

Numerical smoothing and hierarchical approximations for efficient option pricing and density estimation

Christian Bayer* Chiheb Ben Hammouda† Raúl Tempone‡§

Abstract

When approximating the expectation of a functional of a certain stochastic process, the efficiency and performance of deterministic quadrature methods such as sparse grids, and hierarchical variance reduction methods such as multilevel Monte Carlo (MLMC), may be highly deteriorated in different ways by the low regularity of the integrand with respect to the input parameters. To overcome this issue, a smoothing procedure is needed to uncover the available regularity and improve the performance of the aforementioned numerical methods. In this work, we consider cases where we cannot perform an analytic smoothing. Thus, we introduce a novel numerical smoothing technique based on root-finding combined with a one dimensional integration with respect to a single well-chosen variable. We prove that under appropriate conditions, the resulting function of the remaining variables is a highly smooth function, potentially allowing a higher efficiency of adaptive sparse grids quadrature (ASGQ), in particular when combined with hierarchical transformations (Brownian bridge and Richardson extrapolation on the weak error) to treat high dimensionality effectively. Our study is motivated by option pricing problems and our main focus is on dynamics where a discretization of the asset price is needed. Through our analysis and numerical experiments, we illustrate the advantage of combining numerical smoothing with ASGQ compared to the Monte Carlo (MC) approach. Furthermore, we demonstrate how numerical smoothing significantly reduces the kurtosis at the deep levels of MLMC, and also improves the strong convergence rate, when using Euler scheme. Due to the complexity theorem of MLMC, and given a pre-selected tolerance, TOL, this results in an improvement of the complexity from $\mathcal{O}(\text{TOL}^{-2.5})$ in the standard case to $\mathcal{O}(\text{TOL}^{-2} \log(\text{TOL})^2)$. Finally, we show how our numerical smoothing combined with MLMC enables us also to estimate density functions, which standard MLMC (without smoothing) fails to achieve.

Keywords Deterministic quadrature methods, Hierarchical variance reduction methods, Multilevel Monte Carlo, Numerical smoothing, Adaptive sparse grids quadrature, Brownian bridge, Richardson extrapolation, Option pricing, Monte Carlo.

2010 Mathematics Subject Classification 65C05, 65C20, 65D30, 65D32, 65Y20, 91G20, 91G60.

*Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Berlin, Germany.

†King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), Thuwal 23955 – 6900, Saudi Arabia (chiheb.benhammouda@kaust.edu.sa).

‡King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), Thuwal 23955 – 6900, Saudi Arabia (raul.tempone@kaust.edu.sa).

§Alexander von Humboldt Professor in Mathematics for Uncertainty Quantification, RWTH Aachen University, Germany.

1 Introduction

In many applications in quantitative finance, one is usually interested in computing efficiently

$$(1.1) \quad \mathbb{E}[g(X)],$$

where X is a certain stochastic process and g is an observable of the state process X .

Approximating (1.1) is usually challenging due to a combination of two complications:

1. The high dimensionality of the input space, a consequence of i) the time-discretization of a stochastic differential equation (SDE) describing the dynamics of the asset price, ii) path dependence of the option price on the whole trajectory of the underlying (i.e., X is a path of an asset price, not the value at a specific time), or iii) a large number of underlying assets, and so forth.
2. The low regularity of the payoff function, g , with respect to the input parameters.

There are mainly two classes of methods for approximating (1.1), and in this work, we introduce a numerical smoothing technique as means to improve the performance of both approaches.

1. The first class of methods relies on deterministic quadrature methods (sparse grids (SG) quadrature [8], adaptive sparse grids quadrature (ASGQ) [20], quasi Monte Carlo (QMC) [26], etc, ...) to approximate the integral arising from (1.1). Both the high dimension of the input space and the existence of discontinuities¹ in the integrand heavily degrade the performance of deterministic quadrature formulas. Despite the significant progress in SG methods [8] for high-dimensional integration of smooth integrands, few works have been carried out to deal with cases involving integrands with discontinuities.

Some works [17, 18, 19, 6, 27] have addressed similar kinds of problems, characterized by the presence of discontinuities, but with more emphasis on QMC. We note that [17, 18, 19] focus more on theoretical aspects of applying QMC in such a setting. On the other hand, here we focus more on specific practical problems, where we include the adaptivity paradigm. Other works [6, 27, 5] address the low regularity of the integrand by performing analytic smoothing based on conditional expectation tools, before applying quadrature methods. For instance, the authors in [27] imposed very strong assumptions to perform analytic smoothing. In our work, we do not make such strong assumptions, which is why we use numerical smoothing.

In the first part of this work, we consider cases where analytic smoothing cannot be done, and we introduce a novel numerical smoothing technique based on i) identifying the exact location of the discontinuities using root-finding algorithms, ii) employing suitable transformations of the integration domain, and iii) a pre-integration step with respect to the dimension containing the discontinuity. We prove that under appropriate conditions the resulting function of the remaining variables is a highly smooth function, thus potentially allowing a higher efficiency of ASGQ, in particular when it is combined with hierarchical transformations to effectively treat the high dimensionality. We apply the ASGQ method to solve the integration problem. Given that ASGQ benefits from anisotropy, the first representation consists

¹By discontinuities, we refer to either discontinuities in the gradients (kinks) or discontinuities in the function (jumps).

of applying a hierarchical path generation method (PGM) based on Brownian bridge construction, with the aim of reducing the effective dimension. The second technique consists of applying Richardson extrapolation to reduce the bias (weak error), which in turn reduces the number of time steps needed at the coarsest level to achieve a certain error tolerance and consequently the total number of dimensions needed for the integration problem. Our analysis, along with the numerical experiments, illustrates the advantage of our approach, which substantially outperforms the Monte Carlo (MC) approach even for high-dimensional cases and for dynamics where discretization is needed such as the Heston model.

2. The second class of methods relies on MC methods (standard MC, Multilevel Monte Carlo (MLMC) [16],...) to approximate the expectation in (1.1). Although the rate of convergence of standard MC is insensitive to both the dimension of the input space and the regularity of the observable g , it is very slow. On the other hand, MLMC, which is based on a hierarchical representation of the expectation and has a better speed of convergence than standard MC, is badly affected by the low regularity of g . These negative effects consist of i) a large variance and a low strong convergence rate that affect the speed of convergence of the MLMC method (see [15, 4] and our numerical experiments in Section 6) and ii) a high kurtosis at the deep levels of MLMC, affecting the robustness of the estimator (see Section 3.2 and our numerical experiments in Section 6). Furthermore, when g is a Dirac delta function, standard (without smoothing) MLMC or MC fail to approximate the density functions due to the singularity present in g , and the resulting infinite variance.

Avikainen in [4], and Giles, Higham, and Mao in [15] used MLMC for such a task without smoothing and obtained a poor performance for the MLMC estimator. On the other hand, a second approach was suggested in [11, 14], which used implicit smoothing based on the use of conditional expectation tools. There are two potential issues with this 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, *i.e.*, the Milstein scheme, to improve the strong order of convergence, and consequently, the speed of convergence of the MLMC estimator. Such a scheme becomes very computationally expensive for higher dimensional dynamics. Different non-smooth payoff functions were considered in [11, 14] (Asian, barrier, digital) but the only considered dynamics were the geometric Brownian motion (GBM) model. Finally, in [16], the authors suggested a different approach based on parametric smoothing. In fact, they carefully constructed a regularized version of the integrand, based on a regularization parameter that depends on a regularity index of the function of interest, and the tolerance requirement. This approach, despite offering better performance than the standard (without smoothing) MLMC estimator and a clear setting for theoretical analysis, has the practical disadvantage in the difficulty of generalizing it to i) cases where there is no prior knowledge of the degree of smoothness of the function of interest, and ii) more challenging dynamics than GBM dynamics that were tested and analyzed in [16].

In the second part of this work, we propose an alternative approach, based on numerical smoothing technique (as explained previously) combined with the MLMC estimator. Compared to the aforementioned works, our approach can be easily applied to cases where one cannot perform analytic smoothing. Furthermore, compared to the case without smoothing, we improve the robustness of the MLMC estimator by significantly reducing the kurtosis at the deep levels, and also improve the variance and the strong convergence rate, and consequently

the speed of convergence of the MLMC method from $\mathcal{O}(\text{TOL}^{-2.5})$ to $\mathcal{O}(\text{TOL}^{-2} \log(\text{TOL})^2)$, with TOL being a prescribed tolerance. We emphasize that we obtain similar rates of strong convergence and MLMC complexity as in [11, 14], without the need to use higher order schemes such as the Milstein scheme. In addition, our approach, compared to that of [16], is easier to apply for any dynamics and payoff function. Furthermore, our numerical smoothing idea enables us to approximate density functions using MLMC, a task where standard MLMC and MC (without smoothing) fail due to infinite variance. We emphasize that estimating densities using the approach in [16] has a mean square error (MSE) behavior similar to kernel density techniques, which increases exponentially with respect to the dimension of the state, \mathbf{X} (or a vector valued function that depends on the density of \mathbf{X}) (see Remark 6.6). However, thanks to the exact conditional expectation with respect to the Brownian bridge, the error of our approach is only restricted to the error of the root-finding algorithm for identifying the exact location of the discontinuity (see Remark 6.4); therefore, it is insensitive to the dimension of the state \mathbf{X} . Furthermore, although we provide pointwise estimates, our approach can be easily extended to approximate functions using similar ideas introduced in [16, 24] using interpolation grids. Finally, compared to [11, 14, 16], we add numerical results for the Heston model, where discretization is needed unlike the GBM dynamics, which is only considered here as an instructive example to showcase our approach.

We begin by explaining the technique of numerical smoothing and selecting the optimal smoothing direction in Section 2. In Section 3 we explain the different building blocks, constituting our hierarchical methods. We provide in Sections 4.1 and 4.2 an error and work discussion for both methods that we combine with numerical smoothing, namely ASGQ and MLMC. We also provide in Section 4.3 a smoothness analysis of the resulting integrand after performing the numerical smoothing. Finally, we show in Sections 5 and 6 the results obtained through the different numerical experiments conducted for the ASGQ and MLMC methods. The reported results illustrate the significant computational gains achieved by the ASGQ and MLMC methods, both combined with numerical smoothing, over the MC and the standard (without smoothing) MLMC method, respectively.

2 Problem formulation and setting

We start with the continuous time representation of the problem we are addressing. Then, we illustrate the spirit of our approach in the time-stepping setting.

In our context, we work mainly with three possible structures of payoff function g : i) $g(\mathbf{x}) = \max(\phi(\mathbf{x}), 0)$; ii) $g(\mathbf{x}) = \mathbf{1}_{(\phi(\mathbf{x}) \geq 0)}$; iii) $g(\mathbf{x}) = \delta(\phi(\mathbf{x}) = 0)$.²

We introduce the notation \mathbf{x}_{-j} to denote the vector of length $d - 1$ denoting all the variables other than x_j in \mathbf{x} . Furthermore, we assume for some $j \in \{1, \dots, d\}$

$$(2.1) \quad \frac{\partial \phi}{\partial x_j}(\mathbf{x}) > 0, \forall \mathbf{x} \in \mathbb{R}^d \quad (\text{Monotonicity condition})^3$$

$$(2.2) \quad \lim_{x_j \rightarrow +\infty} \phi(\mathbf{x}) = \lim_{x_j \rightarrow +\infty} \phi(x_j, \mathbf{x}_{-j}) = +\infty, \forall \mathbf{x}_{-j} \in \mathbb{R}^{d-1} \text{ or } \frac{\partial^2 \phi}{\partial x_j^2}(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^d \quad (\text{Growth condition}).$$

² $\mathbf{1}_{\mathcal{A}}$ is the indicator function of the set \mathcal{A} , and $\delta(\cdot)$ is the Dirac delta function.

2.1 Continuous time formulation and optimal smoothing direction

In this section, we aim to characterize the optimal smoothing direction through the continuous time formulation. We recall that the purpose of this work is to approximate $\mathbb{E}[g(\mathbf{X}_T)]$, where g is a certain low regular payoff function, and $\mathbf{X} := (X^{(1)}, \dots, X^{(d)})$ is described by the following SDE⁴

$$(2.3) \quad dX_t^{(i)} = a_i(\mathbf{X}_t)dt + \sum_{j=1}^d b_{ij}(\mathbf{X}_t)dW_t^{(j)}.$$

First, we start by hierarchically representing \mathbf{W} . We can write

$$W^{(j)}(t) = \frac{t}{T}W^{(j)}(T) + B^{(j)}(t) = \frac{t}{\sqrt{T}}Z_j + B^{(j)}(t),$$

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

Now, let us denote the optimal smoothing direction by \mathbf{v} , which we characterize in the following. Given the definition of \mathbf{v} , we represent $\mathbf{Z} := (Z_1, \dots, Z_d)$ hierarchically as

$$\mathbf{Z} = \underbrace{P_0\mathbf{Z}}_{\text{One dimensional projection}} + \underbrace{P_{\perp}\mathbf{Z}}_{\text{Projection on the complementary}},$$

where we write⁵ $P_0\mathbf{Z} := (\mathbf{Z}, \mathbf{v})\mathbf{v}$, and $\|\mathbf{v}\| = 1$. We can easily show that $Z_v := (\mathbf{Z}, \mathbf{v})$ is normal with $\mathbb{E}[Z_v] = 0$ and $\text{Var}(Z_v) = 1$ (since \mathbf{v} is a deterministic direction). Furthermore, for any $1 \leq j \leq d$, we can write (with $\mathbf{w} := \mathbf{Z} - Z_v\mathbf{v}$)

$$(2.4) \quad Z_j = Z_v v_j + (P_{\perp}\mathbf{Z})_j = Z_v v_j + w_j.$$

Going back to the SDE (2.3), we have

$$dX_t^{(i)} = a_i(\mathbf{X}_t)dt + \sum_{j=1}^d b_{ij}(\mathbf{X}_t)Z_j \frac{dt}{\sqrt{T}} + \sum_{j=1}^d b_{ij}(\mathbf{X}_t)dB_t^{(j)}.$$

Using (2.4) implies

$$(2.5) \quad dX_t^{(i)} = \left(a_i(\mathbf{X}_t) + \sum_{j=1}^d b_{ij}(\mathbf{X}_t) \frac{Z_v v_j}{\sqrt{T}} \right) dt + \left(\sum_{j=1}^d b_{ij}(\mathbf{X}_t) \frac{w_j}{\sqrt{T}} \right) dt + \sum_{j=1}^d b_{ij}(\mathbf{X}_t)dB_t^{(j)}.$$

If we define $H_{\mathbf{v}}(Z_v, \mathbf{w}) := g(\mathbf{X}(T))$, then observe that, given the representation of \mathbf{X} given by (2.5) and decomposition (2.4), we can write

$$(2.6) \quad \begin{aligned} \mathbb{E}[g(\mathbf{X}(T))] &= \mathbb{E}[\mathbb{E}[H_{\mathbf{v}}(Z_v, \mathbf{w}) \mid \mathbf{w}]] \\ \text{Var}[g(\mathbf{X}(T))] &= \mathbb{E}[\text{Var}[H_{\mathbf{v}}(Z_v, \mathbf{w}) \mid \mathbf{w}]] + \text{Var}[\mathbb{E}[H_{\mathbf{v}}(Z_v, \mathbf{w}) \mid \mathbf{w}]]. \end{aligned}$$

³Without a loss of generality, we show the monotonicity condition for an increasing function. However, the assumption still holds for a decreasing function, which can be the case when considering a spread option.

⁴We assume that the $\{W^{(j)}\}_{j=1}^d$ are uncorrelated and the correlation terms are included in the diffusion terms b_j .

⁵We use (\cdot, \cdot) to denote the scalar product operator.

Therefore, given (2.6), we characterize the optimal smoothing direction as the one that maximizes, at the final time, the smoothing effect, which is essentially the variance of the component orthogonal to the discontinuity. This implies that \mathbf{v} solves the following equivalent optimization problems

$$(2.7) \quad \max_{\substack{\mathbf{v} \in \mathbb{R}^d \\ \|\mathbf{v}\|=1}} \mathbb{E} [\text{Var} [H_{\mathbf{v}}(Z_v, \mathbf{w}) \mid \mathbf{w}]] \iff \min_{\substack{\mathbf{v} \in \mathbb{R}^d \\ \|\mathbf{v}\|=1}} \text{Var} [\mathbb{E} [H_{\mathbf{v}}(Z_v, \mathbf{w}) \mid \mathbf{w}]].$$

Solving the optimization problem (2.7) is a hard task, and the optimal smoothing direction \mathbf{v} is problem dependent. In this work, we aim to determine \mathbf{v} heuristically given the structure of the problem at hand. In the following Section, we provide more insights on how to choose \mathbf{v} and also how to perform the numerical smoothing in the time-stepping setting.

2.2 Motivation and idea of the numerical smoothing

To illustrate our approach, we consider in this section the example of an arithmetic basket call option under the multi-dimensional GBM model. We note that our approach is easily extended to any dynamics and payoff functions (see Sections 5 and 6 for the different examples that we test). We specifically consider the process \mathbf{X} to be the discretized d -dimensional GBM model given by

$$dX_t^{(j)} = \sigma^{(j)} X_t^{(j)} dW_t^{(j)}, \quad 1 \leq j \leq d,$$

where $\{W^{(1)}, \dots, W^{(d)}\}$ are correlated Brownian motions with correlations ρ_{ij} , and $\{\sigma^{(j)}\}_{j=1}^d$ are the volatilities of the different assets.

We also consider the payoff function g given by

$$(2.8) \quad g(\mathbf{X}(T)) = \max \left(\sum_{j=1}^d c_j X^{(j)}(T) - K, 0 \right),$$

with $\{c_j\}_{j=1}^d$ are weights of the basket, K is the exercise price, and T is the time to maturity.

We denote by $(Z_1^{(j)}, \dots, Z_N^{(j)})$ the N Gaussian independent random variables (rdvs) that will be used to construct the approximate path of the j -th asset $\overline{X}^{(j)}$, with N being the number of time steps used in the discretization. We denote $\psi^{(j)} : (Z_1^{(j)}, \dots, Z_N^{(j)}) \rightarrow (B_1^{(j)}, \dots, B_N^{(j)})$ the mapping of the Brownian bridge construction and by $\Phi : (\Delta t, \mathbf{B}) \rightarrow (\overline{X}_T^{(1)}, \dots, \overline{X}_T^{(d)})$, the mapping consisting of the time-stepping scheme, where $\mathbf{B} := (B_1^{(1)}, \dots, B_N^{(1)}, \dots, B_1^{(d)}, \dots, B_N^{(d)})$ is the non-correlated Brownian bridge⁶. Then, we can express the option price as

$$(2.9) \quad \begin{aligned} \mathbb{E} [g(\mathbf{X}(T))] &\approx \mathbb{E} \left[g \left(\overline{X}_T^{(1)}, \dots, \overline{X}_T^{(d)} \right) \right] := \mathbb{E} \left[g(\overline{\mathbf{X}}^{\Delta t}(T)) \right] \\ &= \mathbb{E} \left[g \circ \Phi \left(B_1^{(1)}, \dots, B_N^{(1)}, \dots, B_1^{(d)}, \dots, B_N^{(d)} \right) \right] \\ &= \mathbb{E} \left[g \circ \Phi \left(\psi^{(1)}(Z_1^{(1)}, \dots, Z_N^{(1)}), \dots, \psi^{(d)}(Z_1^{(d)}, \dots, Z_N^{(d)}) \right) \right] \\ &= \int_{\mathbb{R}^{d \times N}} G(z_1^{(1)}, \dots, z_N^{(1)}, \dots, z_1^{(d)}, \dots, z_N^{(d)}) \rho_{d \times N}(\mathbf{z}) dz_1^{(1)} \dots dz_N^{(1)} \dots dz_1^{(d)} \dots dz_N^{(d)}, \end{aligned}$$

where $G := g \circ \Phi \circ (\psi^{(1)}, \dots, \psi^{(d)})$, and $\rho_{d \times N}$ is the multivariate Gaussian density given by

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

⁶Without loss of generality, the correlated Brownian bridge can be obtained through a simple matrix multiplication.

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

$$(2.10) \quad \bar{X}^{(j)}(T) = X_0^{(j)} \prod_{n=0}^{N-1} \left[1 + \frac{\sigma^{(j)}}{\sqrt{T}} Z_1^{(j)} \Delta t + \sigma^{(j)} \Delta B_n^{(j)} \right] = X_0^{(j)} \prod_{n=0}^{N-1} f_n^{(j)}(Z_1^{(j)}), \quad 1 \leq j \leq d.$$

Remark 2.1. Note that (2.10) holds even for stochastic volatility models, with the particularity of $\sigma^{(j)}$ being non constant that changes value at each time step.

2.2.1 Step 1: Root-finding for determining the discontinuity location

The first step of our approach 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 $\mathbf{Z}_1 := (Z_1^{(1)}, \dots, Z_1^{(d)})$

$$(2.11) \quad \mathbf{Y} = \mathcal{A}\mathbf{Z}_1,$$

with \mathcal{A} being a certain $d \times d$ matrix that represents the linear mapping. To make connection with Section 2.1, the smoothing direction \mathbf{v} is then given by the first row of the linear mapping \mathcal{A} .

A general choice for \mathcal{A} should be in the family of rotations, which is problem dependent choice. For instance, if we consider the basket call option, then a sufficiently good choice of matrix \mathcal{A} would 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 the Gram-Schmidt procedure.

From (2.10), we have

$$\bar{X}^{(j)}(T) = X_0^{(j)} \prod_{n=0}^{N-1} f_n^{(j)}((\mathcal{A}^{-1}\mathbf{Y})_j) = X_0^{(j)} \prod_{n=0}^{N-1} F_n^{(j)}(Y_1, \mathbf{Y}_{-1}), \quad 1 \leq j \leq d,$$

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

$$F_n^{(j)}(Y_1, \mathbf{Y}_{-1}) = \left[1 + \frac{\sigma^{(j)} \Delta t}{\sqrt{T}} A_{j1}^{\text{inv}} Y_1 + \frac{\sigma^{(j)}}{\sqrt{T}} \left(\sum_{i=2}^d A_{ji}^{\text{inv}} Y_i \right) \Delta t + \sigma^{(j)} \Delta B_n^{(j)} \right].$$

Given that the irregularity is located at the strike price K (see (2.8))⁸, and in order to determine y_1^* , we need to solve, for fixed \mathbf{y}_{-1} ,

$$K = \sum_{j=1}^d c_j X_0^{(j)} \prod_{n=0}^{N-1} F_n^{(j)}(y_1^*(K), \mathbf{y}_{-1}),$$

which implies that the location of the discontinuity for the approximate problem is equivalent to finding the roots of the polynomial $P(y_1^*(K))$, given by

$$P(Y_1^*(K)) = \left(\sum_{j=1}^d c_j X_0^{(j)} \prod_{n=0}^{N-1} F_n^{(j)}(y_1^*) \right) - K.$$

We use Newton iteration method to find \bar{y}_1^* , approximation of the discontinuity location, y_1^* .

⁷We denote by $\mathbf{1}_{1 \times d}$ the row vector with dimension d , and where all its coordinates are equal to one.

⁸One may have different locations depending on the considered payoff function.

Remark 2.2. Due to using the hierarchical Brownian bridge construction, we suggest that the coarsest factors of the Brownian increments are the most important, compared to the remaining factors. Furthermore, the choice of the linear mapping \mathcal{A} creates a new hierarchy in terms of smoothness.

Remark 2.3. Although we consider the case of the basket call option under the multivariate GBM model to illustrate our numerical smoothing idea, we believe, thanks to our formulation of this step, that this idea can be extended in a straightforward manner to any kind of payoff and dynamics.

2.2.2 Step 2: Pre-Integration (Conditional expectation)

At this stage, we want to perform the pre-integrating step with respect to the direction where we performed the root-finding to determine the discontinuity location, namely y_1^* . In fact, using Fubini's theorem, we have from (2.9)

$$\begin{aligned}
\mathbb{E}[g(\mathbf{X}(T))] &\approx \mathbb{E}\left[g\left(\overline{X}_T^{(1)}, \dots, \overline{X}_T^{(d)}\right)\right] := \mathbb{E}\left[g(\overline{\mathbf{X}}^{\Delta t}(T))\right] \\
&= \int_{\mathbb{R}^{d \times N}} G\left(z_1^{(1)}, \dots, z_N^{(1)}, \dots, z_1^{(d)}, \dots, z_N^{(d)}\right) \rho_{d \times N}(\mathbf{z}) dz_1^{(1)} \dots dz_N^{(1)} \dots dz_1^{(d)} \dots dz_N^{(d)} \\
&= \int_{\mathbb{R}^{dN-1}} I\left(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) \rho_{d-1}(\mathbf{y}_{-1}) d\mathbf{y}_{-1} \rho_{d \times (N-1)}\left(\mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) d\mathbf{z}_{-1}^{(1)} \dots d\mathbf{z}_{-1}^{(d)} \\
(2.12) \quad &:= \mathbb{E}\left[I\left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)}\right)\right] \\
&\approx \mathbb{E}\left[\overline{I}\left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)}\right)\right],
\end{aligned}$$

where

$$\begin{aligned}
I\left(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) &= \int_{\mathbb{R}} G\left(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) \rho_{y_1}(y_1) dy_1 \\
(2.13) \quad &= \int_{-\infty}^{y_1^*} G\left(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) \rho_{y_1}(y_1) dy_1 + \int_{y_1^*}^{+\infty} G\left(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) \rho_{y_1}(y_1) dy_1,
\end{aligned}$$

and \overline{I} its approximation using Newton iteration and Laguerre quadrature, and given by

$$(2.14) \quad \overline{I}\left(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right) := \sum_{k=0}^{N_q} \eta_k G\left(\zeta_k(\overline{y}_1^*), \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}\right),$$

where \overline{y}_1^* is the approximated location of the discontinuity, and N_q is the number of Laguerre quadrature points $\zeta_k \in \mathbb{R}$ with $\zeta_0 = \overline{y}_1^*$ and corresponding weights η_k ⁹.

Since G has a discontinuity, computing h in (2.13) should be carried carefully to not deteriorate its regularity. We generally do not have a closed form for the function I . Therefore, the pre-integration step should be performed numerically after solving the root-finding problem, as explained in Section 2.2.1. In our work, the pre-integration step is performed by summing the terms corresponding to the uni-variate integrals in each region where G is smooth. To this end, we use the Gauss-Laguerre quadrature to obtain \overline{I} , given by (2.14), which is an approximation of I defined in (2.13).

⁹Of course, the points ζ_k have to be chosen in a systematic manner depending on \overline{y}_1^* .

Both approaches that we use in this work, namely ASGQ and MLMC methods (see Section 3) aim to efficiently approximate the resulting expectation, $\mathbb{E} \left[\bar{I} \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right]$, defined in (2.12) and (2.14), and obtained after the numerical smoothing step, as described in Section 2.2. This numerical treatment enables us to uncover the available regularity, and hence achieve a highly smooth integrand, \bar{I} , inside the expectation (see Section 4.3 for a smoothness analysis of \bar{I}). Therefore, applying a deterministic quadrature technique such as ASGQ, with careful handling of the high-dimensional integration problem defined by (2.12), becomes an adequate option for computing the option price, as we will investigate later. On the other hand, we also emphasize that the numerical smoothing step plays a role of variance reduction that improves the performance of the MLMC method, as we will illustrate numerically in Section 6.

Remark 2.4. The pre-integration step described previously can be generalized to the case when there are multiple discontinuities, which can be either due to the payoff structure or because of using the Richardson extrapolation. In this case, one needs a prior knowledge of those multiple locations.

Remark 2.5. It is observed that for most considered payoffs, the discontinuities arise along a lower-dimensional manifold, and the additional computational costs for the zero finding are negligible. There might be, for a given $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})$: i) Case 1: no solution, i.e., the integrand is smooth (*best case*); ii) Case 2: a unique solution; iii) Case 3: multiple solutions.

Generally, we assume that we are in the first or second case. This assumption is reasonable, for instance, if we are in the setting described by conditions ((2.1) and (2.2)). Then for each fixed \mathbf{y}_{-1} , the function G either has a simple root y_1^* or is positive for all $y_1 \in \mathbb{R}$.

In many situations, case 2 (which is thought to include case 1) can be guaranteed by monotonicity (see assumption (2.1)). 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 GBM model and the Cox-Ingersoll-Ross (CIR) process .

Even in the multi-dimensional case, such monotonicity conditions can hold in specific situations. For instance, in the case of a basket option under the multivariate GBM, we can choose a linear combination of the terminal values of the driving Brownian motion, denoted by Y_1 in (2.11), 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.

Remark 2.6. If the ASGQ used for computing the integral of \bar{I} depends on derivatives (or difference quotients) of \bar{I} , then we may also need to make sure that derivatives of \bar{I} are close enough to the derivatives of I . This may require higher order solution methods for determining y_1^* .

3 Hierarchical Integration Methods Combined with Numerical Smoothing

We recall that our goal is to compute the expectation in (2.9). To perform this task, our approach can be seen as a two-stage method. In the first stage, we uncover the available regularity in our

problem through a numerical smoothing procedure, based on a root-finding algorithm to determine the location of the discontinuity with respect to the coarsest factors, followed by a pre-integration step (see Sections 2.2). After performing the first stage of our methodology, we end up with an integration problem (2.12) of a highly regular integrand (see our smoothness analysis in Section 4.3) that lives in $(dN - 1)$ -dimensional space, which becomes very large due to either i) a large number of time steps, N , used in the discretization scheme, or ii) a large number of assets, d . The second stage of our approach consists of efficiently approximating the high-dimensional expectation (integral) (2.12). To this end, we design mainly two methods, namely i) a hierarchical ASGQ, using the same construction as in [20, 5], and ii) the MLMC method, as described in [12, 13].

We briefly describe the ASGQ and MLMC methods in our context in Sections 3.1 and 3.2, respectively. To make effective use of ASGQ, we use a similar idea that was introduced in [5], and we apply two hierarchical representations to overcome the issue of facing a high-dimensional integrand. We first apply a hierarchical path generation method, based on Brownian bridge construction, with the aim of reducing the effective dimension. Then, we apply Richardson extrapolation to reduce the bias, resulting in considerably reducing the dimension of the integration. We refer to [5] for more details on how to apply these two hierarchical representations.

3.1 Adaptive sparse grids quadrature (ASGQ)

Using the same notation and construction as in Section 4.1 in [5], the ASGQ estimator used for approximating (2.12), and using a set of multi-indices $\mathcal{I} \subset \mathbb{N}^{dN-1}$ is given by

$$(3.1) \quad Q_N^{\mathcal{I}} = \sum_{\beta \in \mathcal{I}} \Delta Q_N^{\beta}, \quad \text{with} \quad \Delta Q_N^{\beta} = \left(\prod_{i=1}^{dN-1} \Delta_i \right) Q_N^{\beta},$$

and

$$\Delta_i Q_N^{\beta} := \begin{cases} Q_N^{\beta} - Q_N^{\beta'}, & \text{with } \beta' = \beta - e_i, \text{ if } \beta_i > 0, \\ Q_N^{\beta}, & \text{otherwise,} \end{cases}$$

where Q^{β} is the Cartesian quadrature operator with $m(\beta_i)$ points along the i th dimension.

If we denote by $\mathcal{E}_{\text{total, ASGQ}}$ the total error of approximating $\mathbb{E}[g(\mathbf{X}(T))]$ using the ASGQ estimator, Q_N (defined by (3.1)), then we have a natural error decomposition

$$(3.2) \quad \begin{aligned} \mathcal{E}_{\text{total, ASGQ}} &\leq \underbrace{\left| \mathbb{E}[g(\mathbf{X}(T))] - \mathbb{E}\left[I(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})\right] \right| + \left| \mathbb{E}\left[I(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})\right] - \mathbb{E}\left[\bar{I}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})\right] \right|}_{\text{Bias}} \\ &\quad + \underbrace{\left| \mathbb{E}\left[\bar{I}(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)})\right] - Q_N \right|}_{\text{Quadrature error}} \\ &\leq \mathcal{E}_B(N) + \mathcal{E}_Q(\text{TOL}_{\text{ASGQ}}, N, d), \end{aligned}$$

where \mathcal{E}_Q is the quadrature error, \mathcal{E}_B is the bias, TOL_{ASGQ} is a user-selected tolerance, and I and \bar{I} are given by (2.13) and (2.14). We refer to Section 4.1 for more analysis on the error of the ASGQ method, and to Section 4.1 in [5] for more details on the construction of the ASGQ method.

3.2 Multilevel Monte Carlo (MLMC)

Let \mathbf{X} be a stochastic process and $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a smooth scalar observable. Let us assume that we want to approximate $\mathbb{E}[f(\mathbf{X}(T))]$, but instead of sampling directly from $\mathbf{X}(T)$, we sample

from $\overline{\mathbf{X}}^{\Delta t}(T)$, which are rdvs generated by an approximate method with step size Δt . Let us assume also that the variates $\overline{\mathbf{X}}^{\Delta t}(T)$ are generated with an algorithm with weak order, $\mathcal{O}(\Delta t)$, *i.e.*, $\mathbb{E} \left[f(\mathbf{X}(T)) - f(\overline{\mathbf{X}}^{\Delta t}(T)) \right] = \mathcal{O}(\Delta t)$. Let μ_M be the standard Monte Carlo estimator of $\mathbb{E} \left[f(\overline{\mathbf{X}}^{\Delta t}(T)) \right]$ defined by $(\overline{\mathbf{X}}_{[m]}^{\Delta t}(T))$ are independent and distributed as $\overline{\mathbf{X}}^{\Delta t}(T)$

$$\mu_M := \frac{1}{M} \sum_{m=1}^M f(\overline{\mathbf{X}}_{[m]}^{\Delta t}(T)).$$

Consider now the following decomposition of the global error

$$\mathbb{E} [f(\mathbf{X}(T))] - \mu_M = \left(\mathbb{E} \left[f(\mathbf{X}(T)) - f(\overline{\mathbf{X}}^{\Delta t}(T)) \right] \right) + \left(\mathbb{E} \left[f(\overline{\mathbf{X}}^{\Delta t}(T)) \right] - \mu_M \right).$$

To achieve the desired accuracy, TOL, it is sufficient to take $\Delta t = \mathcal{O}(\text{TOL})$ so that the first term on the right is $\mathcal{O}(\text{TOL})$ and, by the Central Limit Theorem, impose $M = \mathcal{O}(\text{TOL}^{-2})$ so that the statistical error given by the second term on the right is $\mathcal{O}(\text{TOL})$ [10]. As a consequence, the expected total computational work is $\mathcal{O}(\text{TOL}^{-3})$.

The MLMC estimator, introduced by Giles [12], allows us to reduce the total computational work up to $\mathcal{O} \left(\text{TOL}^{-2 - \max(0, \frac{\gamma - \beta}{\alpha})} \log(\text{TOL})^{2 \times \mathbf{1}_{\{\beta = \gamma\}}} \right)$, where (α, β, γ) are weak, strong and work rates respectively. We refer to [9, 13] for the theorem of the computational complexity of the MLMC estimator for different scenarios.

We construct the MLMC estimator as follows: consider a hierarchy of nested meshes of the time interval $[0, T]$, indexed by $\ell = 0, 1, \dots, L$. We denote by Δt_0 the step size used at level $\ell = 0$. The size of the subsequent time steps for levels $\ell \geq 1$ are given by $\Delta t_\ell = K^{-\ell} \Delta t_0$, where $K > 1$ is a given integer constant. In this work, we take $K = 2$. Hereafter, $\overline{\mathbf{X}}_\ell$ denotes the approximate process generated using a step size of Δt_ℓ . We also denote by M_ℓ the number of samples at level ℓ . Consider now the following telescoping decomposition of $\mathbb{E} [f(\overline{\mathbf{X}}_L(T))]$

$$\begin{aligned} \mathbb{E} [g(\overline{\mathbf{X}}_L(T))] &= \underbrace{\mathbb{E} [f(\overline{\mathbf{X}}_0(T))]}_{M_0} + \underbrace{\sum_{\ell=1}^L \mathbb{E} [f(\overline{\mathbf{X}}_\ell(T)) - f(\overline{\mathbf{X}}_{\ell-1}(T))]}_{\gg M_\ell (\searrow \text{ as } \ell \nearrow)} \\ &\text{Var}[f(\overline{\mathbf{X}}_0(T))] \gg \text{Var}[f(\overline{\mathbf{X}}_\ell(T)) - g(\overline{\mathbf{X}}_{\ell-1}(T))] (\searrow \text{ as } \ell \nearrow) \\ &M_0 \gg M_\ell (\searrow \text{ as } \ell \nearrow) \end{aligned}$$

Then, by defining

$$\begin{cases} \widehat{Q}_0 := \frac{1}{M_0} \sum_{m_0=1}^{M_0} f(\overline{\mathbf{X}}_{0,[m_0]}(T)) \\ \widehat{Q}_\ell := \frac{1}{M_\ell} \sum_{m_\ell=1}^{M_\ell} (f(\overline{\mathbf{X}}_{\ell,[m_\ell]}(T)) - f(\overline{\mathbf{X}}_{\ell-1,[m_\ell]}(T))), \quad 1 \leq \ell, L, \end{cases}$$

we arrive at the unbiased MLMC estimator, \widehat{Q} , of $\mathbb{E} [f(\overline{\mathbf{X}}_L(T))]$

$$(3.3) \quad \widehat{Q} := \sum_{\ell=0}^L \widehat{Q}_\ell.$$

We note that the key point here is that both $\overline{\mathbf{X}}_{\ell,[m_\ell]}(T)$ and $\overline{\mathbf{X}}_{\ell-1,[m_\ell]}(T)$ are sampled using different time discretizations but with the same generated randomness.

In our context, our aim is to approximate $\mathbb{E} \left[\bar{I}(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)}) \right]$, defined in (2.12). We denote by I_ℓ and \bar{I}_ℓ the level ℓ approximations of I and \bar{I} expressed in (2.13) and (2.14) (\bar{I}_ℓ computed with N_ℓ time steps; $N_{q,\ell}$ the number of Laguerre quadrature points at level ℓ ; and $\text{TOL}_{\text{Newton},\ell}$ the tolerance of the Newton method at level ℓ). In this case, \widehat{Q}_ℓ , defined in (3.3), is given by

$$\widehat{Q}_\ell := \frac{1}{M_\ell} \sum_{m_\ell=1}^{M_\ell} (\bar{I}_{\ell,[m_\ell]} - \bar{I}_{\ell-1,[m_\ell]})$$

In the context of option pricing or density estimation of asset price dynamics, the standard MLMC estimator (without smoothing) fails or does not have the optimal performance and robustness due to the singularity present in the payoff or the delta functions, implying that the MLMC estimator suffers from i) a high kurtosis at the deep levels, ii) high variance, and iii) low strong convergence rate. We refer to Section 6 for a numerical illustration of these issues.

To explain the effect of the high kurtosis on the robustness and performance of the MLMC estimator, let G denote a rrv, and let G_ℓ denote the corresponding level ℓ numerical approximation. We also define $Y_\ell = G_\ell - G_{\ell-1}$. Then, the standard deviation of the sample variance for the rrv Y_ℓ is given by

$$(3.4) \quad \sigma_{\mathcal{S}^2(Y_\ell)} = \frac{\text{Var}[Y_\ell]}{\sqrt{M}} \sqrt{(\kappa - 1) + \frac{2}{M - 1}},$$

where the kurtosis $\kappa = \frac{\mathbb{E}[(Y_\ell - \mathbb{E}[Y_\ell])^4]}{(\text{Var}[Y_\ell])^2}$.

Hence, $\mathcal{O}(\kappa)$ samples are required to obtain a reasonable estimate for the variance.

For the setting of the MLMC method, accurate estimates for $V_\ell = \text{Var}[Y_\ell]$ are required. Therefore, when the kurtosis of Y_ℓ is high, it affects both the robustness (no reliable estimates of the sample variance, V_ℓ) and also the performance (too many samples are required to control $\sigma_{\mathcal{S}^2(Y_\ell)}$ given by (3.4)) of the multilevel estimator.

In this work, we address the aforementioned issues and propose a novel approach that is based on the MLMC method combined with the numerical smoothing idea, as explained in Sections 2.1 and 2.2. Our approach of numerical smoothing improves the robustness of the MLMC estimator, by reducing significantly the kurtosis at the deep levels, and also improves the complexity of the MLMC estimator by reducing the variance and increasing the strong rate of convergence.

4 Error discussion and smoothness analysis

4.1 Error and work discussion for ASGQ combined with numerical smoothing

In this section, we discuss and analyze the different errors that we have in our approach when using the ASGQ method combined with the numerical smoothing idea. Let us denote by Q_N the ASGQ (as defined in Section 3.1) that we use to approximate $\mathbb{E}[g(X(T))]$, then following the notation of

Section 2.2, we have the following error decomposition

$$\begin{aligned}
(4.1) \quad \mathbb{E} [g(X(T)) - Q_N] &= \underbrace{\mathbb{E} [g(X(T))] - \mathbb{E} [g(\bar{\mathbf{X}}^{\Delta t}(T))]}_{\text{Error I: bias or weak error}} \\
&+ \underbrace{\mathbb{E} \left[I \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right] - \mathbb{E} \left[\bar{I} \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right]}_{\text{Error II: numerical smoothing error}} \\
&+ \underbrace{\mathbb{E} \left[\bar{I} \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right] - Q_N}_{\text{Error III: ASGQ error}}
\end{aligned}$$

Since we use schemes based on forward Euler to simulate asset dynamics then we have

$$(4.2) \quad \text{Error I} = \mathcal{O}(\Delta t).$$

Given the details in Section 3.1, we have, for N_{ASGQ} quadrature points used by the ASGQ method,

$$(4.3) \quad \text{Error III} = \mathcal{O}\left(N_{\text{ASGQ}}^{-p}\right),$$

with $p := p(N, d) > 0$ is related to the degree of regularity of \bar{I} , defined in (2.12) and (2.13), in the $dN - 1$ dimensional space¹⁰. We note that the claimed rate of convergence of Error III is justified by our smoothness analysis in Section 4.3, with the assumption that \bar{I} converges to I for sufficiently large N_q (number of quadrature points used in the one dimensional quadrature in the numerical smoothing procedure) and small tolerance $\text{TOL}_{\text{Newton}}$. Furthermore, since we expect that the regularity index p is monotone decreasing function of the dimension of the problem, which is $dN - 1$, then handling the high dimensionality using hierarchical representations such as Brownian bridge and Richardson extrapolation enable us to work in the region where the smoothness of \bar{I} is not deteriorated.

Regarding Error II in (4.1), let us denote by y_1^* the exact location of the discontinuity and by \bar{y}_1^* the approximated location of the discontinuity obtained by Newton iteration (without loss of generality, we assume that $y_1^* < \bar{y}_1^*$), then we have $|y_1^* - \bar{y}_1^*| = \text{TOL}_{\text{Newton}}$ and

$$\begin{aligned}
(4.4) \quad \text{Error II} &:= \mathbb{E} \left[I \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right] - \mathbb{E} \left[\bar{I} \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right] \\
&\leq \sup_{\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)}} \left| I \left(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)} \right) - \bar{I} \left(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \dots, \mathbf{z}_{-1}^{(d)} \right) \right| \\
&= \mathcal{O}(N_q^{-s}) + \mathcal{O}\left(|y_1^* - \bar{y}_1^*|^{\kappa+1}\right) = \mathcal{O}(N_q^{-s}) + \mathcal{O}(\text{TOL}_{\text{Newton}}^{\kappa+1}),
\end{aligned}$$

where $\kappa \geq 0$ ¹¹, N_q is the number of points used by the Laguerre quadrature for the pre-integration step, and $s > 0$ is related to the degree of regularity of the integrand, G , with respect to y_1 .¹²

The first error contribution in (4.4) is coming from the one dimensional pre-integration step using Laguerre quadrature, as explained in Section 2.2.2. Given that G is a highly smooth function in the parts of the integration domain separated by the discontinuity location, then we have an

¹⁰In this case, the weighted mixed derivatives of I are bounded up to order p .

¹¹The value of κ depends on the payoff, for instance, $\kappa = 0$ for a digital option and $\kappa = 1$ for a call/put payoffs.

¹²In this case, the derivatives of G with respect to y_1 are bounded up to order s .

exponential convergence of the quadrature, which justifies the term N_q^{-s} . The second error contribution in (4.4) is due to the gap created by integrating the function G over domains separated by an approximate location of the discontinuity \bar{y}_1^* instead of y_1^* , which is the exact location.

Given (4.1), (4.2), (4.4) and (4.3), then heuristically, we have as a total error estimate

$$(4.5) \quad \mathcal{E}_{\text{total, ASGQ}} := \mathbb{E}[g(X(T)) - Q_N] = \mathcal{O}(\Delta t) + \mathcal{O}(N_{\text{ASGQ}}^{-p}) + \mathcal{O}(N_q^{-s}) + \mathcal{O}(\text{TOL}_{\text{Newton}}^{\kappa+1}).$$

For an optimal performance of our approach, we need to optimize over the parameters present in (4.5) to achieve a certain error tolerance, TOL, with the least amount of work, by solving (4.6)

$$(4.6) \quad \begin{cases} \min_{(N_{\text{ASGQ}}, N_q, \text{TOL}_{\text{Newton}})} \text{Work}_{\text{ASGQ}} \propto N_{\text{ASGQ}} \times N_q \times \Delta t^{-1} \\ \text{s.t. } \mathcal{E}_{\text{total, ASGQ}} = \text{TOL}. \end{cases}$$

Although we do not solve (4.6) in our experiments in Section 5 (we choose the parameters heuristically such that we achieve a sub-optimal performance of our method), we show in Appendix B that, for a given error tolerance TOL, and under certain conditions for the regularity parameters s and p ($p, s \gg 1$), a lower bound on the computational work the ASGQ method is of order $\text{Work}_{\text{ASGQ}} = \mathcal{O}(\text{TOL}^{-1})$. This is significantly better than the MC method which, in the best scenario, achieves a computational work of order $\mathcal{O}(\text{TOL}^{-3})$.

We emphasize that the optimal performance for ASGQ can be deteriorated i) if p and s are not large enough, or ii) due to the adverse effect of the high dimension that may affect the rates badly. Remember that when using sparse grids (not adaptive), then error III (given by (4.3) in our case) will be $\mathcal{O}(N_{\text{SG}}^{-p} (\log(N_{\text{SG}}))^{(d-1)(p+1)})$ (where d is the dimension of the integration domain, and for functions with bounded mixed derivatives up to order p). In our case, when $p \gg 1$, we claim that error III is of order $\mathcal{O}(N_{\text{ASGQ}}^{-p})$, and the deterioration of the rate in practice may be due to ignoring that log term, implying the curse of dimensionality effect. Furthermore, although we work in the pre-asymptotic regime (small number of time steps, N), we should emphasize that the regularity parameter p may be deteriorated when increasing the dimension of the integration problem by increasing N .

Remark 4.1. Although we carried out our previous analysis without using Richardson extrapolation, we should mention that using this hierarchical representation improves the complexity rate of ASGQ (as we observed through our numerical experiments in Section 5).

4.2 Error and work discussion for MLMC combined with numerical smoothing

In this section, we discuss the different errors when using the MLMC method combined with the numerical smoothing idea. Following the notation of Sections 2.2 and 3.2, we have then the following error decomposition

$$(4.7) \quad \begin{aligned} \mathbb{E}[g(X(T)) - \hat{Q}] &= \underbrace{\mathbb{E}[g(X(T))] - \mathbb{E}[g(\bar{\mathbf{X}}^{\Delta t_L}(T))]}_{\text{Error I: bias or weak error}} \\ &+ \underbrace{\mathbb{E}\left[I_L(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)})\right] - \mathbb{E}\left[\bar{I}_L(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)})\right]}_{\text{Error II: numerical smoothing error}} \\ &+ \underbrace{\mathbb{E}\left[\bar{I}_L(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)})\right] - \hat{Q}}_{\text{Error III: MLMC statistical error}}. \end{aligned}$$

Since we use schemes based on forward Euler to simulate asset dynamics then we have

$$(4.8) \quad \text{Error I} = \mathcal{O}(\Delta t_L).$$

Error II in (4.7) is given by the same structure as in (4.4). Regarding the structure of Error III, we have \widehat{Q} as an unbiased estimator of $\mathbb{E} \left[\bar{I}_L \left(\mathbf{Y}_{-1}, \mathbf{Z}_{-1}^{(1)}, \dots, \mathbf{Z}_{-1}^{(d)} \right) \right]$ and the only type of error is the statistical error, equal to $\sqrt{\sum_{\ell=L_0}^L M_\ell^{-1} V_\ell}$, with $V_\ell := \text{Var} [\bar{I}_\ell - \bar{I}_{\ell-1}]$. We note that the optimal number of samples of the MLMC estimator is $M_\ell^* \propto \sqrt{\frac{V_\ell}{N_{q,\ell} \Delta t_\ell^{-1} \log(\text{TOL}_{\text{Newton},\ell}^{-1})}}$ ¹³, and from our numerical experiments (see Section 6), we observe that $V_\ell \propto \Delta t_\ell \left(d_1 + d_2 N_{q,\ell}^{-r} + d_3 \text{TOL}_{\text{Newton},\ell} \right)$, with $d_{1,2,3}$ being constants and $r \geq 1$. Therefore,

$$\begin{aligned} V_\ell N_{q,\ell} \Delta t_\ell^{-1} \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right) &= \left(d_1 N_{q,\ell} + d_2 N_{q,\ell}^{-r+1} + d_3 \text{TOL}_{\text{Newton},\ell} N_{q,\ell} \right) \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right) \\ &\approx d_1 N_{q,\ell} \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right), \end{aligned}$$

implying that

$$(4.9) \quad \text{Error III} = \mathcal{O} \left(\sqrt{\sum_{\ell=L_0}^L \sqrt{N_{q,\ell} \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right)}} \right).$$

Given (4.7), (4.8), (4.4) and (4.9), then we have as a total error estimate

$$(4.10) \quad \begin{aligned} \mathcal{E}_{\text{total, MLMC}} &:= \mathbb{E} [g(X(T)) - \widehat{Q}] \\ &= \mathcal{O}(\Delta t_L) + \mathcal{O} \left(\sqrt{\sum_{\ell=L_0}^L \sqrt{N_{q,\ell} \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right)}} \right) + \mathcal{O} \left(N_{q,L}^{-s} \right) + \mathcal{O} \left(\text{TOL}_{\text{Newton},L}^{\kappa+1} \right). \end{aligned}$$

For an optimal performance of our approach, we need to optimize the parameters present in (4.10) to achieve a certain error tolerance, TOL, with the least amount of work, by solving (4.11)

$$(4.11) \quad \begin{cases} \min_{(L, L_0, \{M_\ell\}_{\ell=L_0}^L, N_q, \text{TOL}_{\text{Newton}})} \text{Work}_{\text{MLMC}} \propto \sum_{\ell=L_0}^L M_\ell (N_{q,\ell} \Delta t_\ell^{-1}) \\ \text{s.t. } \mathcal{E}_{\text{total, MLMC}} = \text{TOL}. \end{cases}$$

In Section 6, we determine the optimal parameters solving (4.11) heuristically.

4.3 Smoothness analysis

For an optimal performance of the ASGQ method, explained in Section 3.1, we require the integrand to be analytic or highly smooth. In fact, although we face the issue of the ‘‘curse of dimensionality’’ when increasing either the number of time steps, N , or the number of assets, d , the high smoothness of the integrand implies a spectral convergence for ASGQ. For this reason, we attempt in this Section to provide a smoothness analysis of the integrand of interest.

¹³Note that $N_{q,\ell} \Delta t_\ell^{-1} \log \left(\text{TOL}_{\text{Newton},\ell}^{-1} \right)$ corresponds to the cost per sample per level in the MLMC estimator.

4.3.1 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

$$\psi(t) := \begin{cases} 1, & 0 \leq t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq t < 1, \\ 0, & \text{else,} \end{cases}$$

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

$$\psi_{-1}(t) := \mathbb{1}_{[0,1]}(t); \quad \psi_{n,k}(t) := 2^{n/2} \psi(2^n t - k), \quad n \in \mathbb{N}_0, \quad k = 0, \dots, 2^n - 1.$$

We note that $\text{supp } \psi_{n,k} = [2^{-n}k, 2^{-n}(k+1)]$. Moreover, we define a grid $\mathcal{D}^n := \{t_\ell^n \mid \ell = 0, \dots, 2^{n+1}\}$ by $t_\ell^n := \frac{\ell}{2^{n+1}}$. 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

$$\Psi_{-1}(t) := \int_0^t \psi_{-1}(s) ds; \quad \Psi_{n,k}(t) := \int_0^t \psi_{n,k}(s) ds.$$

For an i.i.d. set of standard normal r.v.s (*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

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

and the truncated version

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

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

$$\Delta_\ell^N F := F(t_{\ell+1}^N) - F(t_\ell^N).$$

4.3.2 Smoothness analysis

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

$$(4.12) \quad dX_t = b(X_t) dW_t, \quad X_0 = x \in \mathbb{R}.$$

We assume that b is bounded and has bounded derivatives of all orders. Recall that we want to compute, for some function $g : \mathbb{R} \rightarrow \mathbb{R}$ which is not necessarily smooth, $\mathbb{E}[g(X_T)]$. We also define the solution of the Euler scheme along the grid \mathcal{D}^N by $X_0^N := X_0 = x$ and (For convenience, we also define $X_T^N := X_{2^N}^N$.)

$$(4.13) \quad X_{\ell+1}^N := X_\ell^N + b(X_\ell^N) \Delta_\ell^N W.$$

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

$$(4.14) \quad X_\ell^N = X_\ell^N(Z_{-1}, Z^N)$$

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

We offer a note of caution regarding convergence as $N \rightarrow \infty$: while the sequence of random processes X^N converges to the solution of (4.12) (under the usual assumptions on b): this is not true in any sense for the deterministic functions.

Define the deterministic function $H^N : \mathbb{R}^{2^{N+1}-1} \rightarrow \mathbb{R}$, given by

$$(4.15) \quad H^N(z^N) := E[g(X_T^N(Z_{-1}, z^N))].$$

Then, H^N satisfies Theorem 4.2.

Theorem 4.2. *Assume that X_T^N , defined by (4.13) and (4.14), satisfies Assumptions A.1 and A.3. Then, for any $p \in \mathbb{N}$ and indices n_1, \dots, n_p and k_1, \dots, k_p (satisfying $0 \leq k_j < 2^{n_j}$), the function H^N defined in (4.15), satisfies (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 4.3. We actually expect that H^N is analytic, but a formal proof seems subtle. 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 \blacksquare below) increases fast as p increases due to the product rule of differentiation. Hence, the constant in front 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 4.2 needs to be understood as an assertion about the anisotropy in the variables $z_{n,k}$ rather than a statement on the behavior of higher and higher derivatives of H^N . In fact, we 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 behavior in general.

Sketch of a proof of Theorem 4.2. We apply integration by parts p times as in the proof of Proposition A.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

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

where \blacksquare represents a long sum of products of various terms. However, notice the following structure: ignoring derivatives w.r.t. y , each summand contains all derivatives w.r.t. $z_{n_1, k_1}, \dots, z_{n_p, k_p}$ 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 A.1 and A.3, this implies the claimed result by Lemma A.8. \square

5 Numerical experiments: Numerical smoothing with ASGQ

In this section, we conduct our experiments for three different examples of payoffs: i) a single digital option, ii) a single call option, and iii) a $4d$ -basket call option. The three examples are tested under two different dynamics for the asset price: i) the discretized GBM model (a didactic example), and ii) the discretized Heston model, which is a relevant application of our approach.

Remark 5.1. Although, the discretization of the GBM dynamics is not needed for some of the examples that we consider. We emphasize that we aim to show the efficiency of our proposed approach for dynamics where discretization is required, such as the Heston model that we test in this work. On the other hand, examples under the GBM are only considered here as instructive examples to showcase our approach.

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. While fixing a sufficiently small error tolerance in the price estimates, we compare the computational time needed for both methods to meet the desired error tolerance. For all our numerical experiments, the reported errors are relative errors, normalized by the reference solutions. Furthermore, we conduct our numerical experiments for two different scenarios: i) without Richardson extrapolation, and ii) with level 1 of the Richardson extrapolation. We note that in all cases the actual work (runtime) is obtained using an Intel(R) Xeon(R) CPU E5-268 architecture.

We show the summary of our numerical findings in Table 5.1, which highlights the computational gains achieved by ASGQ combined with numerical smoothing compared to the MC method to meet a certain error tolerance, which we set approximately below 1%. We note that the results are reported using the best configuration with Richardson extrapolation for each method. More detailed results for each case are provided in Sections 5.1.1, 5.1.2, 5.1.3, 5.2.1, and 5.2.2.

| Example | Total relative error | CPU time (ASGQ/MC) in % |
|--------------------------------|----------------------|-------------------------|
| Single digital option (GBM) | 0.7% | 0.7% |
| Single call option (GBM) | 0.5% | 0.8% |
| $4d$ -Basket call option (GBM) | 0.8% | 7.4% |
| Single digital option (Heston) | 0.6% | 6.2% |
| Single call option (Heston) | 0.5% | 17.2% |

Table 5.1: Summary of relative errors and computational gains, achieved by ASGQ combined with numerical smoothing, compared to the MC method, to meet a certain error tolerance. We note that the ratios are computed for the best configuration with Richardson extrapolation for each method.

5.1 Options under the discretized GBM model

In this section, we are interested in the one dimensional lognormal example where, given a standard one-dimensional Brownian motion $\{W_t, 0 \leq t \leq T\}$, the dynamics of the stock are represented by

$$dX_t = \sigma X_t dW_t.$$

5.1.1 Single digital option under the discretized GBM model

The first example that we test is the single digital option under the discretized GBM model, with parameters : $T = 1$, $\sigma = 0.4$ and $S_0 = K = 100$. The exact reference value of this case is 0.42074.

Figure 5.1 shows a comparison of the numerical complexity for each method under the two different scenarios. This Figure shows that, to achieve a relative error below 1%, level 1 of the Richardson extrapolation is the optimal configuration for both the MC and the ASGQ methods.

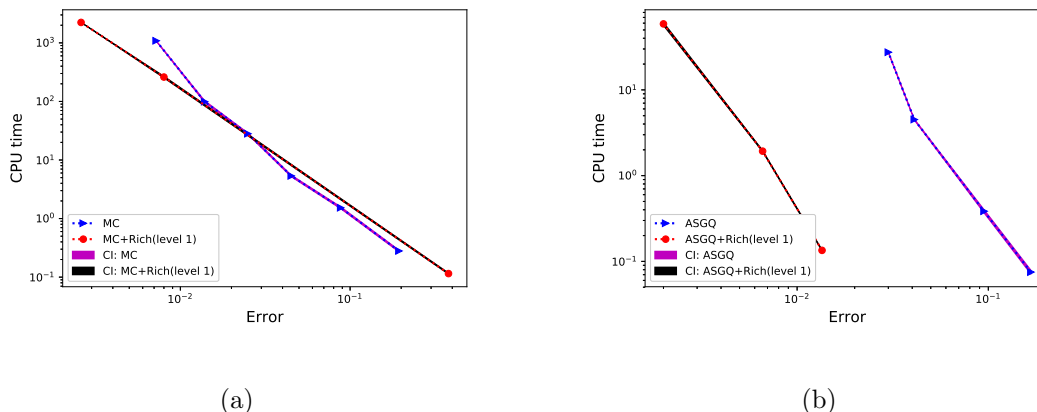


Figure 5.1: Digital option under GBM: Comparing the numerical complexity of the different methods with the different configurations in terms of the level of Richardson extrapolation. CI refers to 95% confidence intervals. a) MC methods. b) ASGQ methods combined with numerical smoothing.

We compare these optimal configurations for each method in Figure 5.2, and we show that ASGQ combined with numerical smoothing outperforms MC in terms of numerical complexity. In particular, to achieve a total relative error around 0.7%, ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation requires approximately 0.7% of the work of MC combined with level 1 of Richardson extrapolation.

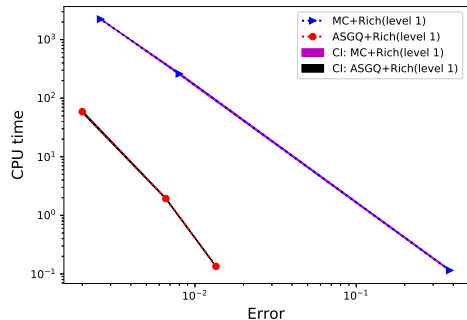


Figure 5.2: Digital option under GBM: Computational work comparison for the different methods with the best configurations, as concluded from Figure 5.1. To achieve a relative error below 1%, the ASGQ method combined with numerical smoothing and level 1 of Richardson extrapolation significantly outperforms the MC method combined with level 1 of the Richardson extrapolation. CI refers to 95% confidence intervals.

5.1.2 Single call option under the discretized GBM model

The second example that we test is the single call option under the discretized GBM model, with parameters: $T = 1$, $\sigma = 0.4$ and $S_0 = K = 100$. The exact reference value of this case is 15.8519.

Figure 5.3 shows a comparison of the numerical complexity for each method under the two scenarios. This Figure shows that, to achieve a relative error of 1%, level 1 of Richardson extrapolation is the optimal configuration for both methods.

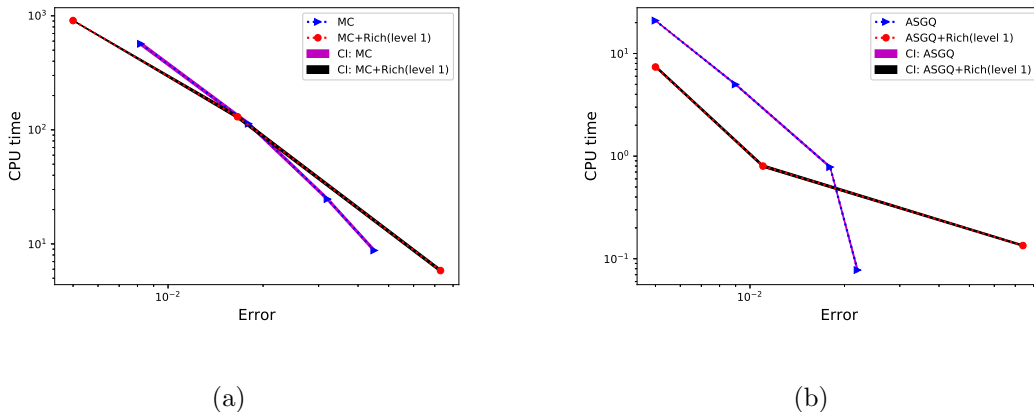


Figure 5.3: Call option under GBM: Comparing the numerical complexity of the different methods with the different configurations in terms of the level of Richardson extrapolation. CI refers to 95% confidence intervals. a) MC methods. b) ASGQ methods combined with numerical smoothing.

We compare these optimal configurations for each method in Figure 5.4, and we show that ASGQ combined with numerical smoothing outperforms MC in terms of numerical complexity. In particular, to achieve a total relative error around 0.5%, ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation requires approximately 0.8% of the work of MC combined

with level 1 of Richardson extrapolation.

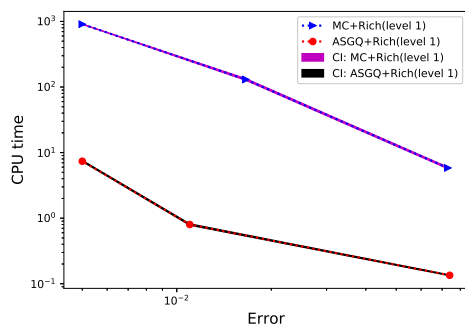


Figure 5.4: Call option under GBM: Computational work comparison for the different methods with the best configurations, as concluded from Figure 5.3. To achieve a relative error below 1%, the ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation significantly outperforms the MC method combined with level 1 of Richardson extrapolation. CI refers to 95% confidence intervals.

5.1.3 Basket call option under discretized GBM model

The third example that we consider under the GBM model is the multi-dimensional basket call option. We consider now the four dimensional basket call option with parameters: $\sigma_{1,2,3,4} = 0.4$, $\rho = 0.3$, $T = 1$, $r = 0$, $S_0^{1,2,3,4} = K = 100$, and $c_{1,2,3,4} = 1/4$. The reference value for those parameters is 11.04. Our experiments show that Richardson extrapolation did not improve the performance of both considered methods, and figure 5.5 shows that ASGQ combined with numerical smoothing requires approximately 7% of the work of MC to achieve a total relative error of around 0.8%.

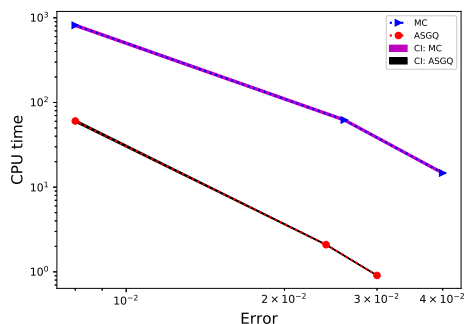


Figure 5.5: 4-dimensional basket call option under GBM: Computational work comparison for the different methods. To achieve a relative error below 1%, ASGQ combined with numerical smoothing significantly outperforms the MC method. CI refers to 95% confidence intervals.

5.2 Options under the discretized Heston model

In this section, we consider testing options under the discretized Heston model [21, 7, 23, 2] whose dynamics are given by

$$(5.1) \quad \begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^S = \mu S_t dt + \rho \sqrt{v_t} S_t dW_t^v + \sqrt{1 - \rho^2} \sqrt{v_t} S_t dW_t \\ dv_t &= \kappa(\theta - v_t) dt + \xi \sqrt{v_t} dW_t^v, \end{aligned}$$

where S_t is the price of the asset; v_t is 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 ; and ξ is the volatility of the volatility.

Given that the SDE for the asset path is now dependent upon the solution of the volatility SDE in (5.1), we must simulate the volatility process first and then use it to simulate the asset path. Therefore, a numerical approximation (discretization) is needed to obtain both paths. To this end, many simulation schemes have been proposed in the literature. In Appendix C, we give an overview of the most popular methods in this context. These methods mainly differ in the way they simulate the volatility process to ensure its positiveness.

Note that we are using ASGQ, which is very sensitive to the smoothness of the integrand. We found numerically (see section C.4) that using a non-smooth transformation to ensure the positiveness of the volatility process deteriorates the performance of the ASGQ. We suggest an alternative way in this work, to guarantee the positiveness of the volatility process, to simulate it as the sum of Ornstein-Uhlenbeck (OU) or Bessel processes as suggested in Appendix C.3.

In the literature [2, 25, 1], 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 differences rates, which is an important feature for an optimal performance of ASGQ (see Appendix C.4.2 for more details), and ii) the behavior of the weak error in order to apply the Richardson extrapolation when it is needed. We refer to Appendices C.4.1 and C.4.2 for more details on the comparison of the different schemes, which suggest that the Heston OU-based scheme, defined in Section C.3, gives the best results according to our criteria. Therefore, in our numerical experiments, we use this scheme with the ASGQ method. For the MC method, we use the full truncation scheme, as explained in Section C.1.

Remark 5.2. Although we compare different simulation schemes for the Heston model, to ensure a sub-optimal performance of the ASGQ combined with numerical smoothing, we emphasize that we do not claim that the chosen scheme is the optimal scheme for all parameters constellations, and a more thorough study on this topic is left as a future work. In this work, we instead aim at highlighting the advantage of our numerical smoothing idea.

Remark 5.3. Although we only illustrate the numerical results for one set of parameters for the Heston model, we note that we obtained almost similar results for many other cases.

5.2.1 Digital option under the discretized Heston model

We consider the digital option under the discretized Heston model with parameters: $S_0 = K = 100$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = 0.0025$. The reference solution is 0.5145.

Figure 5.6 shows a comparison of the numerical complexity for each method under the two different scenarios. From this figure, we can conclude that, to achieve a relative error of 1%, level

1 of Richardson extrapolation is the optimal configuration for the ASGQ method, and without Richardson extrapolation is the optimal configuration for the MC method.

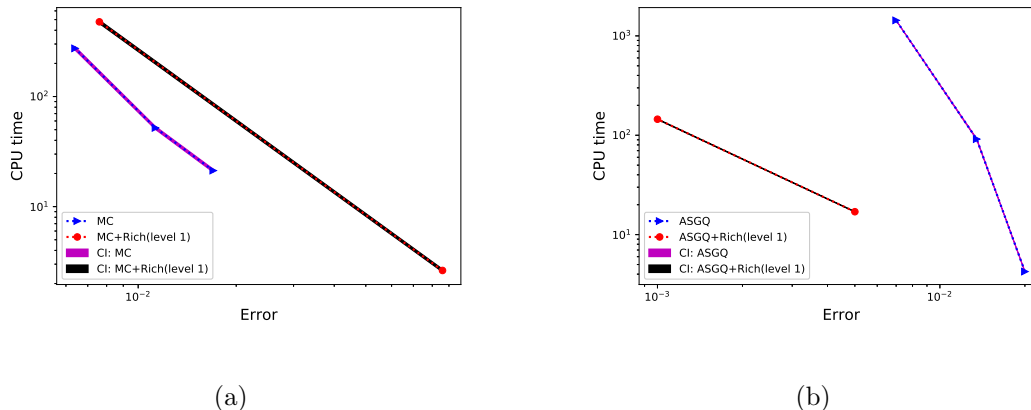


Figure 5.6: Digital option under Heston: Comparing the numerical complexity of the different methods with the different configurations in terms of the level of Richardson extrapolation. CI refers to 95% confidence intervals. a) MC methods. b) ASGQ methods combined with numerical smoothing.

We compare these optimal configurations for each method in Figure 5.7, which shows that, to achieve a total relative error around 0.6%, ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation requires approximately 6% of the work of MC without Richardson extrapolation.

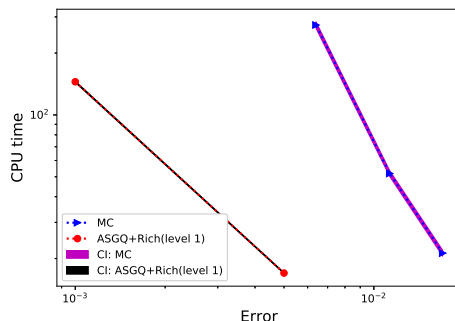


Figure 5.7: Digital option under Heston: Computational work comparison for the different methods with the best configurations, as concluded from Figure 5.6. To achieve a relative error below 1%, the ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation significantly outperforms the MC method without the Richardson extrapolation. CI refers to 95% confidence intervals.

5.2.2 European call option under the discretized Heston model

We consider the call option under the discretized Heston model with parameters: $S_0 = K = 100$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = 0.0025$. The reference solution is 6.332542.

Figure 5.8 shows a comparison of the numerical complexity for each method under the two different scenarios. From this Figure, we conclude that level 1 of Richardson extrapolation is the optimal configuration for both the MC and the ASGQ methods.

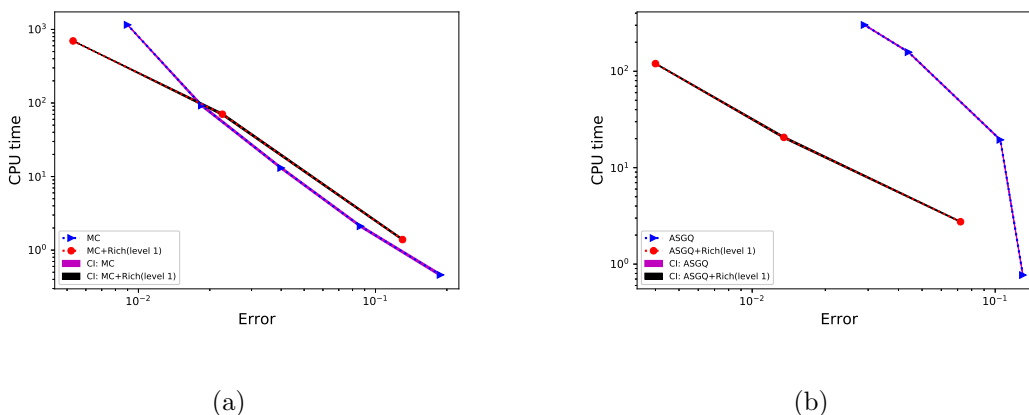


Figure 5.8: Call option under Heston: Comparing the numerical complexity of the different methods with the different configurations in terms of the level of Richardson extrapolation. CI refers to 95% confidence intervals. a) MC methods. b) ASGQ methods combined with numerical smoothing.

We compare these optimal configurations for each method in Figure 5.9, which shows that ASGQ outperforms MC in terms of numerical complexity. In particular, to achieve a total relative error around 0.5%, ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation requires approximately 17% of the work of MC combined with level 1 of Richardson extrapolation.

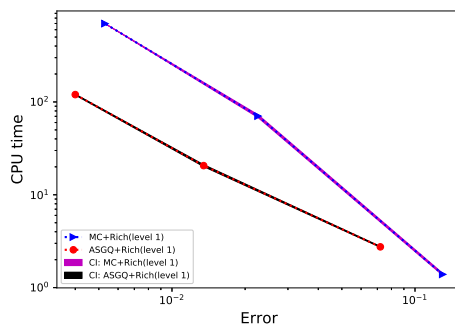


Figure 5.9: Call option under Heston: Computational work comparison for the different methods with the best configurations, as concluded from Figure 5.8. To achieve a relative error below 1%, the ASGQ combined with numerical smoothing and level 1 of Richardson extrapolation significantly outperforms the MC method combined with level 1 of Richardson extrapolation. CI refers to 95% confidence intervals.

6 Numerical experiments: Numerical smoothing with MLMC

In this section, we illustrate the advantage of combining the idea of numerical smoothing with MLMC in the context of quantitative finance. We consider two examples: i) the first one is for computing the price of a digital option under the GBM and Heston models (see Section 6.1), and ii) the second example is for approximating the density function of asset prices, for GBM and Heston models (see Section 6.2). The idea and results of Section 6.1 can be generalized to i) options having low regularity in the payoff function, or ii) computing distribution functions. On the other hand, examples in Section 6.2 can be generalized to: i) computing density functions, which involves the use of Dirac delta functions and whose expectation is hard to approximate using MLMC due to the infinite variance, and ii) computing Greeks for an option with a non-smooth payoff function.

In this section, we mainly compare the different examples: i) standard MLMC estimator (without smoothing), and ii) MLMC estimator combined with numerical smoothing (as explained in Section 2.2). We note that the parameters of the numerical smoothing, $(\text{TOL}_{\text{Newton}}, N_q)$, are chosen heuristically to solve the optimization problem (4.11).

6.1 MLMC for digital options

In this section, we illustrate the advantage of combining our numerical smoothing idea with the MLMC method to compute option prices for non-smooth payoff function. For illustration, we consider the price of the digital option, *i.e.*, we want to approximate (K is the strike)

$$(6.1) \quad \mathbb{E}[g(X)] = \mathbb{E}[\mathbf{1}_{X>K}].$$

When comparing the different methods, we use the Euler-Maruyama scheme for the GBM example, and for the Heston example we use i) the full truncation scheme (we refer to it by the FT scheme) (explained in Section C.1), and ii) the Heston OU-based scheme (we refer to it by the OU scheme) (explained in Section C.3).

6.1.1 Results for the digital option under the GBM model

As an illustration, we choose the digital option under the GBM model with parameters: $S_0 = K = 100$, $T = 1$, $r = 0$, and $\sigma = 0.2$. We summarize the obtained results for approximating the price of the option in Table 6.1 and more details are illustrated by Figures 6.1, 6.2, and 6.3. From these figures and Table 6.1, we can see two main results:

1. A significant reduction of the kurtosis at the finest levels, κ_L , of the MLMC algorithm when using numerical smoothing. In fact, the kurtosis is reduced by a factor of 236 (compare the bottom right plots in Figures 6.2 and 6.3). We stress that this is an important improvement for the robustness of the MLMC estimator, as explained in Section 3.2 and in [13].
2. Numerical smoothing significantly reduces the variance of the coupled levels in MLMC and improves the strong convergence rate, β , from $\beta = 1/2$ to $\beta = 1$ (compare top left plots in Figures 6.2 and 6.3), resulting in a reduction in the order of MLMC numerical complexity from $\mathcal{O}(\text{TOL}^{-2.5})$ to $\mathcal{O}(\text{TOL}^{-2}(\log(\text{TOL}))^2)$ (see Figure 6.1). From Figure 6.1, we see also that MLMC combined with smoothing significantly outperforms standard MLMC in terms of computational work.

| Method | κ_L | α | β | γ | Numerical complexity |
|-------------------------------|------------|----------|---------|----------|--|
| MLMC without smoothing | 709 | 1 | 1/2 | 1 | $\mathcal{O}(\text{TOL}^{-2.5})$ |
| MLMC with numerical smoothing | 3 | 1 | 1 | 1 | $\mathcal{O}(\text{TOL}^{-2}(\log(\text{TOL}))^2)$ |

Table 6.1: Digital option under GBM: Summary of the MLMC numerical results. κ_L is the kurtosis at the finest level of MLMC, and (α, β, γ) are weak, strong and work rates, respectively. TOL is the user-selected MLMC tolerance. These results correspond to Figures 6.1, 6.2, and 6.3, respectively.

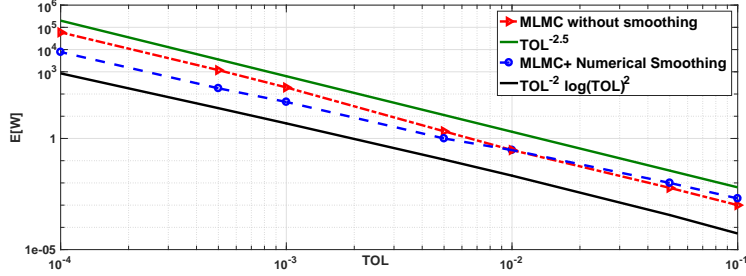


Figure 6.1: Digital option under GBM: Comparison of the numerical complexity (expected work, $E[W]$, vs tolerance, TOL, in a log-log scale) of the different methods i) standard MLMC, ii) MLMC with numerical smoothing. MLMC combined with numerical smoothing outperforms MLMC without smoothing, and also achieves a better numerical complexity rate that is $\mathcal{O}(\text{TOL}^{-2} \log(\text{TOL})^2)$.

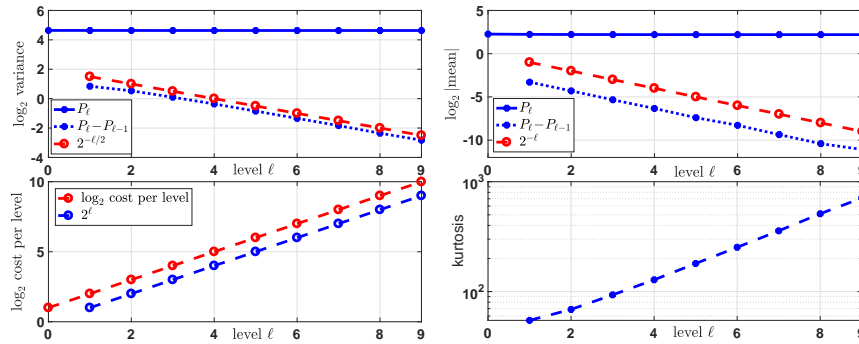


Figure 6.2: Digital option under GBM: Convergence plots for MLMC without smoothing, combined with Euler-Maruyama discretization.

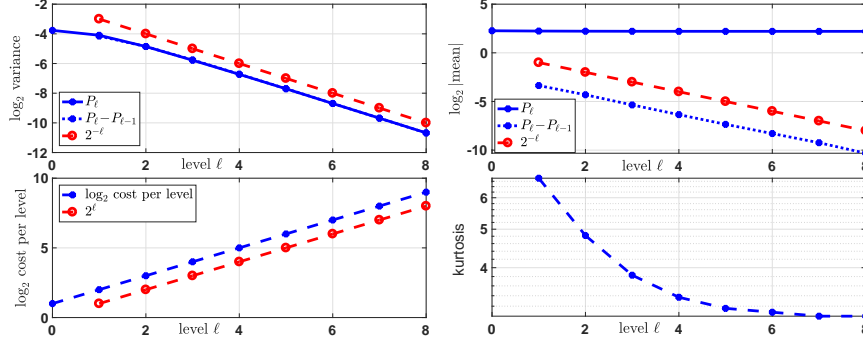


Figure 6.3: Digital option under GBM: Convergence plots for MLMC with numerical smoothing ($\text{TOL}_{\text{Newton}} = 10^{-3}$, $N_q = 8$), combined with Euler-Maruyama discretization.

Remark 6.1. We note that observing a decaying variance of P_ℓ in the top left plot in Figure 6.3 is expected since we use a Brownian bridge for path construction, and the integrand only depends on the terminal value of the Brownian bridge, which has a variance scale of order Δt . This implies that, for the particular case of digital option under the GBM dynamics, we expect the numerical complexity of the MC method with smoothing to be of order $\mathcal{O}(\text{TOL}^{-2})$. Furthermore, we stress that this observation only holds for the the GBM dynamics, which we use it here as a didactic example, and our main interest is to test our approach for dynamics where discretization of the asset price is needed, such as the Heston dynamics, presented in the following Section.

Remark 6.2. We also emphasize that our approach can be extended in a straightforward manner to any kind of dynamics, since it is based on the numerical smoothing idea. In the following Section, we show the advantage of our approach for the Heston model where discretization of the asset price is needed.

6.1.2 Digital option under the Heston model

As an illustration, we choose the digital option under the Heston model with parameters: $T = 1$, $S_0 = K = 100$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = 0.0025$. We summarize the obtained results for approximating the price in Table 6.2 and more details are illustrated by Figures 6.4, 6.5, 6.6, and 6.7. From these figures and Table 6.2, we can make the following observations:

1. The significant reduction of the kurtosis at the finest levels, κ_L , of MLMC when using numerical smoothing (using both OU-based scheme or FT scheme). In fact, the kurtosis is reduced by a factor greater than 27 (compare the bottom right plots in Figures 6.5, 6.6, and 6.7). We stress that this is an important improvement for the robustness of the MLMC estimator, as explained in Section 3.2 and in [13].
2. Numerical smoothing (with OU-based or FT schemes) reduces significantly the variance of coupled levels in MLMC, and improves the strong rate from $\beta = 1/2$ to $\beta = 1$ (compare top left plots in Figures 6.5, 6.6, and 6.7), resulting in reducing the computational work, and the order of MLMC numerical complexity from $\mathcal{O}(\text{TOL}^{-2.5})$ to $\mathcal{O}(\text{TOL}^{-2}(\log(\text{TOL}))^2)$ (see Figure 6.4).

| Method | κ_L | α | β | γ | Numerical complexity |
|---|------------|----------|---------|----------|---|
| MLMC without smoothing + FT scheme | 245 | 1 | 1/2 | 1 | $\mathcal{O}(\text{TOL}^{-2.5})$ |
| MLMC with numerical smoothing + OU-based scheme | 7 | 1 | 1 | 1 | $\mathcal{O}(\text{TOL}^{-2} \log(\text{TOL})^2)$ |
| MLMC with numerical smoothing+ FT scheme | 9 | 1 | 1 | 1 | $\mathcal{O}(\text{TOL}^{-2} \log(\text{TOL})^2)$ |

Table 6.2: Digital option under Heston: Summary of the MLMC numerical results. κ_L is the kurtosis at the finest level of MLMC, (α, β, γ) are weak, strong and work rates, respectively. TOL is the user-selected MLMC tolerance. These results correspond to Figures 6.4, 6.5, 6.6, and 6.7.

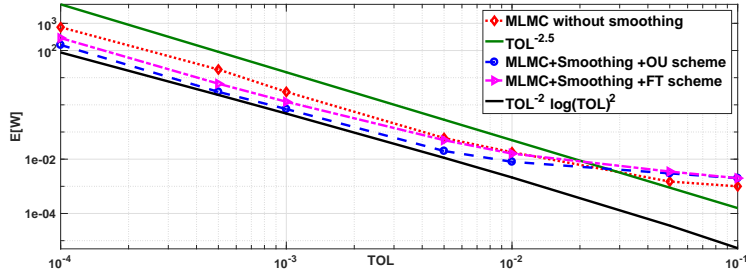


Figure 6.4: Digital option under Heston: Comparison of the numerical complexity (expected work, $E[W]$, vs tolerance, TOL, in a log-log scale) of the different methods i) standard MLMC (based on the FT scheme), ii) MLMC with smoothing (based on OU scheme), and iii) MLMC with smoothing (based on the FT scheme). MLMC combined with numerical smoothing (OU or FT schemes) outperforms standard MLMC, and also achieves a better numerical complexity rate.

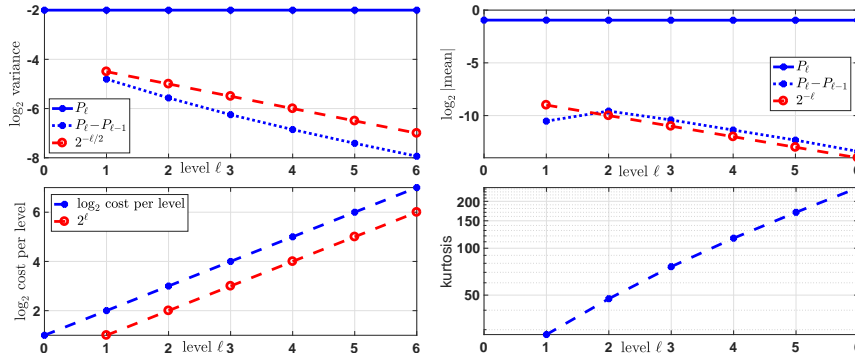


Figure 6.5: Digital option under Heston: Convergence plots for MLMC without smoothing, combined with the FT scheme (explained in Section C.1).

Remark 6.3. Although we just illustrated the benefit of our approach when combined with MLMC for computing the digital option price under the GBM (Section 6.1.1) and Heston dynamics (Section 6.1.2), we emphasize that it can be easily extended to any kind of model dynamics and to any low regular observable, g . For instance, this idea can be applied to approximate distribution functions involving the heavy-side function as the observable g .

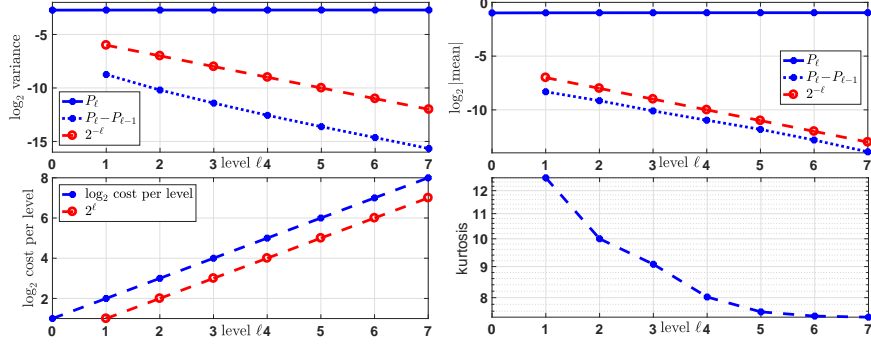


Figure 6.6: Digital option under Heston: Convergence plots for MLMC with numerical smoothing ($\text{TOL}_{\text{Newton}} = 10^{-3}$, $N_q = 32$), combined with the OU scheme (explained in Section C.3).

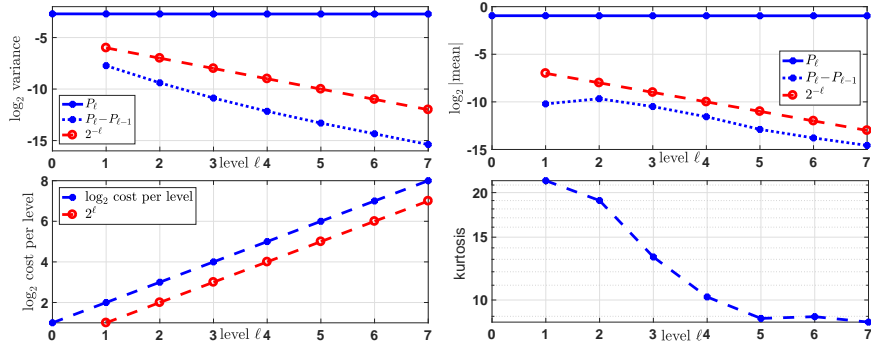


Figure 6.7: Digital option under Heston: Convergence plots for MLMC with numerical smoothing ($\text{TOL}_{\text{Newton}} = 10^{-3}$, $N_q = 32$), combined with the FT scheme (explained in Section C.1).

6.2 MLMC for approximating densities and Greeks

In this section, we explore the idea of combining the numerical smoothing with the MLMC method to compute density functions and Greeks for non-smooth payoff functions. In this context of density estimation, the numerical smoothing procedure combines root finding algorithm, to determine the location of the discontinuity, with a conditional expectation, computed exactly, with respect to the Brownian bridge. We recall 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 , which is given by

$$\rho_X(u) = \mathbb{E}[\delta(X - u)],$$

where δ is the Dirac delta function.

We can easily show, by conditioning with respect to the Brownian bridge, that

$$(6.2) \quad \begin{aligned} \rho_X(u) = \mathbb{E}[\delta(X - u)] &= \frac{1}{\sqrt{2\pi}} \mathbb{E} \left[\exp \left(- (y^*(u))^2 / 2 \right) \frac{dy^*}{dx}(u) \right] \\ &\approx \frac{1}{\sqrt{2\pi}} \mathbb{E} \left[\exp \left(- (\bar{y}^*(u))^2 / 2 \right) \frac{d\bar{y}^*}{dx}(u) \right], \end{aligned}$$

where $y^*(u)$ is the exact discontinuity location, and \bar{y} is the approximated discontinuity location, obtained by solving numerically: $X(T; \bar{y}^*(u), \mathbf{z}_{-1}) = u$, where \mathbf{z} is $N - 1$ ($2N - 1$) Gaussian random vector used for Brownian bridge construction for the GBM dynamics (Heston dynamics), with N being the number of time steps.

Remark 6.4. Formula (6.2) can be easily generalized to the multidimensional case, with the difference that one needs to perform a root finding procedure in the d -dimensional space characterized by the coarsest factor in each dimension. In fact, it is easy to show, for $\mathbf{u} \in \mathbb{R}^d$,¹⁴

$$(6.3) \quad \begin{aligned} \rho_{\mathbf{X}}(\mathbf{u}) &= \mathbb{E}[\delta(\mathbf{X} - \mathbf{u})] = \mathbb{E}[\rho_d(\mathbf{y}^*(\mathbf{u})) \det(\mathbf{J}(\mathbf{u}))] \\ &\approx \mathbb{E}[\rho_d(\bar{\mathbf{y}}^*(\mathbf{u})) \det(\bar{\mathbf{J}}(\mathbf{u}))], \end{aligned}$$

where \mathbf{J} is the Jacobian matrix, with $\mathbf{J}_{ij} = \frac{\partial y_i^*}{\partial x_j}$, $\rho_d(\cdot)$ is the multivariate Gaussian density, $\mathbf{y}^*(\mathbf{u})$ is the exact discontinuity location, and $\bar{\mathbf{y}}^*$ is the approximated discontinuity location, obtained by solving numerically: $\mathbf{X}(T; \bar{\mathbf{y}}^*(\mathbf{u}), \mathbf{z}_{-1}) = \mathbf{u}$, where \mathbf{z} is the Gaussian random vector used for Brownian bridges construction.

We should emphasize that the only error in our approach corresponds to the root finding procedure, and, contrary to approach based on kernel density and parametric regularization (see Remark 6.6), does not depend exponentially on the dimension of the problem.

6.2.1 Approximating density under the GBM model

As an illustration, we choose to compute the density ρ_X at $u = 1$ such that X is a GBM with parameters: $S_0, T = 1, r = 0$, and $\sigma = 0.2$. In this case, as a reference solution, we know that $X(T)$ is lognormally distributed with parameters $r - \sigma^2/2$ and σ .

In Table 6.3, we summarize the obtained results for estimating the density, ρ_X at $u = 1$, using MLMC combined with numerical smoothing, where X follows the GBM dynamics. In Figure 6.9, we show the detailed convergence results for the MLMC estimator combined with the numerical smoothing. From this Figure, we can verify that we obtain a strong convergence rate of order one (see top left plot in Figure 6.9), which results in a numerical complexity of the MLMC estimator to be of order $\mathcal{O}\left(\text{TOL}^{-2} (\log(\text{TOL}))^2\right)$, as confirmed by figure 6.8.

| Method | κ_L | α | β | γ | Numerical complexity |
|---------------------------|------------|----------|---------|----------|--|
| GBM + numerical smoothing | 5 | 1 | 1 | 1 | $\mathcal{O}\left(\text{TOL}^{-2} (\log(\text{TOL}))^2\right)$ |

Table 6.3: Density of GBM: Summary of the MLMC numerical results observed for computing the density ρ_X at $u = 1$, where X follows the GBM dynamics. κ_L is the kurtosis at the finest levels of MLMC with $\Delta t_L = T \cdot 2^{-8}$, and (α, β, γ) are weak, strong and work rates, respectively. TOL is the user-selected MLMC tolerance. These results correspond to Figures 6.8 and 6.9.

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. In the following Section, we show the advantage of our approach for the Heston model where discretization of the asset price is indeed needed.

¹⁴For a matrix \mathcal{A} , we denote by $\det(\mathcal{A})$ its determinant.

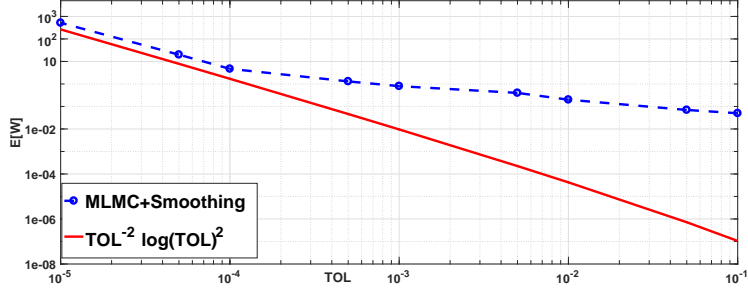


Figure 6.8: Density of GBM: Numerical complexity of MLMC with numerical smoothing for computing the density ρ_X at $u = 1$, where X follows the GBM dynamics.

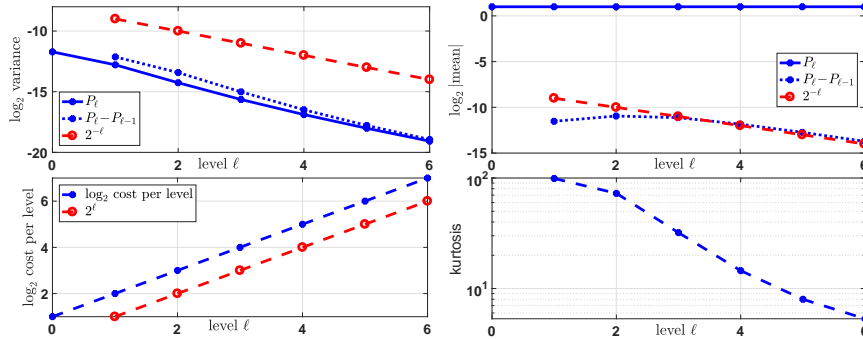


Figure 6.9: Density of GBM: Convergence plots for MLMC with numerical smoothing ($\text{TOL}_{\text{Newton}} = 10^{-2}$), for computing the density ρ_X at $u = 1$, where X follows the GBM dynamics. Remark 6.1 holds in this particular example as well.

6.2.2 Approximating density under the Heston model

As an illustration, we choose to compute the density ρ_X at $u = 1$ such that X is a Heston asset with parameters: $S_0 = 1$, $v_0 = 0.04$, $\mu = 0$, $\rho = -0.9$, $\kappa = 1$, $\xi = 0.1$, $\theta = 0.0025$. In this case, as a reference solution, we obtain the density by applying the Fractional Fourier Transform to the characteristic function, which is known for this model. In Table 6.4, we summarize the obtained results, using MLMC combined with numerical smoothing. In Figure 6.11, we show the detailed convergence results for the MLMC estimator combined with the numerical smoothing, and using the OU scheme (explained in Section C.3). From this Figure, we can verify that we obtain a strong convergence rate of order one (see top left plot in Figure 6.11), which results in a numerical complexity of the MLMC estimator of order $\mathcal{O}\left(\text{TOL}^{-2} (\log(\text{TOL}))^2\right)$, as confirmed by figure 6.10).

Remark 6.5. Although we have only illustrated the benefit of our approach when combined with MLMC for computing the density of the asset price under the GBM (Section 6.2.1) and Heston dynamics (Section 6.2.2), we emphasize that our approach can be easily extended to any kind of model dynamics. Furthermore, our approach can be easily extended to computing financial Greeks.

| Method | κ_L | α | β | γ | Numerical complexity |
|--|------------|----------|---------|----------|--|
| Heston + OU scheme + numerical smoothing | 8 | 1 | 1 | 1 | $\mathcal{O}\left(\text{TOL}^{-2} (\log(\text{TOL}))^2\right)$ |

Table 6.4: Density of Heston: Summary of the MLMC numerical results observed for computing the density ρ_X at $u = 1$, where X follows the Heston dynamics. κ_L is the kurtosis at the finest levels of MLMC with $\Delta t_L = T \cdot 2^{-8}$, (α, β, γ) are weak, strong and work rates, respectively. TOL is the user-selected MLMC tolerance. These results correspond to Figures 6.10 and 6.11.

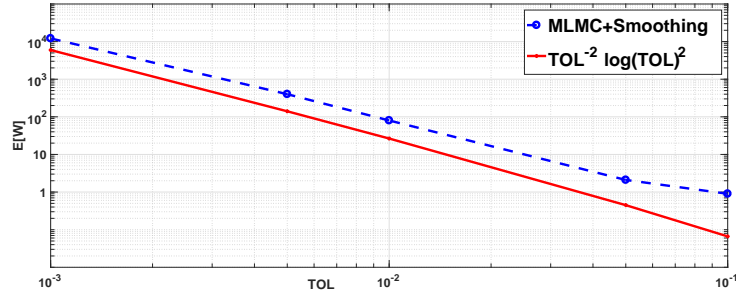


Figure 6.10: Density of Heston: Numerical complexity (expected work, $E[W]$, vs tolerance, TOL) of MLMC with numerical smoothing and combined with the OU scheme (explained in Section C.3) for computing for computing the density ρ_X at $u = 1$, where X follows the Heston dynamics.

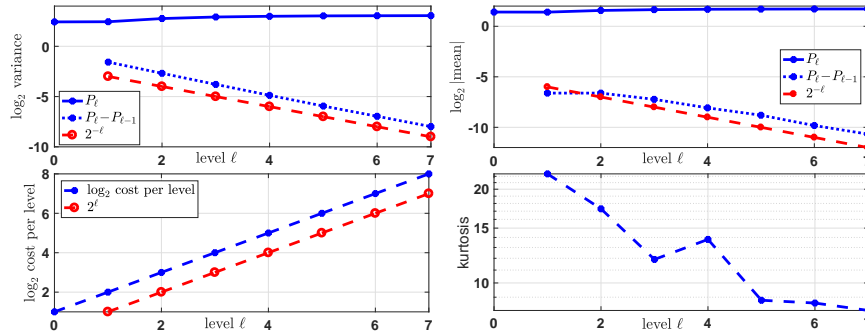


Figure 6.11: Density of Heston: Convergence plots for MLMC with numerical smoothing ($\text{TOL}_{\text{Newton}} = 10^{-2}$) combined with the OU scheme (explained in Section C.3), for computing the density ρ_X at $u = 1$, where X follows the Heston dynamics.

Remark 6.6. We note that as an alternative to our approach, one may use kernel density estimation techniques or a similar approach to that in [16]. However, this class of approaches has a pointwise error that increases exponentially with respect to the dimension of the state vector \mathbf{X} (or a vector valued function that depends on the density of \mathbf{X}). For instance, given a discretization error that should be similar to our approach, and for a d -dimensional problem, a kernel density estimator with a bandwidth matrix, $\mathcal{H} = \text{diag}(h, \dots, h)$, the MSE is of order $c_1 M^{-1} h^{-d} + c_2 h^4$, where M is the number of samples, and c_1 and c_2 are constants. On the other hand, thanks to the exact conditional expectation with respect to the Brownian bridge, the error of our approach is only restricted to the

error of the root-finding algorithm for finding the location of the discontinuity (see Remark 6.4).

Acknowledgments C. Bayer gratefully acknowledges support from the German Research Foundation (DFG), via the Cluster of Excellence MATH+ (project AA4-2) and the individual grant BA5484/1. This work was supported by the KAUST Office of Sponsored Research (OSR) under Award No. URF/1/2584-01-01 and the Alexander von Humboldt Foundation. C. Ben Hammouda and R. Tempone are members of the KAUST SRI Center for Uncertainty Quantification in Computational Science and Engineering.

References Cited

- [1] 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.
- [2] Leif BG Andersen. Efficient simulation of the Heston stochastic volatility model. *Available at SSRN 946405*, 2007.
- [3] Leif BG Andersen and Rupert Brotherton-Ratcliffe. Extended libor market models with stochastic volatility. *Journal of Computational Finance*, 9(1), 2005.
- [4] Rainer Avikainen. On irregular functionals of SDEs and the Euler scheme. *Finance and Stochastics*, 13(3):381–401, 2009.
- [5] Christian Bayer, Chiheb Ben Hammouda, and Raul Tempone. Hierarchical adaptive sparse grids for option pricing under the rough Bergomi model. *arXiv preprint arXiv:1812.08533*, 2018.
- [6] Christian Bayer, Markus Siebenmorgen, and Raúl 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] K Andrew Cliffe, Mike B Giles, Robert Scheichl, and Aretha L Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3, 2011.
- [10] Darrell Duffie, Peter Glynn, et al. Efficient Monte Carlo simulation of security prices. *The Annals of Applied Probability*, 5(4):897–905, 1995.
- [11] Michael B Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 343–358. Springer, 2008.
- [12] Michael B Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [13] Michael B Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

- [14] Michael B Giles, Kristian Debrabant, and Andreas Rößler. Numerical analysis of multilevel Monte Carlo path simulation using the Milstein discretisation. *arXiv preprint arXiv:1302.4676*, 2013.
- [15] 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.
- [16] 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.
- [17] Michael Griebel, Frances Kuo, and Ian Sloan. The smoothing effect of integration in \mathbb{R}^d and the ANOVA decomposition. *Mathematics of Computation*, 82(281):383–400, 2013.
- [18] Michael Griebel, Frances Kuo, and Ian Sloan. Note on the smoothing effect of integration in \mathbb{R}^d and the ANOVA decomposition. *Mathematics of Computation*, 86(306):1847–1854, 2017.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Monique Jeanblanc, Marc Yor, and Marc Chesney. *Mathematical methods for financial markets*. Springer Science & Business Media, 2009.
- [23] Christian Kahl and Peter Jäckel. Fast strong approximation Monte Carlo schemes for stochastic volatility models. *Quantitative Finance*, 6(6):513–536, 2006.
- [24] Sebastian Krumscheid and Fabio Nobile. Multilevel monte carlo approximation of functions. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1256–1293, 2018.
- [25] Roger Lord, Remmert Koekoek, and Dick Van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194, 2010.
- [26] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. Siam, 1992.
- [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.

A Details for the proof of Theorem 4.2 in Section 4.3

We consider a mollified version g_δ of g and the corresponding function H_δ^N (defined by replacing g with g_δ in (4.15)). Tacitly assuming that we can interchange integration and differentiation, we have

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

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

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

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

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

We realize that there is a potential problem looming in the inverse of the derivative w.r.t. y .¹⁵ Before we continue, let us introduce the following notation: for sequences of rrvs F_N, G_N we say that $F_N = \mathcal{O}(G_N)$ if there is a rrv C with finite moments of all orders such that for all N we have $|F_N| \leq C |G_N|$ a.s.

Assumption A.1. There are positive rrvs C_p with finite moments of all orders such that

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

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

Remark A.2. It is probably difficult to argue that a deterministic constant C may exist.

Assumption A.1 is natural, but now we need to make a much more serious assumption, which is probably difficult to verify in practice.

Assumption A.3. For any $p \in \mathbb{N}$ we have that

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

Lemma A.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 .

¹⁵Let us assume that $X_T^N(y, z^N) = \cos(y) + z_{n,k}$. Then (A.1) is generally not integrable.

Proof. First let us note that Assumption A.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 $\text{supp } \psi_{n,k}$ 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

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

we thus have

$$(A.3) \quad \frac{\partial X_T^N}{\partial z_{n,k}}(Z_{-1}, Z^N) = \sum_{\ell=2^{N-n+1}k}^{2^{N-n+1}(k+1)-1} \frac{\partial X_T^N}{\partial \Delta_\ell^N W} \frac{\partial \Delta_\ell^N W}{\partial Z_{n,k}} = 2^{N-n+1} 2^{-(N-n/2)} \mathcal{O}(1) = 2^{-n/2+1} \mathcal{O}(1). \quad \square$$

Lemma A.5. *In the same sense as in Lemma A.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 (A.3) 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

$$(A.4) \quad \frac{\partial^2 X_T^N}{\partial \Delta_\ell^N W \partial \Delta_j^N W} = \frac{\partial^2 X_T^N}{\partial X_{\ell+1}^N \partial X_{j+1}^N} b(X_\ell^N) b(X_j^N) + \mathbf{1}_{j < \ell} \frac{\partial X_T^N}{\partial X_\ell^N} b'(X_\ell^N) \frac{\partial X_\ell^N}{\partial X_{j+1}^N} b(X_j^N) = \mathcal{O}(1)$$

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

Remark A.6. Lemma A.4 and A.5 also hold (mutatis mutandis) for $z_{n,k} = y$ (with $n = 0$).

Proposition A.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

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

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 rdvs, 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 A.4 and Lemma A.5 together with Assumption A.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 A.8. *For any $p \in \mathbb{N}$ and indices n_1, \dots, n_p and k_1, \dots, k_p (satisfying $0 \leq k_j < 2^{n_j}$) 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,

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

By a similar argument as in (A.4) 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 (A.2) 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. □

B More details on the error and work discussion of ASGQ method

In this Section, we show that under certain conditions of the regularity parameters p and s , we can achieve, under the best scenario ($p, s \gg 1$), $\text{Work}_{\text{ASGQ}} = \mathcal{O}(\text{TOL}^{-1})$. In fact, using the method of Lagrange multipliers, we can show that

$$N_{\text{ASGQ}} \propto \Delta t^{\frac{p+s-ps}{p(ps+p+s)}}, \quad \text{and} \quad N_q \propto \Delta t^{\frac{p+s-ps}{s(ps+p+s)}},$$

and using the constraint in (4.6), we can easily show that given an error tolerance, TOL, we have $\Delta t = \mathcal{O}\left(\text{TOL}^{\frac{ps+p+s}{ps-p-s}}\right)$. Therefore, the optimal work $\text{Work}_{\text{ASGQ}}$ solution of (4.6) satisfies

$$\begin{aligned} \text{Work}_{\text{ASGQ}} &\propto N_{\text{ASGQ}} \times N_q \times \Delta t^{-1} \propto \Delta t^{-1} \Delta t^{\frac{p+s-ps}{s(ps+p+s)}} \Delta t^{\frac{p+s-ps}{p(ps+p+s)}} \\ &\propto \text{TOL}^{-1 - \frac{2(p+s)}{ps-p-s} - \frac{1}{p} - \frac{1}{s}} \\ &= \mathcal{O}(\text{TOL}^{-1}), \quad \text{since } p, s \gg 1. \end{aligned}$$

C Schemes to simulate the Heston dynamics

C.1 Fixed Euler scheme

The Forward Euler scheme can be used to simulate the Heston model. To avoid problems with negative values of the volatility process v_t in (5.1), many fixes have been introduced in the literature (see [25]). In Table C.1, we introduce f_1, f_2 , and f_3 , which, with different choices, implies different schemes. Applying forward Euler scheme to discretize (5.1) results in

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

where Z_s and Z_V are two correlated standard normal rdvs with correlation ρ .

| 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 | $ \widehat{V}_t $ | $ \widehat{V}_t $ | $ \widehat{V}_t $ |

Table C.1: Different variants for Forward Euler scheme for Heston model. $\widehat{V}_t^+ = \max(0, \widehat{V}_t)$.

[25] suggest that the full truncation scheme is the optimal option in terms of weak error convergence. Therefore, we use this variant of the Forward Euler scheme.

C.2 Moment matching schemes

We consider two moment matching schemes that were suggested by Andersen and Brotherton-Ratcliffe [3] (we call it the ABR scheme) and by Anderson in [2] (we choose the QE method that was reported to have the optimal results).

C.2.1 The ABR method

The ABR method [3] assumes that the variance v_t is locally lognormal, and the parameters are determined such that the first two moments of the discretization coincide with the theoretical moments, that is

$$(C.1) \quad \begin{aligned} \widehat{V}(t + \Delta t) &= \left(e^{-\kappa \Delta t} \widehat{V}(t) + (1 - e^{-\kappa \Delta t}) \theta \right) e^{-\frac{1}{2} \Gamma(t)^2 \Delta t + \Gamma(t) \Delta W_v(t)} \\ \Gamma(t)^2 &= \Delta t^{-1} \log \left(1 + \frac{\frac{1}{2} \xi^2 \kappa^{-1} \widehat{V}(t) (1 - e^{-2\kappa \Delta t})}{\left(e^{-\kappa \Delta t} \widehat{V}(t) + (1 - e^{-\kappa \Delta t}) \theta \right)^2} \right). \end{aligned}$$

As reported in [25], the scheme, being very easy to implement, is more effective than many of the Euler variants presented in Section C.1; however, it was reported that it has a non-robust weak error behavior with respect to the parameters of the model.

C.2.2 The QE method

Using the idea of moment matching, Anderson in [2] suggested a similar discretization to (C.1) but takes into account the shape of the Heston density function. As reported by [2], the QE scheme has a negligible bias and a better weak error behavior, at the cost of a more complex implementation than the ABR scheme in Section C.2.1. Furthermore, the QE algorithm uses two different distributions to model the volatility depending on the initial value of the volatility. We refer to [2] for more details about the QE scheme. We note that we obtained similar numerical behavior for both ABR and QE schemes; therefore, we chose to illustrate the results obtained by the ABR scheme.

C.3 The Heston OU-based scheme

It is well known that any 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

$$(C.2) \quad dX_t^i = \alpha X_t^i dt + \beta dW_t^i,$$

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

We also define the process Y_t as

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

Then, using the fact that

$$d(X_t^i)^2 = 2X_t^i dX_t^i + 2d\langle X^i \rangle_t = (2\alpha (X_t^i)^2 + \beta^2) dt + 2\beta X_t^i dW_t^i,$$

we can write, using the independence of the Brownian motions,

$$(C.3) \quad dY_t = d\left(\sum_{i=1}^n (X_t^i)^2 \right) = \sum_{i=1}^n d(X_t^i)^2 = (2\alpha Y_t + n\beta^2) dt + 2\beta \sum_{i=1}^n X_t^i dW_t^i.$$

Furthermore, the process, $Z_t = \int_0^t \sum_{i=1}^n X_u^i dW_u^i$, is a martingale with quadratic variation

$$\langle Z \rangle_t = \int_0^t \sum_{i=1}^n (X_u^i)^2 du = \int_0^t Y_u du.$$

Consequently, by Lévy's characterization theorem, the process, $\widetilde{W}_t = \int_0^t \frac{1}{\sqrt{Y_u}} \sum_{i=1}^n X_u^i dW_u^i$, is a Brownian motion.

Finally, we have

$$\begin{aligned} dY_t &= (2\alpha Y_t + n\beta^2) dt + 2\beta\sqrt{Y_t}d\widetilde{W}_t \\ (C.4) \quad &= \kappa(\theta - Y_t) dt + \xi\sqrt{Y_t}dW_t, \end{aligned}$$

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

Equations (C.2), (C.3), and (C.4) show that in order to simulate the CIR process Y_t given by (C.4), we can simulate the OU process \mathbf{X} , with dynamics given by (C.2) 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 can simulate the volatility of the Heston model using a sum of OU processes.

Remark C.1. The previous derivation 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 [22] for details). A second way can be used for generalizing the scheme for any non integer, n^* , by writing $n^* = n + p$, $p \in (0, 1)$, and then we can compute, for any observable g , $E[g(X_{n^*})]$ as

$$E[g(X_{n^*})] \approx (1 - p)E[g(X_n)] + pE[g(X_{n+1})].$$

C.4 On the choice of the simulation scheme of the Heston model

In this section, we determine the optimal scheme for simulating the Heston model defined in (5.1). In our setting, an optimal scheme is characterized by two properties: i) the behavior of mixed rates convergence (see Section C.4.1), which is an important requirement for an optimal performance of ASGQ, and ii) the behavior of the weak error (see Section C.4.2) in order to apply Richardson extrapolation when it is needed.

Although we tested many parameters sets, with consistent numerical observations, for illustration purposes, we only show results for one set of parameters given in table C.2. This set corresponds to $n = 1$, with n being the number of OU processes used in the Heston OU-based scheme, defined in Section C.3. Furthermore, this set does not satisfy the Feller condition, that is $4\kappa\theta > \xi^2$.

| Parameters | Reference solution |
|---|--------------------|
| $S_0 = K = 100, v_0 = 0.04, \mu = 0, \rho = -0.9, \kappa = 1, \xi = 0.1, \theta = \frac{\xi^2}{4\kappa} = 0.0025, (n = 1).$ | 6.332542 |

Table C.2: Reference solution using Premia with `cf_call_heston` method [21]. By n we refer to the number of OU processes for simulating the volatility process in our approach, as shown in Section C.3

C.4.1 Comparison in terms of mixed differences rates

As emphasized in [20], one important requirement to achieve the optimal performance of the ASGQ is to check the error convergence of first and mixed difference operators, as expressed by the error

contribution (C.5), which is a measure of how much the quadrature error would decrease once ΔQ_N^β has been added to the ASGQ estimator $Q_N^{\mathcal{I}}$ (defined in (3.1))

$$(C.5) \quad \Delta E_\beta = \left| Q_N^{\mathcal{I} \cup \{\beta\}} - Q_N^{\mathcal{I}} \right|.$$

An optimal behavior for ASGQ is ensured if: i) ΔE_β decreases exponentially fast with respect to β_i , and ii) ΔE_β has a product structure so that a faster error decay is observed for second differences, compared to corresponding first difference operators.

In this section, we compare the three approaches of simulating Heston dynamics, namely: i) the full truncation scheme (see Section C.1), ii) the ABR scheme (see Section C.2.1), and iii) the Heston OU-based scheme (see Section C.3). The comparison is done in terms of mixed differences convergence. In our numerical experiments, we only observe differences of mixed differences rates related to volatility coordinates, since we use schemes that only differ in the way they simulate the volatility process. Figure C.1 shows a comparison of first differences rates related to volatility coordinates for the different schemes. From this figure, we have: i) the full truncation scheme is the worst scheme, ii) the Heston OU-based scheme and both schemes based on moment matching (ABR and QE schemes defined in Section C.2) show a very good performance in terms of mixed rates convergence, *i.e.*, the error contribution, ΔE_β (defined in (C.5)) decreases exponentially fast with respect to β_i .

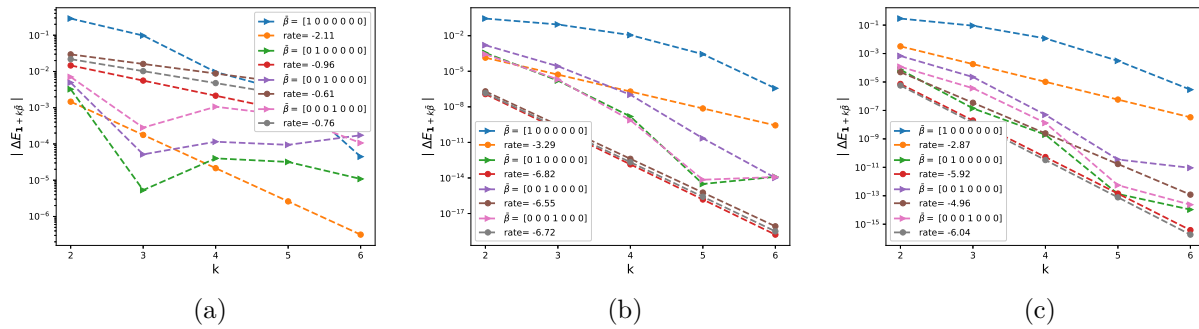
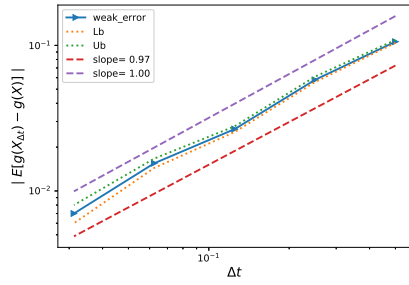


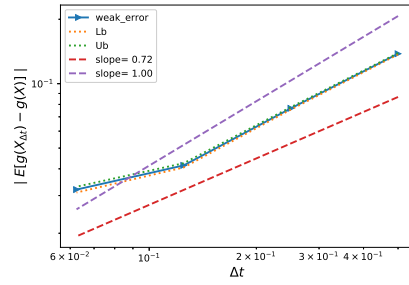
Figure C.1: The rate of error convergence of first order differences $|\Delta E_\beta|$, defined in (C.5), ($\beta = \mathbf{1} + k\bar{\beta}$) for the example of single call option under the Heston model, with parameters given by Set 1 in Table C.2, using $N = 4$ time steps. In this case, we only show the first four dimensions that are used for the volatility noise (mainly dW_v in (5.1)). (a) full truncation scheme as in Section C.1, (b) ABR scheme as in Section C.2.1, (c) Heston OU-based scheme as in Section C.3.

C.4.2 Comparison in terms of the weak error behavior

In this section, we compare i) the ABR scheme discussed in Section C.2.1, and ii) the Heston OU-based scheme discussed in Section C.3, in terms of the behavior of the weak convergence. We select schemes that have a weak error rate of of order one in the pre-asymptotic regime, to apply efficiently Richardson extrapolation, and thus, we can apply Richardson extrapolation in our proposed methods. Figure C.2 shows a comparison of the weak error rates for the different schemes. From this figure, we can check that the Heston OU-based scheme has a better weak convergence rate that is closer to 1 compared to the ABR scheme with a weak error rate of order 0.7.



(a)



(b)

Figure C.2: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 3.2, for the European call option under the discretized Heston model, for parameters in Table C.2. The upper and lower bounds are 95% confidence intervals. (a) Heston OU-based scheme, (b) ABR scheme.