Hybrid Adaptive Multilevel Monte Carlo Algorithm for Non-Smooth Observables of Itô Stochastic Differential Equations.

Thesis by

Nadhir Ben Rached

In Partial Fulfillment of the Requirements

For the Degree of

Masters of Science

King Abdullah University of Science and Technology, Thuwal,

Kingdom of Saudi Arabia

November, 2013

The thesis of Nadhir Ben Rached is approved by the examination committee

Committee Chairperson: Raul Tempone

Committee Member: Marco Scavino

Committee Member: Mohamed-Slim Alouini

ABSTRACT

Hybrid Adaptive Multilevel Monte Carlo Algorithm for Non-Smooth Observables of Itô Stochastic Differential Equations.

Nadhir Ben Rached

The Monte Carlo forward Euler method with uniform time stepping is the standard technique to compute an approximation of the expected payoff of a solution of an Itô Stochastic Differential Equation (SDE). For a given accuracy requirement TOL, the complexity of this technique for well behaved problems, that is the amount of computational work to solve the problem, is $O(TOL^{-3})$.

A new hybrid adaptive Monte Carlo forward Euler algorithm for SDEs with non-smooth coefficients and low regular observables is developed in this thesis. This adaptive method is based on the derivation of a new error expansion with computable leading-order terms. The basic idea of the new expansion is the use of a mixture of prior information to determine the weight functions and posterior information to compute the local error. In a number of numerical examples the superior efficiency of the hybrid adaptive algorithm over the standard uniform time stepping technique is verified. When a non-smooth binary payoff with either Geometric Brownian Motion (GBM) or drift singularity type of SDEs is considered, the new adaptive method achieves the same complexity as the uniform discretization with smooth prob-

lems. Moreover, the new developed algorithm is extended to the Multilevel Monte Carlo (MLMC) forward Euler setting which reduces the complexity from $O(TOL^{-3})$ to $O(TOL^{-2}(\log(TOL))^2)$. For the binary option case with the same type of Itô SDEs, the hybrid adaptive MLMC forward Euler recovers the standard multilevel computational cost $O(TOL^{-2}(\log(TOL))^2)$. When considering a higher order Milstein scheme, a similar complexity result was obtained by Giles using the uniform time stepping for one dimensional SDEs. The difficulty to extend Giles' Milstein MLMC method to the multidimensional case is an argument for the flexibility of our new constructed adaptive MLMC forward Euler method which can be easily adapted to this setting. Similarly, the expected complexity $O(TOL^{-2}(\log(TOL))^2)$ is reached for the multidimensional case and verified numerically.

ACKNOWLEDGEMENTS

Within the framework of this master thesis, I would like to express my deepest gratitude to my supervisor Prof. Raúl Tempone for his support, suggestions, and guidance in almost every step throughout my thesis.

I would like to thank my co-supervisor, Dr. Håkon Hoel, for his precious advice, remarks and assistance. I am deeply grateful to him for the discussions that helped me sort out the technical details of my work.

I am also thankful to all the members of the Stochastic Numerics group and the SRI Center for Uncertainty Quantification in Science and Engineering for their suggestions and constructive comments.

Last but not least, I would like to thank my family for their love and encouragement. Special thank goes to my parents for their support and assistance in all my pursuits.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CDF	Cumulative Distribution Function
CEV	Constant Elasticity of Variance model
CIR	Cox Ingersoll Ross model
CLT	Central Limit Theorem
FP	Fokker-Planck
GBM	Geometric Brownian Motion
MC	Monte Carlo
MLMC	Multilevel Monte Carlo
ODE	Ordinary Differential Equations
PDF	Probability Distribution Function
SDE	Stochastic Differential Equation
SLMC	Single Level Monte Carlo

LIST OF SYMBOLS

α	The singularity point on the SDE
α_C	Risk level in the CLT
$\bar{X}$	Numerical approximation of the stock price
C_C	Confidence constant in the CLT
$C^p(\mathbb{R})$	Vector space of p^{th} differentiable functions
$C^q(\mathbb{R})$	Vector space of q^{th} differentiable functions
Σ	Covariance matrix
δ	Mollification parameter
σ	Volatility
r	Interest rate
ϕ	Dual function
ϵ_δ	Mollification error
ϵ_S	Statistical error
ϵ_T	Time discretization error
$\mathcal{A}_{mlmc}$	MLMC estimator
$\mathcal{F}_{t_{n-1}}$	Filtration of the wiener up to the time t_{n-1}
T	Final time
g_δ	Mollified payoff
H	Kernel density
K	Strike or exercise price
L	Number of MLMC levels
ℓ	Level index
e_n	Local error
M	Number of MC realizations
N	Number of time steps
g	Payoff
ρ_{X_T}	Probability density function of $X(T)$
$\hat{\sigma}_M$	Sample standard deviation
TOL	Tolerance

TOL_ℓ	Tolerance on each MLMC level
TOL_S	statistical tolerance
TOL_T	Time discretization tolerance
w	Weight vector
w_δ	Window function with length δ
X	The underlying stock price
x_0	Initial stock price

LIST OF FIGURES

Chapter 1

Introduction

1.1 Project Framework

Differential Equation models are used not only in natural sciences such as physics and biology, but also in engineering and finance. The aim is to model the evolution of real systems, hence to understand and predict their future behavior.

Differential equations can be used also in the case where the problem in consideration has, by nature, a randomness in its evolution. This gives rise to the so called SDE, which are driven by deterministic and stochastic terms. For example, the stochastic nature of the financial market makes the use of SDE crucial when modeling the evolution of a stock price.

Option pricing is one of the most important applications in financial engineering. An option is a contract between two parties that gives the buyer the right and not the obligation to buy or to sell an underlying underlying asset at a certain price (strike price) on or before an expiration date or exercise date. In this kind of application, the underlying asset is modeled by an SDE and the option price is an expected payoff based on the final underlying stock price. A major revolution of this field was achieved by Fischer Black and Myron Scholes [9] by using SDEs to price various derivatives, including options on commodities and financial assets.

The model in [9] is based on many unrealistic assumptions. For instance, the underlying asset price is assumed to have a log normal distribution which makes the solution of the model unlikely to represent the real evolution. Despite its drawbacks, the Black-Scholes model is still widely used in practice because of its computational simplicity and its robustness to be adjusted to fit real data. Complicated problems are often modeled by relatively complicated SDEs which do not have analytical solutions. Numerical analysis provides a wide range of methods to approximate the unknown solution. Convergence, stability and consistency of any numerical methods are deeply studied by mathematicians. Approximating the exact solution by an Euler scheme, for example, is well studied in [2] and [3].

The Monte Carlo (MC) method, together with a numerical scheme, is the standard way to approximate an option price as the expected payoff g which depends on the stock price X at the final time T, i.e. $\mathbb{E}[g(X(T))]$. It relies on generating M independent and identically distributed realizations of $g(\bar{X}(T))$ and approximating the price as the sample mean

$$\mathbb{E}[g(X(T))] \approx \sum_{j=1}^{M} \frac{g(\bar{X}(T,\omega_j))}{M},$$

where $\bar{X}$(T) denotes the numerical approximation of $X(T)$. The forward Euler method with uniform discretization of the time variable is the standard time stepping scheme. The complexity of the MC forward Euler for a smooth payoff and SDE coefficients is $O(TOL^{-3})$, where TOL is the accuracy of the MC forward Euler approximation.

1.2 Objectives

The aim of this project is to construct an adaptive MC forward Euler algorithm for the approximation of $\mathbb{E}[g(X(T))]$ for singular observables and non-smooth SDE coefficients. Uniform time step discretization will then be more expensive than $O(TOL^{-3})$

for the class of first order numerical schemes. Hence, our goal is to choose the time steps adaptively, based on a new hybrid error estimate, to recover the optimal complexity $O(TOL^{-3})$. The work is thereafter extended to the MLMC setting introduced by Giles [12] where uniform time steps are used. The MLMC algorithm is a variance reduction technique which reduces the complexity from $O(TOL^{-3})$ in the single level setting to $O(TOL^{-2}(\log(TOL^{-1})^2))$. The aim is again to develop an MLMC adaptive algorithm for non-smooth problems to obtain the standard MLMC complexity $O(TOL^{-2}(\log(TOL^{-1})^2))$. Finally, we will apply the MLMC adaptive algorithm to the multidimensional setting to price options written on several underlying assets. Here again, our goal is to reach the standard MLMC complexity.

1.3 Related Work

MC forward Euler adaptivity was introduced in [1] where, assuming a smooth payoff g, the authors developed an adaptive algorithm based on an a posteriori estimate of the weak error. The MLMC setting is introduced in [12] where a clever variance reduction technique was developed using uniform time steps. However, this algorithm is not optimal for the case of a binary option whose payoff is the Heaviside function, see [13] and [14]. Adaptivity in the MLMC setting was derived in [15] using the a posteriori error expansion [1].

1.4 Contribution

In the present work, we introduce a new hybrid adaptive method for the approximation of the binary option price which recovers the complexity $O(TOL^{-3})$. A derivation of a new error expansion which includes prior information is given and put into an adaptive time stepping framework. The novelty of this work is that our hybrid MLMC adaptive method reaches the expected complexity $O(TOL^{-2}(\log(TOL^{-1})^2))$ using the first order forward Euler scheme. A similar result was obtained in [14] when considering the higher order Milstein scheme with smooth SDE coefficients and non-smooth payoff. Whereas extending the Milstein scheme to the multidimensional setting is not straightforward, our algorithm is easy to apply in the higher dimensional case.

1.5 Thesis Organization

The thesis is organized as follows. In Chapter 2, we describe the single level MC forward Euler method based on a posteriori error expansion. Numerical results for low regular problems are discussed. In chapter 3, a new hybrid error expansion is derived. The efficiency of the adaptive algorithm based on the hybrid error expansion is shown using several numerical results. Chapter 4 consists of two parts. First, an extension of the hybrid algorithm to the MLMC setting is described for one dimensional problems. Second, we introduced a technique to express a multidimensional problem in the form of one dimensional problem. Numerical results are provided to assess the performance of the algorithm in both settings.

Chapter 2

A Posteriori Adaptive Algorithm for SDE

The uniform time steps is the most used method for weak approximation of SDE. Under smoothness assumptions, this method gives optimal results. Adaptivity is an alternative algorithm in the case where uniform time steps do not work, especially for singular problems. A posteriori adaptive algorithm, see [1], is introduced in this chapter. Numerical results for both GBM and drift singularity problems are presented.

2.1 Problem Setting

Let us consider the following SDE

$$\begin{aligned} dX(t) &= a(t, X(t))dt + b(t, X(t))dW(t), \quad 0 < t < T, \qquad (2.1) \\ X(0) &= x_0, \end{aligned}$$

where $X(t; \omega)$ is a real stochastic process with randomness generated by the Wiener process $W(t; \omega)$ in $\mathbb{R}$. The functions $a(t, x) \in \mathbb{R}$ and $b(t, x) \in \mathbb{R}$ denote respectively the drift and the diffusion fluxes while x_0 is the initial condition.

The goal is to construct an approximation of the expected value $\mathbb{E}[g(X(T))]$ by a

MC forward Euler method in the final time T and for a given payoff function g. The forward Euler method approximates the unknown process X by $\bar{X}(t_n)$ as follows

$$\begin{aligned} \bar{X}(t_0) &= x_0, \\ \bar{X}(t_{n+1}) &= \bar{X}(t_n) + a(t_n, \bar{X}(t_n))\Delta t_n + b(t_n, \bar{X}(t_n))\Delta W_n, \;\; n = 0, 1, ..., N-1, \end{aligned} \tag{2.2}$$

where $\Delta t_n = t_{n+1} - t_n$ is the time step and $\Delta W_n = W(t_{n+1}) - W(t_n)$ is the Wiener increment. The aim is to choose the time step Δt_n and the number M of independent identically distributed samples $\bar{X}(.;\omega_j), j = 1, 2, ..., M$ such that for a given tolerance TOL

$$\left| \mathbb{E}[g(X(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\bar{X}(T;\omega_j)) \right| \leq TOL \tag{2.3}$$

holds with high probability and as few time steps and realizations as possible. The global error could be divided into two terms

$$\begin{aligned} \left| \mathbb{E}[g(X(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\bar{X}(T;\omega_j)) \right| &\leq \left| \mathbb{E}[g(X(T))] - \mathbb{E}[g(\bar{X}(T))] \right| \\ &+ \left| \mathbb{E}[g(\bar{X}(T))] - \frac{1}{M}\sum_{j=1}^{M} g(\bar{X}(T;\omega_j)) \right| \\ &= \epsilon_T + \epsilon_S, \end{aligned} \tag{2.4}$$

where ϵ_T is the time discretization error and ϵ_S is the statistical error.

The standard way to achieve this aim is to use uniform time steps. It can be proved that, for smooth SDE coefficients and a well behaved payoff function g, the uniform time steps is optimal in terms of computational cost, see [2] and [3]. A forward Euler scheme is first order accurate, we get

$$\epsilon_T = O(\Delta t) = O\left(\frac{1}{N}\right). \tag{2.5}$$

In other words, the number of time steps N required to achieve the accuracy TOL satisfies $N = O(TOL^{-1})$. The order of the sample size M in the MC method is $M = O(TOL^{-2})$. In fact, using the Central Limit Theorem (CLT) we have the following result

$$\lim_{M\to\infty} \mathbb{P}\left[\epsilon_S \leq C_C \hat{\sigma}_M\right] = 1 - \alpha_C, \tag{2.6}$$

where $\hat{\sigma}_M$ is the sample standard deviation of the MC estimator. The confidence constant C_C is related to the risk level α_C as follows

$$\Phi(C_C) = 1 - \alpha_C/2,$$

where Φ is the Cumulative Distribution Function (CDF) of a standard normal random variable. Hence, a robust approximation of the statistical error is to set

$$\epsilon_S \approx \; C_C \hat{\sigma}_M. \tag{2.7}$$

Finally, for smooth problems and using forward Euler with uniform time steps, the computational cost verifies

$$\text{Total cost} = O(TOL^{-3}). \tag{2.8}$$

Smoothness of the SDE coefficients and the payoff g are assumed in order to satisfy (2.5). For non-smooth SDE, the uniform time steps may not give the single level cost (2.8). A solution to recover optimality is to use adaptivity.

2.2 A Posteriori Adaptive Method

The main inspiration of adaptivity is the work by Talay and Tubaro [4], where the authors proved that for uniform time steps the time discretization error has an a priori expansion based on the unknown process X. Kloeden and Platen [2] extend the results of Talay and Tubaro on the existence of a leading-order a priori expansion. Szepessy, Tempone, and Zouraris in [1] developed an efficient adaptive algorithm using a posteriori error information. This a posteriori form was inspired from a posteriori adaptive methods for Ordinary Differential Equations (ODE), see [5] and [6]. The main new idea in [1] is the derivation of a posteriori error expansion with computable leading-order terms on the form

$$\epsilon_T = \mathbb{E}[g(X(T)) - g(\bar{X}(T))] = \mathbb{E}[\sum_{n=1}^{N} \rho(t_n)\Delta t_{n-1}^2] + h.o.t., \tag{2.9}$$

where $\rho(t_n)\Delta t_{n-1}^2$ are computable error indicators which provide information about the improvement of the time mesh and $\rho(t_n)$ is the error density defined as

$$\begin{aligned}\rho(t_n) = \frac{1}{2}\Big[&(a_t + aa_x + \frac{b^2}{2}a_{xx})\phi(t_{n+1}) \\ &+ (bb_t + abb_x + b^2 a_x + \frac{b^3}{2}b_{xx} + \frac{(bb_x)^2}{2})\phi'(t_{n+1}) \\ &+ (b^3 b_x)\phi''(t_{n+1})\Big].\end{aligned} \tag{2.10}$$

All the terms in the previous expression are evaluated on $(\bar{X}(t_n), t_n)$. ϕ , ϕ' and ϕ'' are the weight functions which solve a discrete dual backward problem, see [1]

$$\begin{aligned}&\phi(t_n) = c_x(\bar{X}(t_n), t_n)\phi(t_{n+1}), \quad t_n < T, \\ &\phi(T) = g_x(\bar{X}(T)),\end{aligned}$$

$$
\begin{aligned}
\phi'(t_n) &= c_x^2(\bar{X}(t_n), t_n)\phi'(t_{n+1}) + c_x(\bar{X}(t_n), t_n)\phi(t_{n+1}), \quad t_n < T, \\
\phi'(T) &= g_{xx}(\bar{X}(T)), \\
\phi''(t_n) &= c_x^3(\bar{X}(t_n), t_n)\phi''(t_{n+1}) + 3c_x(\bar{X}(t_n), t_n)c_{xx}(\bar{X}(t_n), t_n)\phi'(t_{n+1}) \\
&\quad + c_{xxx}(\bar{X}(t_n), t_n)\phi(t_{n+1}), \quad t_n < T, \\
\phi''(T) &= g_{xxx}(\bar{X}(T)),
\end{aligned}
$$

where $c(x, t_n) = a(x, t_n)\Delta t_n + b(x, t_n)\Delta W_n$.

It is important to notice that the payoff function is assumed to be smooth (at least three times differentiable) in order to compute the dual functions. So, what can we do for problems with non regular payoffs? One solution is to use mollification. The replacement of g by its mollified version will introduce another error term which will be estimated in the next section.

2.3 Mollification Procedure

For a given function g, the mollified function is defined as follows

$$
g_\delta(y) = \frac{1}{\delta}\int_{\mathbb{R}} g(y-z)H(\frac{z}{\delta})dz, \tag{2.11}
$$

where δ is the mollification parameter and H is a kernel density which has the following properties

1. $\int_{\mathbb{R}} H(z)dz = 1$,
2. $H(z) = H(-z), \quad \forall z \in \mathbb{R}$,
3. $H(z) \geq 0, \quad \forall z \in \mathbb{R}$.

From the convolution properties, we note that if the kernel density is in $C^p(\mathbb{R})$ then our approximate function g_δ is at least in $C^p_{(\mathbb{R})}$. Once we get the approximate payoff function, we need to analyze our approximation error

$$\epsilon_\delta = E[(g - g_\delta)(X(T))]. \tag{2.12}$$

For a given tolerance TOL, the aim is to find the order of δ which guarantees that ϵ_δ is bounded by TOL^2. By considering the Probability Distribution Function (PDF) ρ_{X_T} of $X(T)$, the error can be written as follows

$$\begin{aligned}
\epsilon_\delta &= \int_{\mathbb{R}} (g(y) - g_\delta(y))\rho_{X_T}(y)dy \\
&= \int_{\mathbb{R}} \rho_{X_T}(y) \left[\frac{1}{\delta}\int_{\mathbb{R}} (g(y) - g(y-z))H(\frac{z}{\delta})dz\right] dy \\
&= \int_{\mathbb{R}} \frac{1}{\delta}H(\frac{z}{\delta}) \left[\int_{\mathbb{R}} (g(y) - g(y-z))\rho_{X_T}dy\right] dz \\
&= \int_{\mathbb{R}} \frac{1}{\delta}H(\frac{z}{\delta}) \left[M(0) - M(-z)\right] dz,
\end{aligned}$$

where $M(z) = \int_{\mathbb{R}} g(y+z)\rho_{X_T}dy$. We note that if the probability density ρ_{X_T} is in $C^q(\mathbb{R})$ then $M(z)$ is at least in $C^q_{(\mathbb{R})}$. By making the change of variable $\omega = \frac{z}{\delta}$, we find

$$\epsilon_\delta = \int_{\mathbb{R}} H(\omega)(M(0) - M(-\delta\omega))d\omega. \tag{2.13}$$

Using the Taylor formula, we find

$$M(0) - M(-\delta\omega) = M^{'}(0)(\delta\omega) - \frac{1}{2}M^{''}(0)(\delta\omega)^2 + \frac{1}{6}M^{'''}(0)(\delta\omega)^3 + \cdots . \tag{2.14}$$

Substituting (2.14) in (2.13) , the approximation error becomes

$$\begin{aligned}\epsilon_\delta = & -\frac{1}{2}M''(0)\delta^2\left(\int_{\mathbb{R}}\omega^2 H(\omega)d\omega\right)\\ & -\frac{1}{24}M^{(4)}(0)\delta^4\left(\int_{\mathbb{R}}\omega^4 H(\omega)d\omega\right)\\ & -\frac{1}{6!}M^{(6)}(0)\delta^6\left(\int_{\mathbb{R}}\omega^6 H(\omega)d\omega\right)\\ & -\cdots . \end{aligned} \tag{2.15}$$

Since the odd terms are canceled using the second property of the kernel density, only even terms appear in the previous expansion. The leading term order in (2.15) is δ^2. Consequently, to bound the approximation error by TOL^2, the value of δ should satisfy

$$\delta = O(TOL). \tag{2.16}$$

For some problems, it is advisable to avoid that δ decreases with the same order as the tolerance. In fact, the weights in the a posteriori error expansion are sensitive to the mollification parameter δ. Therefore, for a small tolerance, the weak error may become large for the case of $\delta = O(TOL)$.

Richardson Extrapolation

The aim when using the Richardson extrapolation is to cancel the leading order term δ^2 and to find another approximation of $E[g(X(T))]$ such that the leading order term becomes δ^4. We consider the following notation for the rest of this part: $E_g = \mathbb{E}[g(X(T))]$ and $E_{g_\delta} = \mathbb{E}[g_\delta(X(T))]$. From (2.15), we have

$$E_g - E_{g_\delta} = c_1\delta^2 + c_2\delta^4 + \mathcal{O}(\delta^6). \tag{2.17}$$

Now, we rewrite the previous equation with δ replaced by $\frac{\delta}{2}$

$$E_g - E_{g_{\frac{\delta}{2}}} = c_1 \frac{\delta^2}{4} + c_2 \frac{\delta^4}{16} + \mathcal{O}(\delta^6). \tag{2.18}$$

By multiplying (2.18) by 4 and subtracting (2.17), we find

$$E_g = \frac{4E_{g_{\frac{\delta}{2}}} - E_{g_\delta}}{3} + \hat{c}_2 \delta^4 + \mathcal{O}(\delta^6). \tag{2.19}$$

We remark that the range of δ-values for which the presented extrapolation technique improves the convergence rate depends on the size of the constants c_1 and c_2 in (2.17). From (2.19), we get another mollified function of g which is

$$h_\delta = \frac{4g_{\frac{\delta}{2}} - g_\delta}{3}. \tag{2.20}$$

By using the new approximation, the leading term of the error becomes δ^4. Consequently, the value of δ to be chosen to bound the error by TOL^2 satisfies

$$\delta = O(\sqrt{TOL}). \tag{2.21}$$

As an example, let us mollify the Heaviside function. The Heaviside function is defined as

$$g(y) = \mathbf{1}_{y \geq 0}, \tag{2.22}$$

and

$$g_\delta(y) = \frac{1}{\delta} \int_{\mathbb{R}} g(y-z) H(\frac{z}{\delta}) dz. \tag{2.23}$$

Using for example the standard Gaussian distribution density as a Kernel

$$H(z) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}),$$

the approximate function becomes

$$\begin{aligned} g_\delta(y) &= \frac{1}{\delta} \int_{\mathbb{R}} g(y-z) H(\frac{z}{\delta}) dz \\ &= \frac{1}{\delta} \int_{\mathbb{R}} \mathbf{1}_{y-z\geq 0} \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2\delta^2}) dz \\ &= \frac{1}{\delta} \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2\delta^2}) dz \\ &= \int_{-\infty}^{\frac{y}{\delta}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2}) du. \end{aligned}$$

Finally, the Heaviside approximate function is

$$g_\delta(y) = \Phi(\frac{y}{\delta}), \tag{2.24}$$

where Φ is the CDF of a standard Gaussian random variable.

2.4 Numerical Implementation

Through the mollification procedure introduced in the previous section, we are able to apply the a posteriori adaptive algorithm using the replacement of g by g_δ. In [1], the authors presented two adaptive a posteriori algorithms: one using deterministic time steps and another where the time stepping stochastically depends on the realization. The stochastic time stepping algorithm is more suitable in our case than the deterministic one since we are dealing with problems where singularities are path dependent. Therefore, we will only present the stochastic time stepping method.

For a given tolerance TOL for which we desire that (2.3) is satisfied, we split the tolerance into a time discretization tolerance TOL_T and a statistical tolerance TOL_S such that

$$TOL_T + TOL_S = TOL. \tag{2.25}$$

The average number of time steps is chosen through the control of the time discretization error which has the a posteriori expansion (2.9). This control is based on refinement and stopping criteria. Consider a grid with increments $\Delta t = (\Delta t_0, ..., \Delta t_{N-1})$ which depends on the realization ω. This grid is accepted if the following stopping criterion is satisfied

$$\max_{1 \leq n \leq N} \rho(t_n) \Delta t_{n-1}^2 \leq C_s \frac{TOL_T}{\bar{N}}, \tag{2.26}$$

where $\bar{N}$ is the expected number of time steps. As long as the previous condition is not fulfilled, the n^{th} interval is refined if the following criterion is not satisfied

$$\rho(t_n) \Delta t_{n-1}^2 \leq C_R \frac{TOL_T}{\bar{N}}, \tag{2.27}$$

where C_R and C_S are constants $C_S > C_R$, see [7] and [8]. In our numerical tests, $C_R = 2$ and $C_S = 5$.

For M random samples $\{Y_i\}_{i=1}^{M}$, the sample average is denoted by

$$\mathcal{AV}_M[Y] = \frac{1}{M} \sum_{i=1}^{M} Y_i. \tag{2.28}$$

Similarly, the sample variance is denoted by

$$\nu_M[Y] = \frac{1}{M-1} \sum_{i=1}^{M} (Y_i - \mathcal{AV}_M[Y])^2. \tag{2.29}$$

The number of samples M in the MC method is chosen via the control of the statistical error which is approximated in (2.7). Bounding ϵ_S by TOL_S gives the following stopping criterion

$$\hat{\sigma}_M \leq \frac{TOL_S}{C_C}, \tag{2.30}$$

where $\hat{\sigma}_M$ is the sample standard deviation of the MC estimator, see (2.28) and (2.29), and C_C is the confidence constant, see (2.6).

The algorithm is composed of an inner and an outer loop. The outer loop starts with an initial number of realizations M which is doubled until (2.30) is satisfied. The inner loop constructs M paths by the forward Euler scheme. For each realization, it starts with an initial grid. Using the refinement criterion (2.27) and the stopping criterion (2.26), the corresponding grids are constructed. The a posteriori adaptive technique is described in Algorithm 1 and 2:

Algorithm 1 A Posteriori Adaptive SLMC Algorithm

```
Inputs: $M$, $TOL_S$
Outputs: $\mu$
Compute $\mu$ and $\hat{\sigma}$ by calling Generate Realizations( $\Delta t_{-1}$, $M$, $TOL_T$, $\bar{N}$)
while (2.30) is violated do
    Update the average number of time steps $\bar{N}$
    Update the number of realizations $M = 2M$
    Compute $\mu$ and $\hat{\sigma}$ by calling Generate Realizations( $\Delta t_{-1}$, $M$, $TOL_T$, $\bar{N}$)
end while
```

2.5 Numerical Results

This section presents simulation results using the algorithm described in the previous section. We will consider the binary option whose payoff is the Heaviside function with strike K. Its mollification, which will be used in the a posteriori algorithm, is

$$g_\delta(x) = \frac{4\Phi(2(x-K)/\delta) - \Phi((x-K)/\delta)}{3}, \tag{2.31}$$

Algorithm 2 Generate Realizations

```
**Inputs:** $\Delta t_{-1}$, $M$, $TOL_T$, $\bar{N}$
**Outputs**: $\mu$, $\hat{\sigma}$
for $j = 1, 2, ..., M$ do
    Start from the initial grid $\Delta t_{-1}$ and compute the error indicators.
    while (2.26) is violated do
        Refine the grid by
        for Each interval on the grid do
            if (2.27) is not satisfied then
                Divide the interval into two equal parts
            end if
        end for
        Compute the error indicators on the new refined grid
    end while
    Compute the forward Euler path on the constructed grid
    $\mu = \mu + g_\delta(\bar{X}(T))$
end for
$\mu = \mu/M$ and estimate $\hat{\sigma}$ by sample standard deviation (2.29)
```

where Φ is the CDF of a standard Gaussian.

2.5.1 GBM Problem

The SDE for a GBM problem is defined as

$$\begin{aligned} dX(t) &= rX(t)dt + \sigma X(t)dW(t), \quad 0 < t < T, \qquad (2.32) \\ X(0) &= x_0, \end{aligned}$$

where r is the interest rate and σ is the volatility. The GBM is frequently used in mathematical finance to model stock prices in the Black Scholes model [9]. The problem parameters are $r = 0.05$, $\sigma = 0.2$, $T = 1$, $K = 1$, and $x_0 = 0.9$.

For a range of tolerances, we aim to check if the accuracy condition (2.3) is fulfilled. In Figure 2.1, we plot the global error function of the tolerance. It can be seen that the accuracy condition is satisfied as the global error is always below the tolerance line. Since the red curve has the same order as the blue one, the number of MC samples has the expected order $O(TOL^{-2})$.

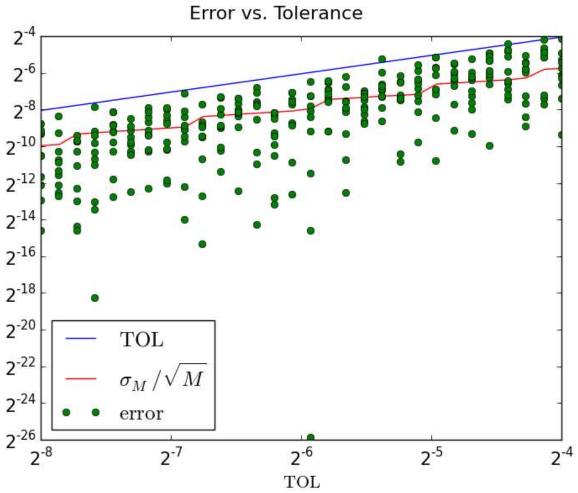

Figure 2.1: Accuracy plot for the a posteriori adaptive SLMC algorithm with the GBM problem and the binary option payoff.

In the second simulation, we are interested in the computational cost measured as the total number of time steps in all MC realizations as function of the tolerances. From Figure 2.2, we point out that the cost is asymptotically $O(TOL^{-3.5})$. This illustrates that the algorithm fails to give the optimal computational cost which is $O(TOL^{-3})$. The jumps observed in Figure 2.2 could be explained by the non-smooth update of the number of realizations M, which is doubled if the stopping rule (2.30) is not fulfilled.

2.5.2 Drift Singularity Problem

We consider a non-smooth SDE where we have a known singularity point on the drift

$$\begin{aligned} &dX_t = r f_\alpha X_t dt + \sigma X_t dW_t, \quad t < T, \\ &X(0) = x_0, \end{aligned} \tag{2.33}$$

where $f_\alpha(t) = |t-\alpha|^{-0.5}$ and α is the singularity, $\alpha = \frac{1}{\pi}$. To avoid an eventual blow up on the numerical implementation due to the singularity, we consider the following

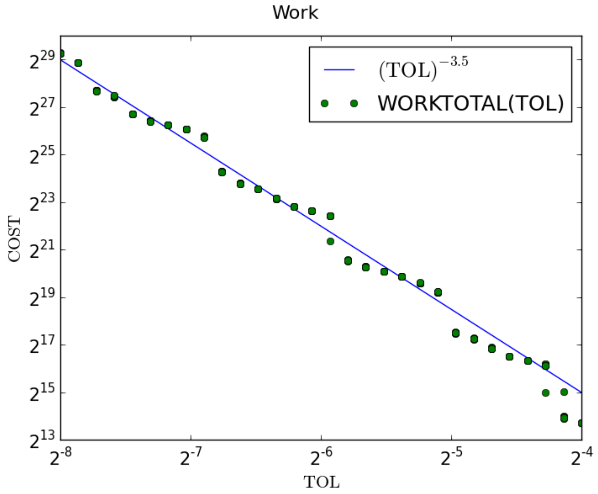

Figure 2.2: Total computational cost for the a posteriori adaptive SLMC algorithm with the GBM problem and the binary option payoff.

modified drift

$$\bar{f}_\alpha(t_n) = \begin{cases} f_\alpha(t_n), & \text{if } f_\alpha(t_n)\Delta t_n < 2\Delta t_n^{1/2}, \\ 2\Delta t_n^{-1/2}, & \text{otherwise.} \end{cases} \tag{2.34}$$

For the same purpose, we do the modification on the first derivative as follows

$$\bar{f}_\alpha(t_n) = \begin{cases} f'_\alpha(t_n), & \text{if } f_\alpha(t_n)\Delta t_n < 2\Delta t_n^{1/2}, \\ f'_\alpha(s^*), & \text{otherwise,} \end{cases} \tag{2.35}$$

where $s^* = \frac{1}{4}\Delta t_n$. Even if these two modifications are not derived from a rigorous theoretical proof, we will show in the numerical part that they still give convincing results and do not introduce a non controllable error.

Comparing to the GBM problem with the Heaviside payoff, this example is more challenging since we add a singularity on the SDE to the non-smoothness of g. We use the same mollified function g_δ as in (2.31) and the same problem parameter values as in the GBM case. The accuracy plot for this example is shown in Figure 2.3. It is clear that again the accuracy is satisfied. Moreover, the number of MC realizations has the expected order, i.e., $O(TOL^{-2})$.

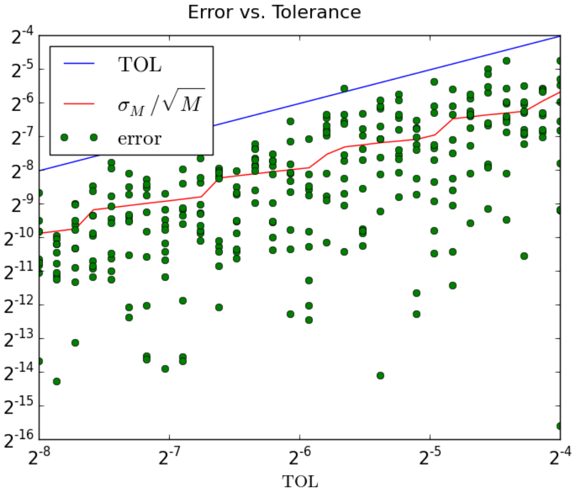

Figure 2.3: Accuracy plot for the a posteriori adaptive SLMC algorithm with the drift singularity problem and the binary option payoff.

Let us examine the computational cost for this problem. In Figure 2.4, we plot the total cost against the range of tolerances. Again, the algorithm has a complexity of order $O(TOL^{-3.5})$ which is different from the expected complexity.

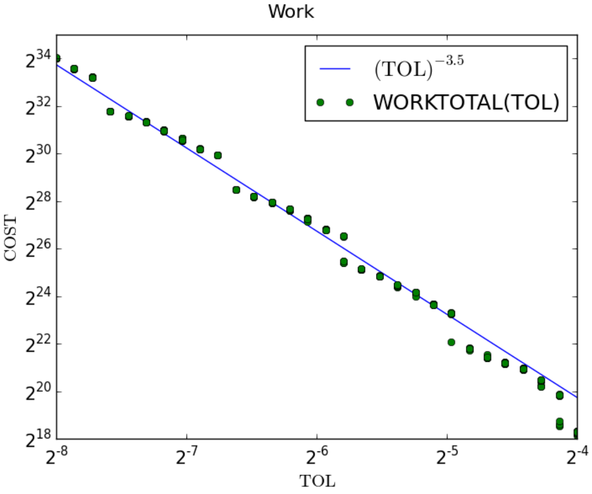

Figure 2.4: Total computational cost for the a posteriori adaptive SLMC algorithm with the drift singularity problem and the binary option payoff.

To conclude, the a posteriori adaptive algorithm does not give the optimal single level computational cost for either GBM or drift singularity problems. This failure is due to the mollification parameter δ. In fact, the dual functions in the a posteriori error expansion are strongly influenced by this parameter which gives, in the final time T, a large value of the error density in the region where we are close to the strike K. Hence, solving the backward problem may result in a grid with too small time steps.

Chapter 3

Hybrid Adaptive Algorithm for SDE

A hybrid adaptive algorithm is presented in this chapter. A derivation of a new hybrid error expansion is described. The expansion is similar to the a posteriori one with a small modification on the weight functions. A brief theoretical proof will show that this new introduced method will give the expected cost $O(TOL^{-3})$ for the GBM and the drift singularity problems. Numerical tests are presented at the end of this chapter to confirm our theoretical results.

3.1 Hybrid Error Expansion

Let $\bar{X}(t_n)$ denote the forward Euler numerical realization (2.2) of the SDE (2.1) on the grid $\{\Delta t_n\}_{n=0}^{N-1}$, and

$$u(x,t) = \mathbb{E}[g(X(T))|X(t) = x].$$

Using the mean value theorem, the weak error can then be written as

$$\begin{aligned}\mathbb{E}[g(X(T))-g(\bar{X}(T))] &= \sum_{n=1}^{N}\mathbb{E}\left[u(X(t_n;\bar{X}(t_{n-1}),t_{n-1}),t_n)-u(\bar{X}(t_n),t_n)\right]\\ &= \sum_{n=1}^{N}\mathbb{E}\left[u_x(\bar{X}(t_n),t_n)e_n+u_{xx}(\bar{X}(t_n)+s_ne_n,t_n)\frac{e_n^2}{2}\right], \quad (3.1)\end{aligned}$$

where $s_n \in [0,1]$ and e_n is the local error defined as :

$$e_n = X(t_n;\bar{X}(t_{n-1}),t_{n-1})-\bar{X}(t_n).$$

Using Itô formula, we obtain

$$\begin{aligned}e_n &= \int_{t_{n-1}}^{t_n} a(t,X(t))-a(t_{n-1},\bar{X}(t_{n-1}))dt+\int_{t_{n-1}}^{t_n} b(t,X(t))-b(t_{n-1},\bar{X}(t_{n-1}))dW_t\\ &= \Delta a_{n-1}+\Delta b_{n-1}\end{aligned}$$

where

$$\begin{aligned}\Delta a_{n-1} &= \int_{t_{n-1}}^{t_n}\int_{t_{n-1}}^{t_n}(a_t+a_xa+\frac{a_{xx}}{2}b^2)(s,X(s))dsdt,\\ \Delta b_{n-1} &= \int_{t_{n-1}}^{t_n}\int_{t_{n-1}}^{t}(a_xb)(s,X(s))dW_sdt+(b_t+b_xa+\frac{b_{xx}}{2}b^2)(s,X(s))dsdW_t\\ &\quad +(b_xb)(s,X(s))dW_sdW_t.\end{aligned}$$

So the local error term has the expression $e_n = \Delta a_{n-1}+\Delta b_{n-1}$. Our objective is to prove that under boundedness assumptions on $u_x(x,t)$, $u_{xx}(x,t)$ and $u_{xxx}(x,t)$ for all $(x,t)\in\mathbb{R}\times[0,T]$, we can develop a hybrid error density for weak approximations of

SDE. Using Taylor expansion for the first term in (3.1) to get

$$\begin{aligned} u_x(\bar{X}(t_n),t_n)e_n &= u_x(\bar{X}(t_n),t_n)\Delta a_{n-1} + \Big[u_x\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right) \\ &+ u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right) b_{n-1}\Delta W_{n-1} \\ &+ u_{xxx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right) \frac{1}{2} b_{n-1}^2 \Delta W_{n-1}^2 + h.o.t.\Big]\Delta b_{n-1}, \end{aligned} \tag{3.2}$$

where $b_{n-1} = b(t_{n-1}, \bar{X}(t_{n-1}))$. The first term in (3.2) satisfies

$$\begin{aligned} &\mathbb{E}\left[u_x\left(\bar{X}(t_n), t_n\right)\Delta a_{n-1}\right] \\ &= \mathbb{E}\left[u_x\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\Delta a_{n-1} + h.o.t.\right] \\ &= \mathbb{E}\left[u_x\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\mathbb{E}\left[\Delta a_{n-1}|\mathcal{F}_{t_{n-1}}\right] + h.o.t.\right] \\ &= \mathbb{E}\left[u_x(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n)\frac{a_t + aa_x + a_{xx}b^2/2}{2}(\bar{X}(t_{n-1}), t_{n-1})\Delta t_{n-1}^2\right] \\ &+ \mathbb{E}\left[o(\Delta t_{n-1}^2)\right]. \end{aligned} \tag{3.3}$$

Let us consider the second summand of (3.2). Its first term is zero,

$$\begin{aligned} &\mathbb{E}\left[u_x\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\Delta b_{n-1}\right] \\ &= \mathbb{E}\left[u_x\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\mathbb{E}\left[\Delta b_{n-1}|\mathcal{F}_{t_{n-1}}\right]\right] \\ &= 0. \end{aligned}$$

For the second term,

$$\begin{aligned} &\mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right) b_{n-1}\Delta W_{n-1}\Delta b_{n-1}\right] \\ &= \mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right) b_{n-1}\mathbb{E}\left[\Delta W_{n-1}\Delta b_{n-1}|\mathcal{F}_{t_{n-1}}\right]\right] \\ &= \mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{a_x b^2 + b_t b + ab_x b + b_{xx}b^3/2}{2}(\bar{X}(t_{n-1}), t_{n-1})\Delta t_{n-1}^2\right] \\ &+ \mathbb{E}\left[o(\Delta t_{n-1}^2)\right]. \end{aligned} \tag{3.4}$$

For the third term, we apply Itô isometry formula, see [2], twice to get

$$\begin{aligned}\mathbb{E}\left[\Delta W_{n-1}^2 \Delta b_{n-1}\right] =&\mathbb{E}\left[\int_{t_{n-1}}^{t_n}\int_{t_{n-1}}^{t_n} 1 dW_s dW_t \int_{t_{n-1}}^{t_n}\int_{t_{n-1}}^{t} (b_x b)(s, X(s)) dW_s dW_t\right]\\ =& \int_{t_{n-1}}^{t_n} \mathbb{E}\left[\int_{t_{n-1}}^{t_n} 1 dW_s \int_{t_{n-1}}^{t} (b_x b)(s, X(s)) dW_s\right] dt\\ =& \int_{t_{n-1}}^{t_n}\int_{t_{n-1}}^{t} \mathbb{E}[(b_x b(s, X(s)))] ds dt.\end{aligned}$$

Consequently,

$$\begin{aligned}&\mathbb{E}\left[u_{xxx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{1}{2}b(t_{n-1}, \bar{X}(t_{n-1}))^2 \Delta W_{n-1}^2 \Delta b_{n-1}\right]\\ &\quad= \mathbb{E}\left[u_{xxx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{1}{2}b_{n-1}^2 \mathbb{E}[\Delta W_{n-1}^2 \Delta b_{n-1} | \mathcal{F}_{t_{n-1}}]\right]\\ &\quad= \mathbb{E}\left[u_{xxx}(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n)\frac{b^3 b_x}{4}(\bar{X}(t_{n-1}, t_{n-1}))\Delta t_{n-1}^2\right] + \mathbb{E}\left[o(\Delta t_{n-1}^2)\right].\end{aligned} \tag{3.5}$$

where $\mathcal{F}_{t_{n-1}}$ is the filtration generated by the Wiener process up to the time t_{n-1}. For the second term in (3.1), we have

$$u_{xx}\left(\bar{X}(t_n) + s_n e_n, t_n\right)\frac{e_n^2}{2} = u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{e_n^2}{2} + h.o.t.$$

Hence,

$$\begin{aligned}&\mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) + s_n e_n, t_n\right)\frac{e_n^2}{2}\right]\\ &\quad= \mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{e_n^2}{2}\right]\\ &\quad= \mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\mathbb{E}[\frac{e_n^2}{2} | \mathcal{F}_{t_{n-1}}]\right]\\ &\quad= \mathbb{E}\left[u_{xx}\left(\bar{X}(t_n) - b_{n-1}\Delta W_{n-1}, t_n\right)\frac{(bb_x)^2}{4}(\bar{X}(t_{n-1}), t_{n-1})\Delta t_{n-1}^2)\right] + \mathbb{E}\left[o(\Delta_{t_{n-1}}^2)\right].\end{aligned} \tag{3.6}$$

Finally, using (3.3), (3.4), (3.5) and (3.6) the time discretization error is written as follows

$$
\begin{aligned}
\Big|\mathbb{E}\left[g(X(T)) - g(\bar{X}(T))\right]\Big| &= \Big|\sum_{n=1}^{N}\mathbb{E}\left[u_x(\bar{X}(t_n),t_n)e_n + u_{xx}(\bar{X}(t_n)+s_ne_n,t_n)\frac{e_n^2}{2}\right]\Big| \\
&\le \sum_{n=1}^{N}\Big|\mathbb{E}\left[u_x(\bar{X}(t_n)-b_{n-1}\Delta W_{n-1},t_n)\frac{a_t + aa_x + a_{xx}b^2/2}{2}\Delta t_{n-1}^2\right]\Big| \\
&+ \Big|\mathbb{E}\left[u_{xx}(\bar{X}(t_n)-b_{n-1}\Delta W_{n-1},t_n)\frac{a_xb^2 + b_tb + ab_xb + (b_{xx}b^3 + (b_xb)^2)/2}{2}\Delta t_{n-1}^2\right]\Big| \\
&+ \Big|\mathbb{E}\left[u_{xxx}(\bar{X}(t_n)-b_{n-1}\Delta W_{n-1},t_n)\frac{b^3b_x}{4}\Delta t_{n-1}^2\right]\Big| + \Big|\mathbb{E}\left[o(\Delta t_{n-1}^2)\right]\Big|,
\end{aligned}
$$

and the boundedness of u_x, u_{xx} and u_{xxx} with respect to the first space variable x leads to

$$
\begin{aligned}
\Big|\mathbb{E}\left[g(X(T)) - g(\bar{X}(T))\right]\Big| &\le \sum_{n=1}^{N}\|u_x(\cdot,t_n)\|_\infty\mathbb{E}\left[\Big|\frac{a_t + aa_x + a_{xx}b^2/2}{2}\Big|\Delta t_{n-1}^2\right] \\
&+ \|u_{xx}(\cdot,t_n)\|_\infty\mathbb{E}\left[\Big|\frac{a_xb^2 + b_tb + ab_xb + b_{xx}b^3/2 + (b_xb)^2/2}{2}\Big|\Delta t_{n-1}^2\right] \\
&+ \|u_{xxx}(\cdot,t_n)\|_\infty\mathbb{E}\left[\Big|\frac{b^3b_x}{4}\Big|\Delta t_{n-1}^2\right] + \Big|\mathbb{E}\left[o(\Delta t_{n-1}^2)\right]\Big| \\
&= \sum_{n=1}^{N}\mathbb{E}\left[\rho(t_n)\Delta t_{n-1}^2\right] + \Big|\mathbb{E}\left[o(\Delta t_{n-1}^2)\right]\Big|,
\end{aligned}
$$

where ρ is the error density

$$
\begin{aligned}
\rho(t_n) &= \|u_x(\cdot,t_n)\|_\infty\Big|\frac{a_t + aa_x + a_{xx}b^2/2}{2}(\bar{X}(t_{n-1}),t_{n-1})\Big| \\
&+ \|u_{xx}(\cdot,t_n)\|_\infty\Big|\frac{a_xb^2 + b_tb + ab_xb + b_{xx}b^3/2 + (b_xb)^2/2}{2}(\bar{X}(t_{n-1}),t_{n-1})\Big| \\
&+ \|u_{xxx}(\cdot,t_n)\|_\infty\Big|\frac{b^3b_x}{4}(\bar{X}(t_{n-1}),t_{n-1})\Big|. \qquad (3.7)
\end{aligned}
$$

This new hybrid error expansion has almost the same form as the a posteriori error expansion derived in [1]. In other words, both the error density in the hybrid and the a posteriori expansions could be written as

$$\rho(t_n) = \sum local\ error(t_n) \times weight(ts_n).$$

We notice that the only difference between the a posteriori and the hybrid expansions is the weights. In fact, in the a posteriori expansion the weights were bounded by the dual functions ϕ, ϕ' and ϕ'' which solve pathwise the discrete dual backward problems. In the hybrid error expansion, the error density weights are the bounds for the first, the second and the third derivatives of u and are usually smoother than their ϕ counterparts.

From the hybrid error expansion, we conclude that if we have reasonable upper bounds for $\|u_x\|_\infty$, $\|u_{xx}\|_\infty$ and $\|u_{xxx}\|_\infty$, then we may construct a reasonably efficient hybrid adaptive method for weak approximations of SDEs. In practice, the upper bounds will be derived through rough a priori analysis of the first, the second and the third variations of the SDE paths and the Fokker-Planck (FP) density.

An advantage of the hybrid algorithm compared to the a posteriori one is the following: in the a posteriori method, we need to solve backward the discrete dual problem and then we go forward to construct paths. In the hybrid method, paths are constructed by going only forward in time. Furthermore, in the a posteriori method we need to start with an initial grid which is refined using the whole path information in order to compute backward the dual functions. For the hybrid method, on the other hand, the time step is directly constructed without any initial guess.

3.2 Error Density: GBM and Drift Singularity

GBM Problem

In this section, we will derive the error density expression for the GBM problem

$$\begin{aligned} dX(t) &= rX(t)dt + \sigma X(t)dW(t), \quad 0 < t < T, \\ X(0) &= x_0. \end{aligned} \tag{3.8}$$

The GBM is a good example to start with in our work due to the prior information it contains. In fact, its FP density of $X(T; x, t)$ takes the explicit form

$$p(y, T; x, t) = \frac{1}{y\sigma\sqrt{2\pi(T-t)}} e^{-\frac{(log(y/x)-(r-\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}}. \tag{3.9}$$

Let us define $X'(T; x, t) = \partial_x X(T; x, t)$ which represents the path's first variation of $X(T; x, t)$. the first variation is governed by the following SDE

$$\begin{aligned} dX'(s; x, t) &= a'(X(s; x, t), s)X'(s; x, t)ds + b'(X(s; x, t), s)X'(s; x, t)dW_s, \\ &= rX'(s; x, t)ds + \sigma X'(s; x, t)dW_s, \quad s > t, \end{aligned}$$

with initial condition $X'(t; x, t) = 1$. When considering the GBM problem, $X'(s; x, t)$ follows the same SDE as $X(s; x, t)$ with

$$X'(s; x, t) = X(s; 1, t) = \frac{X(s; x, t)}{x}.$$

We consider the binary option whose payoff g is the Heaviside function (2.22) with strike K. Let us derive bounds for the derivatives of u. By definition we have

$$\begin{aligned} u_x(x,t) &= \partial_x \mathbb{E}[g(X(T;x,t))] \\ &= \mathbb{E}[g'(X(T;x,t))\partial_x X(T;x,t)] \\ &= \mathbb{E}[\delta_{X(T;x,t)-K}\partial_x X(T;x,t)] \\ &= \mathbb{E}[X'(T;x,t)|X(T;x,t)=K]p(K,T;x,t). \end{aligned}$$

By introducing the variable $z = log(K/x)$ and using (3.9), we obtain

$$\begin{aligned} u_x(x,t) &= \frac{exp(z)}{K\sigma\sqrt{2\pi(T-t)}} e^{-\frac{(z-(r-\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}} \\ &= e^{r(T-t)} \frac{1}{K\sigma\sqrt{2\pi(T-t)}} e^{-\frac{(z-(r+\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}} \\ &=: \hat{u}(z,t) \end{aligned}$$

Consequently, it follows that

$$\begin{aligned} \max_{x\in\mathbb{R}^+} |u_x(x,t)| = \max_{z\in\mathbb{R}} |\hat{u}(z,t)| \\ &= \frac{e^{r(T-t)}}{K\sigma\sqrt{2\pi(T-t)}} \\ &= \frac{C_x(t,T)}{\sqrt{T-t}}. \end{aligned}$$

Now, let us find a bound for the second derivative of u. Using the chain rule we get

$$\begin{aligned} u_{xx}(x,t) &= \hat{u}_z(z,t)\frac{\partial z}{\partial x} \\ &= e^{r(T-t)} \frac{(z-(r+\sigma^2/2)(T-t))}{K^2\sigma^3(T-t)^{3/2}\sqrt{2\pi}} e^{z-\frac{(z-(r+\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}} \\ &= \frac{e^{(2r+\sigma^2)(T-t)}}{K^2\sigma^2(T-t)\sqrt{\pi}} \left(\frac{(z-(r+3\sigma^2/2)(T-t))}{\sqrt{2\sigma^2(T-t)}} + \frac{\sigma\sqrt{T-t}}{\sqrt{2}} \right) e^{-\frac{(z-(r+3\sigma^2/2)(T-t))}{2\sigma^2(T-t)}}, \end{aligned}$$

and thus that

$$\begin{aligned}\|u_{xx}(\cdot,t)\|_\infty &= sup_{z\in\mathbb{R}}|\hat{u}_z(z,t)\frac{\partial z}{\partial x}| \\ &\leq \frac{e^{(2r+\sigma^2)(T-t)}}{K^2\sigma^2(T-t)\sqrt{\pi}}\left(sup_{x\in\mathbb{R}^+}(xe^{-x^2})+\frac{\sigma\sqrt{T-t}}{\sqrt{2}}\right) \\ &= \frac{e^{(2r+\sigma^2)(T-t)}}{K^2\sigma^2(T-t)\sqrt{2\pi}}\left(e^{-1/2}+\sigma\sqrt{T-t}\right) \\ &= \frac{C_{xx}(t,T)}{T-t}.\end{aligned}$$

For the third derivative of u, a similar computation gives

$$\|u_{xxx}(\cdot,t)\|_\infty \leq \frac{C_{xxx}(t,T)}{(T-t)^{3/2}}.$$

Summing up, we find upper bounds for the derivatives of u which will be used in the definition of the error density. Let us include these upper bounds expression into the error density expression. For the GBM problem, the error density has the following upper bound

$$\begin{aligned}\rho(t_n) \leq \frac{1}{2}\Big(&\frac{C_x(t_n,T)}{\sqrt{T-t_n}}r^2|\bar{X}(t_{n-1})| + \frac{C_{xx}(t_n,T)}{T-t_n}(2r\sigma^2+\sigma^4/2)\bar{X}(t_{n-1})^2 \\ &+ \frac{C_{xxx}(t_n,T)}{2(T-t_n)^{3/2}}\sigma^4|\bar{X}(t_{n-1})^3|\Big).\end{aligned}$$

Now, let us determine an upper bound on the time steps Δt such that the weak approximation criterion

$$|\mathbb{E}[g(X(T))-g(\bar{X}(T))]| \leq TOL \tag{3.10}$$

is fulfilled for a given tolerance TOL. Using our new hybrid error expansion, the criterion (3.10) is replaced by

$$\mathbb{E}[\sum_{n=1}^{N} \rho(t_n)\Delta t_{n-1}^2] \leq TOL.$$

The above summands are known as error indicators. In terms of computational cost, the optimal way the criterion (3.10) is met is by ensuring that all error indicator contributions are of the same size, i.e.,

$$\rho(t_1)\Delta t_0^2 = \rho(t_2)\Delta t_1^2 = ... = \rho(t_N)\Delta t_{N-1}^2 = \frac{TOL}{N}.$$

In addition to estimating the weak error, the error indicators give information on where to refine in order to get the optimal mesh. Equal contributions by all the error indicators leads to

$$\Delta t_{n-1} = O\left(\sqrt{\frac{TOL}{\rho(t_n)N}}\right) = O\left(\sqrt{\frac{TOL(T-t_n)^{3/2}}{N}}\right), \quad \forall n.$$

The previous expression was derived by assuming that $\bar{X}(t_{n-1}) = O(1)$ and neglecting the last error indicator which will blow up since the error indicator at the final time is $O((T-t)^{-3/2})$. Finally, using the Cauchy Schwarz inequality, we obtain a bound on the average number of time steps

$$\begin{aligned}
\mathbb{E}[N] &= \mathbb{E}[\int_0^T \Delta t(s)^{-1} ds] \\
&= O\left(TOL^{-1/2}\mathbb{E}[\sqrt{N}\int_0^T (T-s)^{-3/4} ds]\right) \\
&= O\left(TOL^{-1/2}\sqrt{\mathbb{E}[N]}\right),
\end{aligned}$$

which implies that

$$\mathbb{E}[N] = O(TOL^{-1}).$$

This result is a brief theoretical proof confirming that our hybrid method will achieve the computational cost $O(TOL^{-3})$, which is the aim when using adaptivity. It is important to see that the error density at the final time has the order $O((T-t)^{-3/2})$. Therefore, an eventual blow up at the final time will occur. One way to solve this issue is to mollify our payoff function by a convolution with a kernel. In Section 2.3, we have presented a mollification procedure. In the present method, we need to ensure boundedness for $\|u_x(\cdot,t)\|_\infty$, $\|u_{xx}(\cdot,t)\|_\infty$ and $\|u_{xxx}(\cdot,t)\|_\infty$. To do that, we will use a kernel with compact support instead of a Gaussian kernel. One way to construct a compact support n^{th} order differentiable kernel is to convolve the window function to itself n times , see [10]. The resulting approximate function g_δ is presented in Appendix B.

Error density with a mollified payoff version

As it was mentioned above, the aim when using this mollification is to provide upper bounds for $\|u_x(\cdot,t)\|_\infty$, $\|u_{xx}(\cdot,t)\|_\infty$ and $\|u_{xxx}(\cdot,t)\|_\infty$ which do not blow up at the final time. Let us consider the slightly altered problem where g_δ is used instead of g. The first variation of u is then

$$\begin{aligned} u_x(x,t) &= \partial_x \mathbb{E}[g_\delta(X(T;x,t))] \\ &= \mathbb{E}[g_\delta^{'}(X(T;x,t))X'(T;x,t)] \\ &= \mathbb{E}[H_\delta(X(T;x,t))X'(T;x,t)]. \end{aligned}$$

For GBM problem we have shown that

$$X'(s;x,t) = X(s;1,t) = \frac{X(s;x,t)}{x}.$$

Consequently

$$\begin{aligned} u_x(x,t) &= \int_{\mathbb{R}_+} H_\delta(y)\frac{y}{x}P(T,y;x,t)dy \\ &= \int_{K-\delta}^{K+\delta} H_\delta(y)\frac{1}{x\sigma\sqrt{2\pi(T-t)}}e^{-\frac{(log(y/x)-(r-\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}}dy. \end{aligned}$$

From the previous expression we see why the kernel is chosen to have compact support. In fact, the term inside the integral was equal to $u_x(x,t)$ before the mollification with K is now replaced by y. In other words, to obtain bounds for $\|u_x(\cdot,t)\|_\infty$, $\|u_{xx}(\cdot,t)\|_\infty$ and $\|u_{xxx}(\cdot,t)\|_\infty$ we follow the same procedure as in the previous section with K replaced by $K-\delta$.

$$\begin{aligned} \|u_x(\cdot,t)\|_\infty &\le \frac{C_{x,\delta}(T,t)}{\sqrt{T-t}}, \\ \|u_{xx}(\cdot,t)\|_\infty &\le \frac{C_{xx,\delta}(T,t)}{T-t}, \\ \|u_{xxx}(\cdot,t)\|_\infty &\le \frac{C_{xxx,\delta}(T,t)}{(T-t)^{3/2}}. \end{aligned}$$

The mollification will allow us to find alternative bounds and so avoid an eventual blow up at the final time. For a GBM problem we have the following property

$$\int_{\mathbb{R}_+} yP(T,y;x,t)dy = xe^{r(T-t)}.$$

This leads to the bound

$$
\begin{aligned}
|u_x(x,t)| &= |\mathbb{E}[H_\delta(X(T;x,t))\frac{X(T;x,t)}{x}]| \\
&= |\int_{\mathbb{R}_+} H_\delta(y)\frac{y}{x}P(T;x,t)dy| \\
&\le \frac{1}{x\delta}\int_{\mathbb{R}_+} yP(T;x,t)dy \\
&= \frac{e^{r(T-t)}}{\delta} = \frac{D_x(T,t)}{\delta},
\end{aligned}
$$

where we have used the fact that $|H_\delta(y)| \le \frac{1}{\delta}$. For the second derivative of u, The property $X'(s;x,t) = X(s;1,t)$ implies that $X''(s;x,t) = 0$ and that

$$
\begin{aligned}
u_{xx}(x,t) &= \mathbb{E}[g''_\delta(X(T;x,t))(X'(T;x,t))^2] \\
&= \mathbb{E}[H'_\delta(X(T;x,t))(X'(T;x,t))^2].
\end{aligned}
$$

Using the fact that $|H'_\delta(y)| \le \frac{2}{\delta^2}$ and considering $z = log(y/x)$, we get

$$
\begin{aligned}
|u_{xx}(x,t)| &\le \frac{2}{\delta^2}\int_{\mathbb{R}_+} (\frac{y}{x})^2 P(T,y;x,t)dy \\
&= \frac{2}{\delta^2}\int_{\mathbb{R}} \frac{e^{2z}}{\sigma\sqrt{2\pi(T-t)}} e^{-\frac{(z-(r-\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}} dz \\
&= 2\frac{e^{(2r+\sigma^2)(T-t)}}{\delta^2} = \frac{D_{xx}(T,t)}{\delta^2}.
\end{aligned}
$$

Finally, for the third derivative we have

$$
\begin{aligned}
u_{xxx}(x,t) &= \mathbb{E}[g'''_\delta(X(T;x,t))(X'(T;x,t))^3] \\
&= \mathbb{E}[H''_\delta(X(T;x,t))(X'(T;x,t))^3].
\end{aligned}
$$

Consequently,

$$
\begin{aligned}
|u_{xxx}(x,t)| &\leq \frac{4}{\delta^3}\int_{\mathbb{R}_+}(\frac{y}{x})^3 P(T,y;x,t)dy \\
&= \frac{4}{\delta^3}\int_{\mathbb{R}}\frac{e^{3z}}{\sigma\sqrt{2\pi(T-t)}}e^{-\frac{(z-(r-\sigma^2/2)(T-t))^2}{2\sigma^2(T-t)}}dz \\
&= 4\frac{e^{3(r+\sigma^2)(T-t)}}{\delta^3} = \frac{D_{xxx}(T,t)}{\delta^3}.
\end{aligned}
$$

We have used that $|H'''_\delta(y)| \leq \frac{4}{\delta^3}$. To sum up, we have found the following three upper bounds

$$
\begin{aligned}
\|u_x(\cdot,t)\| &\leq \min(\frac{C_{x,\delta}}{\sqrt{T-t}}, \frac{D_x}{\delta}), \\
\|u_{xx}(\cdot,t)\| &\leq \min(\frac{C_{xx,\delta}}{T-t}, \frac{D_{xx}}{\delta^2}), \\
\|u_{xxx}(\cdot,t)\| &\leq \min(\frac{C_{xxx,\delta}}{(T-t)^{3/2}}, \frac{D_{xxx}}{\delta^3}).
\end{aligned}
$$

It is clear that the mollification will ensure that the error density is finite at the final time T, and in a small region of order $O(\delta^3)$ near the final time it will be constant. Let us examine the expected number of time steps using this mollification procedure. For the GBM problem, the resulting error density is

$$
\begin{aligned}
\rho(t_n) &= O\left(\min(\delta^{-3}, (T-t)^{-3/2})\right) \\
&= O\left(\frac{1}{\max(\delta^3, (T-t)^{3/2})}\right).
\end{aligned}
$$

It results into

$$
\Delta t_n = O\left(\sqrt{\frac{TOL}{N\rho(t_n)}}\right) = O\left(\sqrt{\frac{TOL\max(\delta^3, (T-t)^{3/2})}{N}}\right).
$$

Hence, using the Cauchy Schwarz inequality, we get the following bound on the average number of time steps

$$\begin{aligned}
\mathbb{E}[N] &= \mathbb{E}\left[\int_0^T \Delta t(s)^{-1} ds\right] \\
&= CTOL^{-1/2}\mathbb{E}\left[\sqrt{N}\left(\int_0^{T-O(\delta^3)} (T-s)^{-3/4} ds + O(\delta^{3/2})\right)\right] \\
&= O\left(TOL^{-1/2}\sqrt{\mathbb{E}[N]}\right).
\end{aligned}$$

This yields to a desired rate on the number of time steps for the hybrid adaptive forward Euler numerical scheme

$$\mathbb{E}[N] = O(TOL^{-1}).$$

The above analysis shows that whatever $\delta > 0$ that is used in the mollification of the payoff, the expected number of time steps has the right order which guarantees that the hybrid adaptive algorithm will reach the complexity $O(TOL^{-3})$. However, for a small value of δ, the weights might reach big values and thereby generate too small time steps.

Drift Singularity Problem

The drift singularity SDE defined in (2.33) is an example where the complexity $O(TOL^{-3})$ is not obtained using uniform time stepping with a first order numerical method. Therefore, we aim to use our hybrid adaptive method to recover this complexity. Again we need to determine upper bounds for the three derivatives of u. Since the drift singularity SDE is again a GBM, the FP density is also known. In other words, we need to replace the term $r(T-t)$ in the FP expression of the GBM problem by $\int_t^T rf_\alpha(s)ds = 2r(\sqrt{T-\alpha} + \text{sign}(\alpha - t)\sqrt{|t-\alpha|})$

$$p(y,T;x,t) = \frac{1}{y\sigma\sqrt{2\pi(T-t)}}exp\left(-\frac{(\log(y/x) - (r\int_t^T f_{\alpha(s)}ds - \sigma^2/2(T-t)))^2}{2\sigma^2(T-t)}\right).$$

The same replacement should be done then to get the upper bounds expression. In summary we get

$$\begin{aligned}
||u_x(\cdot,t)||_\infty &\leq \frac{C_{\alpha,x,\delta}(T,t)}{\sqrt{T-t}}, \\
||u_{xx}(\cdot,t)||_\infty &\leq \frac{C_{\alpha,xx,\delta}(T,t)}{T-t}), \\
||u_{xxx}(\cdot,t)||_\infty &\leq \frac{C_{\alpha,xxx,\delta}(T,t)}{(T-t)^{3/2}}.
\end{aligned}$$

Again using the mollification technique we get alternative bounds

$$\begin{aligned}
||u_x(\cdot,t)||_\infty &\leq \frac{e^{r\int_t^T f_\alpha(s)ds}}{\delta}, \\
||u_{xx}(\cdot,t)||_\infty &\leq 2\frac{e^{2r\int_t^T f_\alpha(s)ds + \sigma^2(T-t)}}{\delta^2}, \\
||u_{xxx}(\cdot,t)||_\infty &\leq 4\frac{e^{3(r\int_t^T f_\alpha(s)ds + \sigma^2(T-t))}}{\delta^3}.
\end{aligned} \tag{3.11}$$

Finally, a similar analysis as for the GBM case shows that $\mathbb{E}[N] = O(TOL^{-1})$, and hence the hybrid adaptive algorithm will achieve the desired computational cost.

3.3 Numerical Implementation

In this section, we will describe the hybrid adaptive Single Level Monte Carlo (SLMC) algorithm for both GBM and drift singularity problems. The binary option whose payoff is the Heaviside function is considered. As before, the given tolerance TOL is split it into a time discretization and a statistical tolerance as in (2.25). We have proved in the beginning of this chapter that

$$\mathbb{E}[g_\delta(X(T)) - g_\delta(\bar{X}(T))] = \sum_{n=1}^{N} \mathbb{E}[\rho(t_n)\Delta t_{n-1}^2] + h.o.t.,$$

where $\rho(t_n)$ is given by (3.7). The refinement procedure is simple in the hybrid algorithm, the time steps are chosen adaptively and by going forward in time according to the following refinement criterion

$$\rho(t_n)\Delta t_n^2 \leq \frac{TOL_T}{\bar{N}}, \tag{3.12}$$

where $\bar{N}$ denotes the average number of time steps. It is important to note that we are using stochastic time steps where the mesh is path dependent. The algorithm will adaptively update $\bar{N}$ using sample average estimates. To start, we need a first guess of the expected number of time steps.

On the other hand, the number of MC realizations is chosen by controlling the statistical error. The algorithm of the hybrid adaptive MC method is composed of an inner and an outer loop. The outer loop uses the MC technique and update the number of realizations M if (2.30) is not fulfilled. The inner loop generates M realizations using forward Euler method to the accuracy $TOLT$. A description of the hybrid SLMC adaptive algorithm is given in Algorithms 3 and 4.

Algorithm 3 Hybrid Adaptive SLMC Method

```
  Inputs:  $M$, $TOL_S$
  Outputs: $\mu$
  Call Generate SLMC Realizations($M$, $TOL_T$, $\bar{N}$) to compute $\mu$ and $\sigma$
  while 2.30 is not satisfied do
    Update the average number of time steps $\bar{N}$ .
    Set $M = 2M$
    Call Generate SLMC Realizations($M$, $TOL_T$, $\bar{N}$) to compute $\mu$ and $\hat{\sigma}$
  end while
```

Algorithm 4 Generate SLMC Realizations()

```
Inputs: $M$, $TOL_T$, $\bar{N}$
Outputs: $\mu$, $\hat{\sigma}$
for $J = 1, 2, ..., M$ do
    $x = x_0$, $t = 0$,
    $\Delta t = 0.1$,
    while $t < T$ do
        Find the next time steps by dividing $\Delta t$ until 3.12 is satisfied
        if $t + dt > T$ then
            Set $\Delta t = T - t$ and $t = T$
        else
            Set $t = t + \Delta t$
        end if
        Update the number of time steps
        Generate $\bar{X}(t)$ by forward Euler method and set $\Delta t = max(2\Delta t_{old}, 0.1)$
    end while
    $\mu = \mu + g_\delta(\bar{X}(T))$
end for
$\mu = \mu/M$ and estimate $\hat{\sigma}$ by the sample standard deviation (2.29).
```

3.4 Numerical Results

As in the previous chapter, we will carry out some numerical tests using the GBM and the drift singularity problem.

3.4.1 GBM Problem

Let us consider the GBM problem (3.8). Again we consider the binary option with the mollification given in Appendix B. The problem parameters are: $r = 0.05$, $\sigma = 0.2$, $T = 1$, $K = 1$, $x_0 = 0.9$. Essentially, we are interested, as in the a posteriori case, in investigating the accuracy and the complexity of the algorithm.

In Figure 3.1, we present the accuracy plot to check if (2.3) is satisfied. From this plot, we conclude that our algorithm succeeds to meet the accuracy condition since all the points (global error) are below the accuracy line (the range of tolerances).

The important result to check is the algorithm's complexity. Theoretically, we have seen that the hybrid adaptive method will give the complexity $O(TOL^{-3})$. In

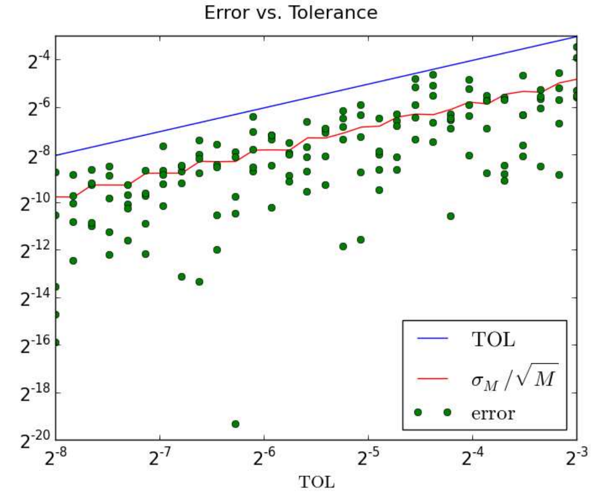

Figure 3.1: Global error for the hybrid adaptive SLMC algorithm with the GBM problem and the binary option payoff.

Figure 3.2, we plot the computational cost against the range of tolerances, and from that we can deduce that numerical result also gives the expected single level complexity.

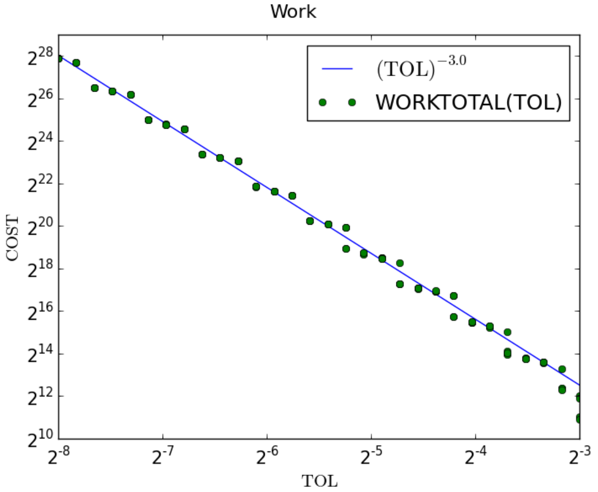

Figure 3.2: Total computational cost for the hybrid adaptive SLMC algorithm with the GBM problem and the binary payoff.

3.4.2 Drift Singularity Problem

For the binary option GBM problem, the MC forward Euler algorithm with uniform time stepping achieves the desired complexity $O(TOL^{-3})$, see [11]. So, to motivate the need of our hybrid adaptive algorithm, we will consider a more difficult problem which in addition to the discontinuous Heaviside payoff also has a singularity in the SDE drift function. The same problem parameters are used as in the GBM case with the mollified function g_δ described in Appendix A, and to avoid possible numerical blow ups, we use the drift modifications (2.34) and (2.35).

Figure 3.3 shows the accuracy plot. We conclude that the hybrid adaptive algorithm meets the accuracy condition (2.3) for this problem as well. The second important result to investigate is the complexity of the method. In Figure 3.4, the computational cost has approximatively the expected order $O(TOL^{-3})$. We remark also that the cost for the drift singularity problem is slightly bigger than the one with a GBM which is a reasonable result since we need to refine more near the singularity point.

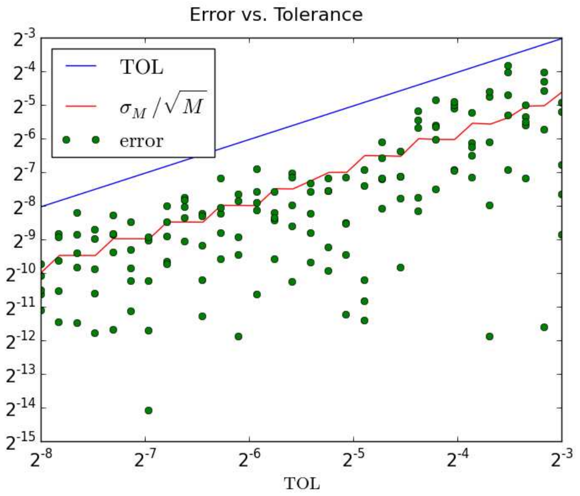

Figure 3.3: Global error for the hybrid adaptive SLMC algorithm with the drift singularity problem and the binary option payoff.

Through our numerical results, we have shown that our hybrid adaptive MC for-

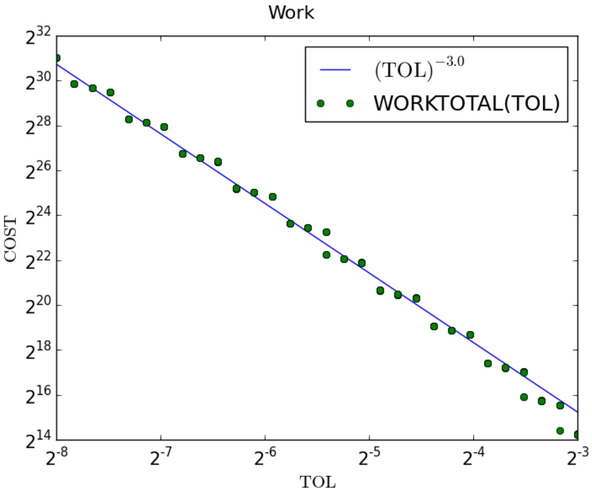

Figure 3.4: Total computational cost for the hybrid adaptive SLMC algorithm with the drift singularity problem and the binary option payoff.

ward Euler method has the optimal computational rate for first order numerical algorithms for SDE. The problem with drift singularity, where the uniform time stepping method is unable to meet the desired cost, is an example which proves the efficiency, in terms of computational cost, of the hybrid adaptive method. In the next chapter, we will extend the hybrid adaptive method to the multilevel Monte Carlo setting.

Chapter 4

Hybrid Adaptive MLMC for SDE

In this chapter, we will introduce the MLMC method, which is a variance reduction technique for the approximation of expected values depending on the solution of SDEs. First, the standard uniform time stepping MLMC method, developed in [12], will be presented. Then, the MLMC with hybrid adaptivity will be introduced to solve the binary payoff problem for both plain GBM and drift singularity SDEs. The main result, which will be presented in the numerical part, is the observed hybrid adaptivity MLMC complexity $O(TOL^{-2}(\log(TOL))^2)$ for the considered non-smooth problems. Finally, we will extend the hybrid adaptive MLMC forward Euler algorithm to the multidimensional case and we show that the desired complexity again is reached.

4.1 The standard MLMC Method

The MLMC method based on uniform time steps was first introduced by Giles in [12]. He developed a clever type of variance reduction technique for the approximation of an expected value of a quantity of interest. The idea of this method is to construct a hierarchy of uniform grids with step sizes

$$\Delta t_\ell = C^{-\ell} \Delta t_0$$

where $C \in \{2, 3, ...\}$ and $\ell \in 0, 1, ..., L$. The smallest time step Δt_L is the time step which will determine the bias error. Giles' source of inspiration for the multilevel algorithm from the multigrid method for the iterative solution of linear systems of equations.

Instead of using the standard SLMC estimator, the idea is to use the following telescopic sum

$$\mathbb{E}[g_L] = \mathbb{E}[g_0] + \sum_{\ell=1}^{L} \mathbb{E}[g_\ell - g_{\ell-1}], \tag{4.1}$$

where $g_\ell = g(\bar{X}_\ell(T))$ and $\bar{X}_\ell(T)$ is a numerical approximation of $X(T)$ using the forward Euler method on the mesh Δt_ℓ. Hence, by generating $\{M_\ell\}_{\ell=0}^{L}$ realizations on each level ℓ and using the telescopic sum, the MLMC estimator $\mathcal{A}_{mlmc}$ of $\mathbb{E}[g_L]$ is defined as follows

$$\begin{aligned}\mathcal{A}_{mlmc}(g(\bar{X}_L(T)) = &\sum_{i=1}^{M_0} \frac{g(\bar{X}_0(T;\omega_{i,0}))}{M_0} \\ &+ \sum_{\ell=1}^{L}\sum_{i=1}^{M_\ell} \frac{g(\bar{X}_\ell(T;\omega_{i,\ell})) - g(\bar{X}_{\ell-1}(T;\omega_{i,\ell}))}{M_\ell}.\end{aligned} \tag{4.2}$$

An important feature of the MLMC estimator is that, for each $\ell > 0$, the pairs $\bar{X}_\ell(T;\omega_{i,\ell})$ and $\bar{X}_{\ell-1}(T;\omega_{i,\ell})$ in the expression $g(\bar{X}_\ell(T;\omega_{i,\ell})) - g(\bar{X}_{\ell-1}(T;\omega_{i,\ell}))$ are generated by the same Brownian path but on two different grids Δt_ℓ and $\Delta t_{\ell-1}$, respectively.

The computational cost for the MLMC algorithm using uniform time steps is

$$\text{Total Cost} = \sum_{\ell=0}^{L} \frac{M_\ell}{\Delta t_\ell}. \tag{4.3}$$

For a given tolerance $TOL > 0$, the main result in [12] is that the computational cost to achieve a mean square accuracy of order TOL is reduced from $O(TOL^{-3})$, in the single level setting, to $O((TOL^{-1}log(TOL^{-1}))^2)$ in the multilevel setting for the first order forward Euler scheme. This cost is achieved when the contribution to the work is of equal size on each level, that is

$$\frac{M_\ell}{\Delta t_\ell} = \frac{M_0}{\Delta t_0}.$$

Hence, the number of realization on each level is computed as $M_\ell = M_{\ell-1}/C$. In the rest of this work, C is chosen to be equal to 2.

4.2 Hybrid Adaptive MLMC

Many assumptions have been assumed in the work by Giles [12] to achieve the complexity $O(TOL^{-2}(\log(TOL))^2)$. In order to control the strong convergence of the forward Euler scheme and to ensure that it gives the right order 1/2, the payoff function, for example, is assumed to be a Lipschitz continuous function. Another assumption in the coefficients of the the SDE have been considered to ensure a first order weak error in the forward Euler scheme (2.5).

The binary option payoff is an example where the standard MLMC does not work since the Lipschitz condition is not valid. In the work by Avikainen [13], it is proved that for the binary option, the forward Euler scheme has strong convergence strictly less than 1/2 . This low convergence rate makes the MLMC algorithm more expensive to control the variance, hence, the desired MLMC cost $O(TOL^{-2}(\log(TOL))^2)$ could not be achieved. In [14], the authors solved the problem of the binary payoff in one dimension by considering a higher order Milstein scheme. The aim of the present section is to extend the hybrid method, introduced in Chapter 3, to the multilevel setting and to show that, using some prior information, it will achieve the standard MLMC computational cost $O(TOL^{-2}(\log(TOL))^2)$ with the forward Euler scheme and without considering any higher order numerical method as in [14].

Adaptivity in the multilevel setting is introduced in [15]. In that work, the a posteriori MLMC algorithm was presented. The work in [15] was inspired from the a posteriori adaptive SLMC method described in Chapter 2. The hybrid adaptive MLMC algorithm, which will be introduced in this section, will be similar to the work in [15], with the difference of considering the hybrid error expansion (3.7) instead of the a posteriori expression (2.10). The accuracy condition for our case is

$$\left|\mathcal{A}_{mlmc}(g_\delta(\bar{X}_L(T)) - \mathbb{E}[g_\delta(X(T))]\right| \leq TOL, \tag{4.4}$$

where g_δ is defined in Appendix B. The choice of the δ value will be discussed in the numerical part. Again, for a given tolerance TOL, we split it into a time discretization tolerance TOL_T and a statistical tolerance TOL_S as in (2.25). The global error (4.4) is divided into two terms, the time discretization error ϵ_T and the statistical error ϵ_S

$$\epsilon_T = \left|\mathbb{E}[g_\delta(X(T))] - \mathbb{E}[g_\delta(\bar{X}_L(T))]\right| \tag{4.5}$$

$$\epsilon_S = \left|\mathcal{A}_{mc}(g_\delta(\bar{X}_L(T)) - \mathbb{E}[g_\delta(\bar{X}_L(T))]\right| \tag{4.6}$$

In the standard MLMC algorithm, the mesh hierarchy is easily constructed since the mesh size is known. In our setting, these sizes are not known since we are using adaptivity. One way to construct the mesh hierarchy is to successively increase the accuracy. In other words, we consider a hierarchy of tolerances defined as follows

$$TOL_\ell = \frac{TOL_0}{2^\ell}, \text{ for } \ell \in \{0, 1, ..., L\} \tag{4.7}$$

and using our refinement procedure described in Chapter 3, the grid Δt_ℓ is constructed adaptively such that for each $\ell \in \{0, 1, ..., L\}$

$$\left|\mathbb{E}[g_\delta(X(T))] - \mathbb{E}[g_\delta(\bar{X}_\ell(T))]\right| \leq TOL_\ell. \tag{4.8}$$

On the finest level L, we have would like to have $TOL_L = TOL_T$, which would ensure that $\epsilon_T \leq TOL_T$. Therefore, we choose the number of levels L as follows

$$L = \left\lceil log_2 \left(\frac{TOL_0}{TOL_T} \right) \right\rceil . \tag{4.9}$$

The MLMC algorithm is constructed, as in the single level setting, with an inner and an outer loop. Given an initial number of realizations M_0, the number of samples on the higher levels are chosen from the relation

$$M_\ell = \frac{M_0}{2^\ell}. \tag{4.10}$$

The outer loop uses the MLMC estimator (4.2) to control the statistical error ϵ_S. The variance of the MLMC estimator is estimated through the sample variance

$$\hat{\sigma}_M^2 = \frac{\nu_{M_0}[g_\delta(\bar{X}_0(T))]}{M_0} + \sum_{\ell=1}^{L} \frac{\nu_{M_\ell}[g_\delta(\bar{X}_\ell(T)) - g_\delta(\bar{X}_{\ell-1}(T))]}{M_\ell}. \tag{4.11}$$

While the stopping criterion (2.30) is not satisfied, the number of samples M_0 is doubled.

The inner loop consists on generating M_ℓ realizations of $\bar{X}_{\ell-1}(T)$ and $\bar{X}_\ell(T)$ by constructing the corresponding grids $\Delta t_{\ell-1}$ and Δt_ℓ to the accuracy $TOL_{\ell-1}$ and TOL_ℓ respectively. Since these pairs should be generated by the same Wiener path, we start by constructing the grid $\Delta t_{\ell-1}$ to the accuracy $TOL_{\ell-1}$ using the refinement criterion (4.12). Then, we use Brownian bridge (see Appendix A) and (4.12) to construct the Δt_ℓ. It is important to ensure that the grid Δt_ℓ on the level ℓ is constructed in the same way as the grid Δt_ℓ on the level ℓ+1, for each $\ell \in \{1, ..., L-1\}$.

For each realization on level ℓ, the grid Δt_ℓ is constructed iteratively from the beginning $t = 0$ to the end $t_N = T$. The criterion to accept a time step is the following

$$\rho(t_n)\Delta t_{n-1}^2 \leq \frac{TOL_\ell}{\bar{N}_\ell}, \tag{4.12}$$

where $\bar{N}_\ell$ denotes the approximate average number of time steps on level ℓ. Until (4.12) is satisfied, the time step is halved. The initial guess of the next time steps is $\Delta t_{ini} = \min(\Delta t_{\max}, 2\Delta t_{old})$ The implementation of the hybrid adaptive MLMC method is described in Algorithms 5 and 6.

Algorithm 5 Hybrid Adaptive MLMC forward Euler Method

```
Inputs: $M_0$, $TOL_S$
Outputs: $\mu$
Call Generate MLMC Realizations($\{M_\ell\}_{\ell=0}^{L}$, $TOL_T$, $\{\bar{N}_\ell\}_{\ell=0}^{L}$)
while (2.30) is not satisfied do
    Update the average number of time steps $\bar{N}$ .
    Set $M = 2M$
    Call Generate MLMC Realizations($\{M_\ell\}_{\ell=0}^{L}$, $TOL_T$, $\{\bar{N}_\ell\}_{\ell=0}^{L}$)
end while
```

4.3 Numerical Results

The hybrid adaptive MLMC algorithm is used in this section to present numerical results. The binary option payoff will be used as a non-smooth observable. The mollified version g_δ of g is used in the numerical implementation. The same SDE will be used as in the previous two chapters: GBM and drift singularity problems. The problem parameters used in all the following simulations are: $r = 0.05$, $\sigma = 0.2$, $x0 = 0.9$, $K = 1$ and $T = 1$.

As a motivation for the use of adaptivity on the binary option type of problems, we will show numerically that the uniform time steps is not optimal, for the class of first order weak error numerical methods, in this setting. This result was proved theoretically in [13]. We consider a GBM type of SDE with the binary payoff. The

Algorithm 6 Generate MLMC Realizations

```
**Inputs:** $\{M_\ell\}_{\ell=0}^{L}$, $TOL_T$, $\{\bar{N}_\ell\}_{\ell=0}^{L}$
**Outputs**: $\mu$, $\hat{\sigma}^2$
Set $\mu = 0$ and $\hat{\sigma}^2 = 0$.
**for** $i = 1, ..., M_0$ **do**
    Evaluate $g_\delta(\bar{X}_0(T;\omega_{i,0}))$ by constructing the grid $\Delta t_0$ to the accuracy $TOL_0$ using the refinement criterion (4.12).
    Set $\mu = \mu + g_\delta(\bar{X}_0(T;\omega_{i,0}))$.
**end for**
Set $\mu = \mu/M_0$ and $\hat{\sigma}^2 = \nu_{M_0}[g_\delta(\bar{X}_0(T))]/M_0$
**for** $\ell = 1, ..., L$ **do**
    Set $\beta = 0$
    **for** $i = 1, ..., M_\ell$ **do**
        Evaluate $g_\delta(\bar{X}_{\ell-1}(T;\omega_{i,\ell}))$ by constructing the the grid $\Delta t_{\ell-1}$ to the accuracy $TOL_{\ell-1}$.
        Evaluate $g_\delta(\bar{X}_\ell(T;\omega_{i,\ell}))$ by constructing the grid $\Delta t_\ell$ using Brownian bridge and the refinement procedure (4.12)
        Set $\beta = \beta + g_\delta(\bar{X}_\ell(T;\omega_{i,\ell})) - g_\delta(\bar{X}_{\ell-1}(T;\omega_{i,\ell}))$.
    **end for**
    Set $\mu = \mu + \beta/M_\ell$ and $\hat{\sigma}^2 = \hat{\sigma}^2 + \nu_{M_\ell}[g_\delta(\bar{X}_\ell(T)) - g_\delta(\bar{X}_{\ell-1}(T))]/M_\ell$.
**end for**
```

algorithm for the standard MLMC is described in [12]. In Figure 4.1, the accuracy plot is presented. It is clear that the accuracy condition (4.4) is met.

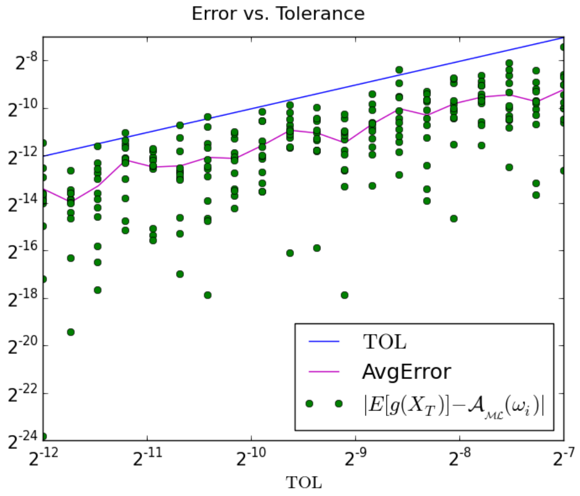

Figure 4.1: Global error for the standard MLMC algorithm with the GBM problem and the binary option payoff.

Let us look on the complexity of the standard MLMC for this problem. In Figure

4.2, the average computational cost is plotted against the tolerances. We point out that the computational cost is not optimal. In other words, the standard MLMC algorithm gives a cost larger than the standard MLMC cost for smooth problems.

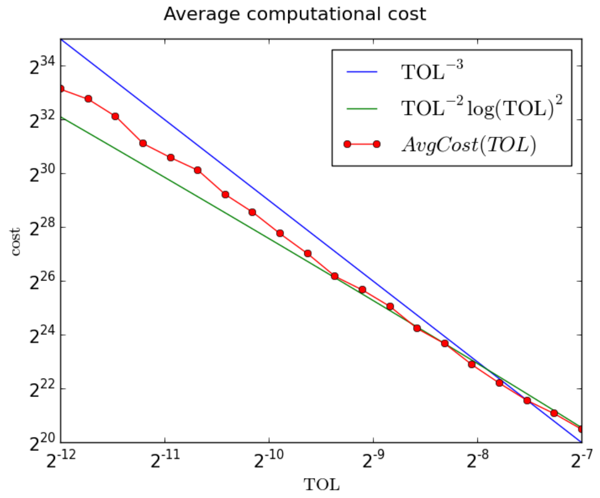

Figure 4.2: Total computational cost for the standard MLMC algorithm with the GBM problem and binary option payoff.

Due to this fact, the hybrid adaptive MLMC is instead used to recover optimality in the cost, $O(TOL^{-2}(\log(TOL))^2)$. At this stage, it is important to note that two ways of the choice of the mollification parameter δ could be tried. Either δ is independent of the level ℓ and chosen to be in relation with only the tolerance on the deepest level L, i.e. $\delta = O(TOL_L)$, or δ is dependent of ℓ and chosen according to the relation $O(TOL_\ell)$. These two methods will not affect the convergence of the algorithm since we conserve the telescopic property of the MLMC estimator. Neither one of them will affect the order of the computational cost but the second method is worthy to try since it is more flexible and it may further reduce the variance of the MLMC estimator. Again we consider the GBM problem with the binary option. The mollified payoff (see Appendix B) is used again. Using the Algorithms 5 and 6 we get the results in Figure 4.3 and Figure 4.4. Figure 4.3 shows that the algorithm converges with the right order of accuracy. The complexity of the hybrid adaptive MLMC algorithm with the forward Euler scheme is shown in Figure 4.4. The novelty of this

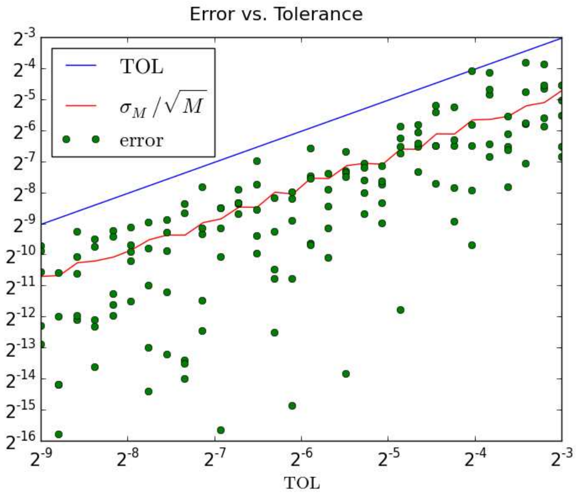

Figure 4.3: Global error for the hybrid MLMC algorithm with the GBM problem and the binary option payoff.

result is that the computational cost has the desired order $O(TOL^{-2}(log(TOL))^2)$. This result shows the efficiency of adaptivity compared to the uniform stepping which is optimal only using Milstein scheme, see [14].

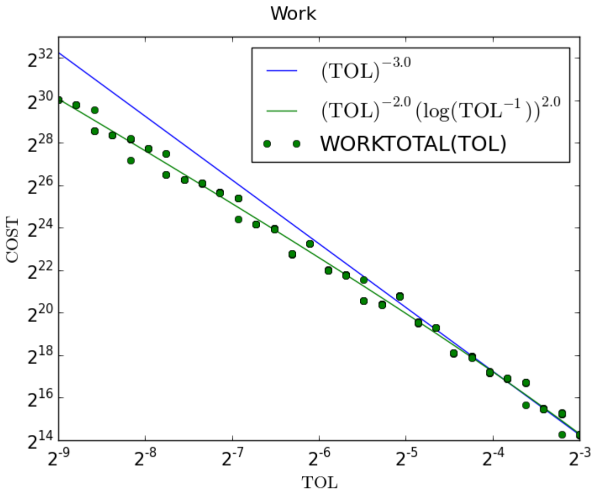

Figure 4.4: Total computational cost for the hybrid MLMC algorithm with the GBM problem and the binary option payoff.

The second SDE is the drift singularity problem. The standard MLMC does not achieve the desired complexity in this case and the hybrid adaptivity is again used. The accuracy plot is in Figure 4.5 where we can conclude that the algorithm is convergent with the right accuracy.

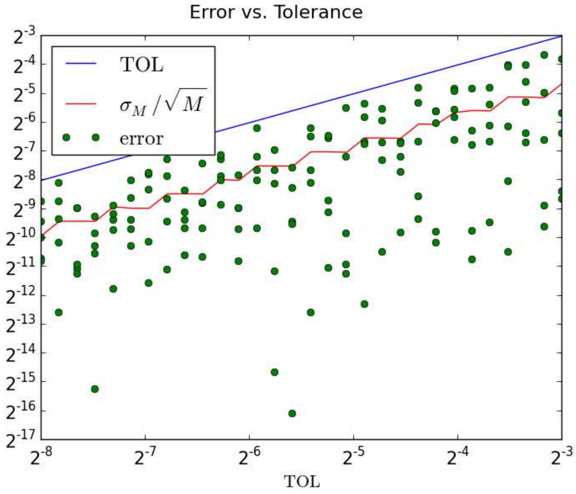

Figure 4.5: Global error for the hybrid MLMC algorithm with the drift singularity problem and the binary option payoff.

The hybrid MLMC is again optimal, see Figure 4.6. The computational cost has the desired MLMC complexity.

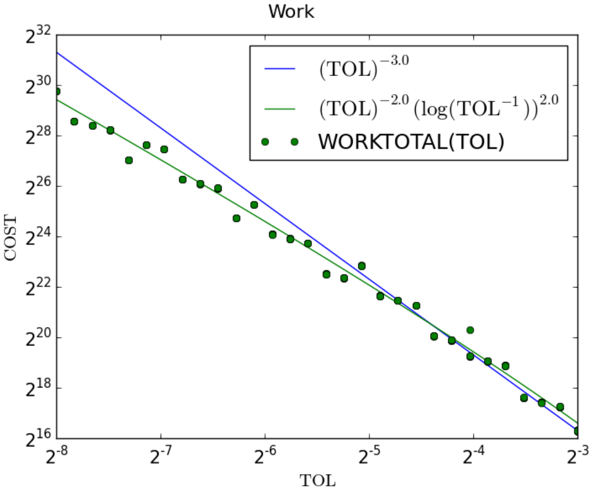

Figure 4.6: Total computational cost for the hybrid MLMC algorithm with the drift singularity problem and the binary payoff.

4.4 Multidimensional Extension

4.4.1 SDE Transformations

It is frequently the case that economic or financial considerations will suggest that a stock price, exchange rate, interest rate or other economic variables evolve in time according an SDE of the form (2.1). More generally, when several related economic variables X_1, X_2,...,X_d are considered, the vector $X = (X_1, X_2, ..., X_d)$ may evolve in time according to a system of SDEs of the form

$$dX_i = rX_i dt + \sum_{j=1}^{d} \sigma_{ij} X_j dW_j, \tag{4.13}$$

where W is a d-dimensional independent Wiener process and σ is the volatility matrix, which represents the interaction between the stock prices. Using the hybrid adaptive MLMC algorithm, the aim is to approximate the price $\mathbb{E}[g(Z(T))]$ of an option written on several underlying assets, g is the binary payoff and Z is defined as

$$Z = \langle X, w \rangle, \tag{4.14}$$

where $\langle , \rangle$ denotes the inner product in $\mathbb{R}^d$ and w is a weight vector whose elements are equal to $1/d$ in this work. In this case, bounds in (3.7) are not easily derived since the FP density of Z is not known. Therefore, a clever modification of the multidimensional problem is instead used to reach our goal. First of all, we use Itô's formula on $Z = \langle X, w \rangle$ to get

$$dZ = rZdt + \sum_{i=1}^{d} w_i \sum_{j=1}^{d} \sigma_{ij} X_j dW_j. \tag{4.15}$$

The previous SDE has different noise sources, the first transformation is to represent the evolution of Z by an SDE with only one noise source

$$d\hat{Z} = r\hat{Z}dt + \sqrt{\sum_{j=1}^{d} X_j^2 (\sum_{i=1}^{d} w_i \sigma_{ij})^2} d\hat{W}. \tag{4.16}$$

Although the paths for (4.15) and (4.16) are different, the approximate SDE (4.16) shares the same statistical properties as the original SDE (4.15). The one dimensional SDE (4.16) could be now approximated by a GBM problem

$$d\hat{Z} = r\hat{Z}dt + \tilde{\sigma}\hat{Z}d\hat{W}, \tag{4.17}$$

where the constant $\hat{\sigma}$ satisfies

$$\tilde{\sigma} \approx \frac{\sqrt{\sum_{j=1}^{d} X_j^2 (\sum_{i=1}^{d} w_i \sigma_{ij})^2}}{\langle X, w \rangle}. \tag{4.18}$$

The actual value of $\tilde{\sigma}$ can be estimated using a prior information of the actual approximate paths of $\bar{X}$.

4.4.2 Implementation and Results

Transforming the multidimensional problem (4.15) into a one dimensional GBM (4.17) will allow us to use the a hybrid error density (3.7) on the new GBM approximation of the evolution of Z. In other words, the refinement process will be performed on the one dimensional GBM (4.17). The implementation of the hybrid MLMC in the multidimensional setting is similar to the implementation described in the previous two sections with the following two modifications:

- The local error in the hybrid error density (3.7) is computed using the GBM problem (4.17). The value of $\tilde{\sigma}$ in (4.18) is updated at each time step using the forward Euler scheme approximations of the underlying assets X, whose components evolve in time according to (4.13).

- There are two alternatives to compute the bounds in the error density expression. Since $\tilde{\sigma}$ is not fixed and it should be updated in each time step, one may choose to consider its lower bound which represent the worst case and then use this value on the expression (3.11), or simply we may work with the dynamic value of $\tilde{\sigma}$ and update the bounds (3.11) on each time step.

It is important to note that for each realization, we have d independent Wiener paths. Besides, since the noise components are independent, Brownian bridges in the refinement stage are performed independently to construct paths for X_i, $i = 1, ..., d$. In numerical example given here, we will use the following volatility matrix

$$\sigma = \begin{pmatrix} 0.2 & 0 & 0 \\ 0.05 & 0.2 & 0 \\ 0.05 & 0.01 & 0.2 \end{pmatrix},$$

and the initial values : $X_1(0) = 0.9$, $X_2(0) = 1$ and $X_3(0) = 1$. For the accuracy plot, we need a reference value of the quantity of interest. In our numerical examples, this reference value is computed using our adaptive algorithm with a tolerance value smaller than the range of tolerances in consideration. In Figure 4.7, we plot the global error function of the range of tolerances. We point out that the accuracy condition is again satisfied. Then, we need to check the complexity of the algorithm. In Figure 4.8, the total computational cost to achieve the accuracy condition is plotted against the range of tolerances. It is clear from this figure that the desired complexity $O(TOL^{-2}(\log(TOL))^2)$ is reached for the multidimensional case also.

In this chapter, we have extended the hybrid adaptive algorithm to the multilevel setting. We have proved numerically the efficiency of the adaptive algo-

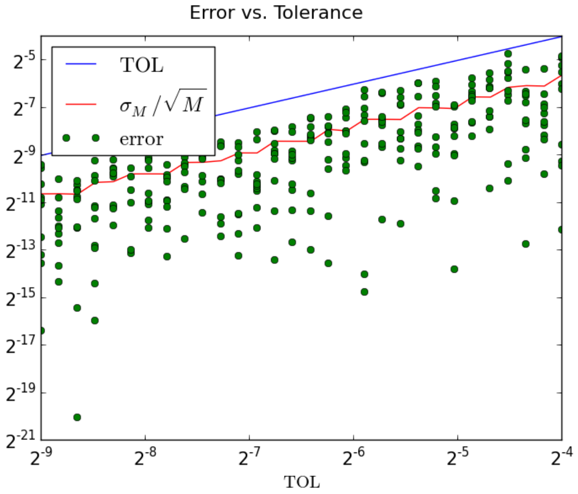

Figure 4.7: Global error for the hybrid MLMC adaptive algorithm in 3D with the binary option payoff.

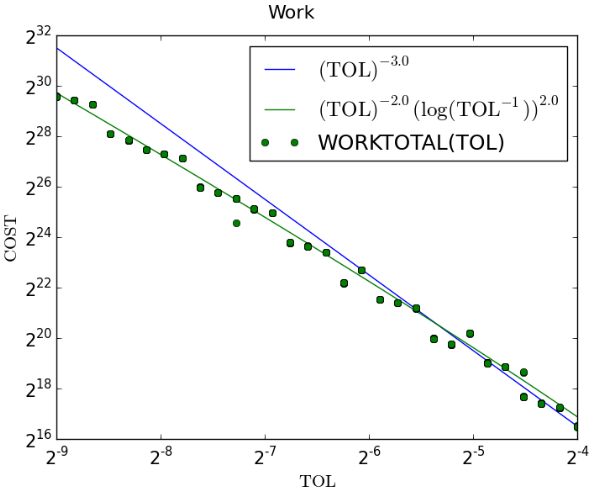

Figure 4.8: Total computational cost for the hybrid MLMC adaptive algorithm in 3D with the binary option payoff.

rithm compared to the standard one which fails to reach the expected MLMC complexity for non-smooth problems. In other words, using the hybrid adaptive forward Euler scheme, our method recovers the desired MLMC computational cost $O(TOL^{-2}(\log TOL)^2)$, which to the best of our knowledge only has been obtained for uniform time steps when applying higher order schemes. Extension to the multidimensional case is also performed through the solution of a nearby problem. The desired optimal computational cost is also reached for this setting through the use of the new introduced MLMC adaptive method.

Chapter 5

Summary

5.1 Conclusion

In this work, we have discussed different adaptive algorithms for the approximation, with a desired computational cost, of an expected value depending on a solution of a SDE and a low regularity observable. A modified version of the a posteriori adaptive method in the SLMC setting has been presented. A mollification procedure with a Gaussian kernel was applied to the non-smooth binary payoff. Numerical results using the a posteriori adaptive SLMC forward Euler method have shown the non optimality of this adaptive method. The reason for these unsatisfactory results is related to the dual function computations which are sensitive to the mollification parameter. In fact, realizations ending in a region near the strike K are characterized by a large value of the error density, hence solving the dual problem backward will result in a small time step.

A hybrid adaptive MC forward Euler method, based on the derivation of a new error expansion, has been then developed. The idea behind this method is to introduce prior information in the a posteriori error expansion. In fact, both expansions share the same a posteriori local error, but the weights for the hybrid method are computed a priori. Numerical results using the hybrid SLMC forward Euler algorithm have shown the computational efficiency of this technique for the binary payoff with either

GBM or drift singularity problems.

The desired complexity in the MLMC setting has also been recovered using an extended version of the hybrid adaptive SLMC method. In this case also, the MLMC hybrid adaptive technique is based on the new derived hybrid error expansion. The construction of the mesh hierarchy and its implementation has been also described. Numerical results have confirmed the computational optimality of the MLMC hybrid technique for non-smooth observable and singular SDE coefficients.

Finally, the multidimensional case has been also discussed. A suitable transformation into a one dimensional case has been introduced. The novelty of the present work resides on achieving the desired complexity, using the hybrid MLMC forward Euler adaptivity, in the multidimensional case.

5.2 Future Research Work

A possible extension of the present work is to consider more challenging SDE problems modeling the evolution of financial instruments. Constant Elasticity of Variance model (CEV) is a relevant model to try. CEV model is a generalized version of the GBM problem where the notion of stochastic volatility is introduced and hence the model is more likely to fit the market. Another important model is the Cox Ingersoll Ross model (CIR) which describes the evolution of the interest rate and is used for bounds pricing.

Both CIR and CEV model include little information compared to the GBM problem where the FP density was a priori known. Hence, the weights in the time discretization error expansion will not be found as straightforwardly as in the GBM case. Therefore, an approximation procedure to compute the weight functions needs to be developed for the CEV, the CIR and more other complicated models.

REFERENCES

[1] A. Szepessy, R. Tempone, and G. E. Zouraris, “Adaptive weak approximation of stochastic differential equations,” *Communications on Pure and Applied Mathematics*, vol. 54, no. 10, pp. 1169–1214, 2001. [Online]. Available: http://dx.doi.org/10.1002/cpa.10000

[2] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, ser. Applications of Mathematics. Springer-Verlag, 1992. [Online]. Available: http://books.google.com.sa/books?id=7bkZAQAAIAAJ

[3] G. Milstein, *Numerical Integration of Stochastic Differential Equations*, ser. Environmental Science and Technology. Kluwer Academic Publishers, 1995. [Online]. Available: http://books.google.com.sa/books?id=o2y8Or_a4W0C

[4] D. Talay and L. Tubaro, “Expansion of the global error for numerical schemes solving stochastic differential equations,” *Stochastic Analysis and Applications*, vol. 8, no. 4, pp. 483–509, 1990. [Online]. Available: http://www.informaworld.com/10.1080/07362999008809220

[5] C. Johnson, “Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations,” *SIAM Journal on Numerical Analysis*, vol. 25, no. 4, pp. pp. 908–926, 1988. [Online]. Available: http://www.jstor.org/stable/2157610

[6] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, “Introduction to Adaptive Methods for Differential Equations,” *Acta Numerica*, vol. 4, no. -1, pp. 105–158, 1995. [Online]. Available: http://dx.doi.org/10.1017/s0962492900002531

[7] K.-S. Moon, A. Szepessy, R. Tempone, and G. E. Zouraris, “Convergence rates for adaptive weak approximation of stochastic differential equations,” *Stochastic*

Analysis and Applications, vol. 23, no. 3, pp. 511–558, 2005. [Online]. Available: http://www.informaworld.com/10.1081/SAP-200056678

[8] K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone, "An adaptive algorithm for ordinary, stochastic and partial differential equations," in *Recent Advances in Adaptive Computation*, ser. Contemporary mathematics. American Mathematical Society, 2005, no. 383, pp. 325–343, qC 20100824.

[9] F. Black and M. S. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–54, May-June 1973. [Online]. Available: http://ideas.repec.org/a/ucp/jpolec/v81y1973i3p637-54.html

[10] G. V. Milovanovic and Z. Udovicic, "Calculation of coefficients of a cardinal b-spline." *Appl. Math. Lett.*, vol. 23, no. 11, pp. 1346–1350, 2010. [Online]. Available: http://dblp.uni-trier.de/db/journals/appml/appml23.html#MilovanovicU10

[11] V. Bally and D. Talay, "The Law of the Euler scheme for stochastic differential equations : I. convergence rate of the distribution function," INRIA, Rapport de recherche RR-2244, 1994. [Online]. Available: http://hal.inria.fr/inria-00074427

[12] M. B. Giles, "Multilevel monte carlo path simulation." *Operations Research*, vol. 56, no. 3, pp. 607–617, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/ior/ior56.html#Giles08

[13] R. Avikainen, "On irregular functionals of SDEs and the Euler scheme," *Finance and Stochastics*, vol. 13, no. 3, pp. 381–401, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1007/s00780-009-0099-7

[14] M. Giles and L. Szpruch, *Multilevel Monte Carlo methods for applications in finance.* World Scientific, 2013, pp. 3–48.

[15] H. Hoel, E. von Schwerin, A. Szepessy, and R. Tempone, "Implementation and analysis of an adaptive multilevel monte carlo algorithm," KTH, Numerical Analysis, NA, Tech. Rep. 2012:6, 2012, qC 20120508.

APPENDICES

A Brownian Bridge Interpolation

Let us consider the Wiener path $W = (W_{t_1}, W_{t_2}, \dots, W_{t_n})$ which has the PDF

$$p_W(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}x^T \Sigma^{-1} x}$$

with covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ where $\Sigma(t_i, t_j) = \min(t_i, t_j)$. The structure of the Cholesky factorized $\sqrt{\Sigma}$ is particularly simple

$$\sqrt{\Sigma} = \begin{pmatrix} \sqrt{t_1} & 0 & \dots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \dots & \sqrt{t_N - t_{N-1}} \end{pmatrix}.$$

Through a simple computation, we can easily show that

$$\Sigma^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{t_1}} & 0 & \dots & 0 \\ -\frac{1}{\sqrt{t_2 - t_1}} & \frac{1}{\sqrt{t_2 - t_1}} & \ddots & 0 \\ 0 & -\frac{1}{\sqrt{t_3 - t_2}} & \frac{1}{\sqrt{t_3 - t_2}} & 0 \\ 0 & 0 & \dots & \dots \end{pmatrix},$$

and hence the PDF of W can be written as

$$\begin{aligned} p_W(x) &= \frac{1}{\sqrt{(2\pi)^n|K|}} e^{-\frac{1}{2}(K^{-1/2}x)^T(K^{-1/2}x)} \\ &= \frac{1}{\sqrt{(2\pi)^n|K|}} e^{-\frac{x_1^2}{2|t_1|}} e^{-\frac{(x_2-x_1)^2}{2(t_2-t_1)}} \dots e^{-\frac{(x_n-x_{n-1})^2}{2(t_n-t_{n-1})}}. \end{aligned}$$

Suppose we want to add one more element vector at a time $t_m < t^* < t_{m+1}$:

$$W^* = (W_{t_1}, W_{t_2}, \dots, W_{t_m}, W_{t^*}, W_{t_{m+1}}, \dots, W_{t_n}).$$

The Brownian bridge is then defined as the conditional distribution of $W(t^*)|W$. Considering $x^* = (x_1, x_2, \dots, x_m, x_*, x_{m+1}, \dots, x_n)$, and letting Σ^* denote W^*'s covariance matrix, we have that

$$\begin{aligned} p_{W(t^*)|W}(x_*) &= p_{W^*}(x^*)/p_W(x) \\ &= \frac{\frac{1}{\sqrt{(2\pi)^{n+1}|K^*|}} e^{-\frac{x_1^2}{2|t_1|}} \dots e^{-\frac{(x_*-x_m)^2}{2(t^*-t_m)}} e^{-\frac{(x_{m+1}-x_*)^2}{2(t_{m+1}-t^*)}} \dots e^{-\frac{(x_n-x_{n-1})^2}{2(t_n-t_{n-1})}}}{\frac{1}{\sqrt{(2\pi)^n|K|}} e^{-\frac{x_1^2}{2|t_1|}} \dots e^{-\frac{(x_{m+1}-x_m)^2}{2(t_{m+1}-t_m)}} \dots e^{-\frac{(x_n-x_{n-1})^2}{2(t_n-t_{n-1})}}}. \end{aligned}$$

By a simple computation, we get

$$p(W_{t^*}|\vec{W} = x_*) = \frac{1}{\sqrt{2\pi\frac{(t_{m+1}-t^*)(t^*-t_m)}{(t_{m+1}-t_m)}}} e^{-\frac{(t_{m+1}-t_m)}{2(t_{m+1}-t^*)(t^*-t_m)}\left(x_*-\left(x_m+\frac{t^*-t_m}{t_{m+1}-t_m}(x_{m+1}-x_m)\right)\right)^2},$$

and we conclude that for $t_m < t^* < t_{m+1}$,

$$W_{t^*}|W \sim \mathcal{N}\left(W_{t_m} + \frac{t^*-t_m}{t_{m+1}-t_m}(W_{t_{m+1}} - W_{t_m}), \frac{(t_{m+1}-t^*)(t^*-t_m)}{t_{m+1}-t_m}\right).$$

So we see that the interpolation value $W_{t^*}|W$ only depends on nearest neighbors, i.e.,

$$p_{W_{t^*}|W} = p_{W_{t^*}|(W_{t_m}, W_{t_{m+1}})}.$$

B Mollified Binary Payoff Function

The window function w_δ is defined by

$$w_\delta(x) = \mathbf{1}_{-\delta \leq x \leq \delta}.$$

A first order differentiable kernel is obtained by

$$H_\delta = w_\delta * w_\delta.$$

In fact, once we do the convolution we determine a scaling factor such that ϕ_δ verifies the property of a density. Then, we obtain

$$H_\delta(y) = \begin{cases} 0, & \text{if } |y-K| > \delta, \\ \frac{1-|y-K|\delta^{-1}}{\delta}, & \text{else.} \end{cases}$$

To obtain a twice differentiable kernel function, we convolve the window function with itself three times

$$H_\delta = w_\delta * (w_\delta * w_\delta).$$

The result of the convolution is then scaled to construct a kernel with compact support in $[-\delta, \delta]$

$$H_\delta(y) = \begin{cases} 0, & \text{if } |y-K| > \delta, \\ \frac{3}{2\delta} - \frac{2}{\delta^2}(y-K) - \frac{2}{\delta^3}(y-K-\delta/2)^2, & \text{if } 0 \leq y-K \leq \frac{\delta}{2}, \\ \frac{3}{2\delta} - \frac{2}{\delta^2}(y-K) + \frac{2}{\delta^3}(y-K-\delta/2)^2, & \text{if } \frac{\delta}{2} \leq y-K \leq \delta, \\ \frac{3}{2\delta} + \frac{2}{\delta^2}(y-K) + \frac{2}{\delta^3}(y-K+\delta/2)^2, & \text{if } -\delta \leq y-K \leq -\frac{\delta}{2}, \\ \frac{3}{2\delta} + \frac{2}{\delta^2}(y-K) - \frac{2}{\delta^3}(y-K+\delta/2)^2, & \text{if } -\frac{\delta}{2} \leq y-K \leq 0. \end{cases}$$

The approximate function which is the convolution of the kernel and the Heaviside function is then

$$g_\delta(x) = \begin{cases} 0, & \text{if } x < K-\delta, \\ 1, & \text{if } x > \delta + K, \\ \frac{5}{12} + \frac{3}{2\delta}(x-K) - \frac{1}{\delta^2}(x-K)^2 - \frac{2}{3\delta^3}(x-K-\delta/2)^3, & \text{if } 0 \leq x-K \leq \frac{\delta}{2}, \\ \frac{5}{12} + \frac{3}{2\delta}(x-K) - \frac{1}{\delta^2}(x-K)^2 + \frac{2}{3\delta^3}(x-K-\delta/2)^3, & \text{if } \frac{\delta}{2} \leq x-K \leq \delta, \\ \frac{7}{12} + \frac{3}{2\delta}(x-K) + \frac{1}{\delta^2}(x-K)^2 + \frac{2}{3\delta^3}(x-K+\delta/2)^3, & \text{if } -\delta \leq x-K \leq -\frac{\delta}{2}, \\ \frac{7}{12} + \frac{3}{2\delta}(x-K) + \frac{1}{\delta^2}(x-K)^2 - \frac{2}{3\delta^3}(x-K+\delta/2)^3, & \text{if } -\frac{\delta}{2} \leq x-K \leq 0. \end{cases}$$