Smoothing the Payoff for Efficient Computation of Option Pricing in Time-Stepping Setting

1 Introduction

Many option pricing problems require the computation of multivariate integrals. The dimension of these integrals is determined by the number of independent stochastic factors (e.g. the number of time steps in the time discretization or the number of assets under consideration). The high dimension of these integrals can be treated with dimension-adaptive quadrature methods to have the desired convergence behavior.

Unfortunately, in many cases, the integrand contains either kinks and jumps. In fact, an option is normally considered worthless if the value falls below a predetermined strike price. A kink (discontinuity in the gradients) is present when the payoff function is continuous, while a jump (discontinuity in the function) exists when the payoff corresponds to a binary or other digital options. The existence of kinks or jumps in the integrand heavily degrades the performance of quadrature formulas. In this work, we are interested in solving this problem by using adaptive sparse grids quadrature (ASGQ) methods coupled with suitable transformations. The main idea is to find lines or areas of discontinuity and to employ suitable transformations of the integration domain. Then by a pre-integration (smoothing) step with respect to the dimension containing the kink/jump, we end up with integrating only over the smooth parts of the integrand and the fast convergence of the sparse grid method can be regained.

One can ignore the kinks and jumps, and apply directly a method for integration over $\mathbb{R}^d$. Despite the significant progress in SG methods [8] for high dimensional integration of smooth integrands, few works have been done to deal with cases involving integrands with kinks or jumps due to the decreasing performance of SG methods in the presence of kinks and jumps.

Some works [15, 6, 16, 17, 27] addressed similar kind of problems, characterized by the presence of kinks and jumps, but with much more emphasis on Quasi Monte Carlo (QMC). In [15, 16, 17], an analysis of the performance of Quasi Monte Carlo (QMC) and SG methods has been conducted, in the presence of kinks and jumps. In [15, 16], the authors studied the terms of the ANOVA decomposition of functions with kinks defined on d-dimensional Euclidean space $\mathbb{R}^d$, and showed that under some assumptions all but the the highest order ANOVA term of the 2^d ANOVA terms can be smooth for the case of an arithmetic Asian option with the Brownian bridge construction. Furthermore, [17] extended the work in [15, 16] from kinks to jumps for the case of an arithmetic average digital Asian option with the principal component analysis (PCA). The main findings in [15, 16] was obtained for an integrand of the form $f(\mathbf{x}) = \max(\phi(\mathbf{x}), 0)$ with ϕ being smooth. In fact, by assuming i) the d-dimensional function ϕ has a positive partial derivative with respect to x_j for some $j \in \{1, \ldots, d\}$, ii) certain growth conditions at infinity are satisfied, the authors showed that the ANOVA terms of f that do not depend on the variable x_j are smooth. We note that

[15, 16, 17] focus more on theoretical aspects of applying QMC in such a setting. On the other hand, we focus more on specific practical problems, where we add the adaptivity paradigm to the picture.

A recent work [27] addresses similar kind of problems using QMC. Being very much related to [6], the authors i) assume that the conditional expectation can be computed explicitly, by imposing very strong assumptions. ii) Secondly, they use PCA on the gradients to reduce the effective dimension. In our work, we do not make such strong assumptions, which is why we need numerical methods, more precisely root finding and the quadrature in the first direction.

Add more details here.

2 Problem formulation and Setting

To motivate our approach, we start by the continuous time representation of the problem we are addressing.Then, we illustrate the spirit of our approach in the time stepping setting, for the particular case of basket option.

2.1 Continuous time formulation

The purpose of this work is to approximate $\mathrm{E}\left[g(\mathbf{X}_T)\right]$, where g is a certain payoff function and $\mathbf{X} = (X_1, \ldots, X_d)$ is described by the following SDE

$$dX_i = a_i(X)dt + \sum_{j=1}^{d} b_{ij}(X)dW_t^{(j)}. \tag{2.1}$$

Without loss of generality, we assume that the $\{W^{(j)}\}_{j=1}^d$ are uncorrelated (the correlation terms can be included in the diffusion terms b_j).

First, we start by representing hierarchically $\mathbf{W}$. In fact, we can write

$$\begin{aligned} W^{(j)}(t) &= \frac{t}{T}W^{(j)}(T) + B_j(t) \\ &= \frac{t}{\sqrt{T}}Z_j + B_j(t), \end{aligned} \tag{2.2}$$

with $Z_j \sim \mathcal{N}(0,1)$ iid and $\{B_j\}_{j=1}^d$ are the Brownian bridges.

Now, we aim to represent $\mathbf{Z} = (Z_1, \ldots, Z_d)$ hierarchically by using a discrete Brownian bridge (Bb) namely

$$\mathbf{Z} = \underbrace{\mathbb{P}_0\mathbf{Z}}_{\text{One dimensional projection}} + \underbrace{\mathbb{P}_\perp\mathbf{Z}}_{\text{Projection on the complementary}}, \tag{2.3}$$

where we write $\mathbb{P}_0\mathbf{Z} = (\mathbf{Z}, \mathbf{v})\mathbf{v}$, with $(.,.)$ denotes the scalar product and $||\ \mathbf{v}\ || = 1$. We can easily show that $Z_v := (Z, v)$ is normal with $\mathrm{E}\left[Z_v\right] = 0$ and $\mathrm{Var}(Z_v) = 1$. Furthermore, we can write

$$\begin{aligned} Z_j &= Z_v v_j + (\mathbb{P}_\perp\mathbf{Z})_j \\ &= Z_v v_j + (Z_v^\perp)_j. \end{aligned} \tag{2.4}$$

The first aim of the work is to determine the optimal direction $\mathbf{v}$. By optimal direction, we mean the direction that maximizes the smoothing effect, that is something like a variance of the component orthogonal to the kink. We find hard to formulate as for now but we may try to come up with a better formulation.

Going back to the SDE (2.1), we have

$$dX_i = a_i(X)dt + \sum_{j=1}^{d} b_{ij}(X)Z_j \frac{dt}{\sqrt{T}} + \sum_{j=1}^{d} b_{ij}(X)dB_j. \tag{2.5}$$

Using (2.4) implies

$$dX_i = \left(a_i(X) + \sum_{j=1}^{d} b_{ij}(X)\frac{Z_v v_j}{\sqrt{T}} \right) dt + \left(\sum_{j=1}^{d} b_{ij}(X)\frac{(Z_v^{\perp})_j}{\sqrt{T}} \right) dt + \sum_{j=1}^{d} b_{ij}(X)dB_j. \tag{2.6}$$

2.1.1 First approach (Brutal)

Assumption 1: We assume that the first term in the right-hand side of (2.6) is the dominant term compared to the remaining terms in the sense of variance of g.

In the following, we try to motivate assumption 1 for a simple case.

Proof of assumption 1. Let us assume from (2.6) that $a_i = 0$, $b_{ij} = \alpha = \text{constant}$, and $g : \mathbb{R}^d \to Id_{\mathbb{R}}$, we try here to motivate Assumption 1.

Let us integrate (2.6) form 0 to T, then we have

$$\begin{aligned} X_T - X_0 &= \left(\sqrt{T}\alpha\right) \left(\sum_{j=1}^{d} v_j \right) Z_v + \left(\sqrt{T}\alpha\right) \left(\sum_{j=1}^{d} v_j (Z_v)_j^{\perp} \right) + \alpha \sum_{j=1}^{d} \int_0^T dB_j \\ &= \left(\sqrt{T}\alpha\right) \left(\sum_{j=1}^{d} v_j \right) Z_v + \left(\sqrt{T}\alpha\right) \left(\sum_{j=1}^{d} v_j (Z_v)_j^{\perp} \right) + \alpha \sum_{j=1}^{d} \left(B_j(T) - B_j(0) \right), \end{aligned}$$

implying (if we neglect the correlation terms)

$$\begin{aligned} \mathrm{Var}[X_T - X_0] &\approx \left(\sqrt{T}\alpha\right)^2 \left(\sum_{j=1}^{d} v_j \right)^2 + \underbrace{\left(\sqrt{T}\alpha\right)^2 \mathrm{Var}\left[\sum_{j=1}^{d} v_j (Z_v)_j^{\perp} \right]}_{=0} + \alpha^2 \sum_{j=1}^{d} \mathrm{Var}\left[B_j(T) - B_j(0) \right], \\ &\approx T\alpha^2 \left(\sum_{j=1}^{d} v_j \right)^2 + \alpha^2 \sum_{j=1}^{d} \underbrace{\mathrm{Var}\left[B_j(T) - B_j(0) \right]}_{=0}, \\ &\approx T\alpha^2 \left(\sum_{j=1}^{d} v_j \right)^2. \end{aligned} \tag{2.7}$$

Therefore, we can conclude from (2.7) that the dominant term in the sense of global variance is coming from $\sum_{j=1}^{d} b_{ij}(X)\frac{Z_v v_j}{\sqrt{T}}dt$, which makes assumption 1 valid. □

Given assumption 1 above, we denote the approximate process $\widehat{X}$, whose dynamics are given by

$$\frac{d\widehat{X}_i}{dt} = a_i(\widehat{X}) + \sum_{j=1}^{d} b_{ij}(\widehat{X})\frac{Z_v v_j}{\sqrt{T}}, \quad 1 \le i \le d, \tag{2.8}$$

with mean

$$\frac{d\mathrm{E}\left[\widehat{X}_i\right]}{dt} = \mathrm{E}\left[a_i(\widehat{X})\right] + \sum_{j=1}^{d} \mathrm{E}\left[b_{ij}(\widehat{X})\frac{Z_v v_j}{\sqrt{T}}\right], \quad 1 \le i \le d,$$

and second moment

$$\begin{aligned}\frac{d\mathrm{E}\left[\widehat{X}_i^2\right]}{dt} &= 2\mathrm{E}\left[\widehat{X}_i\frac{d\widehat{X}_i}{dt}\right] \\ &= 2\left(\mathrm{E}\left[\widehat{X}_i a_i(\widehat{X})\right] + \sum_{j=1}^{d}\mathrm{E}\left[\widehat{X}_i b_{ij}(\widehat{X})\frac{Z_v v_j}{\sqrt{T}}\right]\right), \quad 1 \le i \le d.\end{aligned}$$

Now, let us expand $\widehat{X}$ around Z_v, that is

$$\widehat{X}_{(Z_v)} = \widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v + \dots \tag{2.9}$$

Then, for the first moment we have

$$\begin{aligned}\frac{d\mathrm{E}\left[\widehat{X}_i\right]}{dt} &\approx \mathrm{E}\left[a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\right] + \sum_{j=1}^{d}\mathrm{E}\left[b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\frac{Z_v v_j}{\sqrt{T}}\right], \quad 1 \le i \le d, \\ &\approx a_i(\widehat{X}_{(0)}) + \sum_{j=1}^{d}\left(\sum_{k=1}^{d} b'_{ij,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)}\right)\frac{\mathrm{E}\left[Z_v^2\right]v_j}{\sqrt{T}}, \quad 1 \le i \le d, \\ &= a_i(\widehat{X}_{(0)}) + \sum_{j=1}^{d}\left(\sum_{k=1}^{d} b'_{ij,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)}\right)\frac{v_j}{\sqrt{T}}, \quad 1 \le i \le d. \qquad (2.10)\end{aligned}$$

Similarly, if we approximate the second moment, we get for $1 \le i \le d$

$$\begin{aligned}\frac{d\mathrm{E}\left[\widehat{X}_i^2\right]}{dt} &\approx 2\left(\mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)}Z_v\right)a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\right] + \sum_{j=1}^{d}\mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)}Z_v\right)b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\frac{Z_v v_j}{\sqrt{T}}\right]\right), \qquad (2.11) \\ &\approx 2\left(\widehat{X}_{i,(0)}a_i(\widehat{X}_{(0)}) + \widehat{X}'_{i,(0)}\sum_{k=1}^{d} a'_{i,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d}\widehat{X}'_{i,(0)}b_{ij}(\widehat{X}_{(0)}) + \widehat{X}_{i,(0)}\left(\sum_{k=1}^{d}\widehat{X}'_{k,(0)}b'_{ij,k}(\widehat{X}_{(0)})\right)\right)\frac{v_j}{\sqrt{T}}\right).\end{aligned}$$

Regarding the covariance terms $E[X_i X_l]$, for $1 \leq i \neq l \leq d$, their dynamics can be approximated by

$$\begin{aligned}
\frac{d\mathrm{E}\left[\widehat{X}_i\widehat{X}_l\right]}{dt} &= \mathrm{E}\left[\widehat{X}_i\frac{d\widehat{X}_l}{dt}\right] + \mathrm{E}\left[\widehat{X}_l\frac{d\widehat{X}_i}{dt}\right] \\
&\approx \mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)}Z_v\right)a_l(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\right] + \sum_{j=1}^{d}\mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)}Z_v\right)b_{lj}(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\frac{Z_v v_j}{\sqrt{T}}\right] \\
&+ \mathrm{E}\left[\left(\widehat{X}_{l,(0)} + \widehat{X}'_{l,(0)}Z_v\right)a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\right] + \sum_{j=1}^{d}\mathrm{E}\left[\left(\widehat{X}_{l,(0)} + \widehat{X}'_{l,(0)}Z_v\right)b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)}Z_v)\frac{Z_v v_j}{\sqrt{T}}\right] \\
&\approx \left(\widehat{X}_{i,(0)}a_l(\widehat{X}_{(0)}) + \widehat{X}'_{i,(0)}\sum_{k=1}^{d}a'_{l,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d}\widehat{X}'_{i,(0)}b_{lj}(\widehat{X}_{(0)}) + \widehat{X}_{i,(0)}\left(\sum_{k=1}^{d}\widehat{X}'_{k,(0)}b'_{lj,k}(\widehat{X}_{(0)})\right)\right)\frac{v_j}{\sqrt{T}}\right) \\
&+ \left(\widehat{X}_{l,(0)}a_i(\widehat{X}_{(0)}) + \widehat{X}'_{l,(0)}\sum_{k=1}^{d}a'_{i,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d}\widehat{X}'_{l,(0)}b_{ij}(\widehat{X}_{(0)}) + \widehat{X}_{l,(0)}\left(\sum_{k=1}^{d}\widehat{X}'_{k,(0)}b'_{ij,k}(\widehat{X}_{(0)})\right)\right)\frac{v_j}{\sqrt{T}}\right).
\end{aligned} \tag{2.12}$$

The idea then is to maximize, at the final time, the smoothing effect, where $\widehat{X}_T \sim \mathcal{N}(\mu_T(\mathbf{v}), \Sigma_T(\mathbf{v}))$, such that μ_T, Σ_T are computed using (2.10), (2.11) and (2.12). We note that the computation of $\widehat{X}'$ is cheap and its evolution can be deduced from (2.8), resulting in

$$\begin{aligned}
&\frac{d\widehat{X}'_i}{dt} = \sum_{k=1}^{d}a'_{i,k}(\widehat{X})\widehat{X}'_k + \sum_{j=1}^{d}\left(b_{ij}(\widehat{X})\frac{v_j}{\sqrt{T}} + \sum_{k=1}^{d}\left(b'_{ij,k}(\widehat{X})\widehat{X}'_k\right)\frac{Z_v v_j}{\sqrt{T}}\right), \quad 1 \leq i \leq d, \\
&\widehat{X}'(t=0) = 0, \quad 1 \leq i \leq d,
\end{aligned}$$

which when evaluated at $Z_v = 0$ simplifies to

$$\begin{aligned}
&\frac{d\widehat{X}'_{i,(0)}}{dt} = \sum_{k=1}^{d}a'_{i,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)} + \sum_{j=1}^{d}\left(b_{ij}(\widehat{X}_{(0)})\frac{v_j}{\sqrt{T}}\right), \quad 1 \leq i \leq d, \\
&\widehat{X}'_{(0)}(t=0) = 0, \quad 1 \leq i \leq d.
\end{aligned}$$

In our context, we work mainly with two possible structures of payoff function g. In fact, for the cases of call/put options, the payoff g has a kink and will be of the form

$$g(\mathbf{x}) = \max(\phi(\mathbf{x}), 0). \tag{2.13}$$

One can also encounter jumps in the payoff when working with binary digital options. In this case, g is given by

$$g(\mathbf{x}) = \mathbf{1}_{(\phi(\mathbf{x})\geq 0)}. \tag{2.14}$$

We introduce the notation $\mathbf{x} = (x_j, \mathbf{x}_{-j})$, where $\mathbf{x}_{-j}$ denotes the vector of length $d-1$ denoting all

the variables other than x_j. Furthermore, we assume for some $j \in \{1, \dots, d\}$

$$\frac{\partial \phi}{\partial x_j}(\mathbf{x}) > 0, \ \forall \mathbf{x} \in \mathbb{R}^d \ \textbf{(Monotonicity condition)} \tag{2.15}$$

$$\lim_{x \to +\infty} \phi(\mathbf{x}) = \lim_{x \to +\infty} \phi(x_j, \mathbf{x}_{-j}) = +\infty, \text{ or } \frac{\partial^2 \phi}{\partial x_j^2}(\mathbf{x}) \ \textbf{(Growth condition)}. \tag{2.16}$$

2.2 Discrete time formulation

For practical purposes, we consider the basket option under multi-dimensional GBM model where the process $\mathbf{X}$ is the discretized d-dimensional Black-Scholes model and the payoff function g is given by

$$g(\mathbf{X}(T)) = \max \left(\sum_{j=1}^{d} c_j X^{(j)}(T) - K, 0 \right). \tag{2.17}$$

Precisely, we are interested in the d-dimensional lognormal example where the dynamics of the stock prices are given by

$$dX_t^{(j)} = \sigma^{(j)} X_t^{(j)} dW_t^{(j)}, \tag{2.18}$$

where $\{W^{(1)}, \dots, W^{(d)}\}$ are correlated Brownian motions with correlations ρ_{ij}.

We denote by $(Z_1^{(j)}, \dots, Z_N^{(j)})$ the N Gaussian independent rdvs that will be used to construct the path of the j-th asset $\overline{X}^{(j)}$, where $1 \leq j \leq d$ (d denotes the number of underlyings considered in the basket). We denote $\psi^{(j)} : (Z_1^{(j)}, \dots, Z_N^{(j)}) \to (B_1^{(j)}, \dots, B_N^{(j)})$ the mapping of Brownian bridge construction and by $\Psi : (\Delta t, \widetilde{B}_1^{(1)}, \dots, \widetilde{B}_N^{(1)}, \dots, \widetilde{B}_1^{(d)}, \dots, \widetilde{B}_N^{(d)}) \to \left(\overline{X}_T^{(1)}, \dots, \overline{X}_T^{(d)} \right)$, the mapping consisting of the time-stepping scheme, where $\widetilde{\mathbf{B}}$ is the correlated Brownian bridge that can be obtained from the non correlated Brownian bridge $\mathbf{B}$ through multiplication by the correlation matrix, we denote this transformation by $\mathcal{T} : \left(B_1^{(1)}, \dots, B_N^{(1)}, \dots, B_1^{(d)}, \dots, B_N^{(d)} \right) \to \left(\widetilde{B}_1^{(1)}, \dots, \widetilde{B}_N^{(1)}, \dots, \widetilde{B}_1^{(d)}, \dots, \widetilde{B}_N^{(d)} \right)$. Then, we can express the option price as

$$\begin{aligned}
\mathrm{E}\left[g(\mathbf{X}(T))\right] &\approx \mathrm{E}\left[g\left(\overline{X}_T^{(1)}, \dots, \overline{X}_T^{(d)} \right) \right] \\
&= \mathrm{E}\left[g\left(\Psi \circ \mathcal{T} \right) \left(B_1^{(1)}, \dots, B_N^{(1)}, \dots, B_1^{(d)}, \dots, B_N^{(d)} \right) \right] \\
&= \mathrm{E}\left[g\left(\Psi \circ \mathcal{T} \right) \left(\psi^{(1)}(Z_1^{(1)}, \dots, Z_N^{(1)}), \dots, \psi^{(d)}(Z_1^{(d)}, \dots, Z_N^{(d)}) \right) \right] \\
&= \int_{\mathbb{R}^{d \times N}} G(z_1^{(1)}, \dots, z_N^{(1)}, \dots, z_1^{(d)}, \dots, z_N^{(d)})) \rho_{d \times N}(\mathbf{z}) dz_1^{(1)} \dots dz_N^{(1)} \dots z_1^{(d)} \dots dz_N^{(d)},
\end{aligned} \tag{2.19}$$

where

$$\rho_{d \times N}(\mathbf{z}) = \frac{1}{(2\pi)^{d \times N/2}} e^{-\frac{1}{2} \mathbf{z}^T \mathbf{z}}.$$

In the discrete case, we can show that the numerical approximation of $X^{(j)}(T)$ satisfies

$$\begin{aligned}\overline{X}^{(j)}(T) &= X_0^{(j)} \prod_{i=0}^{N-1} \left[1 + \frac{\sigma^{(j)}}{\sqrt{T}} Z_1^{(j)} \Delta t + \sigma^{(j)} \Delta \widetilde{B}_i^{(j)}\right], \quad 1 \le j \le d \\ &= X_0^{(j)} \prod_{i=0}^{N-1} f_i^{(j)}(Z_1^{(j)}), \quad 1 \le j \le d. \end{aligned} \tag{2.20}$$

2.2.1 Step 1: Numerical smoothing

The first step of our idea is to smoothen the problem by solving the root finding problem in one dimension after using a sub-optimal linear mapping for the coarsest factors of the Brownian increments $\mathbf{Z}_1 = (Z_1^{(1)}, \ldots, Z_1^{(d)})$. In fact, let us define for a certain $d \times d$ matrix $\mathcal{A}$, the linear mapping

$$\mathbf{Y} = \mathcal{A}\mathbf{Z}_1. \tag{2.21}$$

Then from (2.20), we have

$$\begin{aligned}\overline{X}^{(j)}(T) &= X_0^{(j)} \prod_{i=0}^{N-1} f_i^{(j)}(\mathcal{A}^{-1}\mathbf{Y})_j, \quad 1 \le j \le d, \\ &= X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1, \mathbf{Y}_{-1}) \quad 1 \le j \le d \end{aligned} \tag{2.22}$$

where, by defining $\mathcal{A}^{\text{inv}} = \mathcal{A}^{-1}$, we have

$$\begin{aligned} g_i^{(j)}(Y_1, \mathbf{Y}_{-1}) &= \left[1 + \frac{\sigma^{(j)}}{\sqrt{T}} \left(\sum_{i=1}^{d} A_{ji}^{\text{inv}} Y_i\right) \Delta t + \sigma^{(j)} \Delta \widetilde{B}_i^{(j)}\right] \\ &= \left[1 + \frac{\sigma^{(j)} \Delta t}{\sqrt{T}} A_{j1}^{\text{inv}} Y_1 + \frac{\sigma^{(j)}}{\sqrt{T}} \left(\sum_{i=2}^{d} A_{ji}^{\text{inv}} Y_i\right) \Delta t + \sigma^{(j)} \Delta \widetilde{B}_i^{(j)}\right]. \end{aligned} \tag{2.23}$$

Therefore, in order to determine Y_1^*, we need to solve

$$x = \sum_{j=1}^{d} c_j X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*(x), \mathbf{Y}_{-1}), \tag{2.24}$$

which implies that the location of the kink point for the approximate problem is equivalent to finding the roots of the polynomial $P(Y_1^*(K))$, given by

$$P(Y_1^*(K)) = \left(\sum_{j=1}^{d} c_j X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*)\right) - K. \tag{2.25}$$

Using **Newton iteration method**, we use the expression $P' = \frac{dP}{dY_1^*}$, and we can easily show that

$$P'(Y_1^*) = \sum_{j=1}^{d} c_j X_0^{(j)} \frac{\sigma^{(j)} \Delta t A_{j1}^{\text{inv}}}{\sqrt{T}} \left(\prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*)\right) \left[\sum_{i=0}^{N-1} \frac{1}{g_i^{(j)}(Y_1^*)}\right]. \tag{2.26}$$

Remark 2.1. For our purposes, we suggest already that the coarsest factors of the Brownian increments are the most important ones, compared to the remaining factors. Furthermore, the choice of the linear mapping $\mathcal{A}$ creates a new hierarchy in terms of smoothness.

Remark 2.2. A general choice for $\mathcal{A}$ should be in the family of rotations. However, we think that a sufficiently good matrix $\mathcal{A}$ would be the one leading to $Y_1 = \sum_{i=1}^{d} Z_1^{(i)}$ up to re-scaling. Therefore, the first attempt will be to set $\mathcal{A}$ to be a rotation matrix, with first row leading to $Y_1 = \sum_{i=1}^{d} Z_1^{(i)}$ up to re-scaling, and with no constraints for the remaining rows. In practice, we construct $\mathcal{A}$ by fixing the first row to be $\frac{1}{\sqrt{d}}\mathbf{1}_{1\times d}$ and the remaining rows are obtained by Gram-Schmidt procedure.

2.2.2 Step 2: Integration

At this stage, we want to perform the pre-integrating step with respect to y_1^*. In fact, using Fubini's theorem, we have from (2.19)

$$\begin{aligned}
\mathrm{E}\left[g(\mathbf{X}(T))\right] &\approx \int_{\mathbb{R}^{d\times N}} G(z_1^{(1)},\dots,z_N^{(1)},\dots,z_1^{(d)},\dots,z_N^{(d)}))\rho_{d\times N}(\mathbf{z})dz_1^{(1)}\dots dz_N^{(1)}\dots z_1^{(d)}\dots dz_N^{(d)} \\
&= \int_{\mathbb{R}^{dN-1}} \left(\int_{\mathbb{R}} G(y_1,\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1\right)\rho_{d-1}(\mathbf{y}_{-1})d\mathbf{y}_{-1}\rho_{d\times(N-1)}(\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})d\mathbf{z}_{-1}^{(1)}\dots d\mathbf{z}_{-1}^{(d)} \\
&= \int_{\mathbb{R}^{dN-1}} h(\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\rho_{d-1}(\mathbf{y}_{-1})d\mathbf{y}_{-1}\rho_{d\times(N-1)}(\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})d\mathbf{z}_{-1}^{(1)}\dots d\mathbf{z}_{-1}^{(d)}, \\
&= \mathrm{E}\left[h(\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\right],
\end{aligned} \tag{2.27}$$

where

$$\begin{aligned}
h(\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)}) &= \int_{\mathbb{R}} G(y_1,\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1 \\
&= \int_{-\infty}^{y_1^*} G(y_1,\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1 \\
&+ \int_{y_1^*}^{+\infty} G(y_1,\mathbf{y}_{-1},\mathbf{z}_{-1}^{(1)},\dots,\mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1.
\end{aligned} \tag{2.28}$$

We generally do not have a closed form for h. Therefore, the pre-integration step should be performed numerically after solving the root finding problem using the numerical smoothing explained in Section 2.2.1.

Remark 2.3. We note that conditions ((2.15) and (2.16)) imply that for each $\mathbf{Y}_{-1}$, the function G either has a simple root y_1^* or is positive for all $y_1 \in \mathbb{R}$.

There seems to be a certain amount of overlap between section 2 and 3 regarding the hierarchical representation. I will see how to make cleaner.

3 Analyticity and Smoothness Analysis

3.1 Haar construction of Brownian motion revisited

For simplicity we shall assume throughout that we work on a fixed time interval $[0,T]$ with $T = 1$.

With the Haar mother wavelet

$$\psi(t) \coloneqq \begin{cases} 1, & 0 \le t < \frac{1}{2}, \\ -1, & \frac{1}{2} \le t < 1, \\ 0, & \text{else}, \end{cases} \tag{3.1}$$

we construct the Haar basis of $L^2([0,1])$ by setting

$$\psi_{-1}(t) \coloneqq \mathbb{1}_{[0,1]}(t), \tag{3.2a}$$

$$\psi_{n,k}(t) \coloneqq 2^{n/2}\psi\left(2^n t - k\right), \quad n \in \mathbb{N}_0,\ k = 0, \dots, 2^n - 1. \tag{3.2b}$$

We note that $\operatorname{supp} \psi_{n,k} = [2^{-n}k, 2^{-n}(k+1)]$. Moreover, we define a grid $\mathcal{D}^n \coloneqq \left\{ t_\ell^n \mid \ell = 0, \dots, 2^{n+1} \right\}$ by $t_\ell^n \coloneqq \frac{\ell}{2^{n+1}}$. Notice that the Haar functions up to level n are piece-wise constant with points of discontinuity given by $\mathcal{D}^n$.

Next we define the antiderivatives of the basis functions

$$\Psi_{-1}(t) \coloneqq \int_0^t \psi_{-1}(s)ds, \tag{3.3a}$$

$$\Psi_{n,k}(t) \coloneqq \int_0^t \psi_{n,k}(s)ds. \tag{3.3b}$$

For an i.i.d. set of standard normal random variables (*coefficients*) Z_{-1}, $Z_{n,k}$, $n \in \mathbb{N}_0$, $k = 0, \dots, 2^n - 1$, we can then define a standard Brownian motion

$$W_t \coloneqq Z_{-1}\Psi_{-1}(t) + \sum_{n=0}^{\infty} \sum_{k=0}^{2^n-1} Z_{n,k}\Psi_{n,k}(t), \tag{3.4}$$

and the truncated version

$$W_t^N \coloneqq Z_{-1}\Psi_{-1}(t) + \sum_{n=0}^{N} \sum_{k=0}^{2^n-1} Z_{n,k}\Psi_{n,k}(t). \tag{3.5}$$

Note that W^N already coincides with W along the grid $\mathcal{D}^N$. We define the corresponding increments for any function or process F by

$$\Delta_\ell^N F \coloneqq F(t_{\ell+1}^N) - F(t_\ell^N). \tag{3.6}$$

3.2 Stochastic differential equations

For simplicity we consider a one-dimensional SDE X given by

$$dX_t = b(X_t)dW_t, \quad X_0 = x \in \mathbb{R}. \tag{3.7}$$

We assume that b is bounded and has bounded derivatives of all orders. Recall that we want to compute

$$E\left[g\left(X_T\right)\right]$$

for some function $g : \mathbb{R} \to \mathbb{R}$ which is not necessarily smooth. We also define the solution of the Euler scheme along the grid $\mathcal{D}^N$ by $X_0^N \coloneqq X_0 = x$ and

$$X_{\ell+1}^N \coloneqq X_\ell^N + b\left(X_\ell^N\right) \Delta_\ell^N W. \tag{3.8}$$

For convenience, we also define $X_T^N \coloneqq X_{2^N}^N$.

Clearly, the random variable X_ℓ^N is a deterministic function of the random variables Z_{-1} and $Z^N \coloneqq (Z_{n,k})_{n=0,\dots,N,\ k=0,\dots 2^n-1}$. Abusing notation, let us therefore write

$$X_\ell^N = X_\ell^N\left(Z_{-1}, Z^N\right)$$

for the appropriate (now deterministic) map $X_\ell^N : \mathbb{R} \times \mathbb{R}^{2^{N+1}-1} \to \mathbb{R}$. We shall write $y \coloneqq z_{-1}$ and z^N for the (deterministic) arguments of the function X_ℓ^N.

A note of caution is in order regarding convergence as $N \to \infty$: while the sequence of random processes $X_\cdot^N$ converges to the solution of (3.7) (under the usual assumptions on b), this is not true in any sense for the deterministic functions.

Define

$$H^N(z^N) \coloneqq E\left[g\left(X_T^N\left(Z_{-1}, z^N\right)\right)\right]. \tag{3.9}$$

We claim that H^N is analytic.

Let us consider a mollified version g_δ of g and the corresponding function H_δ^N (defined by replacing g with g_δ in (3.9)). Tacitly assuming that we can interchange integration and differentiation, we have

$$\frac{\partial H_\delta^N(z^N)}{\partial z_{n,k}} = E\left[g_\delta'\left(X_T^N\left(Z_{-1}, z^N\right)\right) \frac{\partial X_T^N(Z_{-1}, z^N)}{\partial z_{n,k}}\right].$$

Multiplying and dividing by $\frac{\partial X_T^N(Z_{-1}, z^N)}{\partial y}$ and replacing the expectation by an integral w.r.t. the standard normal density, we obtain

$$\frac{\partial H_\delta^N(z^N)}{\partial z_{n,k}} = \int_{\mathbb{R}} \frac{\partial g_\delta\left(X_T^N(y, z^N)\right)}{\partial y} \left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N) \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy. \tag{3.10}$$

If we are able to do integration by parts, then we can get rid of the mollification and obtain smoothness of H^N since we get

$$\frac{\partial H^N(z^N)}{\partial z_{n,k}} = -\int_{\mathbb{R}} g\left(X_T^N(y, z^N)\right) \frac{\partial}{\partial y}\left[\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N) \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\right] dy.$$

We realize that there is a potential problem looming in the inverse of the derivative w.r.t. y.[1] Before we continue, let us introduce the following notation: for sequences of random variables F_N, G_N we say that $F_N = \mathcal{O}(G_N)$ if there is a random variable C with finite moments of all orders such that for all N we have $|F_N| \le C\,|G_N|$ a.s.

[1] Let us assume that $X_T^N(y, z^N) = \cos(y) + z_{n,k}$. Then (3.10) is generally not integrable.

Assumption 3.1. There are positive random variables C_p with finite moments of all orders such that

$$\forall N \in \mathbb{N},\ \forall \ell_1, \dots, \ell_p \in \{0, \dots, 2^N - 1\} : \left| \frac{\partial^p X_T^N}{\partial X_{\ell_1}^N \cdots \partial X_{\ell_p}^N} \right| \leq C_p \text{ a.s.}$$

In terms of the above notation, that means that $\frac{\partial^p X_T^N}{\partial X_{\ell_1}^N \cdots \partial X_{\ell_p}^N} = \mathcal{O}(1)$.

Remark 3.2. It is probably hard to argue that a deterministic constant C may exist.

Assumption 3.1 is natural, but now we need to make a much more serious assumption, which is probably difficult to verify in practice.

Assumption 3.3. For any $p \in \mathbb{N}$ we have that

$$\left(\frac{\partial X_T^N}{\partial y} \left(Z_{-1}, Z^N \right) \right)^{-p} = \mathcal{O}(1).$$

Lemma 3.4. *We have*

$$\frac{\partial X_T^N}{\partial z_{n,k}}(Z_{-1}, Z^N) = 2^{-n/2+1} \mathcal{O}(1)$$

in the sense that the $\mathcal{O}(1)$ term does not depend on n or k.

Proof. First let us note that Assumption 3.1 implies that $\frac{\partial X_T^N}{\partial \Delta_\ell^N W} = \mathcal{O}(1)$. Indeed, we have

$$\frac{\partial X_T^N}{\partial \Delta_\ell^N W} = \frac{\partial X_T^N}{\partial X_{\ell+1}^N} \frac{\partial X_{\ell+1}^N}{\partial \Delta_\ell^N W} = \mathcal{O}(1) b(X_\ell^N) = \mathcal{O}(1).$$

Next we need to understand which increments Δ_ℓ^N do depend on $Z_{n,k}$. This is the case iff $\operatorname{supp} \psi_{n,k}$ has a non-empty intersection with $]t_\ell^N, t_{\ell+1}^N[$. Explicitly, this means that

$$\ell 2^{-(N-n+1)} - 1 < k < (\ell + 1) 2^{-(N-n+1)}.$$

If we fix N, k, n, this means that the derivative of $\Delta_\ell^N W$ w.r.t. $Z_{n,k}$ does not vanish iff

$$2^{N-n+1} k \leq \ell < 2^{N-n+1}(k+1).$$

Noting that

$$\left| \frac{\partial \Delta_\ell^N W}{\partial Z_{n,k}} \right| = \left| \Delta_\ell^N \Psi_{n,k} \right| \leq 2^{-(N-n/2)}, \tag{3.11}$$

we thus have

$$\frac{\partial X_T^N}{\partial z_{n,k}}(Z_{-1}, Z^N) = \sum_{\ell = 2^{N-n+1} k}^{2^{N-n+1}(k+1)-1} \frac{\partial X_T^N}{\partial \Delta_\ell^N W} \frac{\partial \Delta_\ell^N W}{\partial Z_{n,k}} = 2^{N-n+1} 2^{-(N-n/2)} \mathcal{O}(1) = 2^{-n/2+1} \mathcal{O}(1). \tag{3.12}$$

□

Lemma 3.5. *In the same sense as in Lemma 3.4 we have*

$$\frac{\partial^2 X_T^N}{\partial y \partial z_{n,k}}(Z_{-1}, Z^N) = 2^{-n/2+1}\mathcal{O}(1).$$

Proof. $\Delta_\ell^N W$ is a linear function in Z_{-1} and Z^N, implying that all mixed derivatives $\frac{\partial^2 \Delta_\ell^N W}{\partial Z_{n,k}\partial Z_{-1}}$ vanish. From equation (3.12) we hence see that

$$\frac{\partial^2 X_T^N}{\partial z_{n,k}\partial y}(Z_{-1}, Z^N) = \sum_{\ell=2^{N-n+1}k}^{2^{N-n+1}(k+1)-1} \frac{\partial^2 X_T^N}{\partial \Delta_\ell^N W \partial Z_{-1}} \frac{\partial \Delta_\ell^N W}{\partial Z_{n,k}}.$$

Further,

$$\frac{\partial^2 X_T^N}{\partial \Delta_\ell^N W \partial Z_{-1}} = \sum_{j=0}^{2^{N+1}-1} \frac{\partial^2 X_T^N}{\partial \Delta_\ell^N W \partial \Delta_j^N W} \frac{\partial \Delta_j^N W}{\partial Z_{-1}}.$$

Note that

$$\frac{\partial^2 X_T^N}{\partial \Delta_\ell^N W \partial \Delta_j^N W} = \frac{\partial^2 X_T^N}{\partial X_{\ell+1}^N \partial X_{j+1}^N} b(X_\ell^N) b(X_j^N) + \mathbb{1}_{j<\ell} \frac{\partial X_T^N}{\partial X_\ell^N} b'(X_\ell^N) \frac{\partial X_\ell^N}{\partial X_{j+1}^N} b(X_j^N) = \mathcal{O}(1) \tag{3.13}$$

by Assumption 3.1. We also have $\frac{\partial \Delta_j^N W}{\partial Z_{-1}} = \mathcal{O}(2^{-N})$, implying the statement of the lemma. □

Remark 3.6. Lemma 3.4 and 3.5 also hold (mutatis mutandis) for $z_{n,k} = y$ (with $n = 0$).

Proposition 3.7. *We have* $\frac{\partial H^N(z^N)}{\partial z_{n,k}} = \mathcal{O}(2^{-n/2})$ *in the sense that the constant in front of* $2^{-n/2}$ *does not depend on* n *or* k.

Proof. We have

$$\begin{aligned}
\frac{\partial H^N(z^N)}{\partial z_{n,k}} &= -\int_{\mathbb{R}} g\left(X_T^N(y, z^N)\right) \frac{\partial}{\partial y}\left[\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N) \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\right] dy \\
&= -\int_{\mathbb{R}} g\left(X_T^N(y, z^N)\right) \left[-\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-2} \frac{\partial^2 X_T^N}{\partial y^2}(y, z^N) \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N) + \right. \\
&\quad \left. + \left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial^2 X_T^N}{\partial z_{n,k}\partial y}(y, z^N) - y\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N)\right] \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy.
\end{aligned}$$

Notice that when $F^N(Z_{-1}, Z^N) = \mathcal{O}(c)$ for some deterministic constant c, then this property is retained when integrating out one of the random variables, i.e., we still have

$$\int_{\mathbb{R}} F^N(y, Z^N) \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy = \mathcal{O}(c).$$

Hence, Lemma 3.4 and Lemma 3.5 together with Assumption 3.3 (for $p = 2$) imply that

$$\frac{\partial H^N(z^N)}{\partial z_{n,k}} = \mathcal{O}(2^{-n/2})$$

with constants independent of n and k. □

For the general case we need

Lemma 3.8. *For any $p \in \mathbb{N}$ and indices $n_1, \dots, n_p$ and $k_1, \dots, k_p$ (satisfying $0 \le k_j < 2^{n_j}$) we have (with constants independent from n_j, k_j)*

$$\frac{\partial^p X_T^N}{\partial z_{n_1,k_1} \cdots \partial z_{n_p,k_p}}(Z_1, Z^N) = \mathcal{O}\left(2^{-\sum_{j=1}^p n_j/2}\right).$$

The result also holds (mutatis mutandis) if one or several z_{n_j,k_j} are replaced by $y = z_{-1}$ (with n_j set to 0).

Proof. We start noting that each $\Delta_\ell^N W$ is a linear function of (Z_{-1}, Z^N) implying that all higher derivatives of $\Delta_\ell^N W$ w.r.t. (Z_{-1}, Z^N) vanish. Hence,

$$\frac{\partial^p X_T^N}{\partial Z_{n_1,k_1} \cdots \partial Z_{n_p,k_p}} = \sum_{\ell_1 = 2^{N-n_1+1}k_1}^{2^{N-n_1+1}(k_1+1)-1} \cdots \sum_{\ell_p = 2^{N-n_p+1}k_p}^{2^{N-n_p+1}(k_p+1)-1} \frac{\partial^p X_T^N}{\partial \Delta_{\ell_1}^N W \cdots \partial \Delta_{\ell_p}^N W} \frac{\partial \Delta_{\ell_1}^N W}{\partial Z_{n_1,k_1}} \cdots \frac{\partial \Delta_{\ell_p}^N W}{\partial Z_{n_p,k_p}}.$$

By a similar argument as in (3.13) we see that

$$\frac{\partial^p X_T^N}{\partial \Delta_{\ell_1}^N \cdots \partial \Delta_{\ell_p}^N W} = \mathcal{O}(1).$$

By (3.11) we see that each summand in the above sum is of order $\prod_{j=1}^p 2^{-(N-n_j/2)}$. The number of summands in total is $\prod_{j=1}^p 2^{N-n_j+1}$. Therefore, we obtain the desired result. □

Theorem 3.9. *For any $p \in \mathbb{N}$ and indices $n_1, \dots, n_p$ and $k_1, \dots, k_p$ (satisfying $0 \le k_j < 2^{n_j}$) we have (with constants independent from n_j, k_j)*

$$\frac{\partial^p H^N}{\partial z_{n_1,k_1} \cdots \partial z_{n_p,k_p}}(Z^N) = \mathcal{O}\left(2^{-\sum_{j=1}^p n_j/2}\right).$$

The result also holds (mutatis mutandis) if one or several z_{n_j,k_j} are replaced by $y = z_{-1}$ (with n_j set to 0). In particular, H^N is a smooth function.

Remark 3.10. We actually expect that H^N is analytic, but a formal proof seems difficult. In particular, note that our proof below relies on successively applying the above trick for enabling integration by parts: divide by $\frac{\partial X_T^N}{\partial y}$ and then integrate by parts. This means that the number of terms (denoted by ■ below) increases fast as p increases by the product rule of differentiation. Hence, the constant in fron of the $\mathcal{O}\left(2^{-\sum_{j=1}^p n_j/2}\right)$ term will depend on p and increase in p. In that sense, Theorem 3.9 needs to be understood as an assertion about the anisotropy in the variables $z_{n,k}$ rather than a statement about the behaviour of higher and higher derivatives of H^N. In fact, one can see that in our proof the number of summands increases as $p!$ in p. Therefore, the statement of the theorem does not already imply analyticity. Of course, this problem is an artifact of our construction, and there is no reason to assume such a behaviour in general.

Sketch of a proof of Theorem 3.9. We apply integration by parts p times as in the proof of Proposition 3.7, which shows that we can again replace the mollified payoff function g_δ by the true, non-smooth one g. Moreover, from the procedure we obtain a formula of the form

$$\frac{\partial^p H^N}{\partial z_{n_1,k_1} \cdots \partial z_{n_p,k_p}}(z^N) = \int_{\mathbb{R}} g\left(X_T^N(y, z^N)\right) \blacksquare \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy,$$

where $\blacksquare$ represents a long sum of products of various terms. However, it is quite easy to notice the following structure: ignoring derivatives w.r.t. y, each summand contains all derivatives w.r.t. $z_{n_1,k_1}, \ldots, z_{n_p,k_p}$ exactly once. (Generally speaking, each summand will be a product of derivatives of X_T^N w.r.t. some z_{n_j,k_j}s, possibly with other terms such as polynomials in y and derivatives w.r.t. y included.) As all other terms are assumed to be of order $\mathcal{O}(1)$ by Assumptions 3.1 and 3.3, this implies the claimed result by Lemma 3.8. □

4 Details of our hierarchical method

In the following, we describe our approach which can be seen as a two stage method. In the first stage, we use root finding procedure to perform the numerical smoothing described in Section 2.2.1, then in a second stage we perform the numerical integration to compute (2.27), described in Section 2.2.2, by employing hierarchical adaptive sparse grids quadrature, using the same construction as in [18]. Therefore, the initial integration problem that we are solving lives in $dN-1$-dimensional space, which becomes very large as either the number of time steps N, used in the discretization scheme, increases, or the number the assets increase.

We describe the ASGQ method in our context in Section 4.1. To make an effective use of ASGQ, we apply two transformations to overcome the issue of facing a high dimensional integrand. The first transformation consists of applying a hierarchical path generation method, based on Brownian bridge (Bb) construction, with the aim of reducing the effective dimension as described in Section 4.2. The second transformation consists of applying Richardson extrapolation to reduce the bias, resulting in reducing the maximum number of dimensions needed for the integration problem. Details about Richardson extrapolation are provided in Section 4.3.

Since g can have a kink or jump. Computing h in (2.28) should be carried carefully to not deteriorate the smoothness of h. This can be done by applying a root finding procedure and then computing the uni-variate integral by summing the terms coming from integrating in each region where g is smooth. We provide details about the root finding procedure in Section 4.4.

If we denote by $\mathcal{E}_{\text{tot}}$ the total error of approximating the expectation in (2.27) using the ASGQ estimator, Q_N, then we have a natural error decomposition

$$\begin{aligned}\mathcal{E}_{\text{tot}} &\leq \left|\mathrm{E}\left[g(\mathbf{X}(T))\right] - \mathrm{E}\left[h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\right]\right| + \left|\mathrm{E}\left[h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\right] - Q_N\right| \\ &\leq \mathcal{E}_B(N) + \mathcal{E}_Q(\mathrm{TOL}_{\text{ASGQ}}, N), \end{aligned} \tag{4.1}$$

where $\mathcal{E}_Q$ is the quadrature error, $\mathcal{E}_B$ is the bias.

4.1 Adaptive Sparse Grids

We assume that we want to approximate the expected value $\mathrm{E}[f(Y)]$ of an analytic function $f\colon \Gamma \to \mathbb{R}$ using a tensorization of quadrature formulas over Γ.

To introduce simplified notations, we start with the one-dimensional case. Let us denote by β a non negative integer, referred to as a "stochastic discretization level", and by $m : \mathbb{N} \to \mathbb{N}$ a strictly increasing function with $m(0) = 0$ and $m(1) = 1$, that we call "level-to-nodes function". At level β, we consider a set of $m(\beta)$ distinct quadrature points in $\mathbb{R}$, $\mathcal{H}^{m(\beta)} = \{y_\beta^1, y_\beta^2, \ldots, y_\beta^{m(\beta)}\} \subset \mathbb{R}$, and a set of quadrature weights, $\boldsymbol{\omega}^{m(\beta)} = \{\omega_\beta^1, \omega_\beta^2, \ldots, \omega_\beta^{m(\beta)}\}$. We also let $C^0(\mathbb{R})$ be the set of real-valued continuous functions over $\mathbb{R}$. We then define the quadrature operator as

$$Q^{m(\beta)} : C^0(\mathbb{R}) \to \mathbb{R}, \quad Q^{m(\beta)}[f] = \sum_{j=1}^{m(\beta)} f(y_\beta^j)\omega_\beta^j.$$

In our case, we have in (2.27) a multi-variate integration problem with, $f := h$, $\mathbf{Y} = (\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$, and $\Gamma = \mathbb{R}^{dN-1}$, in the previous notations. Furthermore, since we are dealing with Gaussian densities, using Gauss-Hermite quadrature points is the appropriate choice.

We define for any multi-index $\boldsymbol{\beta} \in \mathbb{N}^{dN-1}$

$$Q^{m(\boldsymbol{\beta})} : C^0(\mathbb{R}^{dN-1}) \to \mathbb{R}, \quad Q^{m(\boldsymbol{\beta})} = \bigotimes_{n=1}^{dN-1} Q^{m(\beta_n)},$$

where the n-th quadrature operator is understood to act only on the n-th variable of f. Practically, we obtain the value of $Q^{m(\boldsymbol{\beta})}[f]$ by considering the tensor grid $\mathcal{T}^{m(\boldsymbol{\beta})} = \times_{n=1}^{dN-1} \mathcal{H}^{m(\beta_n)}$ with cardinality $\#\mathcal{T}^{m(\boldsymbol{\beta})} = \prod_{n=1}^{dN-1} m(\beta_n)$ and computing

$$Q^{m(\boldsymbol{\beta})}[f] = \sum_{j=1}^{\#\mathcal{T}^{m(\boldsymbol{\beta})}} f(\widehat{y}_j)\overline{\omega}_j,$$

where $\widehat{y}_j \in \mathcal{T}^{m(\boldsymbol{\beta})}$ and $\overline{\omega}_j$ are products of weights of the univariate quadrature rules.

A direct approximation $\mathrm{E}[f[\mathbf{Y}]] \approx Q^{\boldsymbol{\beta}}[f]$ is not an appropriate option due to the well-known "curse of dimensionality". We use a hierarchical adaptive sparse grids[2] quadrature strategy, specifically using the same construction as ASGQ, and which uses stochastic discretizations and a classic sparsification approach to obtain an effective approximation scheme for $\mathrm{E}[f]$.

To be concrete, in our setting, we are left with a $dN-1$-dimensional Gaussian random input, which is chosen independently, resulting in $dN-1$ numerical parameters for ASGQ, which we use as the basis of the multi-index construction. For a multi-index $\boldsymbol{\beta} = (\beta_n)_{n=1}^{dN-1} \in \mathbb{N}^{dN-1}$, we denote by $Q_N^{\boldsymbol{\beta}}$, the result of approximating (2.27) with a number of quadrature points in the i-th dimension equal to $m(\beta_i)$. We further define the set of differences $\Delta Q_N^{\boldsymbol{\beta}}$ as follows: for a single index $1 \leq i \leq dN-1$, let

$$\Delta_i Q_N^{\boldsymbol{\beta}} = \begin{cases} Q_N^{\boldsymbol{\beta}} - Q_N^{\boldsymbol{\beta}'}, \text{ with } \boldsymbol{\beta}' = \boldsymbol{\beta} - e_i, \text{ if } \boldsymbol{\beta}_i > 0, \\ Q_N^{\boldsymbol{\beta}}, \quad \text{otherwise,} \end{cases}$$

where e_i denotes the ith $dN-1$-dimensional unit vector. Then, $\Delta Q_N^{\boldsymbol{\beta}}$ is defined as

$$\Delta Q_N^{\boldsymbol{\beta}} = \left(\prod_{i=1}^{dN-1} \Delta_i \right) Q_N^{\boldsymbol{\beta}}.$$

[2]More details about sparse grids can be found in [8].

The ASGQ estimator used for approximating (2.27), and using a set of multi-indices $\mathcal{I} \subset \mathbb{N}^{dN-1}$ is given by

$$Q_N^{\mathcal{I}} = \sum_{\boldsymbol{\beta} \in \mathcal{I}} \Delta Q_N^{\boldsymbol{\beta}}. \tag{4.2}$$

The quadrature error in this case is given by

$$\mathcal{E}_Q(\mathrm{TOL}_{\mathrm{ASGQ}}, N) = \left| Q_N^{\infty} - Q_N^{\mathcal{I}} \right| \leq \sum_{\ell \in \mathbb{N}^{dN-1} \setminus \mathcal{I}} \left| \Delta Q_N^{\ell} \right|. \tag{4.3}$$

We define the work contribution, $\Delta \mathcal{W}_{\boldsymbol{\beta}}$, to be the computational cost required to add $\Delta Q_N^{\boldsymbol{\beta}}$ to $Q_N^{\mathcal{I}}$, and the error contribution, $\Delta E_{\boldsymbol{\beta}}$, to be a measure of how much the quadrature error, defined in (4.3), would decrease once $\Delta Q_N^{\boldsymbol{\beta}}$ has been added to $Q_N^{\mathcal{I}}$, that is

$$\begin{aligned} \Delta \mathcal{W}_{\boldsymbol{\beta}} &= \mathrm{Work}[Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \mathrm{Work}[Q_N^{\mathcal{I}}] \\ \Delta E_{\boldsymbol{\beta}} &= \left| Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - Q_N^{\mathcal{I}} \right|. \end{aligned} \tag{4.4}$$

The construction of the optimal $\mathcal{I}$ will be done by profit thresholding, that is, for a certain threshold value $\overline{T}$, and a profit of a hierarchical surplus defined by

$$P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}},$$

the optimal index set $\mathcal{I}$ for ASGQ is given by $\mathcal{I} = \{\boldsymbol{\beta} : P_{\boldsymbol{\beta}} \geq \overline{T}\}$.

Remark 4.1. The analiticity assumption, stated in the beginning of Section 4.1, is crucial for the optimal performance of our proposed method. In fact, although we face the issue of the "curse of dimensionality" when increasing N, the analiticity of f implies a spectral convergence for sparse grids quadrature. A discussion about the analiticity of our integrand is provided in Section 3.

4.2 Brownian bridge (Bb) construction

In the literature of adaptive sparse grids and QMC, several hierarchical path generation methods (PGMs) or transformation methods have been proposed to reduce the effective dimension. Among these transformations, we cite the Brownian bridge (Bb) construction [24, 25, 9], the principal component analysis (PCA) [1] and the linear transformation (LT) [20].

In our context, the Brownian motion on a time discretization can be constructed either sequentially using a standard random walk construction, or hierarchically using other PGMs as listed above. For our purposes, to make an effective use of ASGQ, which benefits from anisotropy, we use the Bb construction since it produces dimensions with different importance for ASGQ, contrary to a random walk procedure for which all the dimensions of the stochastic space have equal importance. In fact, Bb uses the first several coordinates of the low-discrepancy points to determine the general shape of the Brownian path, and the last few coordinates influence only the fine detail of the path. Consequently, this transformation reduces the effective dimension of the problem, which results in accelerating the ASGQ method by reducing the computational cost.

Let us denote $\{t_i\}_{i=0}^N$ the grid of time steps. Then the Bb construction [14] consists of the following: given a past value B_{t_i} and a future value B_{t_k}, the value B_{t_j} (with $t_i < t_j < t_k$) can be generated according to

$$B_{t_j} = (1-\rho)B_{t_i} + \rho B_{t_k} + \sqrt{\rho(1-\rho)(k-i)\Delta t}z,\ z \sim \mathcal{N}(0,1),$$

where $\rho = \frac{j-i}{k-i}$.

4.3 Richardson extrapolation

Another representation that we couple with the ASGQ and QMC methods is Richardson extrapolation [26]. In fact, applying level K_{R} (level of extrapolation) of Richardson extrapolation dramatically reduces the bias, and as a consequence reduces the number of time steps N needed in the coarsest level to achieve a certain error tolerance. As a consequence, Richardson extrapolation directly reduces the total dimension of the integration problem for achieving some error tolerance.

Let us denote by $(X_t)_{0\le t\le T}$ a certain stochastic process and by $(\widehat{X}_{t_i}^h)_{0\le t_i\le T}$ its approximation using a suitable scheme with a time step h. Then, for sufficiently small h, and a suitable smooth function f, we assume that

$$\mathrm{E}\left[f(\widehat{X}_T^h)\right] = \mathrm{E}\left[f(X_T)\right] + ch + \mathcal{O}\left(h^2\right). \tag{4.5}$$

Applying (4.5) with discretization step $2h$, we obtain

$$\mathrm{E}\left[f(\widehat{X}_T^{2h})\right] = \mathrm{E}\left[f(X_T)\right] + 2ch + \mathcal{O}\left(h^2\right),$$

implying

$$2\mathrm{E}\left[f(\widehat{X}_T^{2h})\right] - \mathrm{E}\left[f(\widehat{X}_T^h)\right] = \mathrm{E}\left[f(X_T)\right] + \mathcal{O}\left(h^2\right).$$

For higher levels of extrapolations, we use the following: Let us denote by $h_J = h_0 2^{-J}$ the grid sizes (where h_0 is the coarsest grid size), by K_{R} the level of the Richardson extrapolation, and by $I(J, K_{\mathrm{R}})$ the approximation of $\mathrm{E}\left[f(X_T)\right]$ by terms up to level K_{R} (leading to a weak error of order K_{R}), then we have the following recursion

$$I(J,K_{\mathrm{R}}) = \frac{2^{K_{\mathrm{R}}} I(J,K_{\mathrm{R}}-1) - I(J-1,K_{\mathrm{R}}-1)}{2^{K_{\mathrm{R}}}-1}, \quad J = 1,2,\ldots, K_{\mathrm{R}} = 1,2,\ldots$$

4.4 Root Finding

From Section 2.2, we denoted the location of irregularity (the kink) by y_1^*, that is G, defined in (2.27), is not smooth at the point $(y_1^*, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$. Let us call R the mapping such that: $R : (\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)}) \to y_1^*$. Generally, there might be, for given $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$

- no solution, i.e., the integrand in the definition of $h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$ above is smooth (*best case*);
- a unique solution;

- multiple solutions.

Generally, we need to assume that we are in the first or second case. Specifically, we need that

$$(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) \mapsto h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) \text{ and } (\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) \mapsto \widehat{h}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$$

are smooth, where $\widehat{h}$ denotes the numerical approximation of h based on a grid containing $R(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$. In particular, R itself should be smooth in $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$. This would already be challenging in practice in the third case. Moreover, in the general situation we expect the number of solutions to increase when the discretization of the SDE gets finer.

In many situations, case 2 (which is thought to include case 1) can be guaranteed by monotonicity (see assumption (2.15)). For instance, in the case of one-dimensional SDEs with z_1 representing the terminal value of the underlying Brownian motion, this can often be seen from the SDE itself. Specifically, if each increment "dX" is increasing in z_1, no matter the value of X, then the solution X_T must be increasing in z_1. This is easily seen to be true in examples such as the Black-Scholes model and the CIR process. (Strictly speaking, we have to distinguish between the continuous and discrete time solutions. In these examples, it does not matter.) On the other hand, it is also quite simple to construct counter examples, where monotonicity fails, for instance SDEs for which the "volatility" changes sign, such as a trigonometric function.[3]

Even in multi-dimensional settings, such monotonicity conditions can hold in specific situations. For instance, in case of a basket option in a multivariate Black Scholes framework, we can choose a linear combination of the terminal values of the driving Bm, denoted by Y_1 in Section 2.2.1, such that the basket is a monotone function of y_1. (The coefficients of the linear combination will depend on the correlations and the weights of the basket.) However, in that case this may actually not correspond to the optimal "rotation" in terms of optimizing the smoothing effect.

5 Error discussion

5.1 Errors in smoothing

For the analysis it is useful to assume that $\widehat{h}$ is a smooth function of $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$, but in reality this is not going to be true. Specifically, if the true location y_1^* of the non-smoothness in the system was available, we could actually guarantee $\widehat{h}$ to be smooth, for instance by choosing

$$\widehat{h}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) = \sum_{k=-K}^{K} \eta_k G\left(\zeta_k(R(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})), \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right),$$

for points $\zeta_k \in \mathbb{R}$ with $\zeta_0 = y_1$ and corresponding weights η_k.[4] However, in reality we have to approximate numerically R by $\overline{R}$ with error $\left|R(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) - \overline{R}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})\right| \leq \delta$. Now, the actual integrand in $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$ becomes

$$\overline{h}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}) \coloneqq \sum_{k=-K}^{K} \eta_k G\left(\zeta_k(\overline{R}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})), \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right),$$

[3] Actually, in every such case the simple remedy is to replace the volatility by its absolute value, which does not change the law of the solution. Hence, there does not seem to be a one-dimensional counter-example.

[4] Of course, the points ζ_k have to be chosen in a systematic manner depending on y_1.

which we cannot assume to be smooth anymore. On the other hand, if $\zeta_k(y)$ is a continuous function of R, and R and $\overline{R}$ are continuous in $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$, then *eventually* we will have

$$\left\|\widehat{h} - \overline{h}\right\|_\infty \leq \text{TOL}, \quad \left\|h - \overline{h}\right\|_\infty \leq \text{TOL},$$

i.e., the smooth functions h and $\widehat{h}$ are close to the integrand $\overline{h}$. (Of course, this may depend on us choosing a good enough quadrature ζ!)

Remark 5.1. If the adaptive collocation used for computing the integral of $\overline{h}$ depends on derivatives (or difference quotients) of its integrand $\overline{h}$, then we may also need to make sure that derivatives of $\overline{h}$ are close enough to derivatives of $\widehat{h}$ or h. This may require higher order solution methods for determining y.

6 Numerical experiments for Numerical smoothing with ASGQ

In this section, we conduct our experiments for four different examples: i) single binary option under discretized GBM model, ii) single call option under discretized GBM model, iii) $2d$-basket call option under discretized GBM model and iv) $4d$-basket call option under discretized GBM model .

For each example, we estimate the weak error (Bias) of MC, then we conduct a comparison between MC and ASGQ in terms of errors and computational time. We show tables and plots reporting the different relative errors involved in the MC method (bias and statistical error[5] estimates), and in ASGQ (bias and quadrature error estimates). While fixing a sufficiently small error tolerance in the price estimates, we also compare the computational time needed for both methods to meet the desired error tolerance. We note that in all cases the actual work (runtime) is obtained using an Intel(R) Xeon(R) CPU E5-268 architecture.

Through our conducted numerical experiments for each parameter set, we follow these steps to achieve our reported results:

i) For a fixed number of time steps, N, we compute an accurate estimate, using a large number of samples, M, of the biased MC solution. This step also provides us with an estimate of the bias error, $\mathcal{E}_B(N)$, defined by (4.1).

ii) The estimated biased solution is used as a reference solution to ASGQ to compute the quadrature error, $\mathcal{E}_Q(\text{TOL}_{\text{ASGQ}}, N)$, defined by (4.3).

iii) In order to compare with MC method, the number of samples, M, is chosen so that the statistical error of the Monte Carlo method, $\mathcal{E}_S(M)$, satisfies

$$\mathcal{E}_S(M) = \mathcal{E}_B(N) = \frac{\mathcal{E}_{\text{tot}}}{2}, \tag{6.1}$$

where $\mathcal{E}_B(N)$ is the bias as defined in (4.1) and $\mathcal{E}_{\text{tot}}$ is the total error.

[5]The statistical error estimate of MC is $C_\alpha \frac{\sigma_M}{\sqrt{M}}$, where M is the number of samples and $C_\alpha = 1.96$ for 95% confidence interval.

We show the summary of our numerical findings in Table 6.1, which highlights the computational gains achieved by ASGQ over MC method to meet a certain error tolerance, which we set approximately to 1%. More detailed results for each case are provided in Sections 6.1.2, 6.1.3 and 6.2.

Along the numerical part below, we compare 3 methods:

i) Adaptive sparse grids quadrature (ASGQ) based on construction as explained in Section 4.1.

ii) Plain Monte Carlo where we apply standard Monte Carlo and we refer to it by "MC".

iii) Monte Carlo applied to the smoothed payoff function after numerical smoothing and we refer to it by "MC+ root finding".

Example	Level of Richardson extrapolation	Total relative error	CPU time (ASGQ/MC)
Single binary option (GBM)	without	2%	25%
Single call option (GBM)	without	0.9%	4.6%
$2d$-Basket call option (GBM)	without	0.9%	4.8%
$4d$-Basket call option (GBM)	without	0.8%	3.7%
$1d$ call option (Heston)	level 1	0.3%	30%

Table 6.1: Summary of relative errors and computational gains, achieved by the different methods. In this table, we highlight the computational gains achieved by ASGQ over MC method to meet a certain error tolerance. We provide details about the way we compute these gains for each case in the following sections.

6.1 Options under the discretized one dimensional GBM model

The first two examples that we will test are the single binary and call options under GBM model where the process X is the discretized one dimensional GBM model and the payoff function g is the indicator or maximum function as given by (2.14) and (2.13) respectively. Precisely, we are interested in the one dimensional lognormal example where the dynamics of the stock are given by

$$dX_t = \sigma X_t dW_t, \tag{6.2}$$

where $\{W_t, 0 \leq t \leq T\}$ is a standard one-dimensional Brownian motion. In the discrete case, the numerical approximation of $X(T)$, using N time steps ($\Delta t = \frac{T}{N}$), satisfies

$$\begin{aligned} \overline{X}_T &= \Psi(\Delta t, z_1, \Delta B_0, \ldots, \Delta B_{N-1}), \\ &= \Psi(\Delta t, \Phi(z_1, \ldots, z_N)), \end{aligned} \tag{6.3}$$

wehre $(z_1, \ldots, z_N)$ are standard Gaussian random variables, for some path function Ψ and Brownian bridge map Φ. As explained in Sections 2.2 and 4, the first step of our approach is determining the location of irregularity (kink). In the following, we want to compare different ways for identifying the location of the kink for this model.

6.1.1 Determining the kink location

Exact location of the kink for the continuous problem

Let us denote y^* an invertible function that satisfies

$$X(T; y^*(x), B) = x. \tag{6.4}$$

We can easily prove that the expression of y^* for model given by (6.2) is given by

$$y^*(x) = \left(\log(x/x_0) + T\sigma^2/2\right) \frac{1}{\sqrt{T}\sigma}, \tag{6.5}$$

and since the kink for Black-Scholes model occurs at $x = K$, where K is the strike price then the exact location of the continuous problem is given by

$$y^*(K) = \left(\log(K/x_0) + T\sigma^2/2\right) \frac{1}{\sqrt{T}\sigma}. \tag{6.6}$$

Location of the kink for the discrete problem

The discrete problem of model (6.2) is solved by simulating

$$\begin{aligned}
\overline{X}_{t_1} &= \overline{X}_{t_0} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_0\right] \\
\overline{X}_{t_2} &= \overline{X}_{t_1} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_1\right] \\
&\vdots = \vdots \\
\overline{X}_{t_N} &= \overline{X}_{t_{N-1}} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_{N-1}\right]
\end{aligned} \tag{6.7}$$

implying that

$$\overline{X}(T) = X_0 \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_i\right]. \tag{6.8}$$

Therefore, in order to determine y^*, we need to solve

$$x = \overline{X}(T; y^*, B) = X_0 \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} y^*(x) \Delta t + \sigma \Delta B_i\right], \tag{6.9}$$

which implies that the location of the kink point for the approximate problem is equivalent to finding the roots of the polynomial $P(y^*(K))$, given by

$$P(y^*(K)) = \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} y^*(K) \Delta t + \sigma \Delta B_i\right] - \frac{K}{X_0}. \tag{6.10}$$

The exact location of the kink can be obtained exactly by solving exactly $P(y^*(K)) = 0$.

In our work, we try to find the roots of polynomial $P(y^*(K))$, given by (6.10), by using **Newton iteration method**. In this case, we need the expression $P' = \frac{dP}{dy^*}$. If we denote $f_i(y) = 1 + \frac{\sigma}{\sqrt{T}} y\Delta t + \sigma\Delta B_i$, then we can easily show that

$$P'(y) = \frac{\sigma\Delta t}{\sqrt{T}} \left(\prod_{i=0}^{N-1} f_i(y) \right) \left[\sum_{i=0}^{N-1} \frac{1}{f_i(y)} \right]. \tag{6.11}$$

Therefore, in this case, the integrand $h(\mathbf{z}_{-1})$, as expressed in (2.28), is given by

$$h(\mathbf{z}_{-1}) = \int_{\mathbb{R}} g\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1})\right) \rho_1(z_1) dz_1. \tag{6.12}$$

We get the kink point by running Newton iteration for root solving of the polynomial P as expressed in (6.10) with a precision of 10^{-10}. We decompose the total integration domain into sub-domains such that the integrand is smooth in the interior of each sub-domain and such that the kink is located along the boundary of these areas. The total integral is then given as the sum of the separate integrals, *i.e.*

$$\begin{aligned} h(\mathbf{z}_{-1}) &:= \int_{\mathbb{R}} g\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1})\right) \rho_1(z_1) dz_1 \\ &= \int_{-\infty}^{y^*} g\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1})\right) \rho_d(z_1) dz_1 + \int_{y^*}^{\infty} g\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1})\right) \rho_d(z_1) dz_1, \end{aligned} \tag{6.13}$$

where we use Gauss-Laguerre quadrature with β points to approximate each part.

6.1.2 Results for the single binary option under discretized GBM model

In this case, the integrand $h(\mathbf{z}_{-1})$ is given by

$$\begin{aligned} h(\mathbf{z}_{-1}) &= \int_{\mathbb{R}} \mathbf{1}_{\Psi\circ\Phi(T;z_1,\mathbf{z}_{-1})>K} \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1 \\ &= P(Y > y_*(K)), \end{aligned} \tag{6.14}$$

where $y_*(x)$, is an invertible function that satisfies

$$\Psi \circ \Phi(T; y_*(x), \mathbf{z}_{-1}) = x. \tag{6.15}$$

We get the kink point by running Newton iteration with a precision of 10^{-10}. The parameters that we used in our numerical experiments are: $T = 1$, $\sigma = 0.4$ and $S_0 = K = 100$. The exact value of this case is 0.42074029.

Figure 6.1 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and ASGQ in Tables 6.2 and 6.3, and Figure 6.2. Our numerical experiments show that ASGQ requires approximately 25% of the work of MC to achieve a total relative error of around 2%.

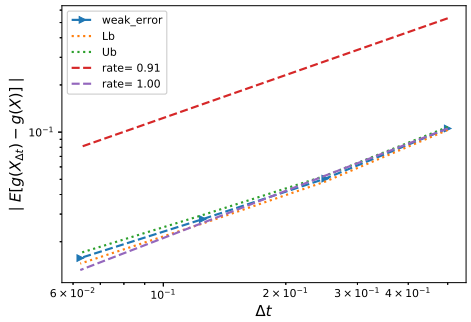


Figure 6.1: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, using MC with $M = 10^4$ samples for the binary option example. The upper and lower bounds are 95% confidence intervals.

Method	Steps		
	4	8	16
ASGQ	**0.05** (0.04,0.01)	**0.02** (0.02,0.002)	**0.01** (0.01,0.002)
MC+root finding	**0.08** (0.04,0.04)	**0.04** (0.02,0.02)	**0.02** (0.01,0.01)
M(# MC samples)	10^1	4×10^1	10^2
MC	**0.1** (0.05,0.05)	**0.05** (0.025,0.025)	**0.02** (0.01,0.01)
M(# MC samples)	10^3	8×10^3	5×10^4

Table 6.2: Total relative error of ASGQ, with different tolerances, and MC to compute binary option price for different number of time steps, without Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps		
	4	8	16
ASGQ	0.8	2	9
MC+root finding	0.2	1	9
MC	0.3	3	29

Table 6.3: Comparison of the computational time of MC and ASGQ, used to compute binary option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.

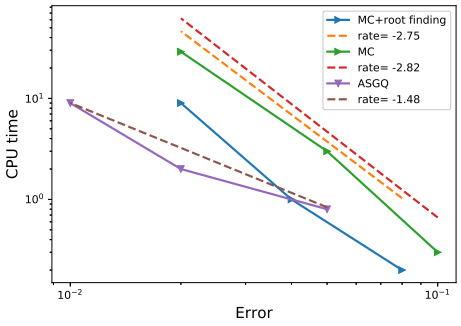


Figure 6.2: Computational work comparison for ASGQ and MC methods, for the case of binary option. This plot shows that to achieve a relative error below 1%, ASGQ outperforms MC method in terms of computational time

6.1.3 Results for the single call option example

In this case, the integrand $h(\mathbf{z}_{-1})$ is given by

$$h(\mathbf{z}_{-1}) = \int_{\mathbb{R}} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1. \tag{6.16}$$

We get the kink point by running Newton iteration with a precision of 10^{-10}. We decompose the total integration domain $\mathbb{R}$ into sub-domains such that the integrand is smooth in the interior of each sub-domain and such that the kink is located along the boundary of these areas. The total integral is then given as the sum of the separate integrals, *i.e.*

$$\begin{aligned} h(\mathbf{z}_{-1}) &:= \int_{\mathbb{R}} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dy \\ &= \int_{-\infty}^{y^*} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1 \\ &+ \int_{y^*}^{\infty} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1, \end{aligned} \tag{6.17}$$

where we use Gauss-Laguerre quadrature with β points to get each part.

The parameters that we used in our numerical experiments are: $T = 1$, $\sigma = 0.4$ and $S_0 = K = 100$. The exact value of this case is 15.85193755.

Figure 6.3 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and ASGQ in Tables 6.4 and 6.5, and Figure 6.4. Our numerical experiments show that ASGQ requires approximately 5% of the work of MC to achieve a total relative error of around 0.9%.

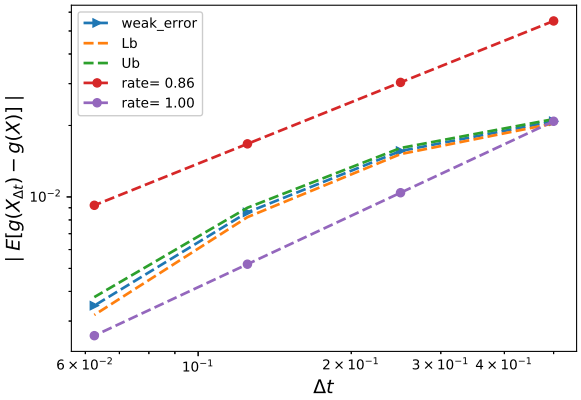


Figure 6.3: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, using MC with $M = 4 \times 10^5$ samples for the call option example. The upper and lower bounds are 95% confidence intervals.

Method	Steps			
	2	4	8	16
ASGQ	**0.022** (0.021,0.001)	**0.018** (0.016,0.002)	**0.009** (0.009,0.0001)	**0.004** (0.004,0.0004)
MC +root finding	**0.041** (0.021,0.02)	**0.032** (0.016,0.016)	**0.018** (0.009,0.009)	**0.008** (0.004,0.004)
M(# MC samples)	10^2	3×10^2	10^3	4×10^3
MC	**0.04** (0.02,0.02)	**0.032** (0.016,0.016)	0.02 (0.01,0.01)	**0.008** (0.004,0.004)
M(# MC samples)	3×10^4	5×10^4	2×10^5	8×10^5

Table 6.4: Total relative error of ASGQ, with different tolerances, and MC to compute call option price for different number of time steps, without Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps			
	2	4	8	16
ASGQ	0.3	3	17	473
MC +root finding	3	16	70	408
MC	3	13	76	380

Table 6.5: Comparison of the computational time of MC and ASGQ, used to compute call option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.

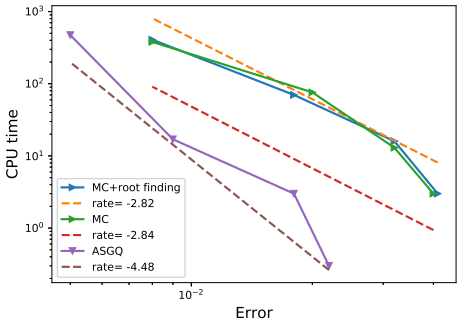


Figure 6.4: Computational work comparison for ASGQ and MC methods, for the case of call option. This plot shows that to achieve a relative error below 1%, ASGQ outperforms significantly MC method in terms of computational time.

6.2 The basket call option under GBM model

The third example that we consider is the multi-dimensional basket call option under GBM.

6.3 $d = 2$

For illustration, we start with the two dimensional basket call option with parameters: $S_0^{1,2} = K = 100$, $\sigma_{1,2} = 0.4$, $\rho = 0.3$, $T = 1$ $r = 0$ and $c_{1,2} = 1/2$, the reference value for those parameters is 12.90. Figure 6.5 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and ASGQ in Tables 6.6 and 6.7, and Figure 6.6. Our numerical experiments show that ASGQ requires approximately 5% of the work of MC to achieve a total relative error of around 0.9%.

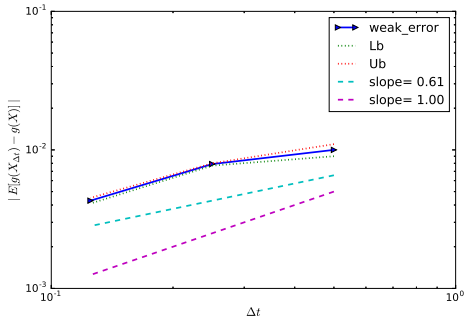


Figure 6.5: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, for the two dimensional basket call option with a number of Laguerre quadrature points $\beta = 128$ and number of samples for MC $M = 10^7$. The upper and lower bounds are 95% confidence intervals.

Method	Steps		
	2	4	8
ASGQ	**0.016** (0.01,0.006)	**0.0134** (0.0079,0.0055)	**0.0085** (0.0043,0.0042)
MC +root finding	**0.02** (0.01,0.01)	**0.0155** (0.0079,0.0076)	**0.0081** (0.0043,0.0038)
M(# MC samples)	2×10^3	5×10^3	2×10^4
MC	**0.02** (0.01,0.01)	**0.0155** (0.0079,0.0076)	**0.0083** (0.0043,0.004)
M(# MC samples)	10^5	2×10^5	8×10^5

Table 6.6: Total relative error of ASGQ, with different tolerances, and MC to compute two dimensional basket call option price for different number of time steps, without Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps		
	2	4	8
ASGQ	4	8	21
MC +root finding	281	814	3888
MC	34	93	442

Table 6.7: Comparison of the computational time of MC and ASGQ, used to compute two dimensional basket call option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.

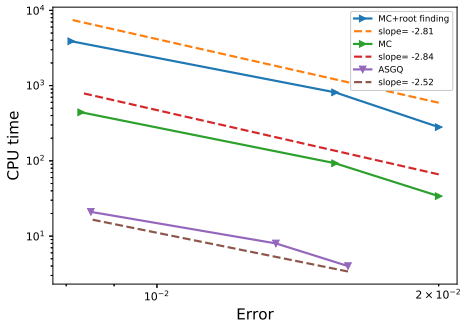


Figure 6.6: Computational work comparison for ASGQ and MC methods, for the case of two dimensional basket call option. This plot shows that to achieve a relative error below 1%, ASGQ outperforms significantly MC method in terms of computational time.

6.4 $d = 4$

We consider now the four dimensional basket call option with parameters: $S_0^{1,2,3,4} = K = 100$, $\sigma_{1,2,3,4} = 0.4$, $\rho = 0.3$, $T = 1$ $r = 0$ and $c_{1,2,3,4} = 1/4$, the reference value for those parameters is 11.04. Figure 6.7 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and ASGQ in Tables 6.8 and 6.9, and Figure 6.8. Our numerical experiments show that ASGQ requires approximately 4% of the work of MC to achieve a total relative error of around 0.8%.

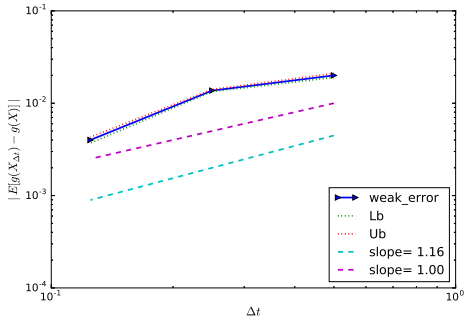


Figure 6.7: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, for the 4-dimensional basket call option with a number of Laguerre quadrature points $\beta = 512$ and number of samples for MC $M = 4 \times 10^6$. The upper and lower bounds are 95% confidence intervals.

Method	Steps		
	2	4	8
ASGQ	**0.04** (0.02,0.02)	**0.027** (0.013,0.014)	**0.008** (0.004,0.004)
MC +root finding	**0.04** (0.02,0.02)	**0.025** (0.013,0.012)	**0.008** (0.004,0.004)
M(# MC samples)	10^3	3×10^3	3×10^4
MC	**0.04** (0.02,0.02)	**0.026** (0.013,0.013)	**0.008** (0.004,0.004)
M(# MC samples)	2×10^4	7×10^4	7×10^5

Table 6.8: Total relative error of ASGQ, with different tolerances, and MC to compute 4-dimensional basket call option price for different number of time steps, without Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps		
	2	4	8
ASGQ	4	13	22
MC +root finding	222	730	10541
MC	10	44	603

Table 6.9: Comparison of the computational time of MC and ASGQ, used to compute 4-dimensional basket call option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.

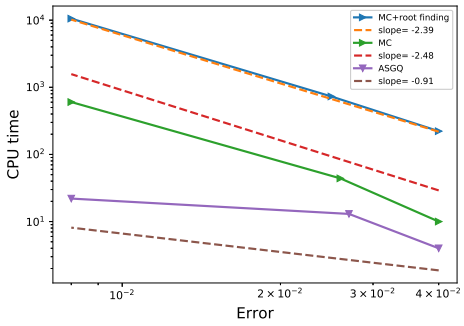


Figure 6.8: Computational work comparison for ASGQ and MC methods, for the case of 4-dimensional basket call option. This plot shows that to achieve a relative error below 1%, ASGQ outperforms significantly MC method in terms of computational time.

7 Options under the discretized the Heston model

In this section, we consider testing options under the discretized Heston model [19, 7, 22, 3] whose dynamics are given by

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^S = \mu S_t dt + \rho\sqrt{v_t} S_t dW_t^v + \sqrt{1-\rho^2}\sqrt{v_t} S_t dW_t \\ dv_t &= \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dW_t^v, \end{aligned} \tag{7.1}$$

where

- S_t is the price of the asset, v_t the instantaneous variance, given as a CIR process.
- W_t^S, W_t^v are correlated Wiener processes with correlation ρ.
- μ is the rate of return of the asset.
- θ is the mean variance.
- κ is the rate at which v_t reverts to θ.

- ξ is the volatility of the volatility, and determines the variance of v_t.

Given that the SDE for the asset path is now dependent upon the solution of the second volatility SDE in (7.1), it is necessary to simulate the volatility process first and then utilize this "volatility path" in order to simulate the asset path. In the case of the original GBM SDE it is possible to use Itô's Lemma to directly solve for S_t. However, we are unable to utilize that procedure here and must use a numerical approximation in order to obtain both paths. In the following, we try to give an overview of the methods that we can use in our context. These methods mainly differs by the way it simulates the volatility process to ensure its positiveness.

7.1 Fixed Euler scheme

In this context, one can use Forward Euler scheme to simulate the Heston model, and to avoid problems with negative values of the volatility process v_t, many fixes were introduced in the literature (see [23]). We introduce f_1, f_2, and f_3, which with different choices implies different schemes, and apply Forward Euler scheme to discretize 7.1 results in

$$\begin{aligned}
\widehat{S}_{t+\Delta t} &= \widehat{S}_t + \mu \widehat{S}_t \Delta t + \sqrt{\widehat{V}_t \Delta t} \widehat{S}_t Z_s \\
\widehat{V}_{t+\Delta t} &= f_1(\widehat{V}_t) + \kappa(\theta - f_2(\widehat{V}_t))\Delta t + \xi \sqrt{f_3(\widehat{V}_t)\Delta t} Z_V \\
\widehat{V}_{t+\Delta t} &= f_3(\widehat{V}_{t+\Delta t}),
\end{aligned} \tag{7.2}$$

where Z_s and Z_V are two correlated standard normal variables with correlation ρ, such that $Z_s = \rho Z_V + \sqrt{1-\rho^2} Z_t$.

Scheme	f_1	f_2	f_3
full truncation scheme	$\widehat{V}_t$	$\widehat{V}_t^+$	$\widehat{V}_t^+$
Partial truncation scheme	$\widehat{V}_t$	$\widehat{V}_t$	$\widehat{V}_t^+$
The reflection scheme	$\lvert\widehat{V}_t\rvert$	$\lvert\widehat{V}_t\rvert$	$\lvert\widehat{V}_t\rvert$

where $\widehat{V}_t^+ = \max(0, \widehat{V}_t)$.

[23] suggest that the Full Truncation method is the optimal option in terms of weak error convergence and robustness with respect to the parameters of the model. Therefore, we use this scheme for this approach.

7.2 Euler schemes with moment matching

We consider two moment matching Euler schemes that were suggested by Andersen and Brotherton-Ratcliffe [4] (we call it ABR scheme) and by Anderson in [3] (we choose the QE method that was reported to have the optimal results).

7.2.1 The ABR method

The ABR method is suggested in [4], where the variance v is assumed to be locally lognormal, where the parameters are determined such that the first two moments of the discretization coincide

with the theoretical moments, that is

$$\widehat{V}(t+\Delta t) = \left(e^{-\kappa\Delta t}\widehat{V}(t) + \left(1-e^{-\kappa\Delta t}\right)\theta\right) e^{-\frac{1}{2}\Gamma(t)^2\Delta t + \Gamma(t)\Delta W_v(t)}$$

$$\Gamma(t)^2 = \Delta t^{-1}\log\left(1 + \frac{\frac{1}{2}\xi^2\kappa^{-1}\widehat{V}(t)(1-e^{-2\kappa\Delta t})}{\left(e^{-\kappa\Delta t}\widehat{V}(t) + (1-e^{-\kappa\Delta t})\theta\right)^2}\right) \tag{7.3}$$

As reported in [23], the scheme, being very easy to implement, is much more effective than many of the Euler fixes, however, it was reported that it has a bad weak error behavior that is not robust with parameters values, which is something that we need to check in our setting!

7.2.2 The QE method

On the other hand, and using the same idea of moment matching, Anderson in[3], suggested more discretisations for the Heston model similar to (7.3) and taking the shape of the Heston density function into account. These schemes have negligible bias, at the cost of a more complex implementation than the one introduced by [4]. As reported by [3], the QE method is the optimal scheme to use and it follows the following algorithm.

1. Select some arbitrary level $\Psi_c \in [1,2]$.
2. Given $\widehat{V}(t)$, compute m and s^2 given by

$$m = \mathrm{E}\left[V(t+\Delta t) \mid V(t) = \widehat{V}(t)\right] = \theta + (\widehat{v}(t)-\theta)e^{-\kappa\Delta t}$$
$$s^2 = \mathrm{Var}\left[V(t+\Delta t) \mid V(t) = \widehat{V}(t)\right] = \frac{\widehat{V}(t)\xi^2 e^{-\kappa\Delta t}}{\kappa}(1-e^{-\kappa\Delta t}) + \frac{\theta\xi^2}{2\kappa}(1-e^{-\kappa\Delta t})^2.$$

3. Compute $\Psi = s^2/m^2$.
4. Draw a uniform random number U_V.
5. If $\Psi \leq \Psi_c$
   i) Compute a and b given by

$$b^2 = 2\Psi^{-1} - 1 + \sqrt{2\Psi^{-1}}\sqrt{2\Psi^{-1}-1}$$
$$a = \frac{m}{1+b^2}$$

   ii) Compute $Z_V = \Phi^{-1}(U_V)$, where Φ^{-1} is the inverse cumulative Gaussian distribution function.
   iii) Set $\widehat{V}(t+\Delta t) = a(b+Z_V^2)^2$.
6. Otherwise, if $\Psi > \Psi_c$

i) Compute p and β according to

$$p = \frac{\Psi - 1}{\Psi - 1}$$
$$\beta = \frac{1-p}{m}$$

ii) Set $\widehat{V}(t+\Delta t) = h^{-1}(U_V; p, \beta)$, where h^{-1} is given by

$$h^{-1}(u; p, \beta) = \begin{cases} 0, & 0 \le u \le p \\ \beta^{-1} \log\left(\frac{1-p}{1-u}\right), & p < u \le 1 \end{cases}$$

Compared to ABR scheme, the QE scheme may have a better weak error behavior with the disadvantage of being more costly. Furthermore, the QE algorithm uses two different distributions to model the volatility depending on the initial value of the volatility!

7.3 Kahl-Jackel Scheme

[22] suggested discretizing the volatility process using an implicit Milstein scheme, coupled with their IJK discretization for the stock process. Specifically, they propose the scheme

$$\begin{aligned} \log(\widehat{X}(t+\Delta t)) = \log(\widehat{X}(t)) &- \frac{\Delta t}{4}\left(\widehat{V}(t+\Delta t) + \widehat{V}(t)\right) + \rho\sqrt{\widehat{V}(t)} Z_V \sqrt{\Delta t} \\ &+ \frac{1}{2}\left(\sqrt{\widehat{V}(t+\Delta t)} + \sqrt{\widehat{V}(t)}\right)\left(Z_x\sqrt{\Delta t} - \rho Z_V \sqrt{\Delta t}\right) + \frac{\xi}{2}\rho\Delta t(Z_V^2 - 1) \end{aligned}$$

$$\widehat{V}(t+\Delta t) = \frac{\widehat{V}(t) + \theta\kappa\Delta t + \xi\sqrt{\widehat{V}(t)} Z_V \sqrt{\Delta t} + \frac{\xi^2}{4}\Delta t(Z_V^2 - 1)}{1 + \kappa\Delta t} \tag{7.4}$$

It is easy to check that this discretization scheme will result in positive paths for the V process if $4\kappa\theta > \xi^2$ (which is rarely satisfied in practice). Unfortunately [22] does not provide a solution for this problem, but it seems reasonable to use a truncation scheme similar to those presented in Section 7.1. Therefore, we do not test this scheme in our numerical experiments since we think that it will have similar issues that we observed for the full truncation scheme presented in Section 7.1. Furthermore, it was shown that the convergence of the weak rate of the IJK-IMM scheme somewhat erratic, especially for cases $4\kappa\theta \le \xi^2$.

7.4 Discretization of Heston model with the volatility process Simulated using the sum of Ornstein-Uhlenbeck (Bessel) processes

Since we are using ASGQ which is very sensitive to the smoothness of the integrand. We found numerically (see section 7.5) that using a non smooth transformation to ensure the positiveness of the volatility process deteriorates the performance of the ASGQ. An alternative way to guarantee the positiveness of the volatility process is to simulate it as the sum of Ornstein-Uhlenbeck (Bessel) processes.

In fact, it is well known that any Ornstein-Uhlenbeck (OU) process is normally distributed. Thus, the sum of n squared OU processes is chi-squared distributed with n degrees of freedom, where $n \in \mathbb{N}$. Let us define $\mathbf{X}$ to be a n-dimensional vector valued OU process with

$$\mathrm{d}X_t^i = \alpha X_t^i \mathrm{d}t + \beta \mathrm{d}W_t^i, \tag{7.5}$$

where $\mathbf{W}$ is a n-dimensional vector of independent Brownian motions.

We define also the process Y_t as

$$Y_t = \sum_{i=1}^{n} \left(X_t^i\right)^2. \tag{7.6}$$

Then, using the fact that

$$\begin{aligned} \mathrm{d}\left(X_t^i\right)^2 &= 2X_t^i \mathrm{d}X_t^i + 2\mathrm{d}\langle X^i \rangle_t \\ &= \left(2\alpha \left(X_t^i\right)^2 + \beta^2\right)\mathrm{d}t + 2\beta X_t^i \mathrm{d}W_t^i, \end{aligned} \tag{7.7}$$

we can write

$$\begin{aligned} \mathrm{d}Y_t &= \mathrm{d}\left(\sum_{i=1}^{n} \left(X_t^i\right)^2\right) \\ &= \sum_{i=1}^{n} \mathrm{d}\left(X_t^i\right)^2 \\ &= \left(2\alpha Y_t + n\beta^2\right)\mathrm{d}t + 2\beta \sum_{i=1}^{n} X_t^i \mathrm{d}W_t^i, \end{aligned} \tag{7.8}$$

where the second step follows from the independence of the Brownian motions. Furthermore, we note that the process

$$Z_t = \int_0^t \sum_{i=1}^{n} X_u^i \mathrm{d}W_u^i \tag{7.9}$$

is a martingale with quadratic variation

$$\begin{aligned} \langle Z \rangle_t &= \int_0^t \sum_{i=1}^{n} \left(X_u^i\right)^2 \mathrm{d}u \\ &= \int_0^t Y_u \mathrm{d}u. \end{aligned} \tag{7.10}$$

Consequently, by Lévy's characterization theorem, we can show that the process

$$\widetilde{W}_t = \int_0^t \frac{1}{\sqrt{Y_u}} \sum_{i=1}^{n} X_u^i \mathrm{d}W_u^i \tag{7.11}$$

is a Brownian motion.

Finally, we have

$$\begin{aligned} \mathrm{d}Y_t &= \left(2\alpha Y_t + n\beta^2\right)\mathrm{d}t + 2\beta\sqrt{Y_t}\mathrm{d}\tilde{W}_t \\ &= \kappa\left(\theta - Y_t\right)\mathrm{d}t + \xi\sqrt{Y_t}\mathrm{d}W_t, \end{aligned} \tag{7.12}$$

where $\kappa = -2\alpha$, $\theta = -n\beta^2/2\alpha$ and $\xi = 2\beta$.

Equations (7.5), (7.8), and (7.12) show in order to simulate the process Y_t given by (7.12), we can simulate the OU process $\mathbf{X}$, with dynamics given by (7.5) such that its parameters (α, β) are expressed in terms of those of the process Y_t, that is

$$\alpha = -\frac{\kappa}{2}, \quad \beta = \frac{\xi}{2}, \quad n = \frac{-2\theta\alpha}{\beta^2} = \frac{4\theta\kappa}{\xi^2}.$$

Therefore, we conclude that we can simulate the volatility of the Heston model using a sum of OU processes.

Remark 7.1. The previous derivation can can be generalized to cases where n^* is not an integer by considering a time-change of a squared Bessel process (see Chapter 6 in [21] for details). A second way that can be used for generalizing the scheme for any non integer, n^*, by writing $n^* = n + \alpha$, $\alpha \in (0, 1)$ and then we can compute, for any observable, g, $\mathrm{E}\left[g(X_{n^*})\right]$ as

$$\mathrm{E}\left[g(X_{n^*})\right] \approx (1-\alpha)\mathrm{E}\left[g(X_n)\right] + \alpha\mathrm{E}\left[g(X_{n+1})\right]$$

7.5 On the choice of the simulation scheme of the Heston model

In this section, we try to determine the optimal scheme for simulating the Heston model defined in 7.1. In the literature [3, 23, 2], more focus was on designing schemes that i) ensures the positiveness of the volatility process and ii) has a good weak error behavior. In our setting, an optimal scheme is defined through two properties: i) the behavior of mixed rates which is an important feature for an optimal performance of ASGQ, and ii) the behavior of the weak error in order to apply Richardson extrapolation when it is needed. In the following section we do the comparison using these two criteria.

7.5.1 Comparison in terms of mixed differences rates

In this section, we try to compare the four approaches of simulating Heston dynamics, which are : i) the full truncation scheme discussed in Section 7.1, ii) the ABR scheme discussed in Section 7.2.1, ii) the QE scheme discussed in Section 7.2.2, and iv) the smooth scheme bases on sum of square of OU processes discussed in Section 7.4. The comparison is done in terms of mixed differences convergence (defined in (7.13)) rates which, as pointed out in [18], are a key determinant for the optimal performance of ASGQ.

$$\Delta E_{\boldsymbol{\beta}} = \left| Q_N^{\mathcal{I}\cup\{\boldsymbol{\beta}\}} - Q_N^{\mathcal{I}} \right|. \tag{7.13}$$

For comparison, we use basically three sets of examples given in table 7.1. The first two sets of parameters we used we test two different cases where the first case corresponds to $n = 1$ and the second one corresponds to $n = 2$. On the other hand, for the last remaining set we used similar parameters tested in [23], where the set 3 stems from Broadie and Kaya [7], and is the hardest of all examples. We also point out that only the second set of parameters satisfies the Feller condition, that is $4\kappa\theta > \xi^2$.

Parameters	Reference solution
Set 1: $S_0 = K = 100$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = \frac{\xi^2}{4\kappa} = 0.0025$, $(n = 1)$.	6.332542
Set 2: $S_0 = K = 100$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = \frac{\xi^2}{4\kappa} = 0.005$, $(n = 2)$.	6.445535
Set 3: $S_0 = K = 100$, $v_0 = 0.09$, $\mu = 0$, $\rho = -0.3$, $\kappa = 2.7778$, $\xi = 1$, $\theta = \frac{\xi^2}{4\kappa} = 0.09$, $(n = 1)$.	10.86117

Table 7.1: Reference solution using Premia with cf_call_heston method as explained in [19], for different parameter constellations. By n we refer to the number of OU processes for simulating the volatility process in approach given by Section 7.4

In our numerical experiments, we only observed differences of mixed differences rates related to volatility coordinates, which is expected since we use basically schemes that only differ by the way it simulates the volatility process. Figures 7.2, 7.3, and 7.4 show a comparison of first differences rates related to volatility coordinates for the different schemes and for the different examples. From these figures, we have the following conclusions

1. In all examples, the full truncation scheme is the worst scheme to use in terms of first differences rates.

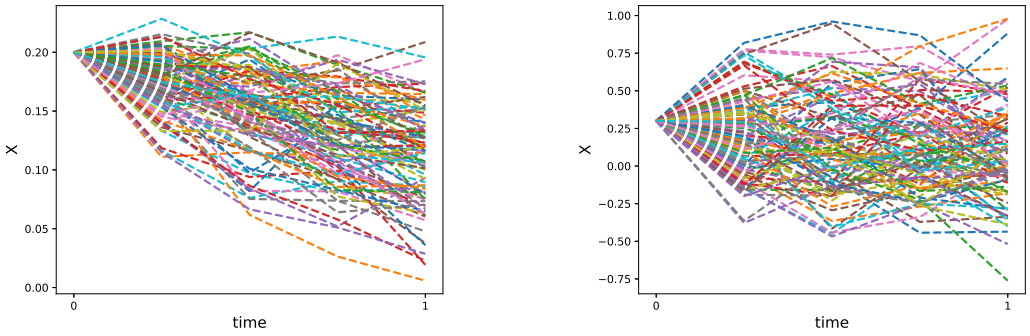
2. The scheme based on the sum of square of OU processes have a very good performance for the first two sets. However, we observed a very bad performance for the example of Set 3 parameters! This can be explained by the fact that when simulating the OU process (given by (7.5)) using Forward Euler scheme, we have the discretized Euler scheme with N time steps given by

$$X_N = (1 + \alpha\Delta t)^N X_0 + \beta \sum_{m=1}^{N} (1 + \alpha\Delta t)^{N-m} \Delta W_n$$

implying

$$Var[X_N] = \beta^2 \sum_{m=1}^{N} (1 + \alpha\Delta t)^{2(N-m)} \Delta t. \tag{7.14}$$

For instance, for set 1 and 2, we have $\beta = 0.05$ and $\alpha = -0.5$ implying $Var[X_4] = 1.8 \times 10^{-3}$, and for the case of set 3, we have $\beta = 0.5$ and $\alpha \approx -1.4$ implying $Var[X_4] \approx 10^{-1}$, meaning that the variability of the stochastic term for small values of number of times steps is very big resulting in very noisy paths around the mean, and therefore a very bad performance for the mixed differences as was observed. In Figure 7.1, we show an illustration for this by comparing paths of the process X given by (7.5) for both sets of parameters 1 and 3 in Table 7.1.



(a) (b)

Figure 7.1: 100 paths of process X given by (7.5) simulated using forward Euler. (a) Case of set 1 parameters in Table 7.1, (b) Case of set 3 parameters in Table 7.1.

3. Both schemes based on moment matching (ABR and QE schemes) have the best performance in terms of mixed rates convergence (almost similar performance with the scheme based on the sum of square of OU processes, for the first two examples in Table 7.1) and also they show robust behavior with respect to all examples with almost similar performance

Mixed differences for the case of Set 1 parameters

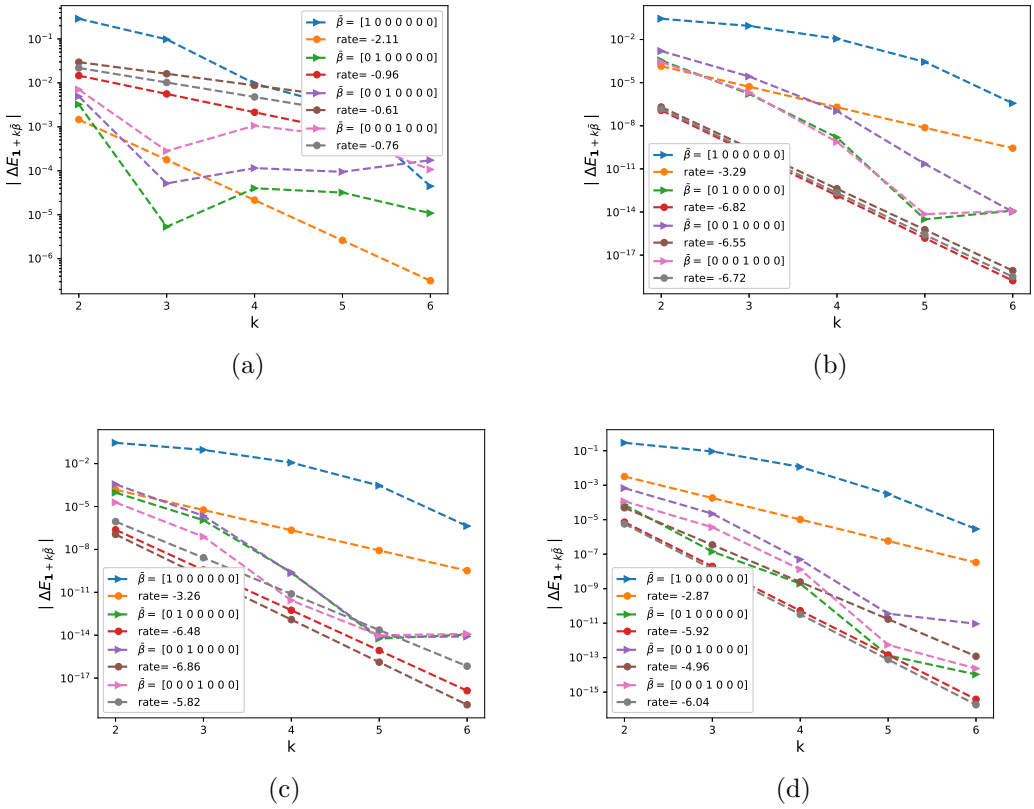


Figure 7.2: The rate of error convergence of first order differences $|\Delta \mathrm{E}_{\boldsymbol{\beta}}|$, defined by (7.13), ($\boldsymbol{\beta} = \mathbf{1} + k\overline{\boldsymbol{\beta}}$) for the example of single call option under Heston model, with parameters given by Set 1 in Table 7.1, using $N = 4$ time steps. In this case, we just show the first 4 dimensions which are used for the volatility noise (mainly dW_v in (7.1)). (a) using full truncation as in Section 7.1, (b) using the ABR scheme as in Section 7.2.1, (c) using the QE scheme as in Section 7.2.2, (d) using the smooth transformation as in Section 7.4.

Mixed differences for the case of Set 2 parameters

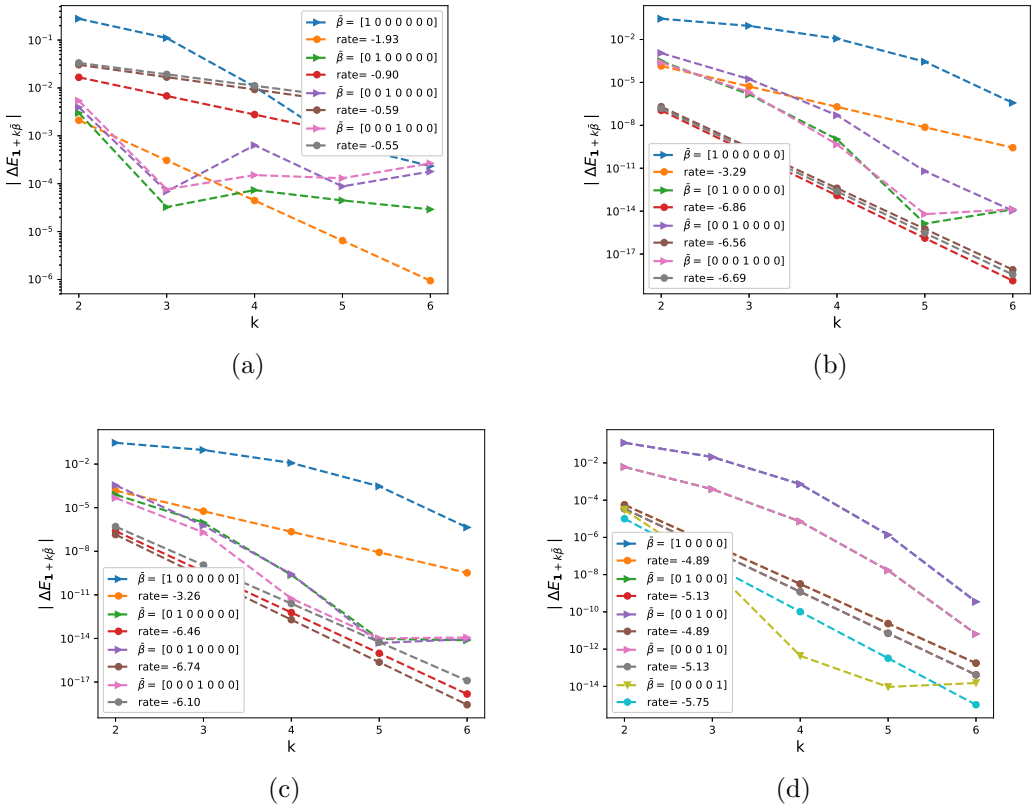


Figure 7.3: The rate of error convergence of first order differences $|\Delta \mathrm{E}_{\boldsymbol{\beta}}|$, defined by (7.13), ($\boldsymbol{\beta} = \mathbf{1} + k\overline{\boldsymbol{\beta}}$) for the example of single call option under Heston model, with parameters given by Set 2 in Table 7.1, using $N = 4$ time steps. In this case, we just show the first 4 dimensions which are used for the volatility noise (mainly dW_v in (7.1)). (a) using full truncation as in Section 7.1, (b) using the ABR scheme as in Section 7.2.1, (c) using the QE scheme as in Section 7.2.2, (d) using the smooth transformation as in Section 7.4, here $N = 2$.

Mixed differences for the case of Set 3 parameters

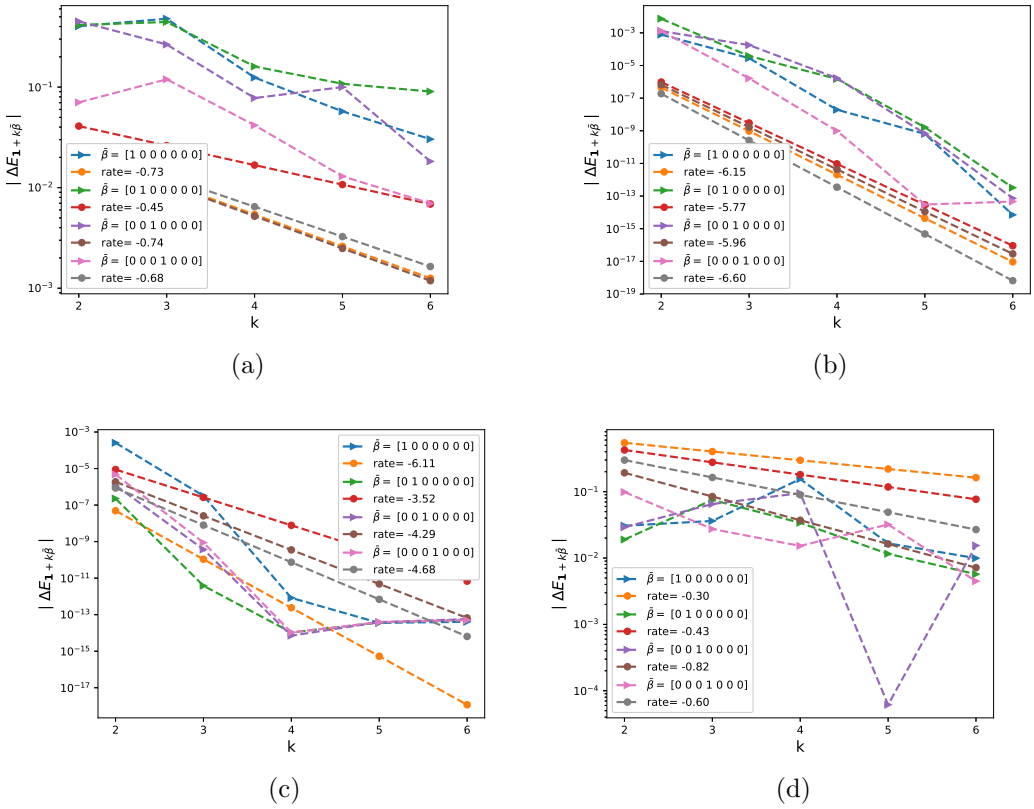


Figure 7.4: The rate of error convergence of first order differences $|\Delta \mathrm{E}_{\boldsymbol{\beta}}|$, defined by (7.13), ($\boldsymbol{\beta} = \mathbf{1} + k\overline{\boldsymbol{\beta}}$) for the example of single call option under Heston model, with parameters given by Set 3 in Table 7.1, using $N = 4$ time steps. In this case, we just show the first 4 dimensions which are used for the volatility noise (mainly dW_v in (7.1)). (a) using full truncation as in Section 7.1, (b) using the ABR scheme as in Section 7.2.1, (c) using the QE scheme as in Section 7.2.2, (d) using the smooth transformation as in Section 7.4.

7.5.2 Comparison in terms of the weak error behavior

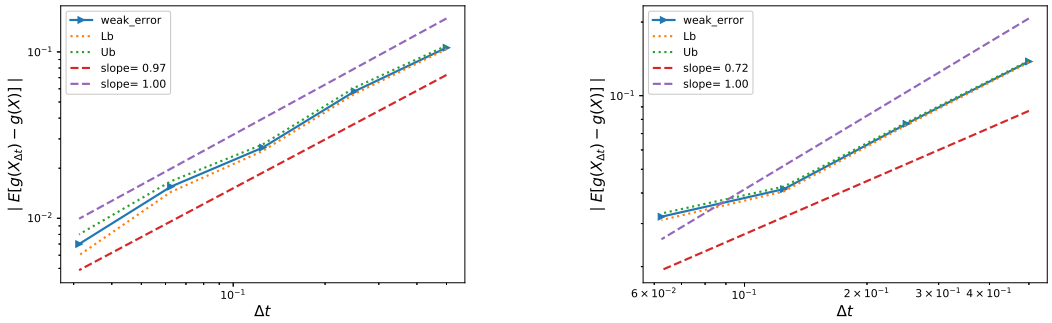
In this section, we try to compare the two approaches, of simulating Heston dynamics, which are: i) the ABR scheme discussed in Section 7.2.1, and ii) the smooth scheme based on sum of square of OU processes discussed in Section 7.4. The comparison is done in terms of the behavior of the weak convergence. In fact, we opt for schemes that has weak error rate of of order 1 in the pre-asymptotic regime so that we can apply Richardson extrapolation in our proposed methods. We point out that we obtained similar behavior for both ABR and QE scheme discussed in Section 7.2.2, therefore, we chose to illustrate the results obtained by the ABR scheme.

For comparison, we considered the three examples of parameters given in table 7.1. Figures 7.5, 7.6, and 7.7 show a comparison of the weak error rates for the different schemes and for the different three examples. We mention that we did not include the weak rate of the smooth scheme based on sum of square of OU processes, for the third example, because, as explained in the previous Section, this set produced bad behavior for the mixed convergence rates for this method. From

these figures, we have the following conclusions

- For the first two examples, the smooth scheme based on sum of square of OU processes has a better weak convergence rate that is close to 1, compared to the ABR scheme having a weak error rate of order 0.7.
- The ABR scheme have a very poor behavior in terms of the weak error for the third example, being almost constant. We emphasize that similar issues was observed for QE scheme, defined in Section 7.2.2. These observation join similar numerical observations in [23], which show that ABR scheme have a very bad behavior for a certain parameters for the Heston model.

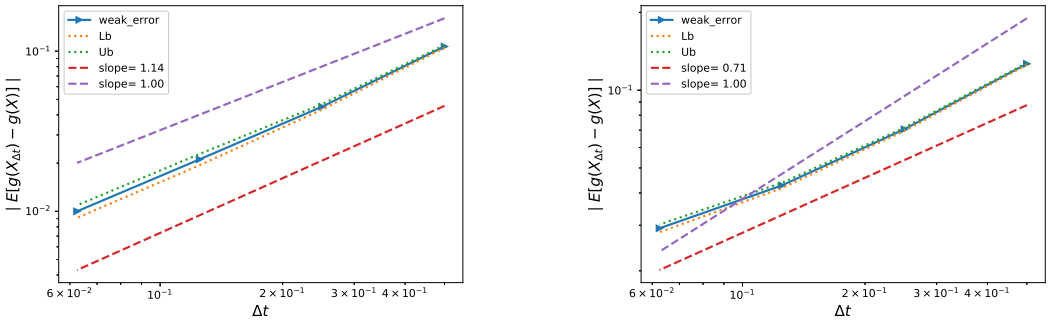
Weak error behavior for the case of Set 1 parameters



(a) (b)

Figure 7.5: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, for the European call option under the discretized Heston model, for Set 1 parameters in Table 7.1. The upper and lower bounds are 95% confidence intervals. (a) using the smooth transformation as in Section 7.4, (b) using the ABR scheme as in Section 7.2.1.

Weak error behavior for the case of Set 2 parameters



(a) (b)

Figure 7.6: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, for the European call option under the discretized Heston model, for Set 2 parameters in Table 7.1. The upper and lower bounds are 95% confidence intervals. (a) using the smooth transformation as in Section 7.4, (b) using the ABR scheme as in Section 7.2.1.

Weak error behavior for the case of Set 3 parameters

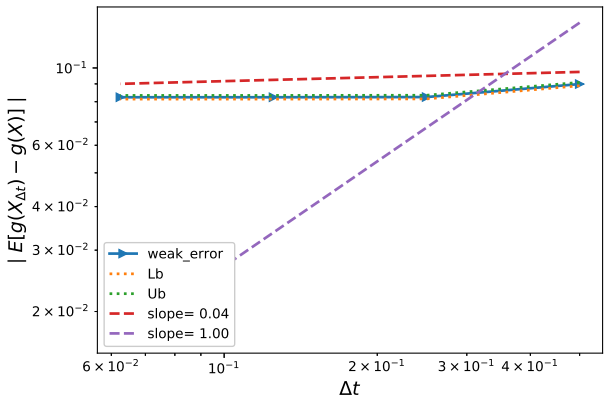


Figure 7.7: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1,using the ABR scheme as in Section 7.2.1, for the European call option under the discretized Heston model, for Set 3 parameters in Table 7.1. The upper and lower bounds are 95% confidence intervals.

7.5.3 Conclusion about the chosen simulation scheme used for ASGQ

Given the results obtained in Sections 7.5.1 and Section 7.5.2, we observed that Set 3 is presenting some issues for all tested schemes, for instance, we observed poor behavior of mixed differences for both the Full truncation scheme and the smooth volatility scheme. On the other hand, applying moment matching schemes for this set produced poor results for the weak error. Therefore, for our aims we excluded this set from our interest, since our main concern in this work is to illustrate the

potential of the numerical smoothing when using ASGQ, and not the investigation of the optimal schemes for the simulation purposes, which is left for the future research. Consequently, we opt for the use of the smooth volatility scheme, defined in Section 7.4, which gives the best results in terms of mixed differences and weak convergence for examples 1 and 2 in Table 7.1.

7.6 European call option under the discretized Heston model

We consider now the European call option under the discretized Heston model with parameters given by Set 1 in Table 7.1. In this case, we use the smooth scheme described in Section 7.4 and with $n = 1$ (that is we just use one OU process to simulate the volatility). Figure 7.5a shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC (we used the Full truncation scheme to simulate the volatility) and ASGQ (we used the smooth approach in Section 7.4 to simulate the volatility) in Tables 7.2 and 7.3 for the case without Richardson extrapolation, and Tables 7.4 and 7.5 for the case of level 1 Richardson extrapolation. We also compare the performance of the two methods for different configurations in Figure 7.8. Our numerical experiments show that Richardson extrapolation improved the performance of both ASGQ and MC, and that to achieve a relative error around 0.3%, ASGQ coupled with level 1 of Richardson extrapolation is the optimal method, and requires approximately 32% of the work of MC coupled with level 1 of Richardson extrapolation.

Method	Steps				
	2	4	8	16	32
ASGQ	**0.115** (0.106,0.009)	**0.0740** (0.058,0.016)	**0.0479** (0.0266,0.0213)	**0.0282** (0.0155,0.0127)	**0.015** (0.007,0.008)
MC	**0.212** (0.106,0.106)	**0.115** (0.058,0.057)	**0.0528** (0.0266,0.0262)	**0.0285** (0.0155,0.013)	**0.0142** (0.007,0.0072)
M(# MC samples)	10^3	3×10^3	2×10^4	6×10^4	2×10^5

Table 7.2: Total relative error of ASGQ, with different tolerances, and MC to compute European call option price under discretized Heston model for different number of time steps, without Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps				
	2	4	8	16	32
ASGQ	14	18	24.5	76.5	351
MC	0.5	1.5	10	50	730

Table 7.3: Comparison of the computational time of MC and ASGQ, used to compute European call option price under discretized Heston model for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.

Method	Steps		
	$1-2$	$2-4$	$4-8$
ASGQ	**0.038** (0.037,0.01)	**0.0142** (0.0077,0.0065)	**0.0032** (0.0015,0.0017)
MC	**0.131** (0.067,0.064)	**0.0123** (0.0061,0.0062)	**0.0031** (0.0016,0.0015)
M(# MC samples)	2×10^3	3×10^5	4×10^6

Table 7.4: Total relative error of ASGQ and MC to compute European call option price under discretized Heston model for different number of time steps, with level 1 of Richardson extrapolation. The values between parentheses correspond to the different errors contributing to the total relative error: for ASGQ we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples,M, is chosen to satisfy (6.1).

Method	Steps		
	$1-2$	$2-4$	$4-8$
ASGQ	4.5	48.5	864
MC	1	145	2764

Table 7.5: Comparison of the computational time of MC and ASGQ, used to compute European call option price under discretized Heston model for different number of time steps, with level 1 of Richardson extrapolation. The average computational time of MC is computed over 10 runs.

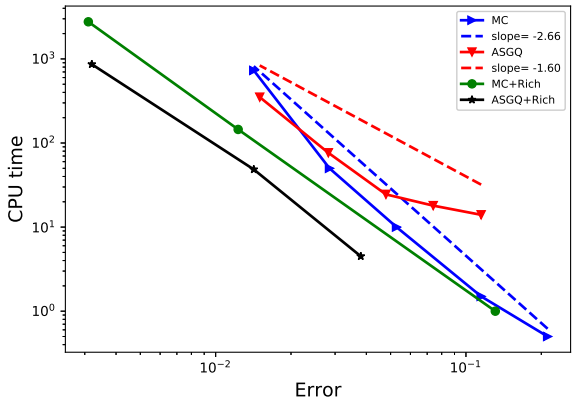


Figure 7.8: Computational work comparison for ASGQ and MC methods, for the case of European call option price under discretized Heston model. This plot shows that to achieve a relative error around 1.4%, using level 1 of Richardson extrapolation is the optimal configuration for both methods, and that ASGQ coupled with level 1 of Richardson extrapolation outperforms MC coupled with level 1 of Richardson extrapolation method in terms of computational time.

Remark 7.2. Although we just illustrate the numerical results for Set 1 parameters in Table 7.1,

we should mention that almost similar results were obtained for the case of the second example in Table 7.1.

8 Numerical smoothing with MLMC

In this section, we want to motivate the advantage of the numerical smoothing idea in the context of MLMC method. For this aim we consider two examples: i) the first one is for computing the price of a digital option (see Section 8.2) , and ii) the second example is for approximating a density function (see Section 8.3). The idea and results of the first example can be generalized to i) any kind of option having low regularity in the payoff function, or ii) for computing distribution functions, since it involves the indicator function. On the other hand, the second example have two important applications: i) computing density functions which involves the use of Dirac delta functions and which is hard to approximate its expectation using MLMC due to the infinite variance, and ii) computing Greeks for an option with a non smooth payoff function.

8.1 Our contributions to the literature

Usually, for such tasks, standard MLMC will fail or not have the optimal performance, due to the singularity present in the delta or the indicator functions, implying either high variance or kurtosis for the MLMC estimator. In the literature, few works tried to address this issue. For instance,

1. Avikainen in [5], and Giles, Higham, and Mao in [11] used MLMC for such a task without smoothing.

2. On the other hand, a second approach was suggested in [13, 10], that used implicit smoothing based on the use of conditional expectations. There are two potential issues with this second approach: i) In general cases, one may have dynamics where it is not easy to derive an analytic expression for the conditional expectation and ii) This approach used a higher order scheme, that is the Milstein scheme, to improve the strong order of convergence, and consequently the complexity of the MLMC estimator. Such a scheme becomes very computationally expensive for higher dimensional dynamics.

3. In [12], the authors suggested a different approach based on parametric smoothing. In fact, they carefully constructed a regularized version of the QoI, based on a regularization parameter that depends on the weak and strong convergence rates and also the tolerance requirement. This approach, despite offering better performance for the MLMC estimator and a better setting for theoretical analysis, it has the practical disadvantage consisting in the difficulty of generalizing it to cases where there is no prior knowledge of the the convergence rates (that is they need to be estimated numerically), and also for each error tolerance, a new regularization parameter needs to be computed.

In this work, we address a similar problem and propose an alternative approach that is based on numerical smoothing. Our approach compared to previous mentioned works, has the following advantages

- It can be easily applied to cases where one can not apply analytic smoothing.

- We obtain similar rates of strong convergence and MLMC complexity as in [13, 10], without the need to use higher order schemes such as Milstein scheme.
- Our approach is parameter free compared to that of [12]. Therefore, in practice it is much easier to apply for any dynamics and QoI.

8.2 MLMC for digital options

In this section, we motivate the idea of using the numerical smoothing idea to compute option prices for non smooth payoff function. For illustration, we consider the price of the digital option, that is we want to approximate

$$\mathrm{E}\left[g(X)\right] = \mathrm{E}\left[\mathbf{1}_{X>K}\right], \tag{8.1}$$

where K is the strike.

Then, using similar idea introduced in Section 6.1, we can show that

$$\begin{aligned} \mathrm{E}\left[g(X)\right] &= \mathrm{E}\left[\mathbf{1}_{X>K}\right] \\ &= \mathrm{E}\left[1 - \Phi(y^*(K))\right], \end{aligned} \tag{8.2}$$

where $y^*(K)$ is the kink location obtained by solving numerically $X(T; y^*(x), \mathbf{z}_{-1})$, $\mathbf{z}$ is the Gaussian random vector used for Brownian bridge construction, and Φ is the cumulative Gaussian distribution function.

As an illustration, we choose the digital option of GBM asset with parameters: $S_0 = K = 100$, $T = 1$, $r = 0$, and $\sigma = 0.2$, and we compare MLMC results with the original payoff function (without numerical smoothing), given by Figure 8.1, and MLMC results after applying our numerical smoothing idea, given by Figure 8.2. From these two Figures we have the following conclusions:

1. The numerical smoothing has improved the rate of strong convergence from 1/2 (without smoothing) to 1 after doing the numerical smoothing (compare both top left plots in Figures 8.1 and Figure 8.2.
2. The numerical smoothing has also dropped significantly the kurtosis by a factor 100, this is can be seen clearly by comparing the middle right plots in both Figures 8.1 and Figure 8.2. We stress that this is an important improvement, since one needs $\mathcal{O}(\kappa)$ samples to obtain a reasonable estimate for the variance. For instance, the standard deviation of the sample variance for a random variable X with zero mean is approximately $\sqrt{(\kappa-1)/N}\mathrm{E}\left[X^2\right]$ where the kurtosis κ is defined as $\kappa = \frac{\mathrm{E}[X^4]}{(\mathrm{E}[X^2])^2}$.
3. Finally, improving the strong error rate after using the numerical smoothing, resulted in an improvement of the complexity rate going from $TOL^{-2.5}$ for the case without smoothing to $TOL^{-2}2$, where TOL is a prescribed tolerance, for the case where MLMC is coupled with numerical smoothing (compare the bottom right plots in Figures 8.1 and 8.2).

Although, for numerical illustration purpose, we just considered GBM dynamics, we emphasize that our approach can be extended in a straightforward manner to any kind of dynamics, since it is based on numerical smoothing based on solving a root finding problem. Furthermore, our approach

can be easily extended to any low regular observable, g. For instance, this idea can be applied for approximating distribution functions involving the indicator function as the observable g. Finally, In all these possible extensions, we expect similar gains to be observed compared to the case of using MLMC without smoothing.

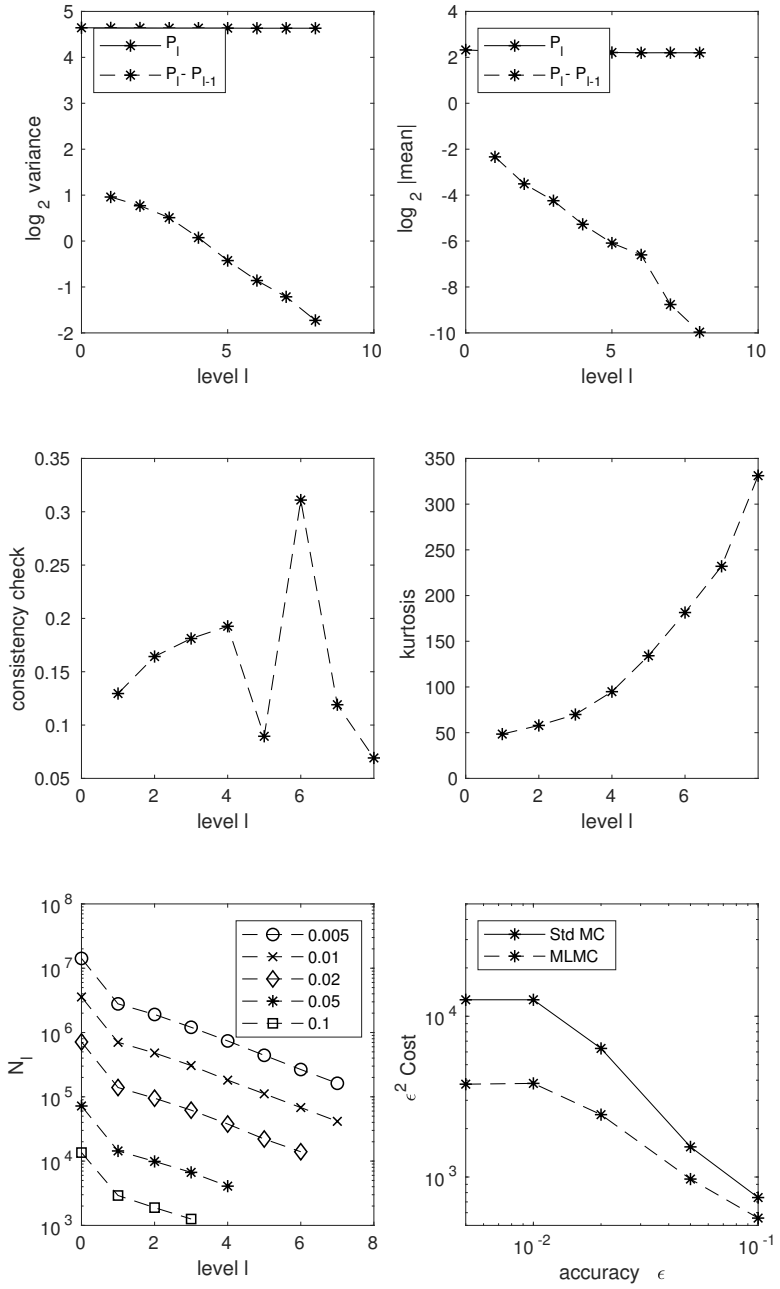


Figure 8.1: Numerical results for a digital call option using the MLMC method coupled with Euler-Maruyama discretisation of the GBM SDE, and without smoothing of the payoff.

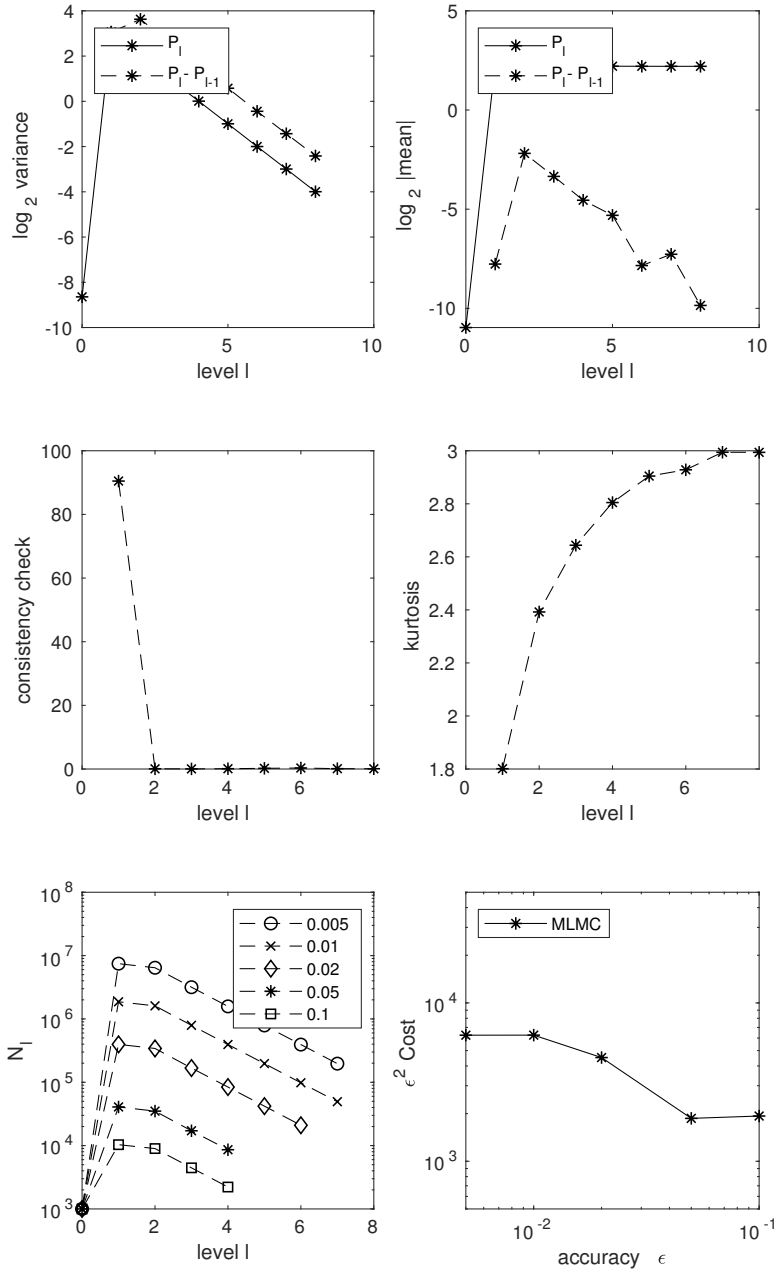


Figure 8.2: Numerical results for a digital call option using the MLMC method coupled with Euler-Maruyama discretisation of the GBM SDE, after applying the numerical smoothing to the payoff. We mention that observing a decaying variance of P_ℓ here is expected since we used Brownian bridge for path construction, and the QoI depends only on the terminal value of the Brownian bridge which has a variance scaled of order Δt.

8.3 MLMC for approximating densities and Greeks

In this section, we motivate the idea of coupling the numerical smoothing with MLMC method to compute density functions and Greeks for non smooth payoff functions. We remind that MLMC without any smoothing will fail due to the infinite variance caused by the singularity of the delta function. The aim in this case is to approximate the density, $\rho_X(u)$, for a given stochastic process X, at point u, and which is given by

$$\rho_X(u) = \mathrm{E}\left[\delta(X-u)\right], \tag{8.3}$$

where δ is the Dirac delta function.

We can show that

$$\begin{aligned}\rho_X(u) &= \mathrm{E}\left[\delta(X-y)\right] \\ &= \exp\left(-(y^*(u))^2/2\right)\frac{dy^*}{dx}(u),\end{aligned} \tag{8.4}$$

where $y^*(K)$ is the kink location obtained by solving numerically $X(T; y^*(x), \mathbf{z}_{-1})$, and $\mathbf{z}$ is the Gaussian random vector used for Brownian bridge construction.

As an illustration, we choose to compute the density ρ_X such that X is a GBM with parameters: $S_0 = K = 1$, $T = 1$, $r = 0$, and $\sigma = 0.2$. In this case, as a reference solution, we know that $X(T)$ is lognormally distributed with parameters $r-\sigma^2$ and σ^2. In Figure 8.3, we show the numerical results for computing the ρ_X at $u = 1$, using MLMC coupled with the numerical smoothing idea. From Figure 8.3, we can check that we obtain a strong convergence rate of order 3/2(see top left plot in Figure 8.3), which results in a complexity of the MLMC estimator to be of order $TOL^{-2}\left|\log TOL\right|^2$, where TOL is a prescribed tolerance.

Although, for numerical illustration purpose, we just considered GBM dynamics, we emphasize that our approach can be extended in a straightforward manner to any kind of dynamics, since it is based on numerical smoothing based on solving a root finding problem. Furthermore, our approach can be easily extended to computing Greeks of a digital options, involving the delta functions.

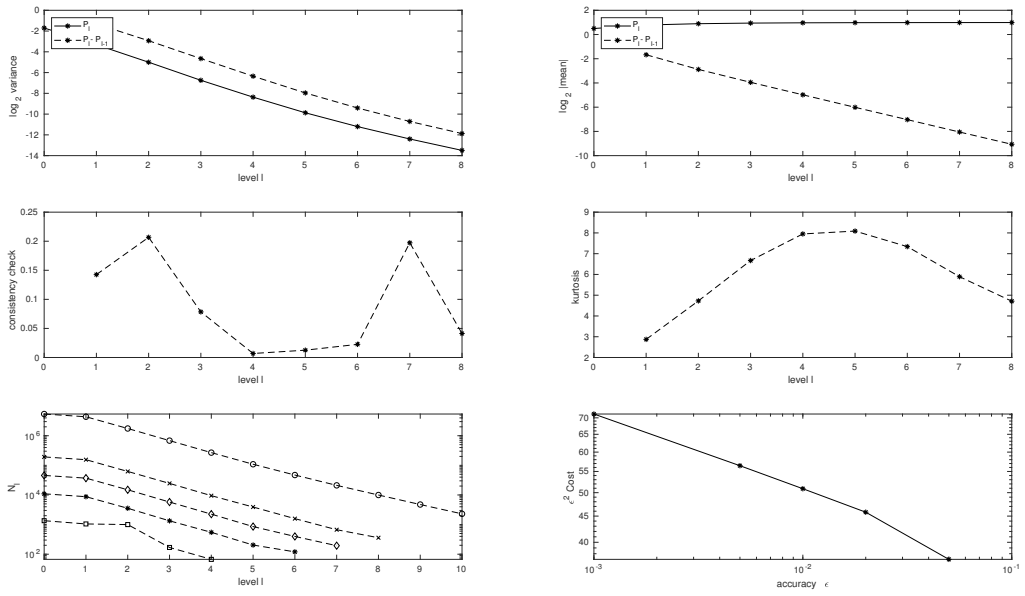


Figure 8.3: Numerical results for density estimation using the MLMC method coupled with Euler-Maruyama discretisation of the GBM SDE, after applying the numerical smoothing. We mention that observing a decaying variance of P_ℓ here is expected since we used Brownian bridge for path construction, and the QoI depends only on the terminal value of the Brownian bridge which has a variance scaled of order Δt.

References Cited

[1] Peter A Acworth, Mark Broadie, and Paul Glasserman. A comparison of some Monte Carlo and quasi Monte Carlo techniques for option pricing. In *Monte Carlo and Quasi-Monte Carlo Methods 1996*, pages 1–18. Springer, 1998.

[2] Aurélien Alfonsi. High order discretization schemes for the cir process: application to affine term structure and heston models. *Mathematics of Computation*, 79(269):209–237, 2010.

[3] Leif BG Andersen. Efficient simulation of the heston stochastic volatility model. *Available at SSRN 946405*, 2007.

[4] Leif BG Andersen and Rupert Brotherton-Ratcliffe. Extended libor market models with stochastic volatility. *Journal of Computational Finance*, 9(1), 2005.

[5] Rainer Avikainen. On irregular functionals of sdes and the euler scheme. *Finance and Stochastics*, 13(3):381–401, 2009.

[6] Christian Bayer, Markus Siebenmorgen, and Rául Tempone. Smoothing the payoff for efficient computation of basket option pricing. *Quantitative Finance*, 18(3):491–505, 2018.

[7] Mark Broadie and Özgür Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations research*, 54(2):217–231, 2006.

[8] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[9] Russel E Caflisch, William J Morokoff, and Art B Owen. *Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension*. 1997.

[10] Michael Giles, Kristian Debrabant, and Andreas Rößler. Numerical analysis of multilevel monte carlo path simulation using the milstein discretisation. *arXiv preprint arXiv:1302.4676*, 2013.

[11] Michael B Giles, Desmond J Higham, and Xuerong Mao. Analysing multi-level monte carlo for options with non-globally lipschitz payoff. *Finance and Stochastics*, 13(3):403–413, 2009.

[12] Michael B Giles, Tigran Nagapetyan, and Klaus Ritter. Multilevel monte carlo approximation of distribution functions and densities. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):267–295, 2015.

[13] Mike Giles. Improved multilevel monte carlo convergence using the milstein scheme. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 343–358. Springer, 2008.

[14] Paul Glasserman. *Monte Carlo methods in financial engineering*. Springer, New York, 2004.

[15] Michael Griebel, Frances Kuo, and Ian Sloan. The smoothing effect of integration in R^{d} and the ANOVA decomposition. *Mathematics of Computation*, 82(281):383–400, 2013.

[16] Michael Griebel, Frances Kuo, and Ian Sloan. Note on the smoothing effect of integration in R^{d} and the ANOVA decomposition. *Mathematics of Computation*, 86(306):1847–1854, 2017.

[17] Andreas Griewank, Frances Y Kuo, Hernan Leövey, and Ian H Sloan. High dimensional integration of kinks and jumps–smoothing by preintegration. *arXiv preprint arXiv:1712.00920*, 2017.

[18] Abdul-Lateef Haji-Ali, Fabio Nobile, Lorenzo Tamellini, and Raul Tempone. Multi-index stochastic collocation for random PDEs. *Computer Methods in Applied Mechanics and Engineering*, 306:95–122, 2016.

[19] Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993.

[20] Junichi Imai and Ken Seng Tan. Minimizing effective dimension using linear transformation. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 275–292. Springer, 2004.

[21] Monique Jeanblanc, Marc Yor, and Marc Chesney. *Mathematical methods for financial markets*. Springer Science & Business Media, 2009.

[22] Christian Kahl and Peter Jäckel. Fast strong approximation monte carlo schemes for stochastic volatility models. *Quantitative Finance*, 6(6):513–536, 2006.

[23] Roger Lord, Remmert Koekkoek, and Dick Van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194, 2010.

[24] William J Morokoff and Russel E Caflisch. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279, 1994.

[25] Bradley Moskowitz and Russel E Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathematical and Computer Modelling*, 23(8):37–54, 1996.

[26] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.

[27] Ye Xiao and Xiaoqun Wang. Conditional quasi-Monte Carlo methods and dimension reduction for option pricing and hedging with discontinuous functions. *Journal of Computational and Applied Mathematics*, 343:289–308, 2018.