamental tool of stochastic calculus. Truncating the Itô-Taylor expansion at an appropriate point produces the Milstein discretisation $\widehat{X}_k \approx x(k\Delta t)$ of equation (2.1) with the ith component

of the form

$$\begin{aligned}\widehat{X}_{i,k+1} = \widehat{X}_{i,k} + f_i(\widehat{X}_k)\Delta t + \sum_{j=1}^{m} g_{ij}(\widehat{X}_k)\Delta w_{j,k} \\ + \sum_{j,n=1}^{m} h_{ijn}(\widehat{X}_k)\left(\Delta w_{j,k}\Delta w_{n,k} - \Upsilon_{jn}\Delta t - A_{jn,k}\right),\end{aligned} \tag{2.4}$$

where

- $h_{ijn}(x)$ is the diffusion tensor defined as

$$h_{ijn}(x) = \frac{1}{2}\sum_{l=1}^{d} g_{ln}(x)\frac{\partial g_{ij}}{\partial x_l}(x),$$

- Υ is the correlation matrix for the driving Brownian paths,

- $A_{jn,k}$ are the Lévy areas defined as

$$A_{jn,k} = \int_{k\Delta t}^{(k+1)\Delta t} \left(\left(w_j(t) - w_j(k\Delta t)\right) dw_n(t) - \left(w_n(t) - w_n(k\Delta t)\right) dw_j(t)\right).$$

We define a straightforward, continuous-time extension of the Milstein scheme in a similar fashion to Euler–Maruyama in equation (2.3).

2.4 Weak Convergence

There are two main concepts in terms of measuring the rate at which the numerical method approaches the solution as we decrease the stepsize $\Delta t \to 0$: *weak convergence* and *strong convergence*. The first concept, *weak convergence*, is less demanding as it measures the rate of decay of the "error of the means." The other one, *strong convergence*, measures the rate at which the "mean of the error" decays as $\Delta t \to 0$. Whenever we mention the expected value we implicitly imply that such a quantity exists.

Given a smooth function $P : \mathbb{R}^d \to \mathbb{R}$, for P from the set of polynomials of degree smaller or equal to R, for some R, we say that X converges to x in a *weak* sense with an order α if

$$|\mathbb{E}[P(x(T))] - \mathbb{E}[P(X_T)]| = \mathcal{O}(\Delta t^{\alpha}). \tag{2.5}$$

Thanks to Assumption 2.1.1 it can be shown that both EM and Milstein have weak order of convergence $\alpha = 1$.

2.5 Strong Convergence

Assumption 2.1.1 allows us to provide the classical strong convergence on the finite time interval $[0, T]$, which is defined as

$$\left(\mathbb{E}\Big[\sup_{0\le k\Delta t\le T} |x(k\Delta t) - X_k|^p\Big]\right)^{1/p} = \mathcal{O}(\Delta t^{\xi}), \quad 2 \le p < \infty. \tag{2.6}$$

Maruyama [77] showed the mean-square convergence of the EM method, while Gihman and Skorohod [35] proved that the strong order of accuracy is $\xi = 1/2$. For the Milstein approximation the rate is improved to $\xi = 1$, that is, we have

$$\left(\mathbb{E}\Big[\sup_{0\le k\Delta t\le T} \left|x(k\Delta t) - \widehat{X}_k\right|^p\Big]\right)^{1/p} = \mathcal{O}(\Delta t), \quad 2 \le p < \infty. \tag{2.7}$$

The continuously extended scheme has a slight degradation in order,

$$\mathbb{E}\Big[\sup_{0\le t\le T} |x(t) - X(t)|^p\Big] = \mathcal{O}(|\, \Delta t \log(\Delta t)\,|^{p/2}), \quad p \ge 2. \tag{2.8}$$

Note that the convergence rate is the same here for the both Euler–Maruyama and Milstein schemes. The inequality in (2.8) was proved in [81] for the case where piecewise linear interpolation is used. A result for the step process then follows via the triangle inequality. Later in Theorems 6.3.2 and 6.3.8 we will show that it is possible to improve the strong convergence rate for a continuously extended Milstein scheme thanks to a Brownian bridge interpolation technique.

2.6 Mean Exit Times

Having specified the open set $O \subset \mathbb{R}^d$, the *stopped exit time* is the first time at which $x(s)$ leaves the open set O, or T if this is smaller. Our quantity of interest in this thesis is the expected value of this random variable. We specifically distinguish between *hitting times* and *exit times*. The former is the case when

the process under consideration starts outside the set and hits the boundary from outside. The latter is the case when the process starts inside the set and leaves it. In this thesis we focus on the latter.

Such exit times are important in many applications, including air traffic management (using the Markov Chain approach) [65], manufacturing flexibility [96], quantum electrodynamics [10], the adoption of technological innovation [55], electronic systems [61, 79], optimal decision making [72, 86], finance, insurance and economics [23, 68, 95, 100]

We then introduce notation for the stopped exit time for the SDE by

$$\tau := (\inf\{s > t \ : \ x(s) \notin O\}) \wedge T.$$

Here $a \wedge b$ denotes $\min(a, b)$.

Similarly, for the continuously extended Euler–Maruyama approximation, we let

$$\nu := (\inf\{s > t \ : \ X(s) \notin O\}) \wedge T.$$

We note that this exit time arises when the natural approximation algorithm is used: record the first discrete time point at which the Euler–Maruyama path exits the set O, or T if this is smaller.

The continuous-time extension (2.3) takes the form of a step process, so ν corresponds to the first grid point where this numerical solution exits the region of interest, or T if this is smaller.

We emphasise that the stopped exit time problem restricts the solution to a compact domain. It follows that we may redefine f and g outside this domain, if necessary, in order to ensure that they are globally Lipschitz. This allows us to derive results that apply to a wide range of nonlinear SDE models, including those arising in finance and chemical kinetics that pose analytical difficulties through non-differentiability of the drift at the origin [2, 93] or superlinear growth of the diffusion at infinity [21, 32, 94].

The work of Gobet and coauthors, [53, Theorem 17] (see also [51] and [52]), gives an optimal rate for the weak convergence of the Euler–Maruyama stopped exit time:

$$\mathbb{E}[\tau] - \mathbb{E}[\nu] = O(\Delta t^{\frac{1}{2}}). \tag{2.9}$$

A proper analysis of the multi-level approach requires an understanding of both the weak and strong convergence rates of the underlying discretisation method. For this reason we present what appears to be the first strong convergence result for the stopped exit time problem (Theorem 4.3.1).

2.7 Summary

In this chapter we introduced the most popular explicit discretisation schemes, explained the difference between weak and strong convergence and presented the mean exit time problem. We stressed the need for the strong convergence result of the numerical scheme for exit times to justify the use of the multi-level Monte Carlo method, which is introduced in the next chapter.

Chapter 3

Multi-level Monte Carlo

> Prediction is very difficult,
> especially about the future.
>
> Niels Bohr, 1885-1962

3.1 Monte Carlo

We start by introducing a standard Monte Carlo approach. Our objective here is to numerically approximate the expected value $\mathbb{E}[Y]$, where $Y = F(x)$ is some functional of a random variable x. In many financial applications we are not able to sample x directly and therefore, in order to perform Monte Carlo simulations, we approximate x with X such that $\mathbb{E}[F(X)] \to \mathbb{E}[F(x)]$, when Δt goes to zero. We have in mind the case where x may represent the solution of a nonlinear SDE at a specified time, and X is a numerical approximation using a stepsize Δt. The standard Monte Carlo estimate is produced when we use X to compute N independent samples,

$$\widehat{Y} = \frac{1}{N} \sum_{i=1}^{N} F(X^{[i]}),$$

where $X^{[i]}$ is the numerical approximation to x on the ith sample path and N is the number of independent simulations of x. For $\Delta t \to 0$ and $N \to \infty$ we achieve a standard Monte Carlo result, $\widehat{Y} \to \mathbb{E}[Y]$ [46]. In practice we choose a certain stepsize $\Delta t > 0$ and a finite number of simulations N and perform Monte Carlo simulations, producing an error to the approximation of $\mathbb{E}[Y]$. Here we are in-

terested in the mean square error (MSE), that is, $\mathbb{E}\left[(\widehat{Y} - \mathbb{E}[Y])^2\right]$. Our objective in Monte Carlo simulations is to estimate Y with a mean square accuracy TOL^2 (so that the desired root-mean-square accuracy is TOL) by minimising the computational complexity required to achieve the target accuracy. The overall error naturally splits into two terms,

$$\begin{aligned}\mathbb{E}\left[(\widehat{Y} - \mathbb{E}[Y])^2\right] &= \mathbb{E}\left[(\widehat{Y} - \mathbb{E}[\widehat{Y}] + \mathbb{E}[\widehat{Y}] - \mathbb{E}[Y])^2\right] \\ &= \mathbb{E}\left[(\widehat{Y} - \mathbb{E}[\widehat{Y}])^2\right] + \left(\mathbb{E}[\widehat{Y}] - \mathbb{E}[Y]\right)^2.\end{aligned} \tag{3.1}$$

The first term on the right-hand side in (3.1) is the Monte Carlo variance; the other one is known as the bias of the numerical approximation. The Monte Carlo variance is proportional to N^{-1},

$$\mathrm{Var}[\widehat{Y}] = \frac{1}{N^2}\mathrm{Var}\left[\sum_{i=1}^{N} F(X^{[i]})\right] = \frac{1}{N}\mathrm{Var}[F(X)].$$

From (2.5) we know that the weak error, $|\mathbb{E}[\widehat{Y}] - \mathbb{E}[Y]| = \mathcal{O}(\Delta t)$, for both Euler–Maruyama and Milstein. Hence, the mean square error for the standard Monte Carlo method is of order $\mathcal{O}(N^{-1}) + \mathcal{O}(\Delta t^2)$. To ensure the root-mean-square error is proportional to TOL, we must have $MSE = \mathcal{O}(\mathrm{TOL}^2)$ and therefore $1/N = \Delta t^2 = \mathcal{O}(\mathrm{TOL}^2)$, which means $N = \mathcal{O}(\mathrm{TOL}^{-2})$ and $\Delta t = \mathcal{O}(\mathrm{TOL})$. The computational cost of standard Monte Carlo is proportional to the number of paths N multiplied by the cost of generating a path, that is, the number of timesteps in each sample path. Thus, the standard Monte Carlo complexity is $\mathcal{O}(\mathrm{TOL}^{-3})$ [29].

3.2 Multi-level Monte Carlo

We first mention that a multi-level Monte Carlo method for parametric integration was developed by Heinrich [56]. A similar two-level strategy was developed slightly earlier by Kebaier [66], and a multi-level approach was under development at the same time by Speight [91, 92]. In the parametric integration setting we are interested in estimating the value of $\mathbb{E}[F(x, \lambda)]$, where x is a finite-dimensional random variable and λ is a parameter. In the simplest case in which λ is a real variable in the range $[0, 1]$, having estimated the value of $\mathbb{E}[F(x, 0)]$ and

$\mathbb{E}[F(x,1)]$, we can use $\frac{1}{2}(F(x,0)+F(x,1))$ as a control variate when estimating the value of $\mathbb{E}[F(x,\frac{1}{2})]$, since the variance of $F(x,\frac{1}{2})-\frac{1}{2}(F(x,0)+F(x,1))$ will usually be less than the variance of $F(x,\frac{1}{2})$. This approach can then be applied recursively for the other intermediate values of λ, yielding large savings if $F(x,\lambda)$ is sufficiently smooth with respect to λ.

Giles' multi-level Monte Carlo path simulation [38] is similar in some ways and different in others. There is no parametric integration, and the random variable is infinite-dimensional, corresponding to a Brownian path in the original paper. However, the control variate viewpoint is very similar. A coarse path simulation is used as a control variate for a more refined fine path simulation, but since the exact expectation for the coarse path is not known, this is in turn estimated recursively using even coarser path simulation as control variates. In many applications the coarsest path in the multi-level method may have only one timestep for the entire interval of interest.

3.2.1 Multi-level Monte Carlo Theorem

A multi-level Monte Carlo simulation uses a number of levels of resolution, $\ell = 0,1,\ldots,L$, with $\ell = 0$ being the coarsest and $\ell = L$ being the finest. In the context of an SDE simulation, level 0 may have just one timestep for the whole time interval $[0,T]$, whereas level L might have 2^L uniform timesteps $\Delta t_L = 2^{-L}T$. The smallest stepsize, Δt_L, is chosen so that the bias in the discretisation method matches the target accuracy of $\mathcal{O}(\mathrm{TOL})$; matching Δt_L with TOL then gives $L = \frac{\log \mathrm{TOL}^{-1}}{\log M}$. The multi-level method was developed for any integer $M > 1$. Level L would then have M^L uniform timesteps $\Delta t_L = M^{-L}T$. We will specify whether the method we are using is for $M = 2$ or for general M.

Let us denote by P the payoff and by P_ℓ its approximation on level ℓ. P is assumed to be a globally Lipschitz function of $x(T)$, for example, $\max(x(T)-E,0)$ for a call option, where by E we denote the strike. We can then write the following trivial identity,

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^{L} \mathbb{E}[P_\ell - P_{\ell-1}]. \tag{3.2}$$

On the left is the exact mean of the high-resolution approximation, which has the required bias. On the right is a telescoping series involving the different levels

of resolution. The idea behind MLMC is to independently estimate each of the expectations on the right-hand side in (3.2) in a way which minimises the overall variance for a given computational cost. Let the first term, $\mathbb{E}[P_0]$, be estimated by a quantity Y_0 that uses the sample mean of N_0 independent paths. Thus,

$$Y_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{[i]}.$$

Each remaining term of the form $\mathbb{E}\left[P_\ell - P_{\ell-1}\right]$ is estimated by a quantity Y_ℓ constructed with N_ℓ independent pairs of paths. Hence,

$$Y_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_\ell^{[i]} - P_{\ell-1}^{[i]} \right). \tag{3.3}$$

The key point here is that $P_\ell^{[i]}$–$P_{\ell-1}^{[i]}$ should come from two discrete approximations for the same Brownian path (see [83]), in order to have a small difference on finer levels (thanks to strong convergence) and as a consequence making the variance small, too. Therefore, very few samples will be required on finer levels to accurately estimate the expected value. From (3.3) we can see that

$$\mathbb{E}[Y_\ell] = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \mathbb{E}[P_\ell^{[i]} - P_{\ell-1}^{[i]}] = \mathbb{E}[P_\ell - P_{\ell-1}].$$

The overall MLMC estimator $\widehat{Y}$ is of the form

$$\widehat{Y} = \sum_{\ell=0}^{L} Y_\ell.$$

The final accuracy depends on the accuracy of the finest level L, even though we are using different levels with different discretisation errors to estimate $\mathbb{E}[P]$,

$$\mathbb{E}[\widehat{Y}] = \sum_{\ell=0}^{L} \mathbb{E}[Y_\ell] = \mathbb{E}[P_0] + \sum_{\ell=1}^{L} \mathbb{E}[P_\ell - P_{\ell-1}] = \mathbb{E}[P_L].$$

The variance is given by

$$\mathrm{Var}[\widehat{Y}] = \sum_{\ell=0}^{L} \mathrm{Var}[Y_\ell] = \sum_{\ell=0}^{L} \frac{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right]}{N_\ell}.$$

Let us denote by C the total computational cost of the multi-level algorithm. Each pair of paths at level ℓ has a cost proportional to $1/\Delta t_\ell$. Hence, the total computational cost is

$$C = \sum_{\ell=0}^{L} \frac{N_\ell}{\Delta t_\ell}.$$

To make the variance of $\widehat{Y}$ less than $\frac{\mathrm{TOL}^2}{2}$ for the computational cost C, we can think of N_ℓ as a continuous variable and use the Lagrange function to find the minimum of

$$\sum_{\ell=0}^{L} \frac{1}{N_\ell} \mathrm{Var}\left[P_\ell - P_{\ell-1}\right] + \lambda \left(\sum_{\ell=0}^{L} \frac{N_\ell}{\Delta t_\ell} - C \right).$$

Thanks to the first order condition we have

$$N_\ell = \lambda^{-\frac{1}{2}} \sqrt{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \Delta t_\ell}.$$

We can then write

$$\begin{aligned} \mathrm{Var}[\widehat{Y}] &= \sum_{\ell=0}^{L} \frac{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right]}{N_\ell} \\ &= \sum_{\ell=0}^{L} \frac{\sqrt{\lambda}}{\sqrt{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \Delta t_\ell}} \mathrm{Var}\left[P_\ell - P_{\ell-1}\right]. \end{aligned} \tag{3.4}$$

Our goal is to achieve $\mathrm{Var}[\widehat{Y}] \leq \frac{\mathrm{TOL}^2}{2}$, so in (3.4) we obtain

$$\lambda^{-\frac{1}{2}} \geq 2\mathrm{TOL}^{-2} \sum_{\ell=0}^{L} \sqrt{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \Delta t_\ell^{-1}}.$$

Hence, the optimal number of samples N_ℓ for level ℓ can be expressed as

$$N_\ell = \left\lceil 2\mathrm{TOL}^{-2} \sqrt{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \Delta t_\ell} \sum_{\ell=0}^{L} \sqrt{\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \Delta t_\ell^{-1}} \right\rceil. \tag{3.5}$$

The smallest stepsize, $\Delta t_L = 2^{-L}T$, is chosen so that the bias in the discretisation method matches the desired accuracy of $\mathcal{O}(\mathrm{TOL})$. Thanks to the $\mathcal{O}(\Delta t)$ weak convergence, we can match Δt_L with TOL and we get

$$L = \frac{\log \mathrm{TOL}^{-1}}{\log 2}.$$

We now present Algorithm 1. We make an initial estimate of variance dependent on tolerance. This way the greater the required precision, the greater the initial sample for estimating variance is.

Algorithm 1 Fix TOL which determines the value of maximum level L.

for $\ell = 0, ..., L$ **do**
  Calculate an initial estimate of variance $\mathrm{Var}[P_\ell - P_{\ell-1}]$ using $N = \mathrm{TOL}^{-1}$ samples.
  Determine optimal number of samples N_ℓ using (3.5).
  Generate additional samples as needed until we have N_ℓ of them.
end for

If we denote by α the order of weak convergence and by β the order of strong convergence, we have then just derived a special case of Theorem 3.1 from [38] for $\alpha = 1$ and $\beta = 1/2$. We now recall a slightly modified version of this theorem from [43] for the reader's convenience.

Theorem 3.2.1. *Let P denote a functional of the solution of a stochastic differential equation (2.1) and let us assume P is globally Lipschitz. Let P_ℓ denote the corresponding level ℓ of a numerical approximation. If there exist independent estimators Y_ℓ based on N_ℓ Monte Carlo samples, and positive constants $\alpha, \beta, \gamma, c_1, c_2, c_3$ such that $\alpha \geq \frac{1}{2}\min(\beta, \gamma)$ and*

i) $|\mathbb{E}[P_\ell - P]| \leq c_1 2^{-\alpha \ell}$,

ii) $\mathbb{E}[Y_\ell] = \begin{cases} \mathbb{E}[P_0], & \ell = 0, \\ \mathbb{E}[P_\ell - P_{\ell-1}], & \ell > 0, \end{cases}$

iii) $\mathrm{Var}[Y_\ell] \leq c_2 N_\ell^{-1} 2^{-\beta \ell}$,

iv) $C_\ell \leq c_3 N_\ell 2^{\gamma \ell}$, *where C_ℓ is the computational complexity of Y_ℓ,*

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

$$Y = \sum_{\ell=0}^{L} Y_\ell,$$

has a mean-square-error with bound

$$\mathbb{E}\left[(Y - \mathbb{E}[P])^2\right] < \mathrm{TOL}^2,$$

with a computational complexity C with bounds

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

We note that we need the order of weak convergence α in condition i) of Theorem 3.2.1. Condition iii) includes the crucial for MLMC rate of strong convergence. We now show that in order to estimate the MLMC variance it is useful to examine the strong convergence property of the numerical scheme. We begin with

$$\mathrm{Var}\left[P_\ell - P_{\ell-1}\right] \leq \mathbb{E}\left[(P_\ell - P_{\ell-1})^2\right] \leq 2\,\mathbb{E}\left[(P_\ell - P)^2\right] + 2\,\mathbb{E}\left[(P - P_{\ell-1})^2\right].$$

For a Lipschitz continuous function P, $|P(x) - P(y)|^2 \leq K\,|x-y|^2$, we can write

$$\mathbb{E}\left[(P_\ell - P)^2\right] \leq K\,\mathbb{E}\left[|x(T) - X_T|^2\right],$$

for some constant K. Remarkably, the multi-level algorithm depends on the strong convergence property even though we compute a weak quantity, that is, an expected value.

3.2.2 Modified Multi-level Monte Carlo

We showed that the crucial step in the MLMC analysis is the estimation of $\mathrm{Var}\left[\mathrm{P}_\ell - \mathrm{P}_{\ell-1}\right]$. This differentiates MLMC from standard Monte Carlo, where we only require a weak error bound for approximations of SDEs. We will, however, demonstrate that it may not be necessary to have the classical strong convergence in order to obtain a good multi-level variance.

In (3.3) we have used the same estimator for the payoff P_ℓ on every level ℓ, and therefore (3.2) becomes a trivial identity due to the telescoping summation on the right-hand side. However, Giles in [37] showed that it can be more desirable to use different estimators for the finer and coarser of the two levels being considered, P_ℓ^f when level ℓ is the finer level, and P_ℓ^c when level ℓ is the coarser level. In this

case, in order for the key equality to hold,

$$E[P_L^f] = \mathbb{E}[P_0^f] + \sum_{\ell=1}^{L} \mathbb{E}[P_\ell^f - P_{\ell-1}^c],$$

we need that for $\ell = 1, \ldots, L$

$$\mathbb{E}[P_\ell^f] = \mathbb{E}[P_\ell^c]. \tag{3.6}$$

Theorem 3.2.1 is still applicable to this modified estimator. The advantage is that it gives us the flexibility to construct approximations for which $P_\ell^f - P_{\ell-1}^c$ is much smaller than the original $P_\ell - P_{\ell-1}$, giving a larger value for β, the rate of variance convergence in condition *iii)* of Theorem 3.2.1. In the next sections we demonstrate how suitable choices of P_ℓ^f and P_ℓ^c can dramatically improve the convergence of the variance of the MLMC estimator.

A good choice of estimators often follows from analysis of the problem under consideration from the distributional point of view. We will demonstrate that methods that had been used previously to improve the weak order of convergence can also improve the order of convergence of the MLMC variance.

3.2.3 Conditional Monte Carlo

Conditional Monte Carlo has been frequently used to improve convergence of the MLMC variance [24, 37]. We recall that our goal is to calculate $\mathbb{E}[P]$. Using conditional expectation, we can write

$$\mathbb{E}[P] = \mathbb{E}\big[\mathbb{E}[P|Z]\big],$$

where Z is a random vector we condition on. Hence, $\mathbb{E}[P|Z]$ is an unbiased estimator of $\mathbb{E}[P]$. Using a general formula for variance decomposition, we obtain

$$\mathrm{Var}\,[P] = \mathbb{E}\big[\mathrm{Var}\,[P|Z]\big] + \mathrm{Var}\,\big[\mathbb{E}[P|Z]\big],$$

and therefore $\mathrm{Var}\,\big[\mathbb{E}[P|Z]\big] \leq \mathrm{Var}\,[P]$. In the multi-level setting we obtain a better variance convergence if we condition on different vectors on fine and coarse levels. This will be explained more carefully in section 6.3 of Chapter 6.

3.3 Financial Applications

In this section we present a number of multi-level Monte Carlo applications in finance. The main application of the multi-level algorithm is to compute the expected payoff of a financial option. We overview results for Euler–Maruyama and Milstein approximations when used for pricing options.

3.3.1 Pricing Options with Euler–Maruyama

Giles et al. [40] analysed all the main option payoffs using the EM scheme. Table 3.1 (taken from [37]) summarises observations from numerical experiments and corresponding bounds derived analytically.

Option	Numerical	Analysis
European	$\mathcal{O}(\Delta t)$	$\mathcal{O}(\Delta t)$
Asian	$\mathcal{O}(\Delta t)$	$\mathcal{O}(\Delta t)$
Lookback	$\mathcal{O}(\Delta t)$	$\mathcal{O}(\Delta t)$
Barrier	$\mathcal{O}(\Delta t^{1/2})$	$\mathcal{o}(\Delta t^{1/2-\varepsilon})$
Digital	$\mathcal{O}(\Delta t^{1/2})$	$\mathcal{O}(\Delta t^{1/2}\log\Delta t)$

Table 3.1: Convergence β of $\mathrm{Var}\left[P_\ell - P_{\ell-1}\right]$: estimates based on numerical experiments and analytical bounds for the *Euler–Maruyama* discretisation; here any $\varepsilon > 0$ can be chosen.

3.3.2 Pricing Options with Milstein

Extending the analysis for EM to the Milstein scheme is not straightforward. This is because the Milstein scheme gives an improved rate of convergence on the grid points compared to the EM scheme, but this is not enough for path-dependent options where the behaviour of the numerical approximation between the grid points is crucial. Recall from (2.8) that when we use a naive continuous extension, the Milstein approximation has the same order of strong convergence as the EM scheme. For this reason we define a Brownian Bridge interpolation for $t \in [k\Delta t, (k+1)\Delta t)$,

$$\widetilde{X}(t) = X_k + \lambda\,(X_{k+1} - X_k) + g(X_k)\,\left(w(t) - w(k\Delta t) - \lambda\,\Delta w_{k+1}\right), \tag{3.7}$$

where $\lambda = \frac{t-k\Delta t}{\Delta t}$. For the Milstein scheme with the Brownian bridge interpolation we have (see, for example, [81])

$$\mathbb{E}\Big[\sup_{0\le t\le T}\big|x(t)-\widetilde{X}(t)\big|^p\Big] = \mathcal{O}(|\,\Delta t\log(\Delta t)\,|^p).$$

Unfortunately, the interpolation $\widetilde{X}(t)$ cannot be implemented because we need to know the whole trajectory $(w(t))_{0\le t\le T}$ in order to construct it. However, combining the Brownian bridge interpolation with a conditional Monte Carlo technique can significantly improve the variance convergence of the multi-level estimator. This is because only distributional knowledge of certain functionals of $(w(t))_{0\le t\le T}$ is needed for suitable multi-level estimators.

A Milstein scheme in a one-dimensional case for complex payoffs was analysed in [24] with numerical tests in [37], a summary of which can be found in Table 3.2. Comparing with Table 3.1, we see that an extra factor of Δt is obtained relative to the EM method.

Option	Numerical	Analysis
European	$\mathcal{O}(\Delta t^2)$	$\mathcal{O}(\Delta t^2)$
Asian	$\mathcal{O}(\Delta t^2)$	$\mathcal{O}(\Delta t^2)$
Lookback	$\mathcal{O}(\Delta t^2)$	$o(\Delta t^{2-\varepsilon})$
Barrier	$\mathcal{O}(\Delta t^{3/2})$	$o(\Delta t^{3/2-\varepsilon})$
Digital	$\mathcal{O}(\Delta t^{3/2})$	$o(\Delta t^{3/2-\varepsilon})$

Table 3.2: Convergence β of $\mathrm{Var}\,[P_\ell - P_{\ell-1}]$: estimates based on numerical experiments and analytical bounds for the *Milstein* discretisation; any $\varepsilon > 0$ can be chosen.

3.3.3 Multi-dimensional Milstein Scheme

We consider the Milstein discretisation defined in (2.4). We note that the rate of strong convergence ξ for the Milstein scheme is double the value we have for the EM scheme and therefore the MLMC variance for Lipschitz payoffs converges twice as fast. This improvement, however, does not come without a price, namely one needs to simulate Lévy areas, which are usually quite expensive in terms of computational effort. Apart from a two-dimensional case, there is not any efficient method to simulate Lévy areas [31, 90, 97].

Clark and Cameron in [19] showed for a particular SDE that it is not possible to achieve a better order of strong convergence than the EM scheme when using just the discrete increments of the underlying Brownian motion. This was confirmed by Müller-Gronbach in [80] who extended this analysis to general SDEs. This means that if we use the standard MLMC method with the Milstein scheme without simulating the Lévy areas, the complexity will be the same as for EM.

Giles and Szpruch demonstrated in [43] that it is possible to obtain a multi-level estimator with variance which decreases at the same rate as for the scalar Milstein estimator, while avoiding computation of Lévy areas. This is achieved by constructing an antithetic estimator. Therefore, the results from Table 3.2 are applicable for the multi-dimensional case, too.

3.3.4 Greeks

Arising in risk management and hedging, sensitivities of the prices to various input parameters, or as they are often called "Greeks", are an important application where Monte Carlo simulations are used. In the multi-level setting three main techniques are used to improve the MLMC variance: payoff smoothing using conditional expectations [45], an approximation of the above technique using path splitting for the final timestep [6] and the use of a hybrid combination of pathwise sensitivity and the Likelihood Ratio Method [39].

We mention work by Burgos et al. [15] whose results for call and digital options are presented in Table 3.3 and Table 3.4, and for lookback and barrier options in Table 3.5 and Table 3.6, respectively. They applied pathwise sensitivity to these smoothed payoffs, with a scalar Milstein scheme used to obtain the penultimate step.

Estimator	β	Complexity
Value	≈ 2.0	$\mathcal{O}(\mathrm{TOL}^{-2})$
Delta	≈ 1.5	$\mathcal{O}(\mathrm{TOL}^{-2})$
Vega	≈ 2.0	$\mathcal{O}(\mathrm{TOL}^{-2})$

Table 3.3: *Call* option. Numerical convergence β of $\mathrm{Var}\left[P_\ell - P_{\ell-1}\right]$ and the MLMC complexity.

Estimator	β	Complexity
Value	≈ 1.4	$\mathcal{O}(\text{TOL}^{-2.0})$
Delta	≈ 0.5	$\mathcal{O}(\text{TOL}^{-2.5})$
Vega	≈ 0.6	$\mathcal{O}(\text{TOL}^{-2.4})$

Table 3.4: *Digital* option. Numerical convergence β of $\text{Var}[P_\ell - P_{\ell-1}]$ and the MLMC complexity.

Estimator	β	Complexity
Value	≈ 1.9	$\mathcal{O}(\text{TOL}^{-2})$
Delta	≈ 1.9	$\mathcal{O}(\text{TOL}^{-2})$
Vega	≈ 1.3	$\mathcal{O}(\text{TOL}^{-2})$

Table 3.5: *Lookback* option. Numerical convergence β of $\text{Var}[P_\ell - P_{\ell-1}]$ and the MLMC complexity.

Estimator	β	Complexity
Value	≈ 1.6	$\mathcal{O}(\text{TOL}^{-2.0})$
Delta	≈ 0.6	$\mathcal{O}(\text{TOL}^{-2.4})$
Vega	≈ 0.6	$\mathcal{O}(\text{TOL}^{-2.4})$

Table 3.6: *Barrier* option. Numerical convergence β of $\text{Var}[P_\ell - P_{\ell-1}]$ and the MLMC complexity.

To avoid difficulties in using conditional expectation to smooth payoffs in practice, authors in [15] use path splitting of every simulated path on the final timestep. We present their results in Table 3.7. Finally, a vibrato Monte Carlo

Estimator	s	β	Complexity
Value	10	≈ 2.0	$\mathcal{O}(\text{TOL}^{-2})$
	500	≈ 2.0	$\mathcal{O}(\text{TOL}^{-2})$
Delta	10	≈ 1.0	$\mathcal{O}(\text{TOL}^{-2}(\log \text{TOL})^2)$
	500	≈ 1.5	$\mathcal{O}(\text{TOL}^{-2})$
Vega	10	≈ 1.6	$\mathcal{O}(\text{TOL}^{-2})$
	500	≈ 2.0	$\mathcal{O}(\text{TOL}^{-2})$

Table 3.7: Splitting. Numerical convergence β of $\text{Var}[P_\ell - P_{\ell-1}]$ and the MLMC complexity for s extra samples on the penultimate step.

technique was applied in [15], numerical results of which are presented in Table 3.8 for the call option with $s = 10$ extra samples on the penultimate step derived using the scalar Milstein scheme.

Estimator	β	Complexity
Value	≈ 2.0	$\mathcal{O}(\text{TOL}^{-2})$
Delta	≈ 1.5	$\mathcal{O}(\text{TOL}^{-2})$
Vega	≈ 2.0	$\mathcal{O}(\text{TOL}^{-2})$

Table 3.8: Vibrato. Numerical convergence β of $\text{Var}\left[P_\ell - P_{\ell-1}\right]$ and the MLMC complexity.

3.3.5 Jump-diffusion Processes

Giles and Xia in [99] successfully adapted the MLMC method to jump-diffusion SDEs (see, for example, [78]), using a jump-adapted approximation from [88]. The implementation of the multi-level method was more difficult for the path-dependent options because the coarse and fine path approximations may have jumps at different times. These differences could lead to a large discrepancy between the coarse and fine path payoffs, and thus greatly increase the multi-level variance. To avoid this, Giles and Xia [99] (see also [98]) modified the simulation approach of Glasserman and Merener [47].

3.3.6 Lévy Processes

Dereich and Heidenreich [26] investigated approximation methods for both finite and infinite activity Lévy driven SDEs with globally Lipschitz payoffs in the context of MLMC. They conclude that the rate of the multi-level variance convergence is closely related to the Blumenthal-Getoor index of the driving Lévy process that measures the frequency of small jumps. If the index is smaller than one, then neglecting jumps which are smaller than a certain value gives good results. For the Blumenthal-Getoor index greater than one, better results are obtained by the use of the Gaussian approximation of small jumps [25].

3.3.7 Stochastic Partial Differential Equations

Parabolic and elliptic stochastic partial differential equations (SPDEs) applications have become a fruitful area for the application of the multi-level method [9, 20, 54]. In particular, the first financial application of SPDE in the multi-level setting [42] results from modelling credit default probabilities. The authors conclude that the overall computational complexity to achieve an $\mathcal{O}(\text{TOL})$ root-mean-square accuracy is $\mathcal{O}(\text{TOL}^{-2})$.

3.3.8 American Options

Belomestny and Schoenmakers in [11] suggested a way to price American options using the MLMC algorithm. Their method is based on the Anderson and Broadie nested Monte Carlo method [4] in which a key component at each timestep is to estimate conditional expectations using a number of sub-paths. They conclude that it is possible to construct a multi-level version of the Andersen and Broadie nested Monte Carlo algorithm that is of the same complexity as the non-nested one.

3.3.9 Multi-level Quasi-Monte Carlo

We note that the dominant computational cost is on the coarsest levels of an approximation if $\beta > \gamma$ in Theorem 3.2.1. This occurs when the rate at which the multi-level variance decays with the increasing grid level is greater than the rate at which the computational cost increases. Since coarse levels of the approximation correspond to a low-dimensional numerical quadrature, it is natural to consider the use of a quasi-Monte Carlo technique. This has been analysed by Giles and Waterhouse in [44] for scalar SDEs with a Lipschitz payoff. They show that for the case of a European call option the computational complexity appears to be reduced from $\mathcal{O}(\text{TOL}^{-2})$ to approximately $\mathcal{O}(\text{TOL}^{-3/2})$, using the Milstein scheme.

3.4 Negative Results

The results discussed above are based on the assumption that the numerical simulation method converges strongly, or, less directly, on the assumption that the coefficients in the underlying SDE satisfy global Lipschitz conditions. It is known that numerical methods, especially explicit methods of the type discussed here, may fail to converge for certain classes of nonlinear SDEs that violate the global Lipschitz requirement. Using these ideas, the authors in [62] showed that the EM-based MLMC approach can break down for some nonlinear SDEs. As discussed earlier, in the mean exit time context that forms the focus of this thesis, the problem is stated on a compact domain and the restriction to global Lipschitz coefficients is therefore natural.

3.5 Summary

In this chapter we presented multi-level Monte Carlo and showed its superiority over standard Monte Carlo in terms of the computational effort. We then introduced conditional Monte Carlo and demonstrated on a number of financial applications how suitable choices of estimators on fine and coarse levels can dramatically improve the variance convergence of the MLMC estimator. In the next chapter we apply the MLMC algorithm for the mean exit time problem.

Chapter 4

Mean Exit Times and the Multi-level Monte Carlo Method

> If people do not believe
> that mathematics is simple,
> it is only because
> they do not realize
> how complicated life is.
>
> John von Neumann, 1903-1957

Parts of this chapter have been published in Higham et al. [59].

4.1 Introduction

We again consider the system of SDEs (2.1) with the same conditions as previously.

Our aim in this chapter is to show that the multi-level idea that Giles [38] introduced for the problem of approximating mean values of the form $\mathbb{E}[f(x(T)]$ may be adapted to the mean exit time context, reducing the computational cost from $\mathcal{O}\left(\mathrm{TOL}^{-4}\right)$ to $\mathcal{O}\left(\mathrm{TOL}^{-3}|\log(\mathrm{TOL})|^{1/2}\right)$. A proper analysis of the multi-level approach requires an understanding of both the weak and strong convergence rates of the underlying discretisation method, and for this reason we present what appears to be the first strong convergence result for the mean exit time problem (Theorem 4.3.1).

The chapter is organised as follows. Section 4.2 outlines the existing approximation results that are relevant to our analysis. In section 4.3, we state and prove a new strong convergence result. Section 4.4 derives the expected computational cost of standard Monte Carlo, defines a multi-level algorithm and shows that it offers improved complexity. In section 4.5 we provide illustrative computational results.

4.2 Current Results

For the purposes of this chapter, we make our notation slightly more informative. We let $x^{t,z}(s)$ denote the general solution of the SDE (2.1) at time s with initial condition $x(t) = z$. So the specific solution of interest, $x(s)$, is shorthand for $x^{0,x_0}(s)$.

The Euler–Maruyama approximation $X_k^{t,z} \approx x^{t,z}(s_k)$ has $X_0 = z$ and

$$X_{k+1}^{t,z} = X_k^{t,z} + f(X_k^{t,z})\Delta t + g(X_k^{t,z})\left(w(s_{k+1}) - w(s_k)\right), \tag{4.1}$$

where $s_k = t + k\Delta t$, and Δt is the stepsize [67]. A straightforward, continuous-time extension $X^{t,z}(s) \approx x^{t,z}(s)$ may then be defined as

$$X^{t,z}(s) = X_k^{t,z}, \quad \text{for } s \in [s_k, s_{k+1}). \tag{4.2}$$

For consistency, we use $X(s)$ as shorthand for our approximation to the given problem; that is $X(s) = X^{0,x_0}(s)$, and we use X_k to denote X_k^{0,x_0}.

Next, we introduce notation for the stopped exit time. Having specified the open set O with boundary ∂O and a fixed future time T, we denote the stopped exit time for the SDE by

$$\tau^{t,z} := \left(\inf\{s > t \ : \ x^{t,z}(s) \notin O\}\right) \wedge T,$$

with τ as shorthand for τ^{0,x_0}.

Similarly, for the continuously extended Euler–Maruyama approximation, we let

$$\nu^{t,z} := \left(\inf\{s > t \ : \ X^{t,z}(s) \notin O\}\right) \wedge T,$$

with ν as shorthand for ν^{0,x_0}.

Our analysis makes use of existing results concerning the finite time strong convergence of the Euler–Maruyama method (2.8) when used to simulate paths of the SDE and also the weak error arising when this method is used to approximate the stopped exit time (2.9).

We now make the following two assumptions throughout this chapter.

Assumption 4.2.1. *(**Strong ellipticity**) for some $c > 0$, $\sum_{ij}(g(x)g^*(x))_{ij}\xi_i\xi_j > c|\xi|^2$, for all $x \in O$, $\xi \in \mathbb{R}^d$.*

Assumption 4.2.2. *(**Regularity of the boundary**) for $d > 1$, $O \subset \mathbb{R}^d$ is a bounded open set with its boundary ∂O being C^2 smooth.*

In many applications we would like to consider SDEs with noise as in (4.15) or (4.19). In these special cases strong ellipticity implies that we restrict our analysis to the domain $O \subset {\mathbb{R}_+}^d$. Particularly, in the one-dimensional case for $O := (\alpha, \beta)$ we set the lower barrier to a positive real number, $\alpha > 0$.

We then give a result that follows directly from the smoothness of the mean exit time as a function of the initial condition.

Lemma 4.2.3. *Under Assumptions 4.2.1 and 4.2.2 there exists a constant K such that for any $z \in O$ we have*

$$\mathbb{E}[\tau^{0,z}] \leq K\left(d_z \wedge T\right),$$

where $d_z = \inf\{|z - y| : y \in \partial O\}$.

Proof

Due to the strong ellipticity in Assumption 4.2.1, the function

$$u(z) := \mathbb{E}[\inf\{s > 0 : x^{0,z}(s) \notin O\}]$$

is known to be $u \in C^2(O) \cap C(\bar{O})$, satisfying the Dirichlet problem

$$Lu + 1 = 0 \text{ on } O, \quad u = 0 \text{ on } \partial O, \tag{4.3}$$

where L is a strongly elliptic operator given by

$$Lu = \frac{1}{2}\mathrm{tr}\{gg^* D^2 u\} + f \cdot Du,$$

with $\mathrm{tr}(A)$ denoting the trace of A [27]. Since the boundary is C^2 smooth, $u \in C^2(\bar{O})$ thanks to [36, Theorem 6.14]. It follows that u is Lipschitz on $\overline{O}$, and we let K denote an appropriate Lipschitz constant. For any given $z \in O$ and $y \in \partial O$ we then have $u(z) = |\, u(z) - u(y)\,| \leq K|z - y|$, and hence

$$u(z) \leq K d_z.$$

By construction, $\mathbb{E}[\tau^{0,z}] \leq u(z)$ and $\mathbb{E}[\tau^{0,z}] \leq T$ and the result follows. □

4.3 Main Theorem

In order to justify a multi-level Monte Carlo approach we first establish a rate of strong convergence for the exit time approximation.

Theorem 4.3.1. *Under Assumptions 2.1.1, 4.2.1 and 4.2.2 we have*

$$\mathbb{E}\left[|\, \tau - \nu\,|^p\right] = \mathcal{O}(|\Delta t \log(\Delta t)|^{\frac{1}{2}}), \ \ \forall p \geq 1.$$

Proof

The proof deals separately with two cases. We first consider the event that the Euler–Maruyama approximation exits before the exact process.

Case I $\nu < \tau$
We have

$$\begin{aligned} \mathbb{E}\left[(\tau - \nu)\,\mathbf{1}_{\{\nu<\tau\}}\right] &= \mathbb{E}\left[\mathbb{E}\left[(\tau - \nu)\,\mathbf{1}_{\{\nu<\tau\}}\,|\,\mathcal{F}_\nu\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\tau^{\nu,x(\nu)}\,|\,\mathcal{F}_\nu\right]\right]. \end{aligned} \tag{4.4}$$

On the right hand side, we have the expected stopped exit time for the exact process, starting from the stopped exit time of the numerical approximation. In order to bound this quantity, we use two properties

- Lemma 4.2.3 tells us that the stopped exit time $\tau^{\nu,x(\nu)}$ will be small in mean if the process starts close to a boundary,
- strong convergence (2.8) tells us that the exact solution is close to the

numerical approximation, which has already reached the boundary or been stopped.

More precisely, applying Lemma 4.2.3 and (2.8) to (4.4) we find

$$\mathbb{E}\left[(\tau-\nu)\,\mathbf{1}_{\{\nu<\tau\}}\right] \le K\mathbb{E}\left[\sup_{0\le s\le T}|x(s)-X(s)|\right] = \mathcal{O}(|\Delta t\log(\Delta t)|^{\frac{1}{2}}). \tag{4.5}$$

Next we consider the event that the exact process exits before the Euler–Maruyama approximation.

Case II $\tau < \nu$
We have

$$\begin{aligned}\mathbb{E}\left[(\nu-\tau)\,\mathbf{1}_{\{\tau<\nu\}}\right] &= \mathbb{E}\left[\mathbb{E}\left[(\nu-\tau)\,\mathbf{1}_{\{\tau<\nu\}}\,|\,\mathcal{F}_\tau\right]\right]\\ &= \mathbb{E}\left[\mathbb{E}\left[\nu^{\tau,X(\tau)}\,|\,\mathcal{F}_\tau\right]\right].\end{aligned}$$

Because Lemma 4.2.3 applies to the exact SDE process, rather than the Euler–Maruyama approximation, the next step is to add and subtract the exact process that runs forward in time from $t=\tau$ with initial condition $X(\tau)$ and apply the triangle inequality, to give

$$\begin{aligned}\mathbb{E}\left[(\nu-\tau)\,\mathbf{1}_{\{\tau<\nu\}}\right] &= \mathbb{E}\left[\mathbb{E}\left[\nu^{\tau,X(\tau)}-\tau^{\tau,X(\tau)}+\tau^{\tau,X(\tau)}\,|\,\mathcal{F}_\tau\right]\right]\\ &\le \left|\mathbb{E}\left[\mathbb{E}\left[\nu^{\tau,X(\tau)}\,|\,\mathcal{F}_\tau\right]\right]-\mathbb{E}\left[\mathbb{E}\left[\tau^{\tau,X(\tau)}\,|\,\mathcal{F}_\tau\right]\right]\right|\\ &\quad+\mathbb{E}\left[\mathbb{E}\left[\tau^{\tau,X(\tau)}\,|\,\mathcal{F}_\tau\right]\right].\end{aligned}$$

The first term on the right hand side concerns the weak error in the mean exit time algorithm, which is known from (2.9) to be $\mathcal{O}(\Delta t^{\frac{1}{2}})$. The second term on the right hand side can be bounded using the same arguments that lead from (4.4) to (4.5), showing that it is $\mathcal{O}(|\Delta t\log(\Delta t)|^{\frac{1}{2}})$. Hence, we find that

$$\mathbb{E}\left[(\nu-\tau)\,\mathbf{1}_{\{\tau<\nu\}}\right] = \mathcal{O}(|\Delta t\log(\Delta t)|^{\frac{1}{2}}). \tag{4.6}$$

Combining (4.5) and (4.6) provides the result for $p=1$. For a more general $p\ge 1$, the required result follows from

$$\mathbb{E}\left[|\,\tau-\nu\,|^p\right] \le T^{p-1}\mathbb{E}\left[\,|\,\tau-\nu\,|\,\right].$$

□

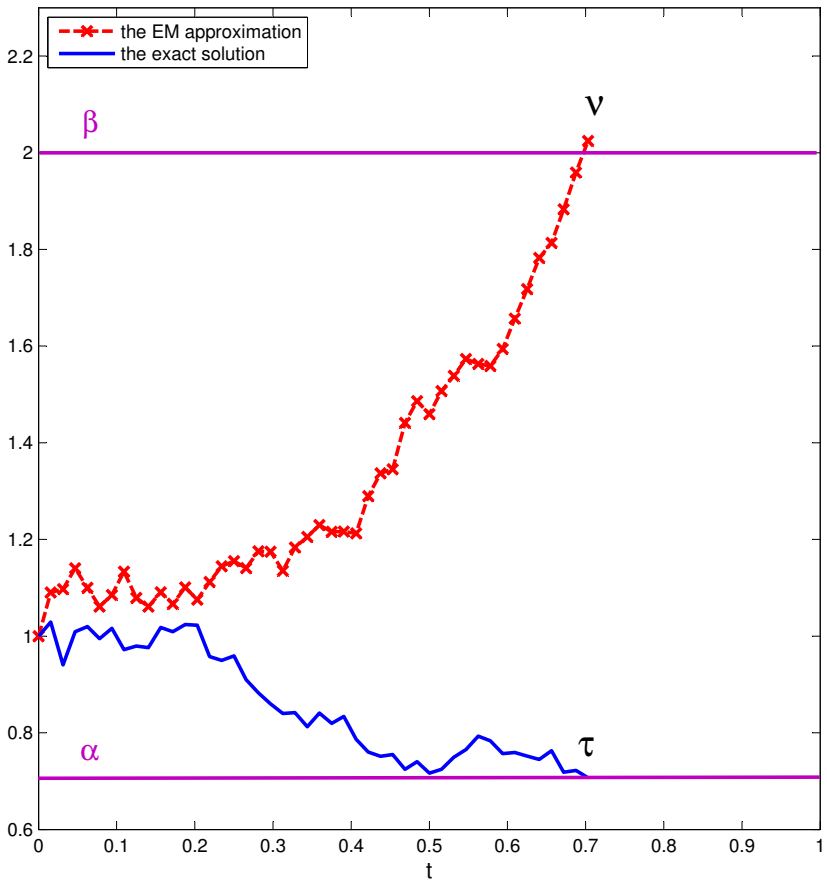

Figure 4.1: Bad approximation scenario for exit time

Strong convergence of Theorem 4.3.1 shows that $\nu \to \tau$ in L^1 with a rate of approximately $\frac{1}{2}$ as $\Delta t \to 0$. However, one may worry about a scenario where the two exit times are close, but the exact solution and numerical approximation exit at different areas of phase space. This is illustrated for the scalar case in Figure 4.1, with $O := (\alpha, \beta)$. In what follows, we show that this event is unlikely.

Corollary 4.3.2. *Under Assumptions 2.1.1, 4.2.1 and 4.2.2 we have*

$$\mathbb{E}[\ |x(\tau) - X(\nu)|^2\ ] = \mathcal{O}(|\Delta t \log(\Delta t)|^{\frac{1}{2}}).$$

Proof. We have

$$\begin{aligned}\mathbb{E}[\ |x(\tau)-X(\nu)|^2\ ] &= \mathbb{E}[\ |x(\tau)-X(\nu)|^2\ \mathbf{1}_{\{\tau\le\nu\}}\ ] + \mathbb{E}[\ |x(\tau)-X(\nu)|^2\ \mathbf{1}_{\{\tau>\nu\}}\ ] \\ &\le 2\ \mathbb{E}[\ |x(\tau)-X(\tau)|^2\ ] + 2\ \mathbb{E}[\ |X(\tau)-X(\nu)|^2\ \mathbf{1}_{\{\tau\le\nu\}}\ ] \\ &\quad + 2\ \mathbb{E}[\ |x(\nu)-X(\nu)|^2\ ] + 2\ \mathbb{E}[\ |x(\tau)-x(\nu)|^2\ \mathbf{1}_{\{\tau>\nu\}}\ ].\end{aligned}$$

From (2.8), the first and third terms on the right hand side are $\mathcal{O}(|\Delta t\log(\Delta t)|)$. Intuitively, the second term is the case where the exact solution is stopped and we want to show that we cannot move with the numerical approximation too far in a small interval of time. The fourth term is a mirror case of the second term, but this time the numerical approximation is stopped and we want to show that we cannot move with the exact solution too far in a small interval of time. First, let us focus on bounding the fourth term,

$$\begin{aligned}\mathbb{E}[\ |x(\tau)-x(\nu)|^2\ \mathbf{1}_{\{\tau>\nu\}}\ ] &= \mathbb{E}\left[\ \mathbf{1}_{\{\tau>\nu\}}\left(\left|\int_\nu^\tau f(x(t))dt + \int_\nu^\tau g(x(t))dw(t)\ \right|^2\right)\ \right] \\ &\le \mathbb{E}\left[\ \left|\int_{\nu\wedge\tau}^\tau f(x(t))dt + \int_{\nu\wedge\tau}^\tau g(x(t))dw(t)\ \right|^2\ \right] \\ &\le 2\ \mathbb{E}\left[\ (\tau-\nu\wedge\tau)\int_{\nu\wedge\tau}^\tau |f(x(t))|^2dt + \int_{\nu\wedge\tau}^\tau |g(x(t))|^2dt\ \right] \\ &\le 2C_1\ \mathbb{E}\left[\ (\tau-\nu\wedge\tau)^2 + (\nu-\nu\wedge\tau)\ \right] \\ &\le 2C_2\ \mathbb{E}[\ |\tau-\nu|\ ] = \mathcal{O}(|\Delta t\log(\Delta t)|^{\frac{1}{2}}),\end{aligned}$$

where in the last line we used Theorem 4.3.1. In a similar manner it can be shown that

$$\mathbb{E}[\ |X(\tau)-X(\nu)|^2\ \mathbf{1}_{\{\tau\le\nu\}}\ ] = \mathcal{O}(|\Delta t\log(\Delta t)|^{\frac{1}{2}}),$$

which finishes the proof. □

4.4 Computational Cost

4.4.1 Standard Monte Carlo

A traditional Monte Carlo approach to the estimation of the mean stopped exit time uses the Euler–Maruyama method to compute independent samples $\nu^{[i]}$

from the distribution of the corresponding random variable ν. Let $\nu^{[i]}$ denote the computed stopped exit time for the ith simulated path. If we use N such paths, then the mean value $\mathbb{E}[\tau]$ is approximated by the sample average

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \nu^{[i]}.$$

The overall error divides naturally into two parts,

$$\mathbb{E}[\tau] - \mu = \mathbb{E}[\tau - \nu + \nu] - \mu = (\mathbb{E}[\tau - \nu]) + (\mathbb{E}[\nu] - \mu)\,. \tag{4.7}$$

The first term in parentheses is the bias; that is, the weak error of the numerical method in terms of its ability to approximate the mean stopped exit time of the SDE. We know from (2.9) that this term is $\mathcal{O}(\Delta t^{\frac{1}{2}})$. The second term in (4.7) concerns the statistical sampling error. This is known to scale like $\mathcal{O}(1/\sqrt{N})$ from the perspective of confidence interval width (see, for example, [45]).

It is natural to measure the computational cost in terms of either

- the total number of evaluations of the coefficients $f(\cdot)$ or $g(\cdot)$ when we use the iteration (4.1), or
- the number of pseudo-random number calls to obtain the Brownian increments in (4.1).

In both cases, the computational cost of each path is proportional to the ratio of the time-span of the numerical approximation, $\nu^{[i]}$, and the stepsize, Δt. The overall expected computational cost of the standard Monte Carlo method is therefore proportional to $N\mathbb{E}[\nu]/\Delta t$, which, from (2.9), may be written

$$\frac{N\left(\mathbb{E}[\tau] + \mathcal{O}(\Delta t^{\frac{1}{2}})\right)}{\Delta t}. \tag{4.8}$$

If we let TOL denote the target level of accuracy, in terms of confidence interval width, then balancing the bias and sampling error in (4.7) gives the scaling $\text{TOL} = \Delta t^{\frac{1}{2}} = 1/\sqrt{N}$, whence the complexity measure (4.8) for the method becomes $\mathcal{O}(\text{TOL}^{-4})$.

4.4.2 Multi-level Monte Carlo

Our aim here is to develop the multi-level methodology in the exit time context.

Following [38], we consider a range of different stepsizes of the form $\Delta t_\ell = M^{-\ell}T$, for $\ell = 0, 1, 2, ..., L$. Here M is fixed integer. The smallest stepsize, $\Delta t_L = M^{-L}T$, is chosen so that the bias in the discretisation method matches the target accuracy of $\mathcal{O}(\text{TOL})$; matching $\Delta t_L^{\frac{1}{2}}$ with TOL then gives

$$L = \frac{\log \text{TOL}^{-2}}{\log M}. \tag{4.9}$$

Intuitively, the multi-level approach exploits the fact that it is not necessary to compute many paths at this high, and expensive, level of resolution. It is sufficient to compute a relative small number of 'high frequency' paths, and then pad out the computation with increasingly more information from the increasingly cheaper, lower resolution stepsizes.

To be more precise, we let the random variable ν_l denote the stopped exit time arising when the Euler–Maruyama approximation is used with stepsize Δt_ℓ. We continue by writing the following trivial identity:

$$\mathbb{E}[\nu_L] = \mathbb{E}[\nu_0] + \sum_{\ell=1}^{L} \mathbb{E}[\nu_\ell - \nu_{\ell-1}]. \tag{4.10}$$

On the left is the exact mean of the high-resolution approximation, which has the required bias. On the right is a telescoping series involving the different levels of resolution. We propose to estimate the expected values on the right hand side of (4.10) as follows. The first term, $\mathbb{E}[\nu_0]$, is estimated by a quantity Z_0 that uses the sample mean of N_0 independent paths; so

$$Z_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} \nu_0^{[i]}.$$

Each remaining term of the form $\mathbb{E}[\nu_\ell - \nu_{\ell-1}]$ is estimated by a quantity Z_ℓ based on N_ℓ independent pairs of paths; so

$$Z_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(\nu_\ell^{[i]} - \nu_{\ell-1}^{[i]} \right). \tag{4.11}$$

Here, the two samples $\nu_\ell^{[i]}$ and $\nu_{\ell-1}^{[i]}$ come from the same Brownian path at the two different levels of resolution. Figure 4.2 illustrates the case where $M = 4$ for a scalar SDE; so the two paths are based on stepsizes that differ in size by a factor of 4. Each Brownian increment for the lower resolution computation is given by the sum of the four increments used for the higher resolution path. Independent paths are used for each i in (4.11), and for each different level $\ell = 1, 2, \ldots, L$.

It remains to control the statistical sampling error by choosing $\{N_\ell\}_{\ell=0}^L$ in order to give an overall variance of $\mathcal{O}(\mathrm{TOL}^2)$ for the estimator.

Using Theorem 4.3.1, we have

$$\mathrm{Var}\left[\nu_\ell - \tau\right] \leq \mathbb{E}[(\nu_\ell - \tau)^2] = \mathcal{O}(|\Delta \mathrm{t}_\ell \log(\Delta \mathrm{t}_\ell)|^{\frac{1}{2}}),$$

and

$$\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] \leq (\sqrt{\mathrm{Var}\left[\nu_\ell - \tau\right]} + \sqrt{\mathrm{Var}\left[\nu_{\ell-1} - \tau\right]})^2 = \mathcal{O}(|\Delta \mathrm{t}_\ell \log(\Delta \mathrm{t}_\ell)|^{\frac{1}{2}}). \quad (4.12)$$

So $\mathrm{Var}\left[\mathrm{Z}_\ell\right] = \mathcal{O}\left(|\Delta \mathrm{t}_\ell \log(\Delta \mathrm{t}_\ell)|^{\frac{1}{2}}/\mathrm{N}_\ell\right)$ and, because the computations are independent over different levels, our overall estimator $Z := Z_0 + \sum_{\ell=1}^L Z_\ell$ has variance

$$\mathrm{Var}\left[\mathrm{Z}\right] = \mathrm{Var}\left[\mathrm{Z}_0\right] + \sum_{\ell=1}^{\mathrm{L}} \mathcal{O}\left(\frac{|\Delta \mathrm{t}_\ell \log(\Delta \mathrm{t}_\ell)|^{\frac{1}{2}}}{\mathrm{N}_\ell}\right).$$

Taking

$$N_0 = \mathcal{O}(\mathrm{TOL}^{-2}), \quad (4.13)$$

$$N_\ell = \mathcal{O}(\mathrm{TOL}^{-2} M^{L/4} \Delta t_\ell^{3/4} |\log(\Delta t_\ell)|^{1/2}), \quad \text{for } \ell = 1, \ldots, L, \quad (4.14)$$

it follows that

$$\begin{aligned}
\mathrm{Var}\left[\mathrm{Z}\right] &= \mathcal{O}(\mathrm{TOL}^2) + \sum_{\ell=0}^{L} \mathcal{O}\left(\frac{|\Delta t_\ell \log(\Delta t_\ell)|^{\frac{1}{2}}}{N_\ell}\right) \\
&= \mathcal{O}(\mathrm{TOL}^2) + \sum_{\ell=0}^{L} \mathcal{O}\left(\frac{|\Delta t_\ell \log(\Delta t_\ell)|^{1/2}}{\mathrm{TOL}^{-2} M^{L/4} \Delta t_\ell^{3/4} |\log(\Delta t_\ell)|^{1/2}}\right) \\
&= \mathcal{O}(\mathrm{TOL}^2) + \mathcal{O}\left(\mathrm{TOL}^2 M^{-L/4} \sum_{\ell=0}^{L} M^{\ell/4}\right) \\
&= \mathcal{O}(\mathrm{TOL}^2),
\end{aligned}$$

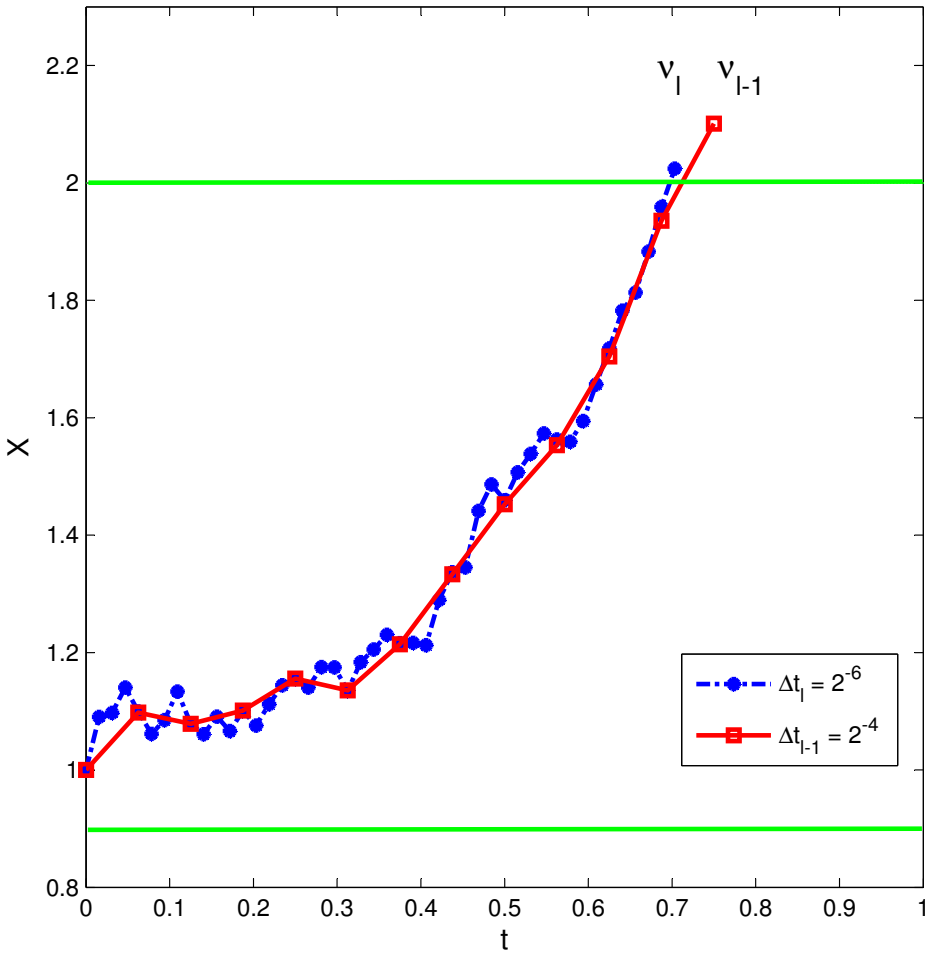

Figure 4.2: Illustration of the estimator Z_ℓ, in the case $M = 4$ for a scalar SDE. The same path is sampled at two different levels of resolution.

as required.

Having specified the algorithm, we may now work out the expected computational cost. Each pair of paths at level ℓ has a cost proportional to $v_\ell^{[i]}/\Delta t_\ell$. Hence, the expected computational cost of the multi-level Monte Carlo method

is

$$\begin{aligned}\sum_{\ell=0}^{L}\frac{N_\ell}{\Delta t_\ell}\mathbb{E}[\nu_\ell] &= \sum_{\ell=0}^{L}\frac{\mathrm{TOL}^{-2}M^{L/4}\Delta t_\ell^{3/4}|\log(\Delta t_\ell)|^{1/2}}{\Delta t_\ell}\left(\mathbb{E}[\tau]+\mathcal{O}(\Delta t_\ell^{1/2})\right)\\ &= \mathcal{O}\left(\mathrm{TOL}^{-2}M^{L/4}|\log(\Delta t_L)|^{1/2}\sum_{\ell=0}^{L}\Delta t_\ell^{-1/4}\right)\\ &= \mathcal{O}\left(\mathrm{TOL}^{-2}M^{L/2}|\log(\mathrm{TOL})|^{1/2}\right).\end{aligned}$$

From (4.9) we see that $M^{L/2} = \mathcal{O}(\mathrm{TOL}^{-1})$, and so the expected computational cost may be written

$$\mathcal{O}\left(\mathrm{TOL}^{-3}|\log(\mathrm{TOL})|^{1/2}\right).$$

This should be compared with the value $\mathcal{O}(\mathrm{TOL}^{-4})$ from section 4.4.1 for standard Monte Carlo.

4.5 Computational Results for Scalar Case

We now present computational results, focussing on the scalar case. Our aims are to test the sharpness of the analysis and to check whether the asymptotically valid improvement in complexity can be observed in a real simulation. Here O takes the form of an open interval, denoted (α, β).

4.5.1 Strong Error in Exit Time

We begin by checking the sharpness of the strong convergence rate from Theorem 4.3.1 on a geometric Brownian motion model

$$dx(s) = fx(s)ds + gx(s)dw(s), \tag{4.15}$$

with constant drift coefficient $f = 0.05$, constant volatility $g = 0.2$ and initial value $x(0) = 2$. We set the boundaries to $\alpha = 1.7$ and $\beta = 2.3$, and the finite cutoff time to $T = 1$. We computed the reference solution $\mathbb{E}[\tau] = 0.4962$ using a numerical solver for the associated boundary value ordinary differential equation. We approximated the strong exit time error $\mathrm{err}_{\Delta t} \coloneqq \mathbb{E}\left[|\tau - \nu|^2\right]$ using the sample mean from $N = 5 \times 10^3$ path simulations, and stepsizes $\Delta t = 2^{-8}, 2^{-9}, 2^{-10}, 2^{-11}$. The error behaviour on a log-log scale is shown in Figure 4.3. A least squares fit for

$\log C$ and q in $\log \mathrm{err}_{\Delta t} = \log C + q \log \Delta t$ produced $q = 0.5167$ with a least squares residual of 0.0289. Error bars representing 95% confidence intervals are small enough to be covered by the stars in the figure. Thus, our results suggest that the strong order of convergence equal to approximately one-half in Theorem 4.3.1 is sharp for $p = 2$.

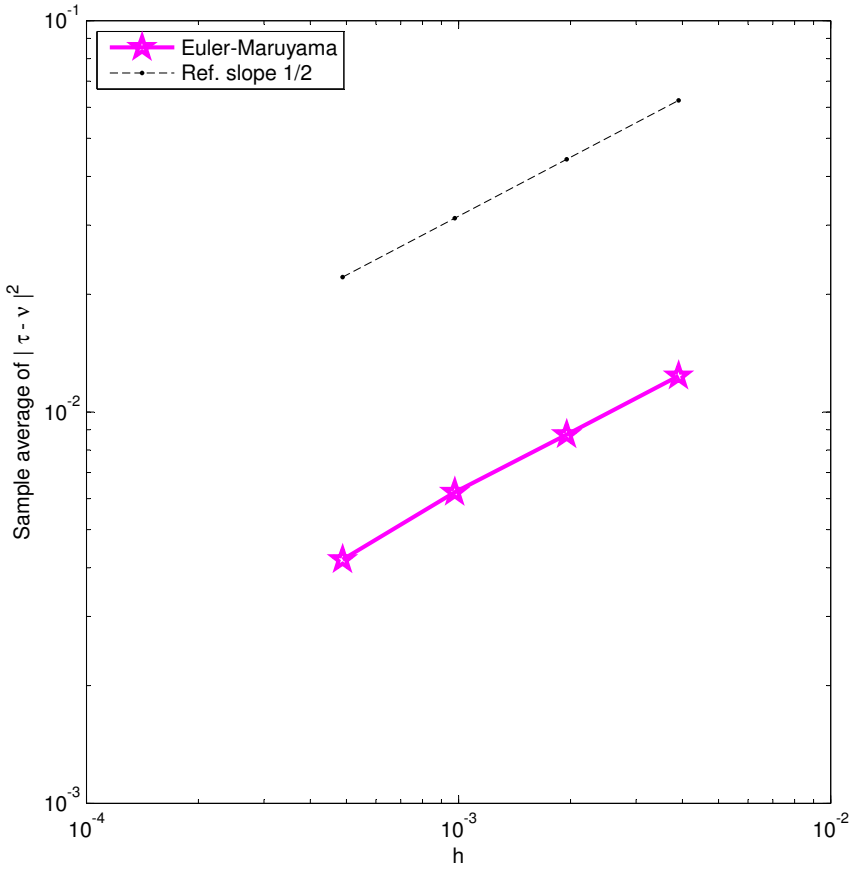

Figure 4.3: Strong error in stopped exit time.

4.5.2 Variance Behaviour

A key step in our analysis of the multi-level Monte Carlo algorithm was the derivation of the variance estimate (4.12). In Figure 4.4 we plot the quantity $\log(\mathrm{Var}[\nu_\ell - \nu_{\ell-1}])/\log(\mathrm{M})$ over a sequence of levels, where the most refined level

corresponds to a user-specified accuracy of TOL = 0.001. If (4.12) is sharp, then this plot will have a slope of approximately −1/2. A least squares fit for the slope gives $q = -0.4984$ with a residual of 0.1040. We also include a line with a slope $-\frac{1}{2}$ for reference. Here we used the linear SDE (4.15) with boundaries changed to $\alpha = 0.9$, $\beta = 1.1$ and initial value $x(0) = 1$, whence $\mathbb{E}[\tau] = 0.2480$.

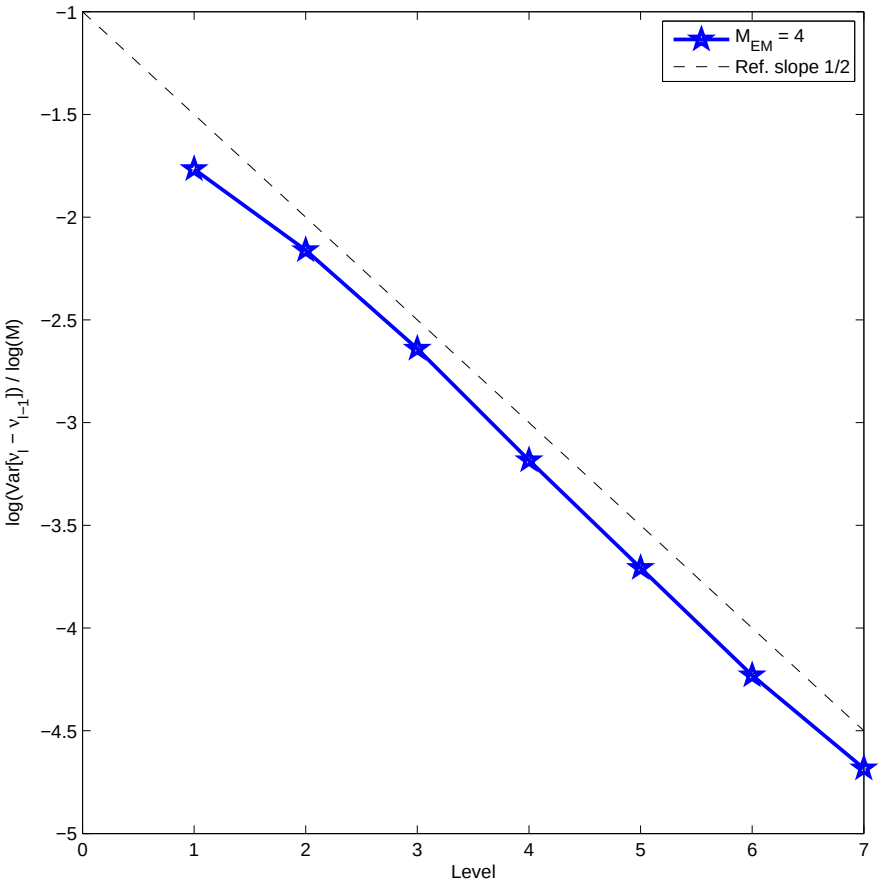

Figure 4.4: Variance of $\nu_\ell - \nu_{\ell-1}$ over different levels.

4.5.3 Complexity for Geometric Brownian Motion

Finally, we compare the computational cost versus accuracy for standard and multi-level mean exit time computation. We measure the computational cost of

the multi-level method as

$$\mathrm{Cost}_{\mathrm{MLMC}} := \left(N_0 + \sum_{\ell=1}^{L} N_\ell M^\ell \right) \frac{d}{T} \mathbb{E}[\tau], \tag{4.16}$$

where d is the dimension of the approximated process. For the complexity analysis in section 4.4.2, we specified an order of magnitude for the number of paths per level, (4.14), and showed that this (a) produces the correct variance and (b) has attractive complexity.

In our implementation, we used the values

$$N_\ell = \left\lceil 2\mathrm{TOL}^{-2} \sqrt{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] \mathrm{M}^{-\ell}} \left(\sum_{\ell=0}^{L} \sqrt{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] \mathrm{M}^{\ell}} \right) \right\rceil, \quad 0 \le \ell \le L. \tag{4.17}$$

More precisely, we used order of magnitude estimates of the variances $\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right]$ that were obtained numerically in a pre-processing step of negligible cost. To justify the choice (4.17), we note that

$$\begin{aligned} \mathrm{Var}\left[\mathrm{Z}\right] &= \sum_{\ell=0}^{L} \left(\frac{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right]}{N_\ell} \right) \\ &= \sum_{\ell=0}^{L} \frac{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right]}{2\mathrm{TOL}^{-2} \sqrt{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] \mathrm{M}^{-\ell}} \left(\sum_{\ell=0}^{L} \sqrt{\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] \mathrm{M}^{\ell}} \right)} \\ &= \frac{1}{2} \mathrm{TOL}^2, \end{aligned}$$

so we achieve a small enough variance.

We also note that the order of magnitude for N_ℓ in (4.14) that we used for the complexity analysis satisfies

$$\begin{aligned} \mathrm{TOL}^{-2} M^{L/4} \Delta t_\ell^{3/4} |\log(\Delta t_\ell)|^{1/2} &\le \mathrm{TOL}^{-2} M^{L/4} \Delta t_\ell^{3/4} |\log(\Delta t_\ell)|^{1/4} |\log(\Delta t_L)|^{1/4} \\ &\le \mathrm{TOL}^{-2} \Delta t_\ell^{3/4} |\log(\Delta t_\ell)|^{1/4} \sum_{\ell=0}^{L} M^{\ell/4} |\log(\Delta t_\ell)|^{1/4}. \end{aligned}$$

Since $\mathrm{Var}\left[\nu_\ell - \nu_{\ell-1}\right] = \mathcal{O}(\Delta \mathrm{t}_\ell^{1/2} |\log(\Delta \mathrm{t}_\ell)|^{1/2})$ it follows that our practical choice of optimal number of paths on level ℓ gives a comparable order of paths per level.

The computational cost of the standard method is measured as

$$\text{Cost}_{\text{stdMC}} := \frac{Nd}{\Delta t}\mathbb{E}[\tau], \tag{4.18}$$

where N is the total number of sample paths, d is the dimension and Δt is the fixed stepsize such that $\Delta t = \mathcal{O}(\text{TOL}^2)$. Here N was chosen adaptively to produce the required $\mathcal{O}(\text{TOL}^2)$ variance. We used three choices of the accuracy parameter, $\text{TOL} = 10^{-1.5}, 10^{-2}, 10^{-2.5}$. For the SDE and parameters used in Figure 4.4, the lower left picture in Figure 4.5 shows the accuracy obtained by the multi-level algorithm as a function of TOL. This confirms that the algorithm produces an error that scales like TOL.

In the top left picture of Figure 4.5 we plot TOL against $\text{TOL}^3 \times \text{Cost}$ for the two methods. The results are consistent with the predictions from our analysis: for the multi-level method this scaled complexity increases slowly like $|\log \text{TOL}|^{1/2}$ whereas for standard Monte Carlo it increases like TOL^{-1}.

In the right of Figure 4.5, we repeat this information in a slightly different format. The upper right hand picture plots the overall cost against the attained error, and the lower right hand picture plots the overall cost against the accuracy parameter. We see that for the most stringent accuracy requirement the multi-level version is about an order of magnitude cheaper.

4.5.4 Complexity for Mean Reverting Square Root Process

As we mentioned in section 4.2, one of the benefits of a mean exit time analysis is that our attention is restricted to a compact domain, so rigorous results can be derived for nonlinear SDEs with coefficients that do not satisfy a global Lipschitz condition. We consider now a mean reverting square root process, proposed by Cox, Ingersoll and Ross (CIR), commonly used to model interest rates [74]

$$dx(s) = f(\mu - x(s))ds + g\sqrt{x(s)}dw(s), \tag{4.19}$$

noting that the diffusion coefficient is not Lipschitz at the origin. Keeping all the parameters the same as for the geometric Brownian motion test in Figure 4.5, additionally we fix $\mu = 1$, which causes the stopped mean exit time to increase

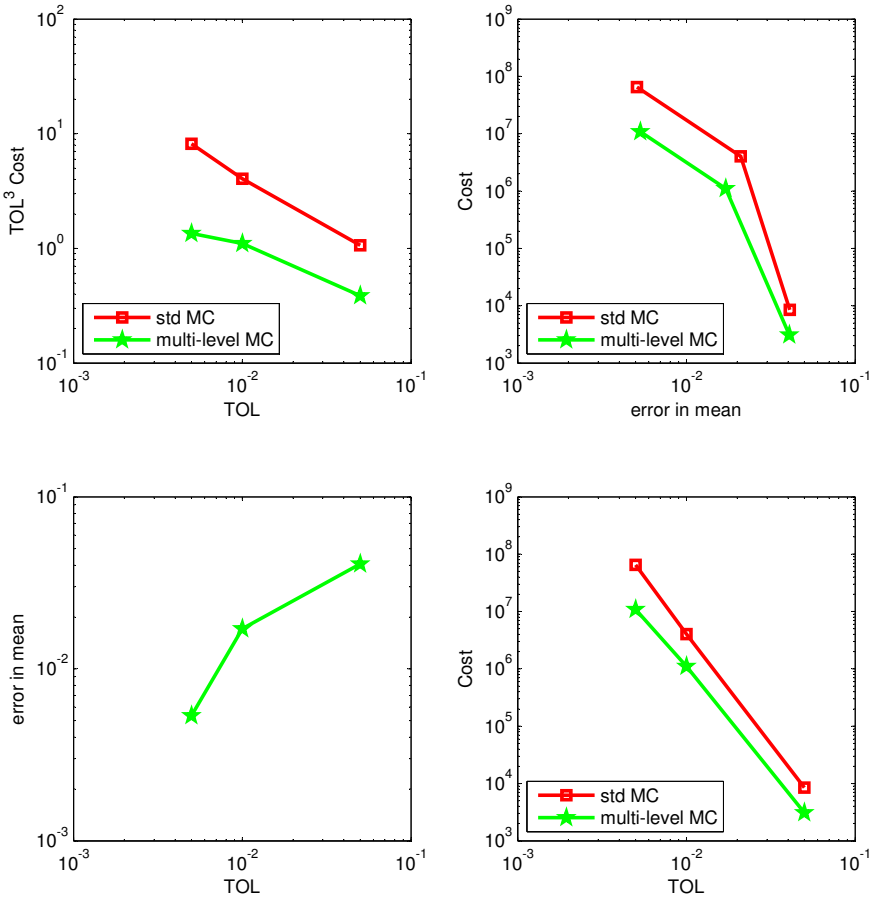

Figure 4.5: Cost versus accuracy for the standard Monte Carlo method and the multi-level Monte Carlo algorithm on geometric Brownian motion.

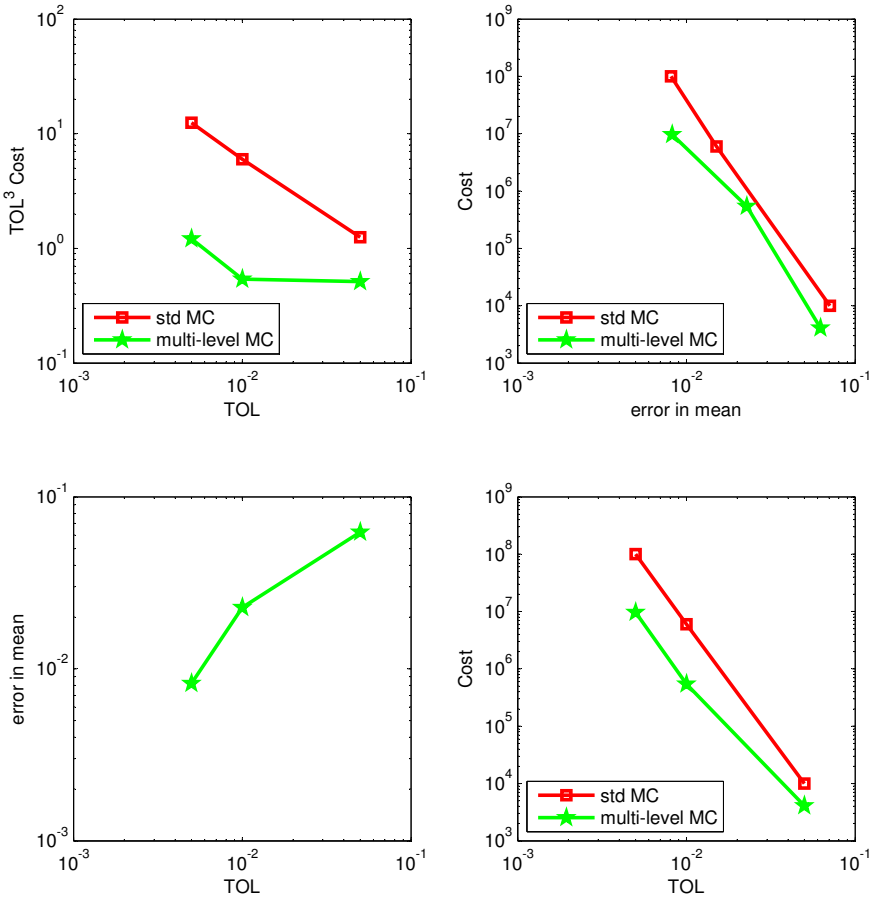

Figure 4.6: Cost versus accuracy for the standard Monte Carlo method and the multi-level Monte Carlo algorithm on the mean reverting square root process.

to 0.2495. The complexity results, shown in Figure 4.6, are consistent with those for the geometric Brownian motion case.

4.6 Summary

In this chapter we showed that it is possible to adapt the MLMC algorithm to the mean exit time problem. We tested the method numerically in one-dimensional cases. In the next chapter we test the algorithm in more dimensions on a range of different applications.

Chapter 5

Computational Results in More Dimensions

> The more you know, the less sure you are.
>
> Voltaire, 1694-1778

Parts of this chapter have been published in Higham and Roj [60]. Our aim here is to present further computational tests in more than one dimension. In the following sections we present computational results for two-dimensional Brownian motion, a simple neural network (two-dimensional correlated Brownian motion with drift) and a first-to-default basket of corporate bonds (five-dimensional geometric Brownian motion.) For each example we confirm that the multi-level approximation converges to the correct value and estimate the order of the variance between fine and coarse approximations of stopped exit times. We also record the computational complexity, in comparison with the standard Monte Carlo method. We therefore add to the numerical tests in Chapter 4 and [59] by considering SDEs in more than one dimension and problems on a non-compact domain. Example 5.1 fulfils all the assumptions of Theorem 4.3.1; examples 5.2 and 5.3 violate the assumption of a non-compact domain. However, we show numerically that the multi-level algorithm performs well in this setting, too.

5.1 Two-dimensional Brownian Motion

We first simulate two-dimensional Brownian motion starting at the origin, with boundary given by a ball of radius R. In the notation of (2.1) we have $d = 2$, $f \equiv 0$ and $g = I$, where I denotes the identity matrix. The finite cutoff time is set to $T = 1$ and simulations are performed for three different radius sizes. By varying the radius we check whether the multi-level algorithm performs well when the majority of sample paths leave the domain in the fixed time T and also when they stay in the domain within this time. The estimated stopped exit time value is compared against a reference value, which in this case was obtained using Monte Carlo simulations with $N = 5 \times 10^7$ samples and a stepsize $\Delta t = 10^{-4}$. The results are presented in Table 5.1. We observe that a more stringent accuracy requirement decreases the error in the multi-level estimate, as expected.

Table 5.1: Two-dimensional Brownian motion. Values in parentheses indicate the half-width of the 95% confidence interval.

Radius	**Reference value**	**TOL**	**MLMC**	**Absolute error %**
$R = 0.5$	0.1279 (0.00003)	0.01	0.1317	2.97
		0.005	0.1354	5.86
		0.001	0.1294	1.17
$R = 1.0$	0.4737 (0.00008)	0.01	0.5037	6.33
		0.005	0.4807	1.48
		0.001	0.4758	0.44
$R = 1.5$	0.7830 (0.00008)	0.01	0.8454	7.97
		0.005	0.7853	0.29
		0.001	0.7847	0.22

In a separate test, we then fix $R = 1$ and check the convergence behaviour. In the left picture of Figure 5.1 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of TOL for TOL $= 10^{-3}$, 5×10^{-3}, 10^{-2} and 5×10^{-2}. We see that the algorithm produces an error that scales like TOL. A line with a slope 1 is included for reference.

In the right picture of Figure 5.1 we plot the quantity $\log(\mathrm{Var}[\nu_\ell - \nu_{\ell-1}])/\log(M)$ over different levels, for a user-specified accuracy of TOL $= 0.001$. A least squares fit for the slope produces $q = -0.5525$ with a residual of 0.1884. A line with a slope $-\frac{1}{2}$ is also included for reference. This agrees with the estimate in section 4.4 of Chapter 4. Finally, we compare the computational complexity of standard Monte

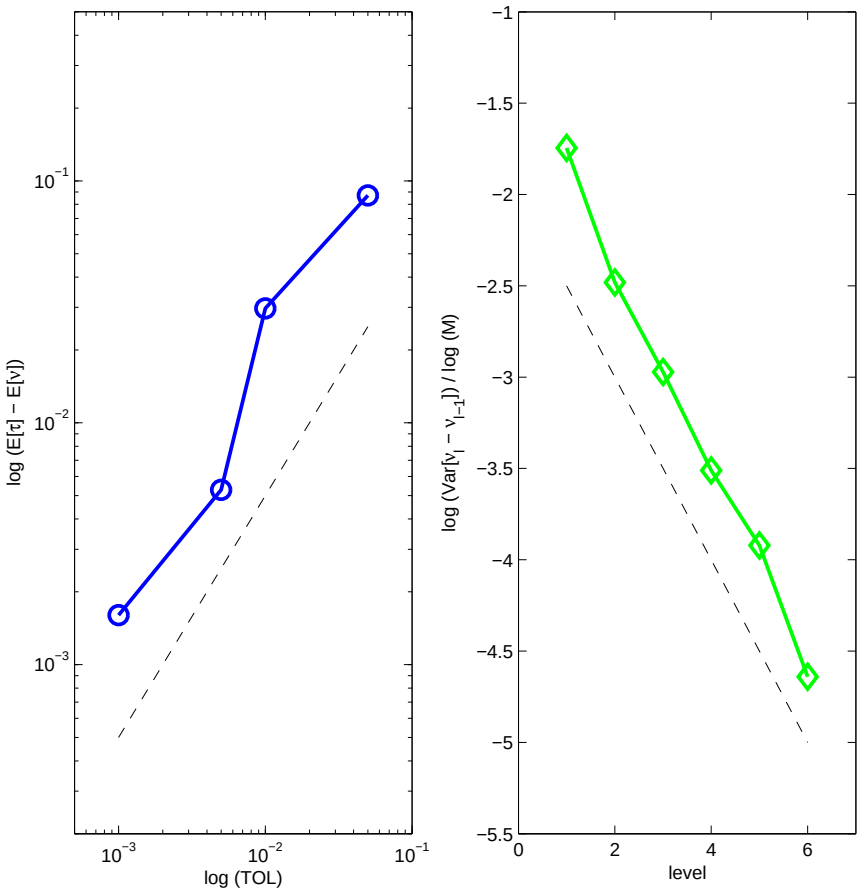

Figure 5.1: Two-dimensional Brownian motion. Left: weak error of the multi-level algorithm. Right: variance of $\nu_l - \nu_{l-1}$ over a sequence of levels.

Carlo with the multi-level version in Figure 5.2. The computational cost of the multi-level version is measured as in (4.16), where N_ℓ is the number of paths used on each level, calculated as

$$N_\ell = \left\lceil 2\mathrm{TOL}^{-2}\sqrt{\mathrm{Var}[\nu_\ell - \nu_{\ell-1}]M^{-\ell}}\left(\sum_{\ell=0}^{L}\sqrt{\mathrm{Var}[\nu_\ell - \nu_{\ell-1}]M^{\ell}}\right)\right\rceil, \quad 0 \le \ell \le L.$$

We measure the computational cost of the standard Monte Carlo method as in (4.18). In Figure 5.2 we plot on a log-log scale the computational cost as a function of the accuracy parameter TOL. For TOL = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7} and 2^{-8} we perform 50 multi-level Monte Carlo computations using different initial states of the pseudo-random number generator, and plot the complexity results with green circles. Then we take averages for each accuracy, indicated as black crosses, and compare them with the complexity of standard Monte Carlo, indicated as red squares. A least squares fit performed on the Monte Carlo slope produces $q = -4.2156$ with a residual 1.0966, and on the multi-level Monte Carlo slope gives $q = -3.1688$ with a residual of 1.2891. This is in agreement with the analytical results quoted in section 4.4: the standard Monte Carlo complexity of $\mathcal{O}(\mathrm{TOL}^{-4})$ and the multi-level Monte Carlo complexity of $\mathcal{O}(\mathrm{TOL}^{-3}|\log(\mathrm{TOL})|^{1/2})$.

5.2 Simple Neural Network

We now apply the multi-level Monte Carlo algorithm in a neural network setting. Various models have been considered for the firing of single neurones, but there is an agreement among researchers that if the electrical state of the neural membrane is stated as a single number, which moves towards or away from the firing potential depending on whether the neurone receives excitatory or inhibitory input, respectively, then the time to firing can be estimated by the first exit time of a certain level for a Brownian motion with drift [30, 34, 63].

Here we are interested in the mean first exit time of 2-dimensional correlated Brownian motion with drift,

$$dx_i(s) = f_i ds + g_i dw_i(s),$$

with constant drift coefficients $f_1 = 0.1$, $f_2 = 0.2$, constant diffusion coefficients

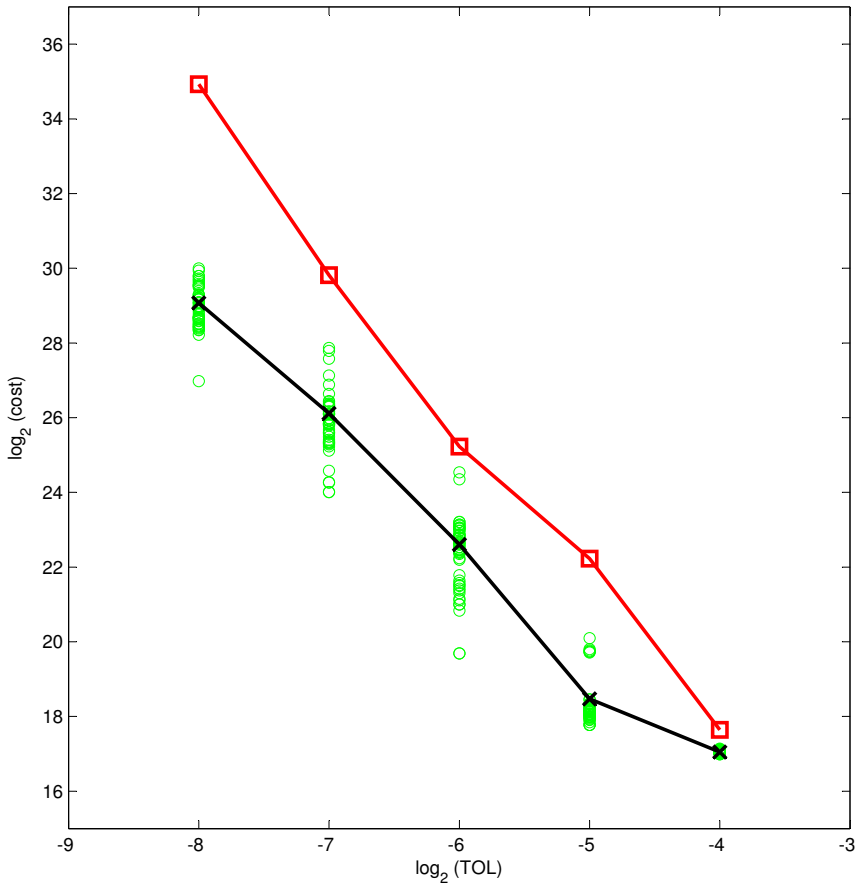

Figure 5.2: Two-dimensional Brownian motion. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages of the multi-level Monte Carlo cost for each accuracy.

$g_i = 1$, $i = 1, 2$, initial values $x_i(0) = 1$, $i = 1, 2$ and correlation coefficient $\rho = -0.5$, producing the variance-covariance matrix $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$. The Cholesky decomposition of Σ gives $C = \begin{bmatrix} 1 & -0.5 \\ 0 & 0.8660 \end{bmatrix}$, which is then used to simulate correlated Brownian motions. For $i = 1, 2$ we define the stopping times

$$\tau_i = \inf\{t > 0 : x_i(t) \leq B_i\},$$

where we set $B_1 = 0.5$, $B_2 = 0.25$. The quantity of interest is then $\mathbb{E}\left[(\tau_1 \wedge \tau_2) \wedge T\right]$. In Figure 5.3 we check the convergence rates of the algorithm and variance. The results are consistent with those in subsection 5.1. A least squares fit for the variance slope gives $q = -0.5025$ with a residual 0.0559.

We also check the computational cost of the algorithm in Figure 5.4. A least squares fit for the standard Monte Carlo slope produces $q = -3.9492$ with a residual 1.1694, whereas the multi-level method gives $q = -3.2004$ with a residual 0.5913, in line with the theoretical complexity outlined in section 4.4.

5.3 First-to-Default Swaps

Finally, we apply the multi-level Monte Carlo algorithm in a financial setting to basket default swaps used in risk management. These financial instruments are derivative securities tied to an underlying basket of corporate bonds or other assets subject to credit risk. A basket default swap gives the protection buyer a type of insurance against the possibility of default in exchange for regular payments made to the protection seller [18]. These instruments are popular mainly because insuring a basket of assets is usually cheaper than insuring each asset separately. We focus on the example of a first-to-default swap in which the protection buyer is compensated if any asset in the basket defaults, at which time the contract expires [64].

Without trying to price the protection and value legs at several dates, here we are interested in a mean first default time of the first-to-default basket of 5 corporate bonds with maturity $T = 2$ years. We note that it is common to use structural models (see, for example, [12]) with a geometric Brownian motion to

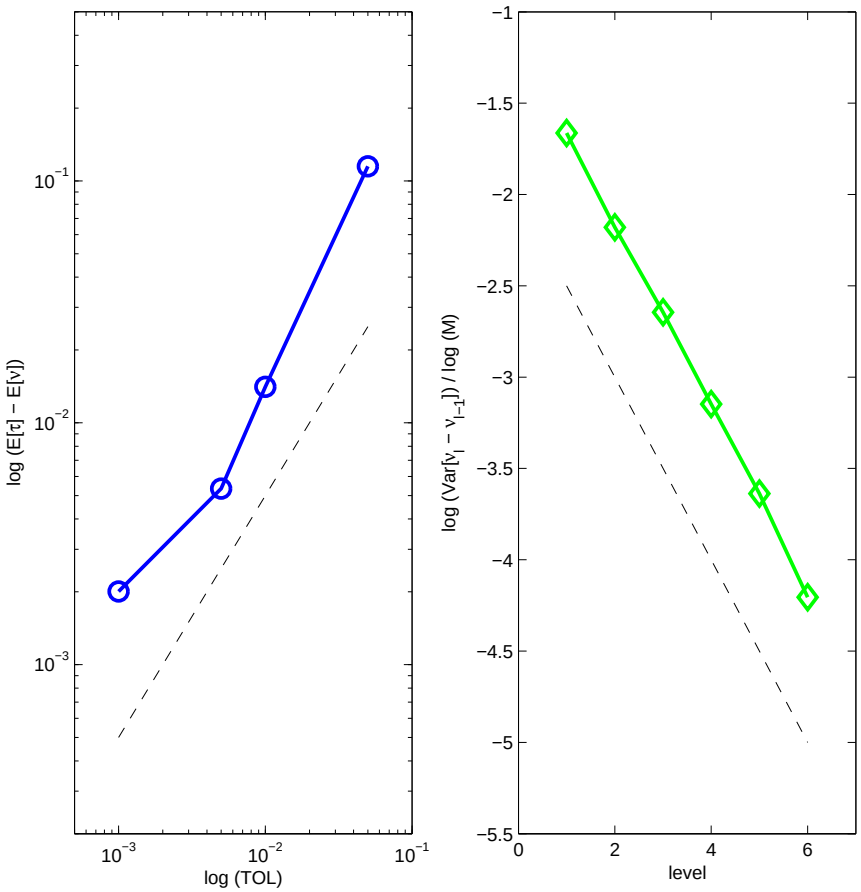

Figure 5.3: Neural network. Left: weak error of the multi-level algorithm. Right: variance of $\nu_\ell - \nu_{\ell-1}$ over a sequence of levels.

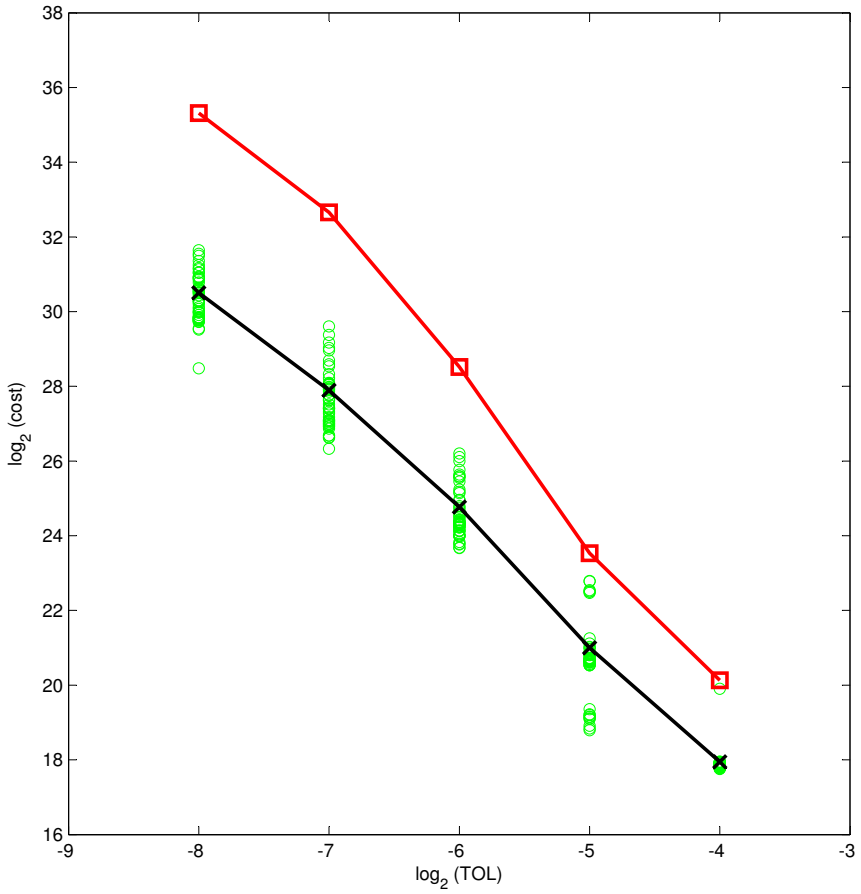

Figure 5.4: Neural network. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages over 50 simulations of multi-level Monte Carlo cost for each accuracy.

describe dynamics for the value of the underlying assets [22]. Hence, we simulate the underlying processes using a geometric Brownian motion model

$$dx_i(s) = f_i x_i(s)ds + g_i x_i(s)dw_i(s),$$

with constant drift coefficients $f_1 = 0.11$, $f_2 = 0.09$, $f_3 = 0.07$, $f_4 = 0.04$, $f_5 = 0.02$, and constant volatilities $g_1 = 0.4$, $g_2 = 0.6$, $g_3 = 0.8$, $g_4 = 0.8$, $g_5 = 1$. The initial value of each company is set to 1, i.e. $x_i(0) = 1$, $i = 1, \ldots, 5$. The lower boundary, which is a default level, is set to $B = 0.1$. We are interested in estimating the quantity $\mathbb{E}\left[(\tau_1 \wedge \tau_2 \wedge \tau_3 \wedge \tau_4 \wedge \tau_5) \wedge T\right]$. We used Monte Carlo simulations with a fixed timestep $\Delta t = 10^{-3}$ and $N = 5 \times 10^6$ samples to obtain a reference value of $\mathbb{E}[\tau \wedge T] = 1.2359$ for the mean first default time of the basket. The half-interval width of the 95% confidence interval is 0.0005, making the statistical error negligible.

We use the same format as in the previous two examples. In the right hand picture of Figure 5.5 a least squares fit for the slope produces $q = -0.6049$ with a residual of 0.5252.

In Figure 5.6 a least squares fit performed on the Monte Carlo slope produces $q = -4.1018$ with a residual 2.9224, and on the multi-level Monte Carlo slope gives $q = -2.7704$ with a residual of 1.4031. The results are therefore consistent with the theoretical predictions.

5.4 Summary

In this chapter we presented more examples for the mean exit time problem. The multi-level algorithm seem to perform very well in more than one dimension and for the cases with a compact and non-compact domain. Next, we present a new method for approximating exit times and adapt it to the multi-level setting.

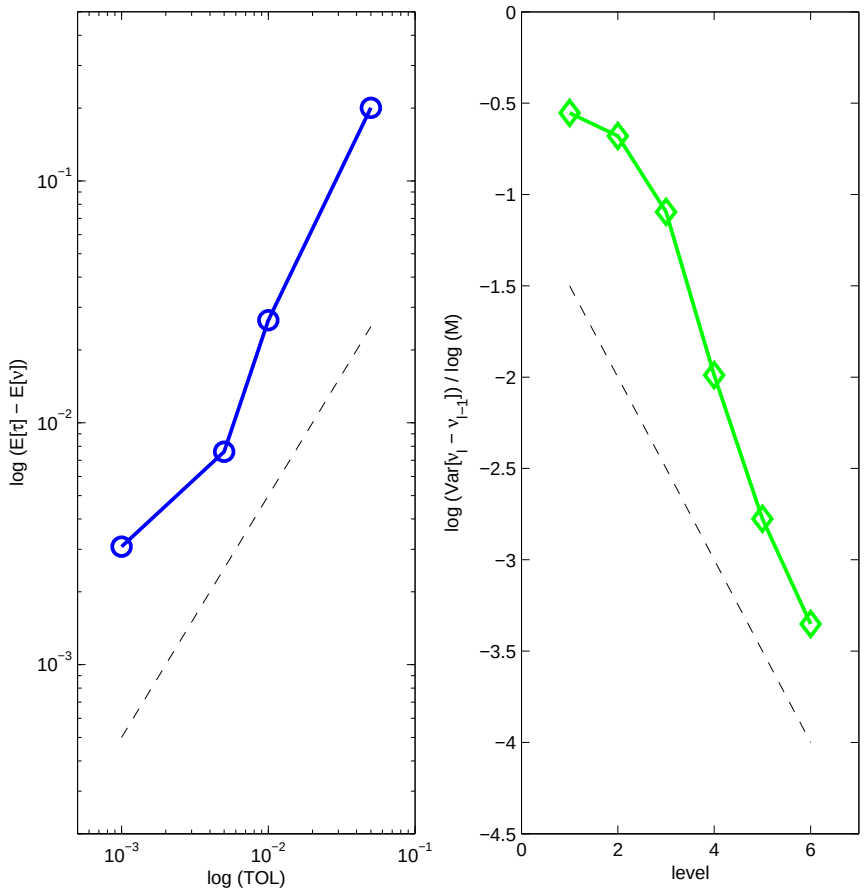

Figure 5.5: First-to-default. Left: weak error of the multi-level algorithm. Right: variance of $\nu_l - \nu_{l-1}$ over a sequence of levels.

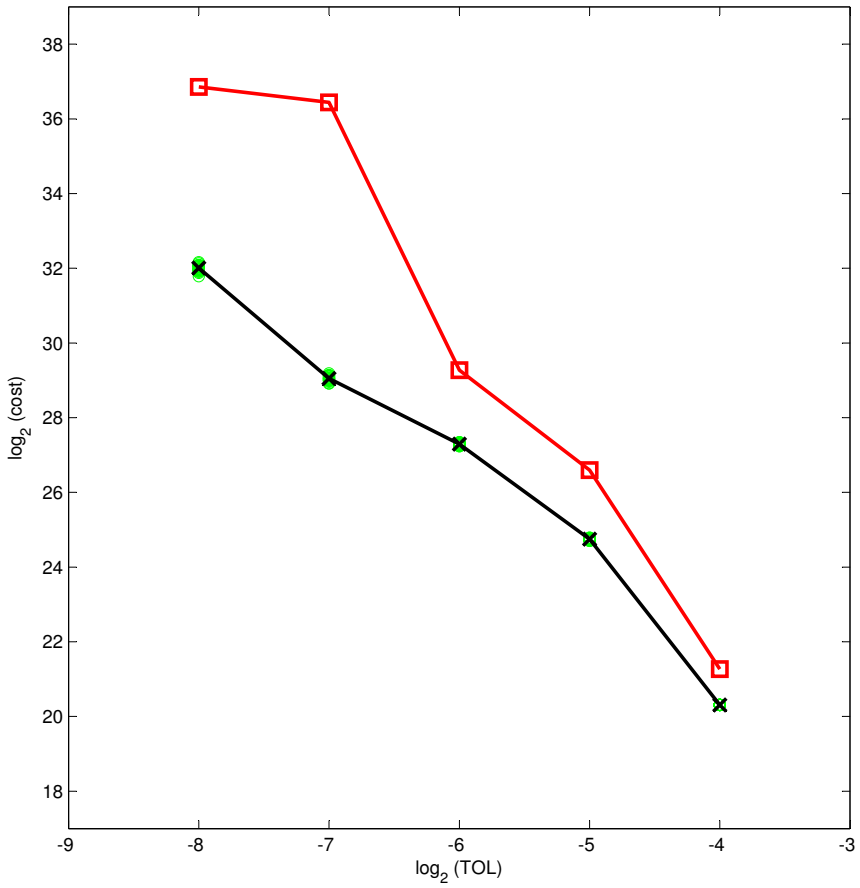

Figure 5.6: First-to-default. Computational cost of standard Monte Carlo (red squares) and multi-level Monte Carlo (green circles). Black crosses indicate averages over 30 simulations of the multi-level Monte Carlo cost for each accuracy.

Chapter 6

Multi-level Monte Carlo Using Brownian Bridge Interpolation for Expected Exit Times

> It is remarkable that a science which began with the consideration of games of chance should have become the most important object of human knowledge.
>
> Pierre Simon Laplace, 1749-1827

Parts of this chapter form a working paper [41].

6.1 Introduction

We consider a general d-dimensional Itô process as defined in (2.1) with the same conditions as previously. We also introduce a half-space $B \subset \mathbb{R}^d$ and deterministic initial condition $x_0 \in B$.

Although popular and adopted by many authors [3,13,73], straightforward implementation of Monte Carlo simulations for estimating expected stopping times turns out to be inefficient. This is due in part to the fact that by sampling discrete points we can miss the exit and stay in the domain after the true process has left the domain. In order to avoid completely missing an exit, we can explore the

properties of the Brownian bridge interpolation between two grid points. This conditional Monte Carlo approach is based on replacing an estimator with its conditional expectation [13, 45] and has been applied in the barrier options context by Gobet in [48, 50] and by Giles in [37]. Here, we use the same technique for estimating expected exit times.

For a given root-mean-square accuracy, TOL, we saw in Chapter 4 that combining Euler–Maruyama samples in a standard Monte Carlo approach leads to a computational cost of $\mathcal{O}(\mathrm{TOL}^{-4})$. A multi-level alternative was developed in Chapter 4, which has a cost of $\mathcal{O}(\mathrm{TOL}^{-3}|\log \mathrm{TOL}|^{1/2})$. The new sampling method that we develop in this chapter, which is based on the Brownian bridge interpolation, leads to a cost of $\mathcal{O}(\mathrm{TOL}^{-3})$ when used with standard Monte Carlo, and $\mathcal{O}(\mathrm{TOL}^{-2})$ in a suitable multi-level setting.

Some of the ideas behind the new algorithm were tested numerically in [89]. Our aim here is to develop and rigorously analyse a complete algorithm. We begin by introducing notation in section 6.2. The key result is spread over Theorems 6.3.2 and 6.3.8, which are presented in section 6.3. In section 6.4 we compare the expected computational cost of the new method in the Monte Carlo and multi-level Monte Carlo settings. Finally, this chapter is concluded with numerical results in section 6.5 and 6.6.

6.2 Notation

We begin with the Milstein discretisation defined as in (2.4). We drop the hat for clarity, that is, we use $X_k^\ell \approx x(k\Delta t_\ell)$ of equation (2.1) with initial data $X_0 = x_0$ and stepsize $\Delta t_\ell = 2^\ell T$. We make the following assumptions throughout this chapter.

Assumption 6.2.1.

*A*1 ***Commutativity condition:*** *the diffusion tensor satisfies*

$$h_{ijn}(x) = h_{inj}(x),$$

for all $j, n = 1, ..., m$, $i = 1, ..., d$, $x \in \mathbb{R}^d$.

A2 **C^2 *continuity:*** *the drift coefficient f and diffusion coefficient g have two continuous bounded derivatives on $B \subset \mathbb{R}^d$.*

A3 ***Strong ellipticity:*** *for some $c > 0$, $\sum_{ij}(gg^T(x))_{ij}\xi_i\xi_j > c|\xi|^2$, for all $x \in B$, $\xi \in \mathbb{R}^d$.*

A4 ***Regularity of the boundary:*** *$B \subset \mathbb{R}^d$ is a half-space.*

We note that the commutativity condition in Assumption 6.2.1 is satisfied by SDEs with additive noise, diagonal noise in case of $d = m$, and linear noise [67]. Under this assumption, because the Lévy areas are anti-symmetric, $A^\ell_{jn,k} = -A^\ell_{nj,k}$, it follows that

$$h_{ijn}(X^\ell_k)A^\ell_{jn,k} + h_{inj}(X^\ell_k)A^\ell_{nj,k} = 0.$$

This means that the terms in (2.4) involving the Lévy areas cancel out and it is not necessary to simulate them. The ith component of the Milstein scheme can then be written as

$$X^\ell_{i,k+1} = X^\ell_{i,k} + f_i(X^\ell_k)\Delta t_\ell + \sum_{j=1}^{m} g_{ij}(X^\ell_k)\Delta w^\ell_{j,k} + \sum_{j,n=1}^{m} h_{ijn}(X^\ell_k)\left(\Delta w^\ell_{j,k}\Delta w^\ell_{n,k} - \Omega_{jn}\Delta t_\ell\right). \tag{6.1}$$

We now recall some classical results from differential geometry, which can be found, for example, in [36, pp.381-384]. For $x \in \partial B$, we denote by $n(x)$ the unit inward normal vector at x ($\|n(x)\| = 1$ for $x \in \partial B$.) We call a function $x \longrightarrow \pi_B(x)$ a projection of x on the boundary ∂B. The signed normal distance of x to ∂B is a function $x \longrightarrow d_B(x)$, such that

$$d_B(x) = \begin{cases} d(x, \partial B), & x \in B \\ 0, & x \in \partial B \\ -d(x, \partial B), & x \notin B, \end{cases}$$

where $d(x, \partial B)$ is the Euclidean distance between $x \in \mathbb{R}^d$ and the boundary ∂B. Here $d_B : \mathbb{R}^d \to \mathbb{R}^1$ is a Lipschitz continuous function with gradient 1 and Hessian matrix equal to the zero matrix [36]. Also, d_B is a C^∞ function on a closed neighborhood of the boundary and can be easily extended to a C^∞ function on $\mathbb{R}^d$ with bounded derivatives [51].

Thanks to the regularity of the boundary condition in Assumption 6.2.1, for

any x, there are unique $\pi_B(x) \in \partial B$ and $d_B(x) \in \mathbb{R}$ such that

$$x = \pi_B(x) + d_B(x)n(\pi_B(x)).$$

The C^2 continuity condition in Assumption 6.2.1 allows us to use the classical finite time strong convergence result for the Milstein scheme (as in (2.7).) The following Lemma is standard and therefore we omit the proof [81].

Lemma 6.2.2. *Let Assumption 6.2.1 hold. There exist a constant c_q, independent of Δt_ℓ, such that*

$$\left(\mathbb{E}\left[\max_{0\le k\le 2^\ell} \left|X_{k+1}^\ell - X_k^\ell\right|^q\right]\right)^{1/q} \le c_q \left|\Delta t_\ell \log \frac{1}{\Delta t_\ell}\right|^{1/2}, \quad 1 \le q < \infty,$$

and

$$\left(\mathbb{E}\left[\sup_{|t-s|\le \Delta t_\ell} |x(s) - x(t)|^q\right]\right)^{1/q} \le c_q \left|\Delta t_\ell \log \frac{1}{\Delta t_\ell}\right|^{1/2}, \quad 1 \le q < \infty.$$

When sampling directly from a discrete process, there is a positive probability that the process has left the domain B even though $X_k^\ell \in B$ and $X_{k+1}^\ell \in B$. Using a Brownian bridge technique, we can calculate the conditional probability of the process leaving the domain B conditional on the points X_k^ℓ and X_{k+1}^ℓ.

First, let us define the Brownian bridge interpolation with the ith component of the form,

$$\widetilde{X}_i(t) = X_{i,k}^\ell + \lambda_\ell \left(X_{i,k+1}^\ell - X_{i,k}^\ell\right) + \sum_{j=1}^m g_{ij}(X_k^\ell)\left(w(t) - w_{j,k}^\ell - \lambda_\ell \left(w_{j,k+1}^\ell - w_{j,k}^\ell\right)\right), \tag{6.2}$$

where $\lambda_\ell = \frac{t-k\Delta t_\ell}{\Delta t_\ell}$. We now state Proposition 6.2.3 (a proof can be found, for example, in [69, Lemma 2].) We note that the formula from Proposition 6.2.3 is used in [49, equation (11)] and [50] for the simulation of killed diffusions in an n-dimensional half-space.

Proposition 6.2.3. *Let Assumption 6.2.1 hold. For the half-space B, we have an explicit formula for calculating probability of an exit of the Brownian bridge interpolation, fixed at points $X_k^\ell \in B$ and $X_{k+1}^\ell \in B$, which is given by*

$$\mathbb{P}\left(\inf_{k\Delta t_\ell \le t \le (k+1)\Delta t_\ell} d_B(\widetilde{X}(t)) \le 0 \mid X_k^\ell, X_{k+1}^\ell\right) = \exp\left(-2\frac{d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|g(X_k^\ell)n(\pi_B(X_k^\ell))\right\|^2 \Delta t_\ell}\right).$$

This result is true for half-spaces. For more general domains, there is no explicit formula for the conditional probability. It can, however, be accurately approximated using an asymptotic expansion in Δt_ℓ [8] or by locally approximating the domain by an appropriate half-space [50].

Let us introduce the following notation for the conditional probability of the exit of the Brownian bridge interpolation

$$p_k^\ell \equiv \begin{cases} 1, & \text{if } X_k^\ell \text{ or } X_{k+1}^\ell \text{ are outside B,} \\ \exp\left(\frac{-2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\|g(X_k^\ell)n(\pi_B(X_k^\ell))\|^2\Delta t_\ell}\right), & \text{otherwise.} \end{cases} \tag{6.3}$$

We can express the expected bounded exit time of the SDEs (2.1) from the half-space B as

$$\mathbb{E}[\tau] = \int_0^T \mathbb{P}_\tau(\tau > t)dt,$$

where

$$\tau \equiv \inf\{t \geq 0 : d_B(x(t)) \leq 0\} \wedge T.$$

Since the process $\{x(t)\}_{0\leq t\leq T}$ is Markovian

$$\begin{aligned} \mathbb{E}[\tau] &= \mathbb{E}[\mathbb{E}[\tau \mid x(0), x(\Delta t_\ell), \ldots, x(T)]] \\ &= \mathbb{E}\left[\int_0^T \mathbb{P}_\tau(\tau > t \mid x(0), x(\Delta t_\ell), \ldots, x(T))dt\right] \\ &= \mathbb{E}\left[\sum_{k=0}^{N-1} \int_{k\Delta t_\ell}^{(k+1)\Delta t_\ell} \mathbb{P}_\tau(\tau > t \mid x(k\Delta t_\ell), x((k+1)\Delta t_\ell))dt\right]. \end{aligned}$$

Using the trapezoidal integration rule on this formulation we find

$$\mathbb{E}[\mathbb{E}[\tau \mid x(0), x(\Delta t_\ell), \ldots, x(T)]] \approx \mathbb{E}\left[\sum_{k=0}^{N-1} \frac{1}{2}\Delta t_\ell \left(\widehat{\Pi}_k^\ell + \widehat{\Pi}_{k+1}^\ell\right)\right], \tag{6.4}$$

where $\widehat{\Pi}_k^\ell = \mathbb{P}_\tau(\tau > k\Delta t_\ell \mid x(k\Delta t_\ell), x((k+1)\Delta t_\ell))$ is the conditional probability of the process not leaving the domain B until time $k\Delta t_\ell$. Denoting by $\tau_{\Delta t_\ell}$ an approximation of τ using the Milstein scheme (6.1) we take

$$\mathbb{E}[\mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]] = \mathbb{E}\left[\sum_{k=0}^{N-1} \frac{1}{2}\Delta t_\ell \left(\Pi_k^\ell + \Pi_{k+1}^\ell\right)\right], \tag{6.5}$$

where

$$\Pi_k^\ell = \prod_{i=0}^{k-1}(1 - p_i^\ell),$$

with p_i^ℓ defined in (6.3). The next part of this chapter is devoted to proving convergence of (6.5) to $\mathbb{E}[\tau]$.

6.3 New Algorithm and Multi-level Monte Carlo

Multi-level Monte Carlo simulations use a number of different levels of resolution, $0 \le \ell \le L$. Here, $\ell = 0$ is the coarsest level and $\ell = L$ is the finest. For a range of different stepsizes of the form $\Delta t_\ell = M^{-\ell}T$, the smallest stepsize, Δt_L, is chosen so that the bias in the discretisation method matches a user-specified root mean square accuracy of $\mathcal{O}(\mathrm{TOL})$. In this chapter we take $M = 2$.

For clarity of the exposition we will express the coarse timestep approximation at each level in terms of the fine time-step, that is $\mathcal{P}^f_{\Delta t_\ell} := \{k\Delta t_\ell : k = 0, 1, 2, \ldots, 2^\ell\}$. The partition for the coarse approximation is given by $\mathcal{P}^c_{\Delta t_{\ell-1}} := \{k\Delta t_\ell : k = 0, 2, 4, \ldots, 2^\ell\}$. Hence, $X_k^{\ell-1}$ is associated with X_k^ℓ for $k = 0, 2, 4, \ldots, 2^\ell$. The multi-level idea is based on a straightforward telescoping identity, which in our context could be written

$$\mathbb{E}[\tau_L] = \mathbb{E}[\tau_0] + \sum_{\ell=1}^{L} \mathbb{E}[\tau_\ell - \tau_{\ell-1}].$$

In practice, following the approach in [37], we find it useful to use different estimators $\mathbb{E}[\tau_\ell^f]$ and $\mathbb{E}[\tau_\ell^c]$ on fine and coarse meshes, respectively, provided

$$\mathbb{E}[\tau_\ell^f] = \mathbb{E}[\tau_\ell^c], \quad \ell = 1, \ldots, L. \tag{6.6}$$

Condition (6.6) allows to exploit identity

$$\mathbb{E}[\tau_L^f] = \mathbb{E}[\tau_0^f] + \sum_{\ell=1}^{L} \mathbb{E}[\tau_\ell^f - \tau_{\ell-1}^c], \tag{6.7}$$

in the same way as for classical MLMC. On the left we have the finest resolution approximation with the required bias. On the right is a telescoping summation involving different levels of resolution. We independently estimate each of the

expectations on the right-hand side in (6.7) in a way which minimises the overall variance.

We continue by introducing notation for exit times of the fine and coarse paths. Let us denote by $\tau^f_{\Delta t_\ell}$ the exit time of the new algorithm approximating the fine path over the time interval $[0,T]$. The form of the estimator (6.5) allows to focus our attention on the conditional expectation itself. On the fine level we then obtain

$$\mathbb{E}[\tau^f_{\Delta t_\ell} \mid X^\ell_0, X^\ell_1, \dots, X^\ell_{2^\ell}] = \sum_{k=0}^{N-1} \frac{1}{2}\Delta t_\ell \left(\Pi^{\ell,f}_k + \Pi^{\ell,f}_{k+1}\right). \tag{6.8}$$

We could define the estimator for the coarse level in a similar fashion but in order to couple the fine and coarse level better and, as a consequence, to reduce variance even further, for k-even we introduce an additional middle point $\widetilde{X}^{\ell-1}_{k+1}$ for the coarse approximation using equation (6.2), at a time $(k+1)\Delta t_\ell$, with the ith component of the form

$$\widetilde{X}^{\ell-1}_{i,k+1} = \frac{1}{2}\left(X^{\ell-1}_{i,k} + X^{\ell-1}_{i,k+2} - \sum_{j=1}^{m} g_{ij}(X^{\ell-1}_k)\left((w^\ell_{j,k+2} - w^\ell_{j,k+1}) - (w^\ell_{j,k+1} - w^\ell_{j,k})\right)\right). \tag{6.9}$$

This way we have added $N/2$ simulation points on the coarse path and as a result we obtain a new estimator on the coarse level

$$\mathbb{E}[\tau^c_{\Delta t_\ell} \mid X^{\ell-1}_0, \widetilde{X}^{\ell-1}_1, X^{\ell-1}_2, \dots, X^{\ell-1}_{2^\ell}] = \sum_{\substack{k=0 \\ k-\text{even}}}^{N-2} \Delta t_\ell \left(\Pi^{\ell-1,c}_k + \Pi^{\ell-1,c}_{k+2}\right), \tag{6.10}$$

where $\Pi^{\ell-1,c}_k$ and $p^{\ell-1}_k$ are probabilities for which computation involves referring to the interpolated point. Because the increments $(w^\ell_{k+2} - w^\ell_{k+1})$ and $(w^\ell_{k+1} - w^\ell_k)$ are mutually independent we have

$$\Pi^{\ell-1,c}_k = \prod_{i=0}^{k-1}(1 - p^{\ell-1}_i) \equiv \prod_{i=0}^{k-1}(1 - p^{\ell-1}_{i,1})(1 - p^{\ell-1}_{i,2}),$$

where $(1 - p^{\ell-1}_{i,1})$ and $(1 - p^{\ell-1}_{i,2})$ are the probabilities of not leaving the domain on

the first and second half of the coarse interval, respectively, and they satisfy

$$p_{i,1}^{\ell-1} \equiv \begin{cases} 1, & \text{if } X_k^{\ell-1} \text{ or } \widetilde{X}_{k+1}^{\ell-1} \text{ are outside B,} \\ \exp\left(\frac{-2d_B(X_k^{\ell-1})d_B(\widetilde{X}_{k+1}^{\ell-1})}{\|g(X_k^{\ell-1})n(\pi_B(X_k^{\ell-1}))\|^2\Delta t_\ell}\right), & \text{otherwise.} \end{cases}$$
$$p_{i,2}^{\ell-1} \equiv \begin{cases} 1, & \text{if } \widetilde{X}_{k+1}^{\ell-1} \text{ or } X_{k+2}^{\ell-1} \text{ are outside B,} \\ \exp\left(\frac{-2d_B(\widetilde{X}_{k+1}^{\ell-1})d_B(X_{k+2}^{\ell-1})}{\|g(X_k^{\ell-1})n(\pi_B(X_k^{\ell-1}))\|^2\Delta t_\ell}\right), & \text{otherwise.} \end{cases} \tag{6.11}$$

It is crucial that we use $g(X_k^{\ell-1})$ on both timesteps, having used the Brownian bridge with the diffusion term $g(X_k^{\ell-1})$ to derive both minima. The key equality (6.6) would have been violated if we changed $g(X_k^{\ell-1})$ to $g(\widetilde{X}_{k+1}^{\ell-1})$ in calculating the probability $p_{i,2}^{\ell-1}$, as we would have used different Brownian bridges on the first and second half of the coarse timestep. When generating Brownian increments for the fine sample path, we use $\left(w_{k+1}^\ell - w_k^\ell\right)$ as the increment for the first fine timestep and $\left(w_{k+2}^\ell - w_{k+1}^\ell\right)$ for the second one. We then build the increment for the coarse path by summing the Brownian increments from two fine timesteps, i.e., $\left(w_{k+1}^\ell - w_k^\ell\right) + \left(w_{k+2}^\ell - w_{k+1}^\ell\right)$ is the increment of the coarse timestep. Note that in (6.9) the increments $\left(w_{k+2}^\ell - w_{k+1}^\ell\right) - \left(w_{k+1}^\ell - w_k^\ell\right)$ are independent of the Brownian increments used to generate the coarse sample path, $\left(w_{k+2}^\ell - w_{k+1}^\ell\right) + \left(w_{k+1}^\ell - w_k^\ell\right)$. Hence, when implementing the algorithm we can reuse the pseudo-random numbers $\left(w_{k+2}^\ell - w_{k+1}^\ell\right)$ and $\left(w_{k+1}^\ell - w_k^\ell\right)$ to calculate extra points on the coarse sample path.

Clearly, we have not changed the expectation by introducing the Brownian bridge interpolation in the coarse timestep estimator and the equality (6.6) is satisfied. Indeed, we have

$$\mathbb{E}[\tau_{\Delta t_\ell}^c] = \mathbb{E}[\mathbb{E}[\tau_{\Delta t_\ell}^c \mid X_0^\ell, X_2^{\ell-1}, \ldots, X_{2^\ell}^{\ell-1}]] = \mathbb{E}[\mathbb{E}[\tau_{\Delta t_\ell}^c \mid X_0^{\ell-1}, \widetilde{X}_1^{\ell-1}, X_2^{\ell-1}, \ldots, X_{2^\ell}^{\ell-1}]].$$

We now show that the interpolated point on the coarse path is close to the point from the fine approximation.

Lemma 6.3.1. *Let Assumption 6.2.1 hold. There exists a constant c, independent of Δt_ℓ, such that*

$$\left(\mathbb{E}\left[\sup_{k=0,2,\ldots,2^{\ell-1}-2} \left\|X_{k+1}^\ell - \widetilde{X}_{k+1}^{\ell-1}\right\|^q\right]\right)^{1/q} \leq c\,|\Delta t_\ell \log \frac{1}{\Delta t_\ell}|, \quad 1 \leq q < \infty.$$

Proof. From the definition of the Milstein scheme (6.1), by combining $X_{i,k}^{\ell-1}$ and $X_{i,k+2}^{\ell-1}$

$$X_{i,k+1}^{\ell-1} = \frac{1}{2}\left(X_{i,k}^{\ell-1} + X_{i,k+2}^{\ell-1} - \sum_{j=1}^{m} g_{ij}(X_k^{\ell-1})(\Delta w_{j,k+1}^{\ell-1} - \Delta w_{j,k}^{\ell-1})\right) + R_{k+1}, \tag{6.12}$$

where

$$\begin{aligned} R_{k+1} =& (f_i(X_k^{\ell-1}) - f_i(X_{k+1}^{\ell-1}))\Delta t_{\ell-1} + \sum_{j=1}^{m}(g_{ij}(X_k^{\ell-1})) - g_{ij}(X_{k+1}^{\ell-1}))\Delta w_{j,k+1}^{\ell-1} \\ &- \sum_{j,n=1}^{m} h_{ijn}(X_{k+1}^{\ell-1})\left(\Delta w_{j,k+1}^{\ell-1}\Delta w_{n,k+1}^{\ell-1} - \Omega_{jn}\Delta t_{\ell-1}\right) \\ &+ \sum_{j,n=1}^{m} h_{ijn}(X_k^{\ell-1})\left(\Delta w_{j,k}^{\ell-1}\Delta w_{n,k}^{\ell-1} - \Omega_{jn}\Delta t_{\ell-1}\right). \end{aligned}$$

Subtracting (6.9) from (6.12), using the Lipschitz continuity of $f(\cdot)$ and $g(\cdot)$ and applying Lemma 6.2.2 completes this proof. □

Thanks to Lemma 6.3.1 we will now write $X_{k+1}^{\ell-1} \equiv \widetilde{X}_{k+1}^{\ell-1}$ for $k = 0, 2, \ldots, 2^\ell - 2$ without introducing ambiguity.

In order to justify the multi-level Monte Carlo approach we establish a rate of strong convergence for the exit time approximation of fine and coarse paths. The result is spread over Theorem 6.3.2 and Theorem 6.3.8. We first state a universal, scheme indifferent, theorem in which for some mesh we bound the difference between conditional expectations of exit times for high and low resolution paths by the difference between probabilities of not leaving the domain on these paths. We then work towards Theorem 6.3.8 in which a rate of strong convergence of almost 3/2 is achieved.

Theorem 6.3.2. *Let Assumption 6.2.1 hold. For some constant c we have*

$$\begin{aligned} &\mathbb{E}\left[\left(\mathbb{E}[\tau_{\Delta t_\ell}^f \mid X_0^\ell, \ldots, X_{2^\ell}^\ell] - \mathbb{E}[\tau_{\Delta t_\ell}^c \mid X_0^{\ell-1}, \ldots, X_{2^\ell}^{\ell-1}]\right)^2\right] \\ &\qquad \leq c\left(\max_k \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2\right] + \Delta t_\ell^2\right). \end{aligned}$$

Proof. Using equations (6.8) and (6.10), we have

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathbb{E}[\tau^f_{\Delta t_\ell} \mid X^\ell_0,\dots,X^\ell_{2^\ell}] - \mathbb{E}[\tau^c_{\Delta t_\ell} \mid X^{\ell-1}_0,\dots,X^{\ell-1}_{2^\ell}]\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{2}\Delta t_\ell \sum_{k=0}^{N-1}\left(\Pi^{\ell,f}_k + \Pi^{\ell,f}_{k+1}\right) - \Delta t_\ell \sum_{\substack{k=0\\ k-\text{even}}}^{N-2}\left(\Pi^{\ell-1,c}_k + \Pi^{\ell-1,c}_{k+2}\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{2}\Delta t_\ell \sum_{k=0}^{N-1}\left[\left(\Pi^{\ell,f}_k - \Pi^{\ell-1,c}_k\right) + \left(\Pi^{\ell,f}_{k+2} - \Pi^{\ell-1,c}_{k+2}\right)\right]\right. \right. \\
&\quad \left.\left. - \frac{1}{2}\Delta t_\ell \sum_{\substack{k=0\\ k-\text{even}}}^{N-2}\left(\Pi^{\ell-1,c}_k - 2\Pi^{\ell-1,c}_{k+1} + \Pi^{\ell-1,c}_{k+2}\right)\right)^2\right].
\end{aligned} \tag{6.13}$$

Due to the fact that Π_k is a non-increasing function of k we have

$$-(\Pi^{\ell-1,c}_k - \Pi^{\ell-1,c}_{k+2}) \le \Pi^{\ell-1,c}_k - 2\Pi^{\ell-1,c}_{k+1} + \Pi^{\ell-1,c}_{k+2} \le \Pi^{\ell-1,c}_k - \Pi^{\ell-1,c}_{k+2}, \quad k \ge 0.$$

Hence

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathbb{E}[\tau^f_{\Delta t_\ell} \mid X^\ell_0,\dots,X^\ell_{2^\ell}] - \mathbb{E}[\tau^c_{\Delta t_\ell} \mid X^{\ell-1}_0,\dots,X^{\ell-1}_{2^\ell}]\right)^2\right] \\
&\le \frac{1}{2}\Delta t^2_\ell\left(\mathbb{E}\left(\sum_{k=0}^{N-1}\left(\Pi^{\ell,f}_k - \Pi^{\ell-1,c}_k\right) + \left(\Pi^{\ell,f}_{k+2} - \Pi^{\ell-1,c}_{k+2}\right)\right)^2 + \mathbb{E}\left(\sum_{\substack{k=0\\ k-\text{even}}}^{N-2}\Pi^{\ell-1,c}_k - \Pi^{\ell-1,c}_{k+2}\right)^2\right) \\
&\le \frac{1}{2}\Delta t^2_\ell\left(2N\mathbb{E}\left(\sum_{k=0}^{N-1}\left(\Pi^{\ell,f}_k - \Pi^{\ell-1,c}_k\right)^2 + \left(\Pi^{\ell,f}_{k+2} - \Pi^{\ell-1,c}_{k+2}\right)^2\right) + \mathbb{E}\left(\Pi^{\ell-1,c}_0 - \Pi^{\ell-1,c}_N\right)^2\right) \\
&\le 2T^2 \max_k \mathbb{E}\left[\left|\Pi^{\ell,f}_k - \Pi^{\ell-1,c}_k\right|^2\right] + \frac{1}{2}\Delta t^2_\ell.
\end{aligned} \tag{6.14}$$

□

We will now outline the main idea of proof of the rate of convergence of the MLMC variance of the mean exit time estimator. Let us recall that

$$p^\ell_k \equiv \begin{cases} 1, & \text{if } X^\ell_k \text{ or } X^\ell_{k+1} \text{ are outside B,} \\ \exp\left(\frac{-2d_B(X^\ell_k)d_B(X^\ell_{k+1})}{\|g(X^\ell_k)n(\pi_B(X^\ell_k))\|^2\Delta t_\ell}\right), & \text{otherwise.} \end{cases}$$

If $d_B(X_k^\ell) > \Delta t_\ell^{1/2-\varepsilon_1}$ and $d_B(X_{k+1}^\ell) > \Delta t_\ell^{1/2-\varepsilon_2}$, then

$$0 \le p_k^\ell \le \exp\left(\frac{-2\Delta t_\ell^{1-\varepsilon_1-\varepsilon_2}}{\left\|g(X_k^\ell)n(\pi_B(X_k^\ell))\right\|^2 \Delta t_\ell}\right) \to 0, \quad \text{when } \Delta t_\ell \to 0. \tag{6.15}$$

That is, p_k^ℓ tends to 0 exponentially fast. When $d_B(X_k^\ell) < \Delta t_\ell^{1/2-\varepsilon_1}$ and $d_B(X_{k+1}^\ell) < \Delta t_\ell^{1/2-\varepsilon_2}$, then

$$0 \leftarrow \exp\left(\frac{-2\Delta t_\ell^{1-\varepsilon_1-\varepsilon_2}}{(\left\|g(X_k^\ell)n(\pi_B(X_k^\ell))\right\|^2 \Delta t_\ell}\right) \le p_k^\ell \le 1, \quad \text{when } \Delta t_\ell \to 0.$$

So if X_k^ℓ and X_{k+1}^ℓ are within a $\Delta t_\ell^{1/2-\varepsilon}$-neighborhood of the boundary, the probability of leaving the set B could be anything between 0 and 1. This behaviour of the function p_k^ℓ indicates that the main difficulty in proving Theorem 6.3.8 is in the case where processes under consideration are close to the boundary. For this reason we split the probability space. Let $\varepsilon > 0$. We introduce the following event $\Omega = \Omega^1 \cup \Omega^2$, where

$$\Omega^1 = \{\omega : \inf_{0\le t\le T} d_B(x(t)) > \Delta t_\ell^{1/2-\varepsilon}\},$$
$$\Omega^2 = \{\omega : \inf_{0\le t\le T} d_B(x(t)) < -\Delta t_\ell^{1/2-\varepsilon}\},$$

and hence

$$\Omega^c = \{\omega : \inf_{0\le t\le T} |d_B(x(t))| \le \Delta t_\ell^{1/2-\varepsilon}\}.$$

We then write

$$\mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2\right] = \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_\Omega\right] + \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^c}\right].$$

The first summand on the right-hand side of the above inequality converges to 0 exponentially fast. For the second summand we show in Theorem 6.3.8 that on the set Ω^c we have

$$\mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2\right] = \mathcal{O}(\Delta t_\ell^{1-\varepsilon}),$$

but if we chose Ω^c such that $\mathbb{P}(\Omega^c) = \mathcal{O}(\Delta t_\ell^{1/2-\varepsilon})$, then the overall multi-level variance would be of order $\mathcal{O}(\Delta t_\ell^{3/2-\varepsilon})$. This is exactly what we are pursuing here.

For this reason we introduce a new assumption which bounds the probability of the solution of the SDE entering a neighborhood of the boundary ∂B for any time $0 \le t \le T$.

Assumption 6.3.3. *There exists a constant c such that the probability of the SDE solution $\{x(t)\}_{0\le t\le T}$ being within the neighborhood of the set ∂B has a bound*

$$\mathbb{P}\left(\inf_{0\le t\le T}|d_B(x(t))| \le \epsilon\right) \le c\ \epsilon, \quad \text{for all } \epsilon > 0.$$

In a one-dimensional context, Assumption 6.3.3 corresponds to an assumption of a locally bounded density of $\inf_{0\le t\le T} x(t)$.

The next two results, Lemma 6.3.4 and Theorem 6.3.5, tell us that we can find a set of almost full measure on which path properties of SDEs (2.1) defined by the set Ω are preserved by the numerical approximation (6.1). These results rely on the strong convergence property specified in (2.7).

Lemma 6.3.4. *Let Assumption 6.2.1 hold and let $\varepsilon > 0$. For $\eta \in (0,1)$ define $E = \bigcap_{i=1}^4 A_i$, where*

$$\begin{aligned}
A_1 &= \{\omega : \sup_{0\le k\le 2^\ell} \left\|x(k\Delta t_\ell) - X_k^\ell\right\| \le \eta \Delta t_\ell^{1-\varepsilon}\},\\
A_2 &= \{\omega : \sup_{0\le k\le 2^\ell-2} \left\|X_{k+1}^\ell - X_k^\ell\right\| \le \eta \Delta t_\ell^{1/2-\varepsilon}\},\\
A_3 &= \{\omega : \sup_{|t-s|\le \Delta t_\ell} \|x(t) - x(s)\| \le \eta \Delta t_\ell^{1/2-\varepsilon}\},\\
A_4 &= \{\omega : \sup_{0\le k\le 2^\ell} \left\|X_k^\ell - X_k^{\ell-1}\right\| \le \eta \Delta t_\ell^{1-\varepsilon}\}.
\end{aligned} \tag{6.16}$$

There exists a constant $c_{\eta,\varepsilon,p}$, independent of Δt_ℓ, such that

$$\mathbb{P}(E^c) \le c_{\eta,\varepsilon,p} \Delta t_\ell^p.$$

Proof. By subadditivity of the probability measure $\mathbb{P}(E^c) \le \sum_{i=1}^4 \mathbb{P}(A_i^c)$. Now

using Lemma 6.2.2 and the Markov inequality, we obtain

$$\begin{aligned}\mathbb{P}(A_2^c) &= \mathbb{P}(\{\omega : \sup_{0\le k\le 2^\ell-2} \left\|X_{k+1}^\ell - X_k^\ell\right\| > \eta\Delta t_\ell^{1/2-\varepsilon}\}) \\ &\le \frac{\mathbb{E}\left[\sup_{0\le k\le 2^\ell-2} \left|X_{k+1}^\ell - X_k^\ell\right|^q\right]}{\eta^q \Delta t_\ell^{q(1/2-\varepsilon)}} \\ &\le \frac{c_q \gamma^{-q}(\Delta t_\ell^{q(1/2-\gamma)} - \Delta t_\ell^{q/2})}{\eta^q \Delta t_\ell^{q(1/2-\varepsilon)}} \\ &\le c_{\eta,\gamma,q}\Delta t_\ell^{q(\varepsilon-\gamma)},\end{aligned}$$

where we used the inequality that for $x, \gamma > 0$ we have

$$\log x^{-1} \le \gamma^{-1}(x^{-\gamma} - 1).$$

Thanks to Lemma 6.2.2 we can similarly estimate probabilities of the events A_1, A_3 and A_4, and by choosing $\gamma < \varepsilon$ and q such that $q(\varepsilon - \gamma) = p$, proof is complete. □

Let us now define $\Omega^N = \Omega^{1,N} \cup \Omega^{2,N}$ such that

$$\begin{aligned}\Omega^{1,N} &= \{\omega : \inf_{0\le k\le 2^\ell} d_B(X_k^\ell) > \Delta t_{\ell-1}^{1/2-\varepsilon}\}, \\ \Omega^{2,N} &= \{\omega : \inf_{0\le k\le 2^\ell} d_B(X_k^\ell) < -\Delta t_{\ell-1}^{1/2-\varepsilon}\},\end{aligned}$$

and hence

$$\left(\Omega^N\right)^c = \{\omega : \inf_{0\le k\le 2^\ell} \left|d_B(X_k^\ell)\right| \le \Delta t_{\ell-1}^{1/2-\varepsilon}\}.$$

Theorem 6.3.5. *Let Assumption 6.2.1 hold. Let $E \subset \Omega$ be defined as in Lemma 6.3.4. The following inclusions hold:*

i) $\Omega^1 \cap E \subset \Omega^{1,N}$,

ii) $\Omega^2 \cap E \subset \Omega^{2,N}$,

iii) $\Omega^c \cap E \subset (\Omega^N)^c$.

Proof. Let $\eta \in (0,1)$ be such that $(1/2^{1/2-\varepsilon} < 1 - \eta\Delta t_\ell^{1/2}) \cap (\eta\Delta t_\ell^{1/2} + \eta - 1 \le -1/2^{1/2-\varepsilon}) \cap (\eta\Delta t_\ell^{1/2} + \eta + 1 < 2^{1/2-\varepsilon})$. First, we prove $i)$. Let $\omega \in \Omega^1 \cap E$. By the

Lipschitz continuity of the function $d_B(\cdot)$,

$$d_B(x(k\Delta t_\ell)) \le \left|d_B(x(k\Delta t_\ell)) - d_B(X_k^\ell)\right| + d_B(X_k^\ell) \le \sup_{0\le k\le 2^\ell} \left\|x(k\Delta t_\ell) - X_k^\ell\right\| + d_B(X_k^\ell),$$

and the fact that $\omega \in \Omega \cap A_1$,

$$\Delta t_\ell^{1/2-\varepsilon} < \inf_{0\le t\le T} d_B(x(t)) \le \inf_{0\le k\le 2^\ell} d_B(x(k\Delta t_\ell)) \le \eta \Delta t_\ell^{1-\varepsilon} + \inf_{0\le k\le 2^\ell} d_B(X_k^\ell),$$

we have

$$\Delta t_\ell^{1/2-\varepsilon}(1 - \eta \Delta t_\ell^{1/2}) < \inf_{0\le k\le 2^\ell} d_B(X_k^\ell).$$

We can choose η such that

$$\inf_{0\le k\le 2^\ell} d_B(X_k^\ell) > \Delta t_{\ell-1}^{1/2-\varepsilon}.$$

For the remaining part of this chapter we often perform a similar action without specifically mentioning the constant. Second, we proceed to proving $ii)$. Let $\omega \in \Omega^2 \cap E$. Then

$$d_B(X_k^\ell) \le \left|d_B(x(k\Delta t_\ell)) - d_B(X_k^\ell)\right| + d_B(x(k\Delta t_\ell)),$$

and

$$\inf_{0\le k\le N} d_B(x(k\Delta t_\ell)) \le \sup_{0\le k\le N} \sup_{k\Delta t_\ell \le t \le (k+1)\Delta t_\ell} |d_B(x(k\Delta t_\ell)) - d_B(x(t))| + \inf_{0\le t\le T} d_B(x(t)),$$

which proves that

$$\inf_{0\le k\le N} d_B(X_k^\ell) < (\eta \Delta t_\ell^{1/2} + \eta - 1)\Delta t_\ell^{1/2-\varepsilon}.$$

Our proof of implication $iii)$ is almost the same as the proof of $ii)$. Let $\omega \in \Omega^c \cap E$. Then

$$\inf_{0\le k\le N} \left|d_B(X_k^\ell)\right| \le (\eta \Delta t_\ell^{1/2} + \eta + 1)\Delta t_\ell^{1/2-\varepsilon}.$$

□

Let us consider an event $\omega \in \Omega^c \cap E = \Omega^c \cap (\Omega^N)^c \cap E$. Therefore, we know that there exist k^* and s^*, $0 \le k^*, s^* \le N$ such that $\left|d_B(X_{k^*}^\ell)\right| < \Delta t_\ell^{1/2-\varepsilon}$ and

$|d_B(X^\ell_{s^*})| < \Delta t_\ell^{1/2-\varepsilon}$. The following Lemma tells us that if the fine approximation is in a $\Delta t_\ell^{1/2-\varepsilon}$-neighborhood of the boundary at a time $\Delta t_\ell k^*$, then the coarse approximation at the same time $\Delta t_\ell k^*$ is also in the $\Delta t_\ell^{1/2-\varepsilon}$-neighborhood of the boundary. We use convention that $\Delta t_{-1} = 2\ \Delta t_0$.

Lemma 6.3.6. *Let Assumption 6.2.1 hold. For $\omega \in \Omega^c \cap E$, the following inclusions hold:*

i) $\{k : \left|d_B(X^\ell_k)\right| < \Delta t_\ell^{1/2-\varepsilon}\} \subset \{k : \left|d_B(X^{\ell-1}_k)\right| < \Delta t_{\ell-1}^{1/2-\varepsilon}\}$,

ii) $\{k : \left|d_B(X^{\ell-1}_k)\right| < \Delta t_{\ell-1}^{1/2-\varepsilon}\} \subset \{k : \left|d_B(X^{\ell}_k)\right| < \Delta t_{\ell-2}^{1/2-\varepsilon}\}$,

iii) $\{k : \left|d_B(X^\ell_k)\right| < \Delta t_\ell^{1/2-\varepsilon}\} \subset \{k : \left|d_B(X^{\ell}_{k+1})\right| < \Delta t_{\ell-1}^{1/2-\varepsilon}\}$.

Proof. Let us choose $\eta \in (0,1)$ as in proof of Theorem 6.3.5. Let $\omega \in \Omega^c \cap E$. From

$$\left|d_B(X^{\ell-1}_k)\right| \le \left|d_B(X^{\ell-1}_k) - d_B(X^\ell_k)\right| + \left|d_B(X^\ell_k)\right|,$$

and the fact that $\omega \in \Omega^c \cap A_4$ we have

$$\left|d_B(X^{\ell-1}_k)\right| \le (\eta \Delta t_\ell^{1/2} + 1)\Delta t_\ell^{1/2-\varepsilon}.$$

Inclusion *ii*) follows by the same argument. Regarding inclusion *iii*) we have

$$\left|d_B(X^{\ell}_{k+1})\right| \le \left|d_B(X^{\ell}_{k+1}) - d_B(X^\ell_k)\right| + \left|d_B(X^\ell_k)\right|.$$

From the fact that $\omega \in \Omega^c \cap A_2$ we conclude

$$\left|d_B(X^{\ell}_{k+1})\right| \le (\eta + 1)\Delta t_\ell^{1/2-\varepsilon},$$

by selecting the constant η appropriately. □

We are now close to proving the main result of this chapter. In the remaining analysis we often use the following identity which we present here for the reader's convenience, namely

$$x_1y_1 - x_2y_2 = \frac{1}{2}(y_1+y_2)(x_1-x_2) + \frac{1}{2}(x_1+x_2)(y_1-y_2), \quad \text{for all } x_1, y_1, x_2, y_2. \tag{6.17}$$

In order to obtain the following pathwise inequality we require the strong order of convergence to be one. This is in fact the only theorem where the higher order strong convergence is needed in order to obtain the results from this chapter. This also suggests that our result cannot be generalised to the Euler-Maruyama method. From Lemma 6.3.6 we can find the set of indexes α for which both fine and coarse approximations are within the $\Delta t_{\ell-1}^{1/2-\varepsilon}$-neighborhood of the boundary and the set of indexes $\alpha^c = \{0, 1, \dots, 2^\ell\} \setminus \alpha$, for which both approximations are at least $\Delta t_{\ell-1}^{1/2-\varepsilon}$ far from the boundary. For convenience let us define $G_k^{\ell,f} \equiv g(X_k^\ell)n(\pi_B(X_k^\ell))$ for $k = 0, 1, \dots, 2^\ell$. Due to the Brownian bridge interpolation on the coarse level we have

$$G_k^{\ell-1,c} \equiv \begin{cases} g(X_k^{\ell-1})n(\pi_B(X_k^{\ell-1})), & \text{for } k = 0, 2, \dots, 2^\ell, \\ g(X_{k-1}^{\ell-1})n(\pi_B(X_{k-1}^{\ell-1})), & \text{for } k = 1, 3, \dots, 2^\ell + 1. \end{cases}$$

Theorem 6.3.7. *Let Assumption 6.2.1 hold and $2\Delta t_0 \le 1$. For $\omega \in (\Omega^N)^c \cap E$ we have the following pathwise inequalities*

$$\sup_{k\in\alpha} \left| \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|G_k^{\ell,f}\right\|^2 \Delta t_\ell} - \frac{2d_B(X_k^{\ell-1})d_B(X_{k+1}^{\ell-1})}{\left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} \right| \le \Delta t_{\ell-2}^{1/2-3\varepsilon}, \tag{6.18}$$

and

$$\left| \prod_{s\in\alpha}(1 - p_s^{\ell-1}) - \prod_{s\in\alpha}(1 - p_s^\ell) \right| \le \Delta t_{\ell-2}^{1/2-3\varepsilon}. \tag{6.19}$$

Proof. Let $\omega \in (\Omega^N)^c \cap E$. We obtain

$$\begin{aligned} &\left| \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|G_k^{\ell,f}\right\|^2 \Delta t_\ell} - \frac{2d_B(X_k^{\ell-1})d_B(X_{k+1}^{\ell-1})}{\left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} \right| \\ &= \left| \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|G_k^{\ell,f}\right\|^2 \Delta t_\ell} - \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} \right. \\ &\quad \left. + \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)}{\left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} - \frac{2d_B(X_k^{\ell-1})d_B(X_{k+1}^{\ell-1})}{\left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} \right| \\ &= \left| \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell)\left(G_k^{\ell-1,c} - G_k^{\ell,f}\right)^T\left(G_k^{\ell-1,c} + G_k^{\ell,f}\right)}{\left\|G_k^{\ell,f}\right\|^2 \left\|G_k^{\ell-1,c}\right\|^2 \Delta t_\ell} \right. \\ &\quad \left. + \frac{2d_B(X_k^\ell)d_B(X_{k+1}^\ell) - 2d_B(X_k^{\ell-1})d_B(X_{k+1}^{\ell-1})}{\left\|G_k^{\ell-1,c}\right\| \Delta t_\ell} \right|. \end{aligned} \tag{6.20}$$

We first focus our attention on bounding the first term on the right-hand side in (6.20). By Assumption 6.2.1 and the fact that $\omega \in (\Omega^N)^c \cap E$

$$\left\| G_k^{\ell-1,c} - G_k^{\ell,f} \right\| \leq c\,\eta\,\Delta t_\ell^{1-\varepsilon}, \quad \text{for } k = 0, 2, \ldots, 2^\ell. \tag{6.21}$$

For $k = 1, 3, \ldots, 2^\ell + 1$, again by Assumption 6.2.1 and the fact that $\omega \in (\Omega^N)^c \cap E$, we have

$$\begin{aligned}\left\| G_k^{\ell-1,c} - G_k^{\ell,f} \right\| &\leq \left\| g(X_{k-1}^{\ell-1}) n(\pi_B(X_{k-1}^{\ell-1})) - g(X_{k-1}^{\ell}) n(\pi_B(X_{k-1}^{\ell})) \right\| \\ &\quad + \left\| g(X_{k-1}^{\ell}) n(\pi_B(X_{k-1}^{\ell})) - g(X_k^{\ell}) n(\pi_B(X_k^{\ell})) \right\| \\ &\leq c\,\eta\,\Delta t_{\ell-1}^{1/2-\varepsilon} + c\,\eta\,\Delta t_\ell^{1-\varepsilon}.\end{aligned}$$

We then can find $\eta \in (0, 1)$ such that

$$\sup_{k \in \alpha} \left| \frac{2|d_B(X_k^\ell)| \cdot |d_B(X_{k+1}^\ell)| \left\| G_k^{\ell-1,c} - G_k^{\ell,f} \right\| \left\| G_k^{\ell-1,c} + G_k^{\ell,f} \right\|}{\left\| G_k^{\ell,f} \right\|^2 \left\| G_k^{\ell-1,c} \right\|^2 \Delta t_\ell} \right| \leq \Delta t_{\ell-2}^{1/2-3\varepsilon}. \tag{6.22}$$

For the second term on the right-hand side in (6.20) we use the inequality (6.17) and obtain

$$\begin{aligned}&\sup_{k \in \alpha} \left| \frac{2 d_B(X_k^\ell) d_B(X_{k+1}^\ell) - 2 d_B(X_k^{\ell-1}) d_B(X_{k+1}^{\ell-1})}{\left\| G_k^{\ell-1,c} \right\| \Delta t_\ell} \right| \\ &= \sup_{k \in \alpha} \left| \frac{(d_B(X_{k+1}^\ell) + d_B(X_{k+1}^{\ell-1}))(d_B(X_k^\ell) - d_B(X_k^{\ell-1}))}{\left\| G_k^{\ell-1,c} \right\| \Delta t_\ell} \right. \\ &\quad \left. + \frac{(d_B(X_k^\ell) + d_B(X_k^{\ell-1}))(d_B(X_{k+1}^\ell) - d_B(X_{k+1}^{\ell-1}))}{\left\| G_k^{\ell-1,c} \right\| \Delta t_\ell} \right| \\ &\leq c\,\eta \Delta t_\ell^{1/2-2\varepsilon}.\end{aligned} \tag{6.23}$$

From (6.22) and (6.23) we conclude that (6.18) in Theorem 6.3.7 holds.

We now focus our attention on proving (6.19). From the inequality

$$e^x - e^y \leq e^x (x - y), \quad \text{for } x \geq y,$$

and the pathwise inequality (6.18) we deduce that

$$1 - p_k^{\ell-1} \leq (1 - p_k^\ell) + p_k^\ell \Delta t_{\ell-2}^{1/2-3\varepsilon}.$$

Let us define a function

$$h(\Delta t_{\ell-2}) \equiv \prod_{s\in\alpha}\left((1-p_s^\ell)+p_s^\ell\Delta t_{\ell-2}^{1/2-3\varepsilon}\right)-\prod_{s\in\alpha}\left(1-p_s^\ell\right)-\Delta t_{\ell-2}^{1/2-3\varepsilon}.$$

Note that this function has the following properties:

- it is convex,
- $h(0)=0$,
- $h(1)=-\prod_{s\in\alpha}(1-p_s^\ell)\le 0$.

This allows us to conclude that for $\Delta t_{\ell-2}\in[0,1]$, we have

$$h(\Delta t_{\ell-2})\le 0.$$

As a consequence

$$\prod_{s\in\alpha}\left((1-p_s^\ell)+p_s^\ell\Delta t_{\ell-2}^{1/2-3\varepsilon}\right)\le\prod_{s\in\alpha}\left(1-p_s^\ell\right)+\Delta t_{\ell-2}^{1/2-3\varepsilon}.$$

This proves that

$$\prod_{s\in\alpha}(1-p_s^{\ell-1})\le\prod_{s\in\alpha}(1-p_s^\ell)+\Delta t_{\ell-2}^{1/2-3\varepsilon}.$$

By symmetry we can show that

$$\prod_{s\in\alpha}(1-p_s^\ell)\le\prod_{s\in\alpha}(1-p_s^{\ell-1})+\Delta t_{\ell-2}^{1/2-3\varepsilon},$$

and hence

$$\left|\prod_{s\in\alpha}(1-p_s^{\ell-1})-\prod_{s\in\alpha}(1-p_s^\ell)\right|\le\Delta t_{\ell-2}^{1/2-3\varepsilon}.$$

□

Theorem 6.3.8. *Let Assumptions 6.2.1, 6.3.3 hold and* $2\Delta t_0\le 1$. *Given any* $\varepsilon^*>0$, *there exists a constant* c_{ε^*}, *independent of* Δt_ℓ, *such that*

$$\mathbb{E}[(\Pi_k^{\ell,f}-\Pi_k^{\ell-1,c})^2]\le c_{\varepsilon^*}\Delta t_\ell^{3/2-\varepsilon^*}.$$

Proof. Using notation from Lemma 6.3.4 and Theorem 6.3.5, and noting that $0 \leq \Pi_k^{\ell,f}, \Pi_k^{\ell-1,c} \leq 1$, we have

$$\begin{aligned}\mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2\right] \leq \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^1 \cap E}\right] + \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^2 \cap E}\right] \\ + \mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^c \cap E}\right] + c\Delta t_\ell^p.\end{aligned} \tag{6.24}$$

Considering the first summand of the above inequality, Theorem 6.3.5 gives us $\Omega^1 \cap E \subset \Omega^{1,N} \cap E$. Therefore, from (6.15) we immediately conclude that there exists a constant c such that

$$\mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^1 \cap E}\right] \leq c\,\Delta t_\ell^p, \quad \text{for any } p > 0.$$

Similarly, for the second summand in (6.24), Theorem 6.3.5 gives us $\Omega^2 \cap E \subset \Omega^{2,N} \cap E$, and as a consequence

$$\mathbb{E}\left[(\Pi_k^{\ell,f} - \Pi_k^{\ell-1,c})^2 \mathbf{1}_{\Omega^2 \cap E}\right] = 0.$$

To complete the proof we need to deal with the third summand in (6.24) for which Theorem 6.3.5 gives us $\Omega^c \cap E = \Omega^c \cap (\Omega^N)^c \cap E$. Recalling that by α we denoted the set of indexes for which both approximations are within the $\Delta t_{\ell-1}^{1/2-\varepsilon}$-neighborhood to the boundary, we arrive at

$$\begin{aligned}&\mathbb{E}\left[\left(\prod_{s\in\alpha}(1-p_s^f)\prod_{s\in\alpha^c}(1-p_s^f) - \prod_{s\in\alpha}(1-p_s^c)\prod_{s\in\alpha^c}(1-p_s^c)\right)^2 \mathbf{1}_{\Omega^c\cap(\Omega^N)^c\cap E}\right] \\ &\quad\leq 2\mathbb{E}\left[\left(\prod_{s\in\alpha}(1-p_s^f) - \prod_{s\in\alpha}(1-p_s^c)\right)^2 \mathbf{1}_{\Omega^c\cap(\Omega^N)^c\cap E}\right] + c\Delta t_\ell^p,\end{aligned}$$

where we used the identity (6.17) and the fact that for indexes α^c both products are exponentially small. Using Assumption 6.3.3, Theorem 6.3.7 and Hölder's inequality we have

$$\mathbb{E}\left[\left(\prod_{s\in\alpha}(1-p_s^f) - \prod_{s\in\alpha}(1-p_s^c)\right)^2 \mathbf{1}_{\Omega^c\cap(\Omega^N)^c\cap E}\right] \leq c\Delta t_\ell^{1-6\varepsilon}[P(\Omega^c)]^{1-\varepsilon},$$

which completes the proof. □

6.4 Expected Computational Complexity

In this section we show that the new multi-level algorithm improves the expected computational complexity by one order of magnitude, in terms of the required accuracy, comparing with the new Milstein-Monte Carlo method.

6.4.1 New Milstein-Monte Carlo

We now consider a Monte Carlo approach to the estimation of the mean exit time using the Milstein method with probabilities based on Brownian bridges to compute independent samples $\mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]^{[i]}$ from the distribution of the corresponding random variable $\mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]$. Let $\mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]^{[i]}$ denote the computed exit time for the ith simulated path. If we use M such paths, then the mean value $\mathbb{E}[\tau]$ is approximated by the sample average

$$\mu = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]^{[i]}.$$

We can break down the overall error naturally into two parts,

$$\mathbb{E}[\tau] - \mu = \mathbb{E}[\tau - \tau_{\Delta t_\ell} + \tau_{\Delta t_\ell}] - \mu = \left(\mathbb{E}[\tau - \tau_{\Delta t_\ell}]\right) + \left(\mathbb{E}[\tau_{\Delta t_\ell}] - \mu\right). \tag{6.25}$$

The first term in parentheses is the bias; that is, the weak error of the new algorithm in terms of its ability to approximate the mean exit time of the SDE. Gobet in [48] proved that the weak error for the Euler Brownian bridge is of $\mathcal{O}(\Delta t_\ell)$. Knowing that the Euler–Maruyama and Milstein discretisation have the same order of weak error, we deduce that

$$\mathbb{E}[\tau] - \mathbb{E}[\tau_{\Delta t_\ell}] = \mathcal{O}(\Delta t_\ell). \tag{6.26}$$

The second term in (6.25) concerns the statistical sampling error. From the perspective of confidence interval width, this is known to scale like $\mathcal{O}(1/\sqrt{M})$. As previously, it is natural to measure the computational cost in terms of either (a) the total number of evaluations of the drift and diffusion coefficients $f(\cdot)$ or $g(\cdot)$, or (b) the number of pseudo-random number calls to obtain the Brownian increments Δw_k^ℓ. In both cases, the computational cost of each path

is proportional to the ratio of the time-span of the numerical approximation, $\mathbb{E}[\tau_{\Delta t_\ell} \mid X_0^\ell, X_1^\ell, \ldots, X_N^\ell]^{[i]}$, and the stepsize, Δt_ℓ. The overall expected computational cost of the new Milstein-Monte Carlo method is therefore proportional to $M\mathbb{E}[\tau_{\Delta t_\ell}]/\Delta t_\ell$, which, from (6.26), may be written

$$\frac{M\left(\mathbb{E}[\tau] + \mathcal{O}(\Delta t_\ell)\right)}{\Delta t_\ell}. \tag{6.27}$$

If we let TOL denote the user-specified level of accuracy, in terms of confidence interval width, so that the root mean square error is $\mathcal{O}(\text{TOL})$, then balancing the bias and sampling error in (6.25) gives the scaling $\text{TOL} = \Delta t_\ell = 1/\sqrt{M}$, whence the complexity measure (6.27) for the method becomes $\mathcal{O}(\text{TOL}^{-3})$. With reference to the complexity results derived in Chapter 4, we see that this compares favourably with the $\mathcal{O}(\text{TOL}^{-4})$ complexity of standard Monte Carlo with EM, and is comparable with the $\mathcal{O}(\text{TOL}^{-3}|\log(\text{TOL})|^{1/2})$ complexity of the multi-level EM method.

6.4.2 New Milstein-MLMC

Theorem 3.1 of [38] provides a general complexity result for multi-level Monte Carlo. Using that framework, in Theorems 6.3.2 and 6.3.8 we achieved the convergence rate of variance $\beta = 3/2 - \varepsilon$ and we can conclude that complexity of the multi-level Monte Carlo algorithm is $\mathcal{O}(\text{TOL}^{-2})$, giving one order of magnitude saving comparing to the new Milstein-Monte Carlo method, in terms of the required accuracy TOL. This can also be compared to the crudest and most popular Euler-Monte Carlo approach which has complexity $\mathcal{O}(\text{TOL}^{-4})$, as discussed in Chapter 4. The new Milstein-MLMC method therefore yields significant computational saving of two orders of magnitude over the previous state of the art. A more detailed description of the new Milstein-MLMC algorithm that specifies an optimal number of paths N_ℓ and a termination criterion can be found in section 6.5.

6.5 Computational Results

Numerical results are presented in this section. We first describe the precise numerical algorithm used in the simulations. We then check the sharpness of the rate of weak convergence in (6.26) for a one-dimensional case with a lower barrier, the rate of strong convergence in Theorem 6.3.2 for a two-dimensional case, and finally the asymptotic improvement in complexity in a more realistic simulation.

Algorithm 2 below specifies the computation for one sample path. Full implementation of the multi-level method requires estimating variance with an initial small number of sample paths, calculating the optimal number of sample paths and then generating additional samples as needed for the optimal number of sample paths, if necessary. Level ℓ is increased until a required bias is achieved. The distance between two points is measured by the Euclidean norm.

Algorithm 2 Fix $\Delta t_\ell > 0$, $\ell > 0$ and set $\Delta t_{\ell-1} = 2\Delta t_\ell$.

Set $X_k^\ell = X_0$, $X_k^{\ell-1} = X_0$, $\mathbb{E}[\tau^f] = 0$, $\mathbb{E}[\tau^c] = 0$, $\Pi^{\ell,f} = 1$, $\Pi^{\ell-1,c} = 1$.
for $k = 1, ..., 2^{\ell-1}$ **do**
  Generate dw_1^ℓ, dw_2^ℓ.
  for $m = 1, 2$ **do**
    $X_{old}^\ell \leftarrow X_{k+m-1}^\ell$.
    Evaluate X_{k+m}^ℓ using dw_m^ℓ and Δt_ℓ.
    Calculate p_{k+m}^ℓ using X_{old}^ℓ and X_{k+m}^ℓ in equation (6.3).
    if $d_B(X_{k+m}^\ell) \le 0$ **then**
      Set $p_{k+m}^\ell = 1$.
    end if
    Update $\mathbb{E}[\tau^f]$ and $\Pi^{\ell,f}$.
  end for
  $X_{old}^{\ell-1} \leftarrow X_k^{\ell-1}$.
  Calculate $dw^{\ell-1} = dw_1^\ell + dw_2^\ell$.
  Evaluate $X_{k+2}^{\ell-1}$ using $dw^{\ell-1}$ and $\Delta t_{\ell-1}$.
  Evaluate $X_{k+1}^{\ell-1}$ using $dw_{k+1}^{\ell-1} = dw_2^\ell - dw_1^\ell$ in equation (6.9).
  Calculate $p_1^{\ell-1}, p_2^{\ell-1}$ using $X_{old}^{\ell-1}, X_{k+1}^{\ell-1}$ and $X_{k+2}^{\ell-1}$ in equations (6.11).
  if $d_B(X_{k+1}^{\ell-1}) \le 0$ or $d_B(X_{k+2}^{\ell-1}) \le 0$ **then**
    Set $p_{k+2}^{\ell-1} = 1$.
  end if
  Update $\mathbb{E}[\tau^c]$ and $\Pi^{\ell-1,c}$.
end for
$\mathbb{E}[\tau^f] = \mathbb{E}[\tau^f] + \Pi^{\ell,f} \times T$.
$\mathbb{E}[\tau^c] = \mathbb{E}[\tau^c] + \Pi^{\ell-1,c} \times T$.

We start by examining the sharpness of the rate of weak convergence in (6.26) on a geometric Brownian motion model

$$dx(s) = fx(s)ds + gx(s)dw(s), \tag{6.28}$$

with constant drift $f = 0.02$, constant volatility $g = 0.3$, initial value $x(0) = 1$, lower boundary $B = 0.9$ and a finite cutoff time $T = 1$. In this one-dimensional case $\mathbb{E}[\tau] = 0.4326$, which was calculated by approximating the integral and the tail probability numerically. We approximated the weak exit time error $\text{err}_{\Delta t_\ell} = |\mathbb{E}[\tau] - \mathbb{E}[\tau_{\Delta t_\ell}]|$ using the sample mean from $N = 10^6$ path simulations and a variety of stepsizes. Figure 6.1 shows the error behaviour on a log-log scale. A least squares fit for $\log C$ and q in $\log \text{err}_{\Delta t_\ell} = \log C + q \log \Delta t_\ell$ produced $q = 1.0068$ with a least squares residual of 0.2710. This is consistent with the weak error order of convergence equal to one.

We now proceed to check whether the multi-level algorithm converges to the correct value and whether the bound in Theorems 6.3.2 and 6.3.8 is sharp. For a two-dimensional version of the SDE (6.28) with constant drift coefficients $f_i = 0.05$, $i = 1, 2$, constant volatilities $g_i = 0.3$, $i = 1, 2$, initial values $x_i(0) = 1$, $i = 1, 2$, and finite cutoff time $T = 1$, we divide the plane into two half-planes with equation $y = -0.2x + 1$ and set the domain to be the half-plane not including the origin $(0, 0)$.

We first check the weak convergence behaviour. In the left picture of Figure 6.2 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of TOL for TOL $= 2^{-5}, 2^{-6}, 2^{-7}$ and 2^{-8}. We observe that the algorithm produces an error that scales like TOL. A line with a slope 1 is included for reference. We used Monte Carlo simulations with a fixed timestep $\Delta t_\ell = 10^{-3}$ and $N = 5 \times 10^5$ samples to obtain a reference value of $\mathbb{E}[\tau] = 0.592$. The half-interval width of the 95% confidence interval is 0.0009, making the statistical error negligible.

For simplicity, we denote the estimator for the fine level defined in (6.8) by $\widehat{P}_\ell^f$ and for the coarse one defined in (6.10) by $\widehat{P}_{\ell-1}^c$. In the right picture of Figure 6.2, for the target accuracy TOL $= 2^{-8}$ we plot the quantity $\log\left(\text{Var}\left[\text{P}_\ell^{\text{f}} - \text{P}_{\ell-1}^{\text{c}}\right]\right)$ over a sequence of levels. We also include a line with a slope $-\frac{3}{2}$ for reference. A least squares fit for the slope produces $q = -1.4040$ with a residual of 0.9580.

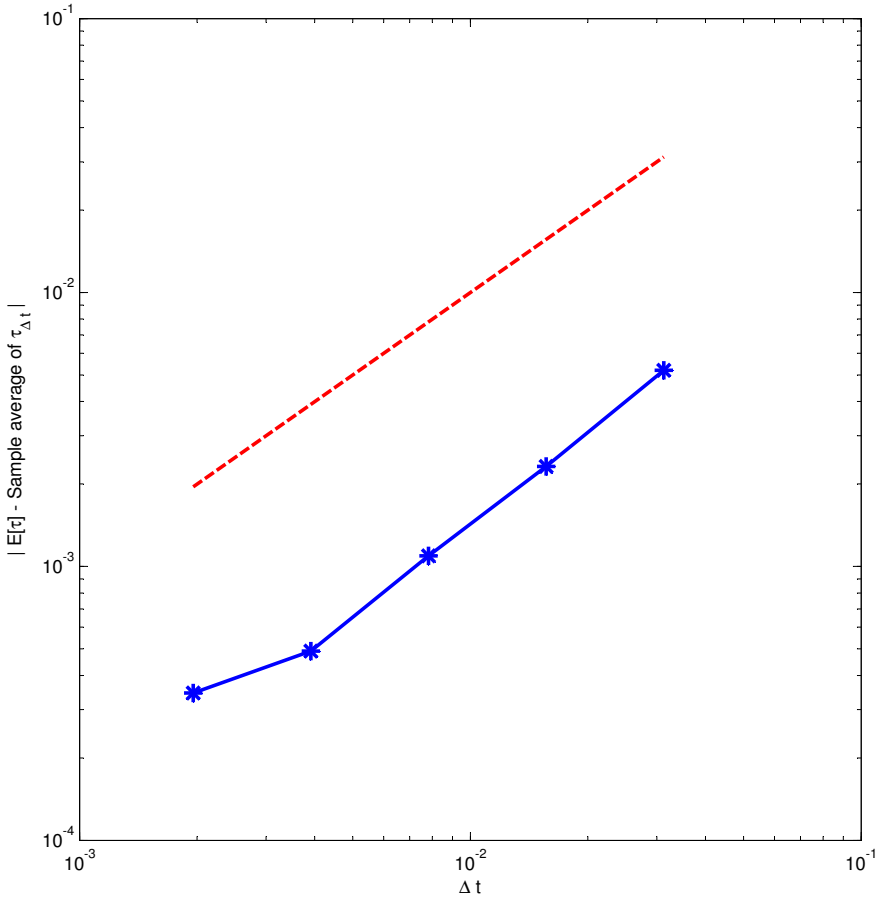

Figure 6.1: Weak error of the new Milstein-Monte Carlo method.

Finally, we compare the computational complexity of the new Milstein-Monte Carlo algorithm with the multi-level version in Figure 6.3. The computational cost of the multi-level version is measured as in (4.16), where N_ℓ is the number of paths used on each level, calculated as

$$N_\ell = \left\lceil 2\mathrm{TOL}^{-2}\sqrt{\mathrm{Var}[\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c]2^{-\ell}}\left(\sum_{\ell=0}^{L}\sqrt{\mathrm{Var}[\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c]2^{\ell}}\right)\right\rceil,$$

for $0 \le \ell \le L$. This choice of N_ℓ gives the $\frac{1}{2}\mathrm{TOL}^2$ upper bound on the variance of the estimator. To achieve a TOL^2 upper bound on the mean square error (so that

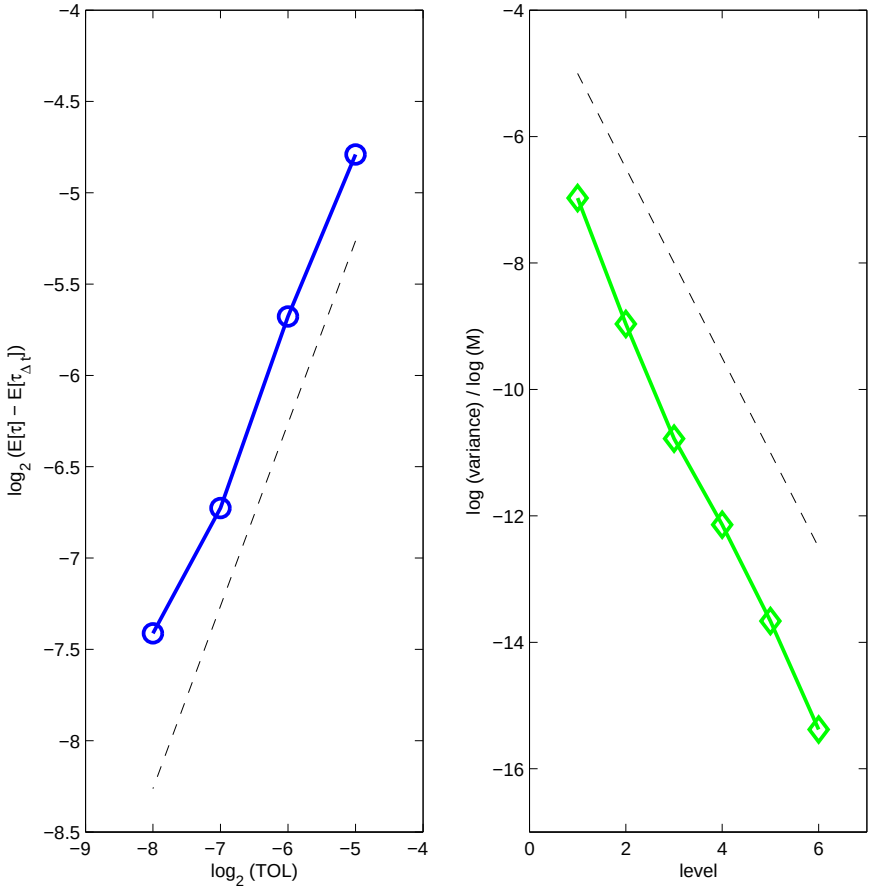

Figure 6.2: Left: weak error of the multi-level algorithm. Right: variance of $\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c$ over different levels.

the root mean square error is $\mathcal{O}(\text{TOL})$), we impose a $\frac{1}{2}\text{TOL}^2$ upper bound on the square bias error. We carefully choose the finest discretisation timestep Δt_L (and therefore the maximum level L) to make sure that $\mathbb{E}[\tau - \widehat{P}_L^f]$ is smaller than $\frac{\text{TOL}}{\sqrt{2}}$. From (6.26) we can approximate the remaining bias with a linear function

$$\mathbb{E}[\tau - \widehat{P}_\ell^f] = a\Delta t_\ell,$$

as $\ell \longrightarrow \infty$. Knowing that $2\Delta t_L = \Delta t_{L-1}$, we then have

$$\begin{aligned}\mathbb{E}\left[\widehat{P}_L^f - \widehat{P}_{L-1}^c\right] &= \mathbb{E}\left[\widehat{P}_L^f - \tau + \tau - \widehat{P}_{L-1}^c\right] \\ &= \mathbb{E}\left[\widehat{P}_L^f - \tau\right] + \mathbb{E}\left[\tau - \widehat{P}_{L-1}^c\right] \\ &= a\Delta t_L \\ &= \mathbb{E}\left[\tau - \widehat{P}_L^f\right].\end{aligned} \tag{6.29}$$

If we denote by $Z_L = \frac{1}{N_L}\sum_{i=1}^{N_L}\left(\widehat{P}_L^{[i]} - \widehat{P}_{L-1}^{[i]}\right)$ the estimator for $\mathbb{E}\left[\widehat{P}_L^f - \widehat{P}_{L-1}^c\right]$, we estimate the remaining bias testing

$$|Z_L| \leq \frac{\text{TOL}}{\sqrt{2}}.$$

From (6.29) we can consider $\mathbb{E}\left[\widehat{P}_L^f - \widehat{P}_{L-1}^c\right]$ as a function of Δt_L which converges to zero as the finest timestep goes to zero. We also can consider Z_L as a discrete sample of the value of this function. If we sample the function at a point Δt_L near its root, we can obtain a Z_L which is misleadingly small. This scenario has been fixed in [38] where the author proposes to estimate the remaining bias with the estimates of the two finest timesteps. In the numerical implementation we then use the following termination criterion

$$\max\left\{\frac{|Z_{L-1}|}{2}, |Z_L|\right\} \leq \frac{\text{TOL}}{\sqrt{2}}.$$

We measure the computational cost of the new Milstein-Monte Carlo method as in (4.18). In Figure 6.3 we plot on a log-log scale the quantity $\text{TOL}^2 \times \text{cost}$ as a function of the accuracy TOL. For $\text{TOL} = 2^{-5}$, 2^{-6}, 2^{-7} and 2^{-8} we perform 30 new Milstein-Monte Carlo and new Milstein-MLMC computations using different initial states of the pseudo-random number generator, and plot the complexity

results with red triangles (new Milstein-Monte Carlo) and green circles (new Milstein-MLMC). We then take averages for each accuracy, indicated as black crosses, for both methods. A least squares fit performed on the Monte Carlo slope produces $q = -0.9676$ with a residual 0.0757, and on the multi-level Monte Carlo slope gives $q = -0.1045$ with a residual of 0.1968. This agrees with the analytical results quoted in section 6.4 : the new Milstein-Monte Carlo complexity of $\mathcal{O}(\mathrm{TOL}^{-3})$ and the new Milstein-MLMC complexity of $\mathcal{O}(\mathrm{TOL}^{-2})$. For the most stringent test, the multi-level method is up to 100 times more efficient than the new Milstein-Monte Carlo method.

6.6 Further Computational Results

In this section we present more computational results using the method outlined in the previous section. This time, however, the domain is a set with curved boundaries. We recall from this chapter that the result from Proposition 6.2.3 is not true for such domains, but it can be used to give a good approximation so long as the boundary is smooth enough.

We then start from a two-dimensional example in which the domain is a circle with radius R=0.5. For a two-dimensional version of the SDE (6.28) with constant drift coefficients $f_i = 0.05$, $i = 1, 2$, constant volatilities $g_i = 0.3$, $i = 1, 2$, initial values $x_i(0) = 1$, $i = 1, 2$, and finite cutoff time $T = 1$, we first check the behaviour of weak error and variance. In the left picture of Figure 6.4 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of TOL for $\mathrm{TOL} = 2^{-5}$, 2^{-6}, 2^{-7} and 2^{-8}. We observe that the algorithm produces an error that scales like TOL, as required. A line with a slope 1 is included. A referenced value was obtained using the standard Monte Carlo method at high accuracy.

In the right picture of Figure 6.4, for the target accuracy $\mathrm{TOL} = 2^{-8}$ we plot the quantity $\log\left(\mathrm{Var}\left[\mathrm{P}_\ell^{\mathrm{f}} - \mathrm{P}_{\ell-1}^{\mathrm{c}}\right]\right)$ over a sequence of levels. We also include a line with a slope $-\frac{3}{2}$ for reference. A least squares fit for the slope produces $q = -1.3663$ with a residual of 0.1029. In Figure 6.5 we present complexity results in the same way as in Figure 6.3. We plot on a log-log scale the quantity $\mathrm{TOL}^2 \times \mathrm{cost}$ as a function of the accuracy TOL. For $\mathrm{TOL} = 2^{-5}$, 2^{-6}, 2^{-7} and 2^{-8} we perform 20 standard Monte Carlo, new Milstein-Monte Carlo and new Milstein-MLMC computations using different initial states of the pseudo-random number generator, and plot

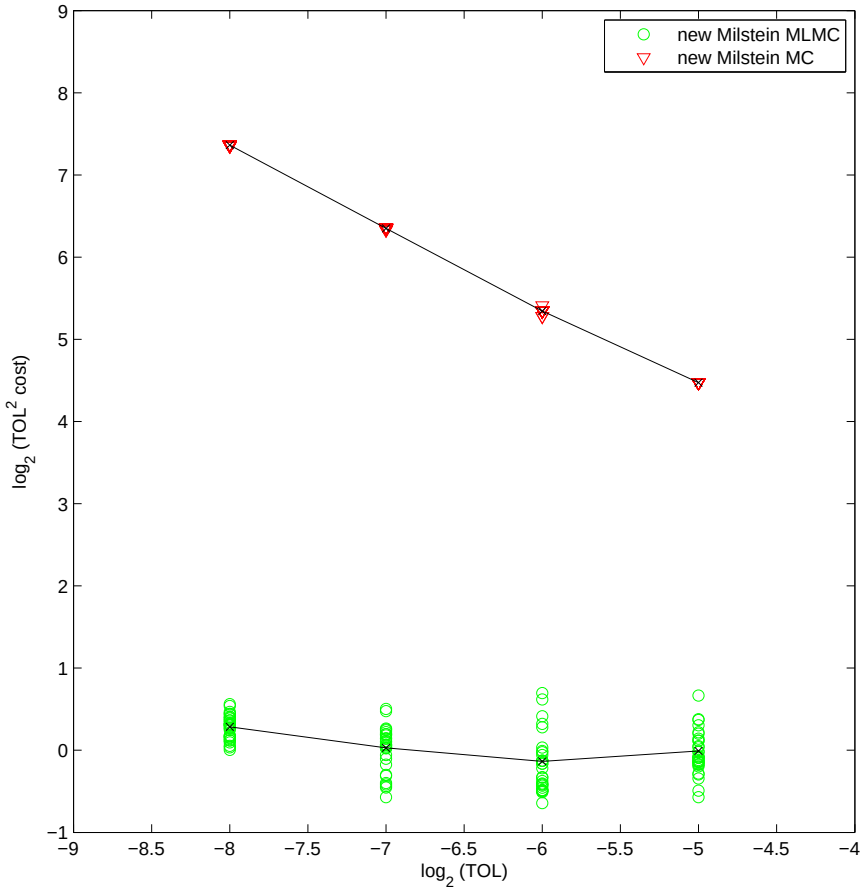

Figure 6.3: Computational effort of new Milstein-Monte Carlo (red triangles) and new Milstein-MLMC (green circles). Black crosses indicate averages of cost of both methods for each accuracy.

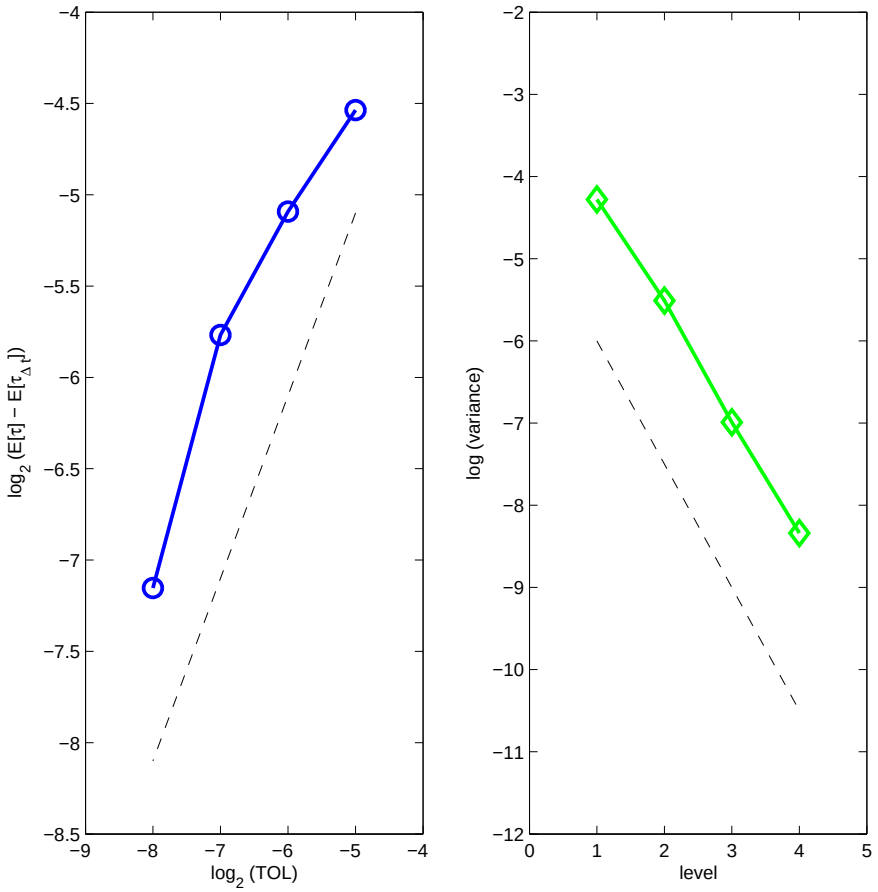

Figure 6.4: Exit of GBM from a 2d ball. Left: weak error of the multi-level algorithm. Right: variance of $\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c$ over different levels.

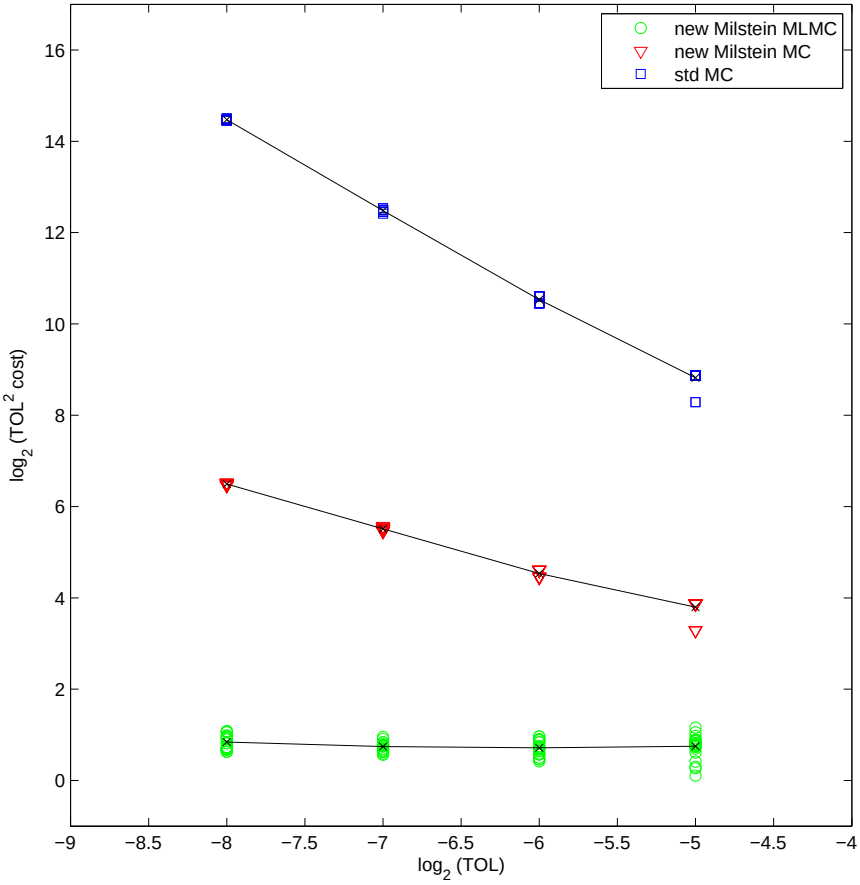

Figure 6.5: Exit of GBM from a 2d ball. Computational effort of standard Monte Carlo (blue squares), new Milstein-Monte Carlo (red triangles) and new Milstein-MLMC (green circles). Black crosses indicate averages of cost of all three methods for each accuracy.

the complexity results with blue squares (standard Monte Carlo), red triangles (new Milstein-Monte Carlo) and green circles (new Milstein-MLMC). We then take averages for each accuracy, indicated as black crosses, for all three methods. A least squares fit performed on the standard Monte Carlo slope produces $q = -1.8899$ with a residual 0.1459, on the new Milstein-Monte Carlo slope gives $q = -0.8151$ with a residual 0.1757 and on the multi-level Monte Carlo slope we obtain $q = -0.0304$ with a residual of 0.0668.

We now proceed to the second example. For a two-dimensional version of the SDE (6.28) with constant drift coefficients $f_i = 0.05$, $i = 1, 2$, constant volatilities $g_i = 0.3$, $i = 1, 2$, initial values $x_i(0) = 1$, $i = 1, 2$, and finite cutoff time $T = 1$, we divide the plane into two parts with equation $y = e^{0.5x} - 1$ and set the domain to be the one not including the origin $(0, 0)$. In the left picture of Figure 6.6 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of the accuracy parameter for TOL $= 2^{-5}$, 2^{-6}, 2^{-7} and 2^{-8}. We can see that the algorithm produces an error that scales like accuracy - a least squares fit for the slope produces $q = 0.9668$ with a residual of 0.1153. A line with a slope 1 is included for reference. A referenced value was obtained using standard Monte Carlo at high accuracy.

In the right picture of Figure 6.6, for the target accuracy TOL $= 2^{-8}$ we plot the quantity $\log\left(\text{Var}\left[\text{P}_\ell^{\text{f}} - \text{P}_{\ell-1}^{\text{c}}\right]\right)$ over a sequence of levels. We also include a line with a slope $-\frac{3}{2}$ for reference. A least squares fit for the slope produces $q = -1.3394$ with a residual of 0.0752.

In Figure 6.7 we present complexity results in the same way as in Figure 6.3. We plot on a log-log scale the quantity $\text{TOL}^2 \times \text{cost}$ as a function of the accuracy TOL. For TOL $= 2^{-5}$, 2^{-6}, 2^{-7} and 2^{-8} we perform 30 new Milstein-Monte Carlo and new Milstein-MLMC computations using different initial states of the pseudo-random number generator, and plot the complexity results with red triangles (new Milstein-Monte Carlo) and green circles (new Milstein-MLMC). We then take averages for each accuracy, indicated as black crosses, for both methods. A least squares fit performed on the new Milstein-Monte Carlo slope gives $q = -0.9023$ with a residual 0.1346 and on the multi-level Monte Carlo slope we obtain $q = -0.0549$ with a residual of 0.0218.

We finish this section with the third example. For a two-dimensional version of the SDE (4.19) with constant drift coefficients $f_i = 0.05$, $i = 1, 2$, constant

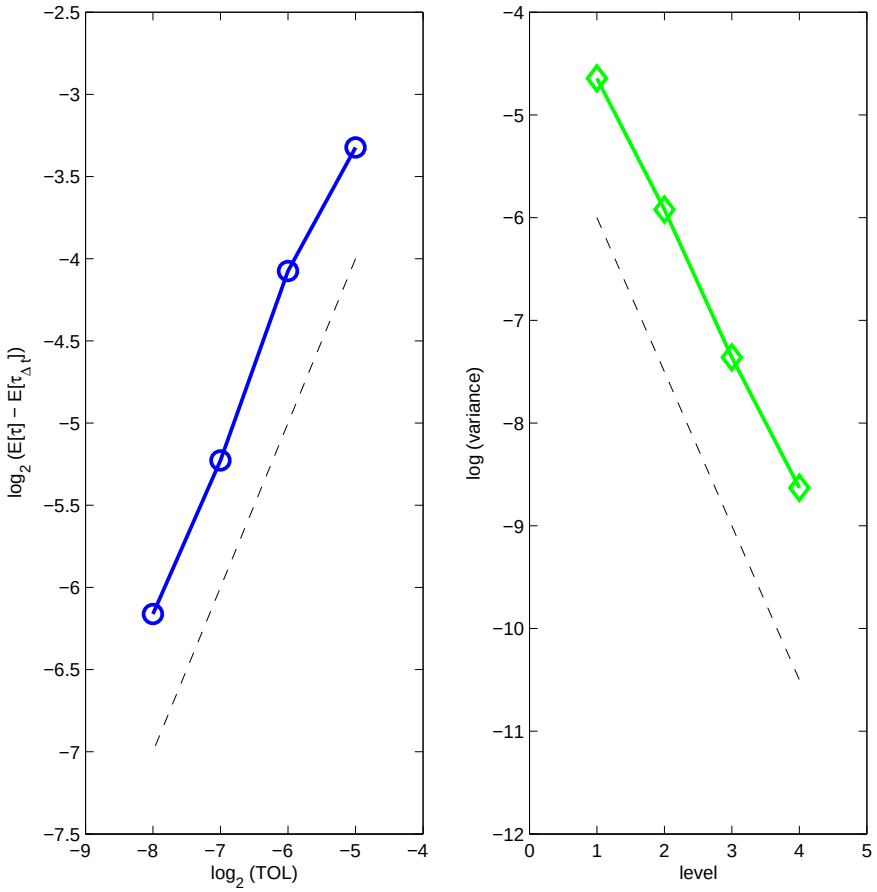

Figure 6.6: Exit of GBM from a curved boundary. Left: weak error of the multi-level algorithm. Right: variance of $\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c$ over different levels.

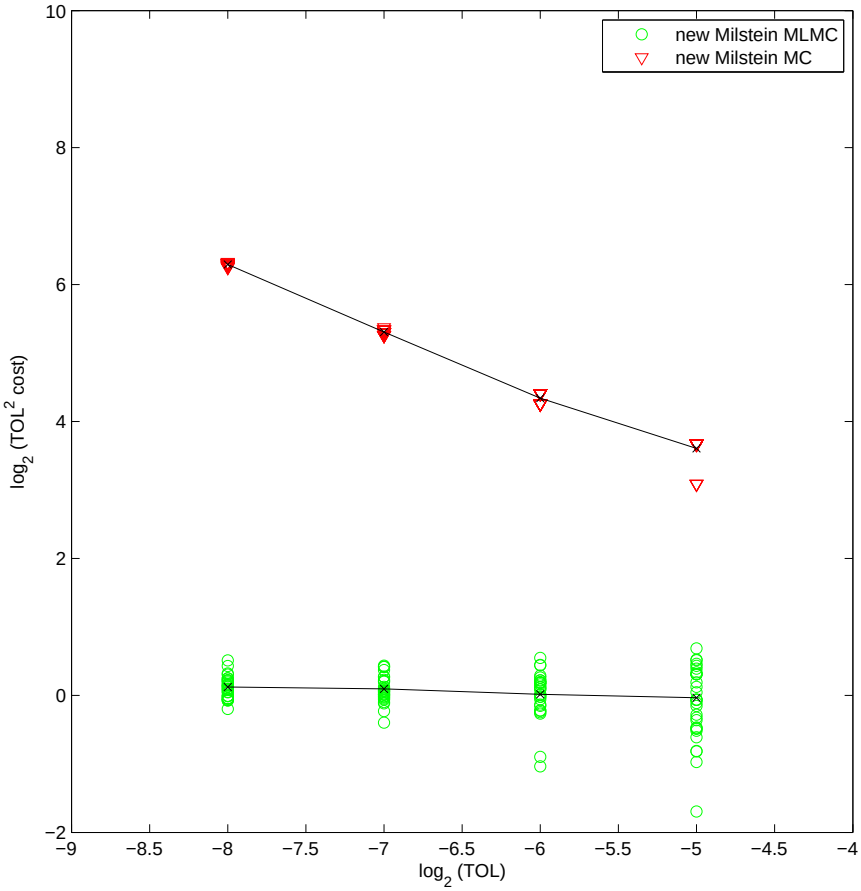

Figure 6.7: Exit of GBM from a curved boundary. Computational effort of new Milstein-Monte Carlo (red triangles) and new Milstein-MLMC (green circles). Black crosses indicate averages of cost of both methods for each accuracy.

volatilities $g_i = 0.3$, $i = 1, 2$, initial values $x_i(0) = 1$, $i = 1, 2$, finite cutoff time $T = 1$, and a mean reversion parameter $\mu = 1$, we set the domain to be a circle with radius R=0.5. We recall that the SDE (4.19) is nonlinear and non-Lipschitzian and is widely used in mathematical finance as an alternative to geometric Brownian motion. In the left picture of Figure 6.8 we plot on a log-log scale the accuracy obtained by the multi-level algorithm as a function of the accuracy parameter for TOL = 2^{-5}, 2^{-6}, 2^{-7} and 2^{-8}. We can see that the algorithm produces an error that scales like accuracy. A line with a slope 1 is included for reference and a referenced value was obtained using standard Monte Carlo at high accuracy.

In the right picture of Figure 6.8, for the target accuracy TOL = 2^{-8} we plot the quantity $\log\left(\mathrm{Var}\left[\mathrm{P}_\ell^{\mathrm{f}} - \mathrm{P}_{\ell-1}^{\mathrm{c}}\right]\right)$ over a sequence of levels. We also include a line with a slope $-\frac{3}{2}$ for reference. A least squares fit for the slope produces $q = -1.3801$ with a residual of 0.0933.

In Figure 6.9 we present complexity results in the same way as in Figure 6.3. We plot on a log-log scale the quantity $\mathrm{TOL}^2 \times \mathrm{cost}$ as a function of the accuracy TOL. For TOL = 2^{-5}, 2^{-6}, 2^{-7} and 2^{-8} we perform 30 standard Monte Carlo, new Milstein-Monte Carlo and new Milstein-MLMC computations using different initial states of the pseudo-random number generator, and plot the complexity results with blue squares (standard Monte Carlo), red triangles (new Milstein-Monte Carlo) and green circles (new Milstein-MLMC). We then take averages for each accuracy, indicated as black crosses, for all three methods. A least squares fit performed on the standard Monte Carlo slope produces $q = -1.9211$ with a residual 0.0828, on the new Milstein-Monte Carlo slope gives $q = -0.9129$ with a residual 0.1058 and on the multi-level Monte Carlo slope we obtain $q = -0.0213$ with a residual of 0.0552.

Three numerical examples from this section suggest that a straightforward implementation of the new multi-level algorithm for the process exiting from a curved boundary does not yield complexity results that are as good as those for the exit from a half-space. This also suggests that the interpolated point on the coarse path could be chosen in a more effective way so the variance convergence is closer to the desired order 3/2.

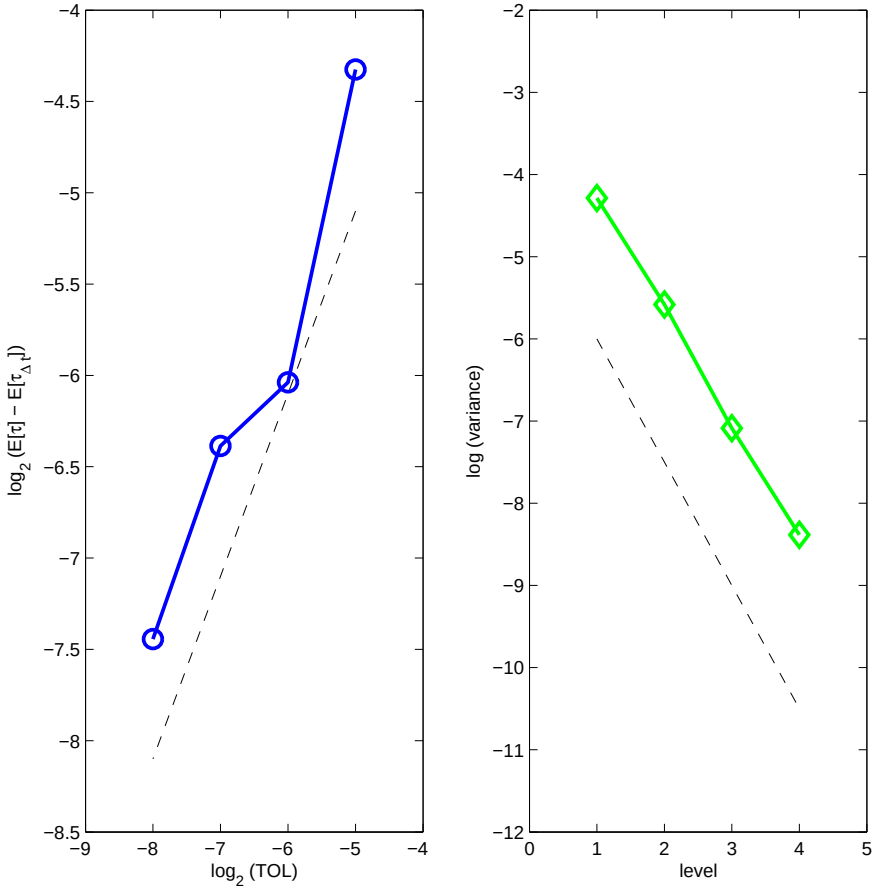

Figure 6.8: Exit of the CIR process from a 2d ball. Left: weak error of the multi-level algorithm. Right: variance of $\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c$ over different levels.

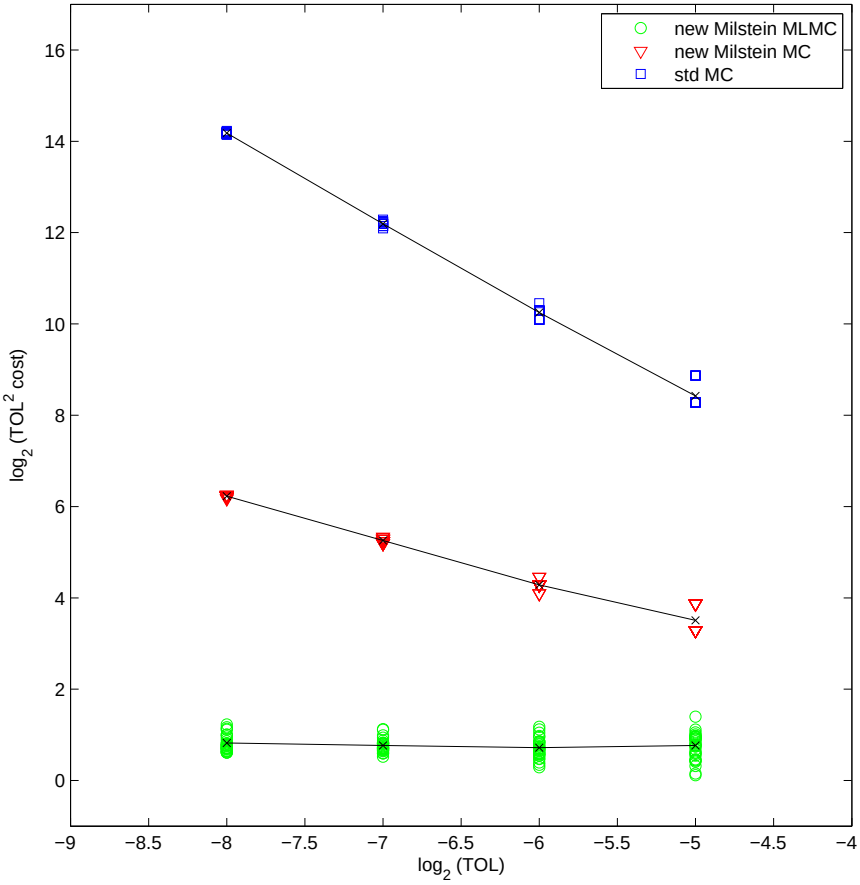

Figure 6.9: Exit of the CIR process from a 2d ball. Computational effort of standard Monte Carlo (blue squares), new Milstein-Monte Carlo (red triangles) and new Milstein-MLMC (green circles). Black crosses indicate averages of cost of all three methods for each accuracy.

6.7 Summary

In this chapter we designed and analysed a multi-level version of a new algorithm for approximating mean exit times. A key step in the design was to provide a different representation for a mean exit time; the main objective in the analysis was to establish a rate of strong convergence for variance between coarse and refined computations. We then showed that it is possible to reduce the computational complexity by two orders of magnitude, compared with the standard Euler-Monte Carlo method. Finally, we presented more computational examples for which there is no theoretical background (the assumption of a half-space is violated) and obtained promising results.

Chapter 7

Conclusion and Future Research

> Knowing is not enough; we must apply.
> Willing is not enough; we must do.
>
> Johann Wolfgang von Goethe, 1749-1832

In the past 5 years substantial progress has been made with the multi-level Monte Carlo method, not only for financial options based on underlying assets described by the Brownian diffusions and Lévy processes but also for the problem when the time of a process exiting a certain set is an estimated quantity.

The multi-level approach is based on a simple concept. It may be described as a recursive control variate method, where we use a coarse path simulation as a control variate for a fine path simulation, relying on strong convergence properties to ensure a very strong coupling between the two. In practical applications the main challenge is to correlate the fine and coarse path simulations as closely as possible, without excessive cost.

In this work we successfully applied the multi-level approach to the problem of mean exit times. We thereby derived and analysed the first multi-level algorithm in this setting. In its basic form, where the exit time is approximated via a numerical scheme leaving the domain, the new method generates savings of almost one order of magnitude in computational cost over the standard Monte Carlo approach. In its most advanced form, in which we calculate probabilities of a Brownian bridge interpolation leaving the domain between two discrete points, a new method yields significant savings of two orders of magnitude compared to the crude Monte Carlo approach.

In the future, fascinating areas for further research include:

- extending the method to more complicated boundaries in the multi-dimensional setting [50], for which illustrative numerical examples have been presented in section 6.6,
- considering the use of quasi-Monte Carlo methods,
- researching on multi-level techniques for early-exercise options such as American and Bermudan options,
- fine-tuning the algorithm to high performance computing architectures.

References

[1] G. B. Ahn, D.H., *A Parametric Nonlinear Model of Term Structure Dynamics*, The Review of Financial Studies, 12 (1999), pp. 721–762.

[2] Y. Ait-Sahalia, *Testing Continuous-time Models of the Spot Interest Rate*, The Review of Financial Studies, 9 (1996), pp. 385–426.

[3] L. Andersen, *A Simple Approach to the Pricing of Bermudan Swaptions in the Multi-factor Libor Market Model*, Journal of Computational Finance, 3 (2000), pp. 5–32.

[4] L. Andersen and M. Broadie, *A Primal-Dual Simulation Algorithm for Pricing Multi-Dimensional American Options*, Management Science, 50 (2004), pp. 1222–1234.

[5] T. Andersen, L. Benzoni, and J. Lund, *An Empirical Investigation of Continuous-time Equity Return Models*, The Journal of Finance, 57 (2002), pp. 1239–1284.

[6] A. Asmussen and P. Glynn, *Stochastic Simulation*, Springer, New York, 2007.

[7] A. Bahar and X. Mao, *Stochastic Delay Population Dynamics*, International Journal of Pure and Applied Mathematics, 11 (2004), pp. 377–400.

[8] P. Baldi, *Exact Asymptotics for the Probability of Exit from a Domain and Applications to Simulation*, The Annals of Probability, 23 (1995), pp. 1644–1670.

[9] A. Barth, C. Schwab, and N. Zollinger, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numerische Mathematik, 119 (2011), pp. 123–161.

[10] D. Bedingham, *State Reduction Dynamics in a Simplified QED model*, Journal of Physics A: Mathematical and Theoretical, 41 (2008), pp. 495–205.

[11] D. Belomestny and J. Schoenmakers, *Multilevel Dual Approach for Pricing American Style Derivatives*, Preprint 1647, WIAS, 2011.

[12] F. Black and J. Cox, *Valuing Corporate Securities: Some Effects of Bond Indenture Provisions*, The Journal of Finance, 31 (1976), pp. 351–367.

[13] P. Boyle, M. Broadie, and P. Glasserman, *Monte Carlo Methods for Security Pricing*, Journal of Economic Dynamics and Control, 21 (1997), pp. 1267–1321.

[14] M. Broadie and Ö. Kaya, *Exact Simulation of Stochastic Volatility and other Affine Jump Diffusion Processes*, Operations Research, 54 (2006), pp. 217–231.

[15] S. Burgos and M. Giles, *Computing Greeks Using Multilevel Path Simulation*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, L. Plaskota and H. Woźniakowski, eds., Springer-Verlag, 2012.

[16] J. Campbell, A. Lo, A. MacKinlay, and R. Whitelaw, *The Econometrics of Financial Markets*, Macroeconomic Dynamics, 2 (1998), pp. 559–562.

[17] K. Chan, G. Karolyi, F. Longstaff, and A. Sanders, *An Empirical Comparison of Alternative Models of the Short-term Interest Rate*, The Journal of Finance, 47 (1992), pp. 1209–1227.

[18] Z. Chen and P. Glasserman, *Fast Pricing of Basket Default Swaps*, Operations Research, 56 (2008), pp. 286–303.

[19] J. Clark and R. Cameron, *The Maximum Rate of Convergence of Discrete Approximations for Stochastic Differential Equations*, in Stochastic Differential Equations, no. 25 in Lecture Notes in Control and Information Sciences, B. Grigelionis, ed., Springer-Verlag, 1980.

[20] K. Cliffe, M. Giles, R. Scheichl, and A. Teckentrup, *Multilevel Monte Carlo Methods and Applications to Elliptic PDEs with Random*

Coefficients, Computing and Visualization in Science, 14 (2011), pp. 3–15.

[21] T. Conley, L. Hansen, E. Luttmer, and J. Scheinkman, *Short-term Interest Rates as Subordinated Diffusions*, The Review of Financial Studies, 10 (1997), pp. 525–577.

[22] M. Crouhy, D. Galai, and R. Mark, *A Comparative Analysis of Current Credit Risk Models*, Journal of Banking & Finance, 24 (2000), pp. 59–117.

[23] J. Cvitanić and I. Karatzas, *Hedging Contingent Claims with Constrained Portfolios*, The Annals of Applied Probability, 3 (1993), pp. 652–681.

[24] K. Debrabant, M. B. Giles, and A. Rossler, *Numerical Analysis of Multilevel Monte Carlo Path Simulation Using Milstein Discretization: Scalar Case*, Technical Report, (2011).

[25] S. Dereich, *Multilevel Monte Carlo Algorithms for Lévy-driven SDEs with Gaussian Correction*, Annals of Applied Probability, 21 (2011), pp. 283–311.

[26] S. Dereich and F. Heidenreich, *A Multilevel Monte Carlo Algorithm for Lévy-driven Stochastic Differential Equations*, Stochastic Processes and their Applications, 121 (2011), pp. 1565–1587.

[27] J. Doob, *Classical Potential Theory and its Probabilistic Counterpart*, vol. 262, Springer, 1984.

[28] J. Duan, *A Specification Test for Time Series Models by a Normality*, in Econometric Society 2004 North American Winter Meetings, no. 467, Econometric Society, 2004.

[29] D. Duffie and P. Glynn, *Efficient Monte Carlo Simulation of Security Prices*, The Annals of Applied Probability, 5 (1995), pp. 897–905.

[30] S. Fienberg, *Stochastic Models for Single Neuron Firing Trains: a Survey*, Biometrics, 30 (1974), pp. 399–427.

[31] J. Gaines and T. Lyons, *Random Generation of Stochastic Integrals*, SIAM Journal of Applied Mathematics, 54 (1994), pp. 1132–1146.

[32] A. Gallant and G. Tauchen, *Estimation of Continuous-time Models for Stock Returns and Interest Rates*, Macroeconomic Dynamics, 1 (2005), pp. 135–168.

[33] T. Gard, *Introduction to Stochastic Differential Equations*, New York, 1988.

[34] G. Gerstein and B. Mandelbrot, *Random Walk Models for the Spike Activity of a Single Neuron*, Biophysical Journal, 4 (1964), pp. 41–68.

[35] I. Gihman and A. Skorohod, *Stochastic Differential Equations*, 1972.

[36] D. Gilbarg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer, 2001.

[37] M. Giles, *Improved Multilevel Monte Carlo Convergence Using the Milstein Scheme*, in Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich, and H. Niederreiter, eds., Springer-Verlag, 2008, pp. 343–358.

[38] M. Giles, *Multilevel Monte Carlo Path Simulation*, Operations Research, 56 (2008), pp. 607–617.

[39] M. Giles, *Multilevel Monte Carlo for Basket Options*, in Proceedings of the 2009 Winter Simulation Conference, M. Rossetti, R. Hill, B. Johansson, A. Dunkin, and R. Ingalls, eds., IEEE, 2009, pp. 1283–1290.

[40] M. Giles, D. Higham, and X. Mao, *Analysing Multi-level Monte Carlo for Options with Non-globally Lipschitz Payoff*, Finance and Stochastics, 13 (2009), pp. 403–413.

[41] M. Giles, D. Higham, M. Roj, and L. Szpruch, *Multi-level Monte Carlo Using Brownian Bridge Interpolation for Expected Exit Times*, working paper, to appear (2013).

[42] M. Giles and C. Reisinger, *Stochastic Finite Differences and Multilevel Monte Carlo for a Class of SPDEs in Finance*, SIAM Journal of Financial Mathematics, to appear (2012).

[43] M. Giles and L. Szpruch, *Antithetic Multilevel Monte Carlo Estimation for Multi-dimensional SDEs without Lévy Area Simulation*, Arxiv preprint arXiv:1202.6283, (2012).

[44] M. Giles and B. Waterhouse, *Multilevel Quasi-Monte Carlo Path Simulation*, in Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics, de Gruyter, 2009, pp. 165–181.

[45] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer Verlag, 2004.

[46] ——, *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2004.

[47] P. Glasserman and N. Merener, *Convergence of a Discretization Scheme for Jump-diffusion Processes with State–dependent Intensities*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 460 (2004), pp. 111–127.

[48] E. Gobet, *Weak Approximation of Killed Diffusion Using Euler Schemes*, Stochastic Processes and their Applications, 87 (2000), pp. 167–197.

[49] ——, *Euler Schemes and Half-space Approximation for the Simulation of Diffusion in a Domain.*, ESAIM: Probability and Statistics, 5 (2001), pp. 261–297.

[50] ——, *Advanced Monte Carlo Methods for Barrier and Related Exotic Options*, Handbook of Numerical Analysis, 15 (2009), pp. 497–528.

[51] E. Gobet and S. Menozzi, *Exact Approximation Rate of Killed Hypoelliptic Diffusions Using the Discrete Euler Scheme*, Stochastic Processes and their Applications, 112 (2004), pp. 201–223.

[52] ——, *Discrete Sampling of Functionals of Itô Processes*, Séminaire de Probabilités XL, XL (2007), pp. 355–374.

[53] ——, *Stopped Diffusion Processes: Boundary Corrections and Overshoot*, Stochastic Processes and their Applications, 120 (2010), pp. 130–162.

[54] S. Graubner, *Multi-level Monte Carlo Methoden für Stochastiche Partial Differentialgleichungen*, Diplomarbeit, TU Darmstadt, 2008.

[55] S. Grenadier and A. Weiss, *Investment in Technological Innovations: An Option Pricing Approach*, Journal of Financial Economics, 44 (1997), pp. 397–416.

[56] S. Heinrich, *Multilevel Monte Carlo Methods*, in Lecture Notes in Large Scale Scientific Computing, Springer-Verlag, 2001, pp. 58–67.

[57] S. Heston, *A Simple New Formula for Options with Stochastic Volatility*, Course notes of Washington University in St. Louis, Missouri, (1997).

[58] D. Higham, *An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*, SIAM Review, (2001), pp. 525–546.

[59] D. Higham, X. Mao, M. Roj, Q. Song, and G. Yin, *Mean Exit Times and the Multilevel Monte Carlo Method*, SIAM/ASA Journal on Uncertainty Quantification, 1 (2013), pp. 2–18.

[60] D. Higham and M. Roj, *Computing Mean First Exit Times for Stochastic Processes Using Multi-level Monte Carlo*, in Proceedings of the 2012 Winter Simulation Conference, C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. Uhrmacher, eds., Institute of Electrical and Electronics Engineers, Inc., 2012.

[61] A. Hussain, *Multisensor Distributed Sequential Detection*, Aerospace and Electronic Systems, IEEE Transactions on Aerospace Electronic Systems, 30 (2002), pp. 698–708.

[62] M. Hutzenthaler, A. Jentzen, and P. Kloeden, *Divergence of the Multilevel Monte Carlo Euler Method for Nonlinear Stochastic Differential Equations*, Annals of Applied Probability, to appear (2013).

[63] S. Iyengar, *Hitting Lines with Two-dimensional Brownian Motion*, SIAM Journal on Applied Mathematics, (1985), pp. 983–989.

[64] T. Joro, A. Niu, and P. Na, *A Simulation-Based First-to-Default (FtD) Credit Default Swap (CDS) Pricing Approach under Jump-Diffusion*, in Proceedings of the 2004 Winter Simulation Conference, R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds., vol. 2, Piscataway, New Jersey, Dec. 2004, Institute of Electrical and Electronics Engineers, Inc., pp. 1632–1636.

[65] N. Kantas, A. Lecchini-Visintini, and J. Maciejowski, *Simulation-based Bayesian Optimal Design of Aircraft Trajectories for Air Traffic Management*, International Journal of Adaptive Control and Signal Processing, 24 (2010), pp. 882–899.

[66] A. Kebaier, *Statistical Romberg Extrapolation: a New Variance Reduction Method and Applications to Options Pricing*, Annals of Applied Probability, 14 (2005), pp. 2681–2705.

[67] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, 1999.

[68] H. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer Verlag, 2001.

[69] H. Lerche and D. Siegmund, *Approximate Exit Probabilities for a Brownian Bridge on a Short Time Interval, and Applications*, Advances in Applied Probability, (1989), pp. 1–19.

[70] A. Lewis, *Option Valuation under Stochastic Volatility*, Finance Press, 2000.

[71] F. Longstaff and E. Schwartz, *Valuing American Options by Simulation: A Simple Least-squares Approach*, Review of Financial Studies, 14 (2001), pp. 113–147.

[72] S. Majd and R. Pindyck, *Time to Build, Option Value, and Investment Decisions*, Journal of Financial Economics, 18 (1987), pp. 7–27.

[73] R. Mannella, *Absorbing Boundaries and Optimal Stopping in a Stochastic Differential Equation*, Physics Letters A, 254 (1999), pp. 257–262.

[74] X. Mao, *Stochastic Differential Equations and Applications*, Horwood, Chichester, second ed., 2007.

[75] X. Mao, G. Marion, and E. Renshaw, *Environmental Brownian Noise Suppresses Explosions in Population Dynamics*, Stochastic Process. Appl, 97 (2002), pp. 95–110.

[76] X. Mao, S. Sabanis, and E. Renshaw, *Asymptotic Behaviour of the Stochastic Lotka–Volterra Model*, Journal of Mathematical Analysis and Applications, 287 (2003), pp. 141–156.

[77] G. Maruyama, *Continuous Markov Processes and Stochastic Equations*, Rendiconti del Circolo Matematico di Palermo, 4 (1955), pp. 48–90.

[78] R. Merton, *Option Pricing when Underlying Stock Returns are Discontinuous*, Journal of Finance, 3 (1976), pp. 125–144.

[79] H. Meyr and G. Polzer, *A Simple Method for Evaluating the Probability Density Function of the Sample Number for the Optimum Sequential Detector*, IEEE Transactions on Communications, 35 (2002), pp. 99–103.

[80] T. Müller-Gronbach, *Strong Approximation of Systems of Stochastic Differential Equations*, Habilitation Thesis, TU Darmstadt, 2002.

[81] ——, *The Optimal Uniform Approximation of Systems of Stochastic Differential Equations*, The Annals of Applied Probability, 12 (2002), pp. 664–690.

[82] B. Oksendal, *Stochastic Differential Equations: an Introduction with Applications*, Springer Verlag, 2003.

[83] G. Pagès, *Multi-step Richardson-Romberg Extrapolation: Remarks on Variance Control and Complexity*, Monte Carlo Methods and Applications, 13 (2007), pp. 37–70.

[84] S. Pang, F. Deng, X. Mao, et al., *Asymptotic Properties of Stochastic Population Dynamics*, Dynamics of Continuous Discrete and Impulsive Systems Series A, 15 (2008), pp. 603–620.

[85] A. Pedersen, *Consistency and Asymptotic Normality of an Approximate Maximum Likelihood Estimator for Discretely Observed Diffusion Processes*, Bernoulli, 1 (1995), pp. 257–279.

[86] R. Pindyck, *Irreversible Investment, Capacity Choice, and the Value of the Firm*, The American Economic Review, 78 (1988), pp. 969–985.

[87] E. Platen, *An Introduction to Numerical Methods for Stochastic Differential Equations*, Acta Numerica, 8 (1999), pp. 197–246.

[88] E. Platen and N. Bruti-Liberati, *Numerical Solution of Stochastic Differential Equations With Jumps in Finance*, Springer, 2010.

[89] T. Primožič, *Estimating Expected First Passage Times Using Multilevel Monte Carlo Algorithm*, Master's Thesis, University of Oxford (2011).

[90] T. Rydén and M. Wiktorsson, *On the Simulation of Iterated Itô Integrals*, Stochastic Processes and their Applications, 91 (2001), pp. 151–168.

[91] A. Speight, *A Multilevel Approach to Control Variates*, Journal of Computational Finance, 12 (2009), pp. 1–25.

[92] ——, *Multigrid Techniques in Economics*, Operations Research, 58 (2010), pp. 1057–1078.

[93] L. Szpruch and D. Higham, *Comparing Hitting Time Behaviour of Markov Jump Processes and their Diffusion Approximations*, Multiscale Model. Simul., 8 (2010), pp. 605–621.

[94] L. Szpruch, X. Mao, D. Higham, and J. Pan, *Numerical Simulation of a Strongly Nonlinear Ait-Sahalia-type Interest Rate Model*, BIT Numerical Mathematics, (2010), pp. 1–21.

[95] M. Taksar, *Optimal Risk and Dividend Distribution Control Models for an Insurance Company*, Mathematical Methods of Operations Research, 51 (2000), pp. 1–42.

[96] A. Triantis and J. Hodder, *Valuing Flexibility as a Complex Option*, Journal of Finance, 45 (1990), pp. 549–565.

[97] M. Wiktorsson, *Joint Characteristic Function and Simultaneous Simulation of Iterated Itô Integrals for Multiple Independent Brownian Motions*, Annals of Applied Probability, 11 (2001), pp. 470–487.

[98] Y. Xia, *Multilevel Monte Carlo Method for Jump-diffusion SDEs*, Arxiv preprint arXiv:1106.4730, (2011).

[99] Y. Xia and M. Giles, *Multilevel Path Simulation for Jump-diffusion SDEs*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, L. Plaskota and H. Woźniakowski, eds., Springer-Verlag, 2012.

[100] G. Yin, C. Xu, and L. Wang, *Optimal Remapping in Dynamic Bulk Synchronous Computations via a Stochastic Control Approach*, IEEE Transactions on Parallel and Distributed Systems, 14 (2003), pp. 51–62.