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

1 Introduction

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

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

One can ignore the kinks and jumps, and apply directly a method for integration over $\mathbb{R}^d$. Despite the significant progress in SG methods [3] for high dimensional integration of smooth integrands, few works have been done to deal with cases involving integrands with kinks or jumps due to the decreasing performance of SG methods in the presence of kinks and jumps.

Some works [6, 2, 7, 8, 14] adressed similar kind of problems, characterized by the presence of kinks and jumps, but with much more emphasis on Quasi Monte Carlo (QMC). In [6, 7, 8], an analysis of the performance of Quasi Monte Carlo (QMC) and SG methods has been conducted, in the presence of kinks and jumps. In [6, 7], the authors studied the terms of the ANOVA decomposition of functions with kinks defined on d-dimensional Euclidean space $\mathbb{R}^d$, and showed that under some assumptions all but the the highest order ANOVA term of the 2^d ANOVA terms can be smooth for the case of an arithmetic Asian option with the Brownian bridge construction. Furthermore, [8] extended the work in [6, 7] from kinks to jumps for the case of an arithmetic average digital Asian option with the principal component analysis (PCA). The main findings in [6, 7] was obtained for an integrand of the form $f(\mathbf{x}) = \max(\phi(\mathbf{x}), 0)$ with ϕ being smooth. In fact, by assuming i) the d-dimensional function ϕ has a positive partial derivative with respect to x_j for some $j \in \{1, \ldots, d\}$, ii) certain growth conditions at infinity are satisfied, the authors showed that the ANOVA terms of f that do not depend on the variable x_j are smooth. We note that [6, 7, 8]

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

A recent work [14] adresses similar kind of problems using QMC. Being very much related to [2], the authors i) assume that the conditional expectation can be computed explicitly, by imposing very strong assumptions. ii) Secondly, they use PCA on the gradients to reduce the effective dimension. In our work, we do not make such strong assumptions, which is why we need numerical methods, more precisely root finding and the quadrature in the first direction.

Add more details here.

2 Problem formulation and Setting

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

2.1 Continuous time formulation

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

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

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

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

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

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

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

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

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

$$
Z_j = Z_v v_j + (\mathbb{P}_\perp \mathbf{Z})_j
$$
$$
(2.4) \qquad\qquad = Z_v v_j + (Z_v^\perp)_j.
$$

2

The first aim of the work is to determine the optimal direction $\mathbf{v}$. By optimal direction, we mean the direction that maximizes the smoothing effect, that is the one given by

$$(2.5) \qquad \sup_{\substack{\mathbf{v} \in \mathbb{R}^{d \times 1} \\ ||\mathbf{v}||=1}} \left(\mathrm{Var}\left[g(\widehat{X}_T) \right] \right),$$

where $\widehat{X}$ is defined in (2.9).

Maybe criterion given by (2.5) is not the right choice since decomposition (2.3) should leave the distribution, and hence the variance invariant. Therefore, maybe the true quantity to maximize is something like a variance of the component orthogonal to the kink. If it is the case then maybe we should re-write criterion (2.5).

Going back to the SDE (2.1), we have

$$(2.6) \qquad dX_i = a_i(X)dt + \sum_{j=1}^{d} b_{ij}(X)Z_j \frac{dt}{\sqrt{T}} + \sum_{j=1}^{d} b_{ij}(X)dB_j.$$

Using (2.4) implies

$$(2.7) \qquad dX_i = \left(a_i(X) + \sum_{j=1}^{d} b_{ij}(X)\frac{Z_v v_j}{\sqrt{T}} \right) dt + \left(\sum_{j=1}^{d} b_{ij}(X)\frac{(Z_v^{\perp})_j}{\sqrt{T}} \right) dt + \sum_{j=1}^{d} b_{ij}(X)dB_j.$$

2.1.1 First approach (Brutal)

Assumption 1: We assume that the first term in the right-hand side of (2.7) is the dominant term compared to the remaining terms in the sense of variance of g.

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

Proof of assumption 1. Let us assume from (2.7) that $a_i = 0$, $b_{ij} = \alpha = $ constant, and $g : \mathbb{R}^d \to Id_{\mathbb{R}}$, we try here to motivate Assumption 1.

Let us integrate (2.7) form 0 to T, then we have

$$X_T - X_0 = \left(\sqrt{T}\alpha \right) \left(\sum_{j=1}^{d} v_j \right) Z_v + \left(\sqrt{T}\alpha \right) \left(\sum_{j=1}^{d} v_j (Z_v)_j^{\perp} \right) + \alpha \sum_{j=1}^{d} \int_0^T dB_j$$

$$= \left(\sqrt{T}\alpha \right) \left(\sum_{j=1}^{d} v_j \right) Z_v + \left(\sqrt{T}\alpha \right) \left(\sum_{j=1}^{d} v_j (Z_v)_j^{\perp} \right) + \alpha \sum_{j=1}^{d} \left(B_j(T) - B_j(0) \right),$$

implying (if we neglect the correlation terms)

3

$$\text{Var}[X_T - X_0] \approx \left(\sqrt{T}\alpha\right)^2 \left(\sum_{j=1}^{d} v_j\right)^2 + \left(\sqrt{T}\alpha\right)^2 \underbrace{\text{Var}\left[\sum_{j=1}^{d} v_j(Z_v)_j^{\perp}\right]}_{=0} + \alpha^2 \sum_{j=1}^{d} \text{Var}\left[B_j(T) - B_j(0)\right],$$

$$\approx T\alpha^2 \left(\sum_{j=1}^{d} v_j\right)^2 + \alpha^2 \sum_{j=1}^{d} \underbrace{\text{Var}\left[B_j(T) - B_j(0)\right]}_{=0},$$

$$(2.8) \qquad \approx T\alpha^2 \left(\sum_{j=1}^{d} v_j\right)^2.$$

Therefore, we can conclude from (2.8) that the dominant term in the sense of global variance is coming from $\sum_{j=1}^{d} b_{ij}(X)\frac{Z_v v_j}{\sqrt{T}} dt$, which makes assumption 1 valid. $\qquad\square$

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

$$(2.9) \qquad \frac{d\widehat{X}_i}{dt} = a_i(\widehat{X}) + \sum_{j=1}^{d} b_{ij}(\widehat{X})\frac{Z_v v_j}{\sqrt{T}}, \quad 1 \le i \le d,$$

with mean

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

and second moment

$$\frac{d\text{E}\left[\widehat{X}_i^2\right]}{dt} = 2\text{E}\left[\widehat{X}_i \frac{d\widehat{X}_i}{dt}\right]$$

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

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

$$(2.10) \qquad \widehat{X}_{(Z_v)} = \widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v + \dots$$

Then, for the first moment we have

$$\frac{d\text{E}\left[\widehat{X}_i\right]}{dt} \approx \text{E}\left[a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v)\right] + \sum_{j=1}^{d} \text{E}\left[b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v)\frac{Z_v v_j}{\sqrt{T}}\right], \quad 1 \le i \le d,$$

$$\approx a_i(\widehat{X}_{(0)}) + \sum_{j=1}^{d} \left(\sum_{k=1}^{d} b'_{ij,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)}\right) \frac{\text{E}\left[Z_v^2\right] v_j}{\sqrt{T}}, \quad 1 \le i \le d,$$

$$(2.11) \qquad = a_i(\widehat{X}_{(0)}) + \sum_{j=1}^{d} \left(\sum_{k=1}^{d} b'_{ij,k}(\widehat{X}_{(0)})\widehat{X}'_{k,(0)}\right) \frac{v_j}{\sqrt{T}}, \quad 1 \le i \le d.$$

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

$$\frac{d\mathrm{E}\left[\widehat{X}_i^2\right]}{dt} \approx 2 \left(\mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)} Z_v\right) a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v)\right] + \sum_{j=1}^{d} \mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)} Z_v\right) b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v) \frac{Z_v v_j}{\sqrt{T}}\right] \right),$$

(2.12)

$$\approx 2 \left(\widehat{X}_{i,(0)} a_i(\widehat{X}_{(0)}) + \widehat{X}'_{i,(0)} \sum_{k=1}^{d} a'_{i,k}(\widehat{X}_{(0)}) \widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d} \widehat{X}'_{i,(0)} b_{ij}(\widehat{X}_{(0)}) + \widehat{X}_{i,(0)} \left(\sum_{k=1}^{d} \widehat{X}'_{k,(0)} b'_{ij,k}(\widehat{X}_{(0)}) \right) \right) \frac{v_j}{\sqrt{T}} \right).$$

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

$$\frac{d\mathrm{E}\left[\widehat{X}_i \widehat{X}_l\right]}{dt} = \mathrm{E}\left[\widehat{X}_i \frac{d\widehat{X}_l}{dt}\right] + \mathrm{E}\left[\widehat{X}_l \frac{d\widehat{X}_i}{dt}\right]$$

$$\approx \mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)} Z_v\right) a_l(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v)\right] + \sum_{j=1}^{d} \mathrm{E}\left[\left(\widehat{X}_{i,(0)} + \widehat{X}'_{i,(0)} Z_v\right) b_{lj}(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v) \frac{Z_v v_j}{\sqrt{T}}\right]$$

$$+ \mathrm{E}\left[\left(\widehat{X}_{l,(0)} + \widehat{X}'_{l,(0)} Z_v\right) a_i(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v)\right] + \sum_{j=1}^{d} \mathrm{E}\left[\left(\widehat{X}_{l,(0)} + \widehat{X}'_{l,(0)} Z_v\right) b_{ij}(\widehat{X}_{(0)} + \widehat{X}'_{(0)} Z_v) \frac{Z_v v_j}{\sqrt{T}}\right]$$

$$\approx \left(\widehat{X}_{i,(0)} a_l(\widehat{X}_{(0)}) + \widehat{X}'_{i,(0)} \sum_{k=1}^{d} a'_{l,k}(\widehat{X}_{(0)}) \widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d} \widehat{X}'_{i,(0)} b_{lj}(\widehat{X}_{(0)}) + \widehat{X}_{i,(0)} \left(\sum_{k=1}^{d} \widehat{X}'_{k,(0)} b'_{lj,k}(\widehat{X}_{(0)}) \right) \right) \frac{v_j}{\sqrt{T}} \right)$$

(2.13)

$$+ \left(\widehat{X}_{l,(0)} a_i(\widehat{X}_{(0)}) + \widehat{X}'_{l,(0)} \sum_{k=1}^{d} a'_{i,k}(\widehat{X}_{(0)}) \widehat{X}'_{k,(0)} + \left(\sum_{j=1}^{d} \widehat{X}'_{l,(0)} b_{ij}(\widehat{X}_{(0)}) + \widehat{X}_{l,(0)} \left(\sum_{k=1}^{d} \widehat{X}'_{k,(0)} b'_{ij,k}(\widehat{X}_{(0)}) \right) \right) \frac{v_j}{\sqrt{T}} \right).$$

The idea then is to maximize, at the final time, the smoothing effect, represented by (2.5), where $\widehat{X}_T \sim \mathcal{N}(\mu_T(\mathbf{v}), \Sigma_T(\mathbf{v}))$, such that μ_T, Σ_T are computed using (2.11), (2.12) and (2.13). We note that the computation of $\widehat{X}'$ is cheap and its evolution can be deduced from (2.9), resulting in

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

$$\widehat{X}'(t=0) = 0, \quad 1 \leq i \leq d,$$

which when evaluated at $Z_v = 0$ simplifies to

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

$$\widehat{X}'_{(0)}(t=0) = 0, \quad 1 \leq i \leq d.$$

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

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

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

$$(2.15) \qquad\qquad g(\mathbf{x}) = \mathbf{1}_{(\phi(\mathbf{x}) \geq 0)}.$$

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

$$(2.16) \qquad \frac{\partial \phi}{\partial x_j}(\mathbf{x}) > 0, \ \forall \mathbf{x} \in \mathbb{R}^d \ \textbf{(Monotonicity condition)}$$

$$(2.17) \qquad \lim_{x \to +\infty} \phi(\mathbf{x}) = \lim_{x \to +\infty} \phi(x_j, \mathbf{x}_{-j}) = +\infty, \ \text{or} \ \frac{\partial^2 \phi}{\partial x_j^2}(\mathbf{x}) \ \textbf{(Growth condition)}.$$

In the following, we motivate the optimization problem (2.5) for the call and the binary examples.

Example 2.1 (Binary option). We consider the case of one dimensional binary option, such that g is given by (2.15) and $\phi(x) = x - K$, where K is the strike price.

In this case we have

$$\text{Var}\left[g(\widehat{X}_T)\right] = p(1-p)$$

where $p = \mathbb{P}(\phi(\widehat{X}_T) \geq 0)$, and can be approximated using the normal approximation based on the moment expansion where the first and second moments are provided by (2.11) and (2.12). In fact, we have

$$
\begin{aligned}
p = \mathbb{P}\left(\phi(\widehat{X}_T) \geq 0\right) &= \mathbb{P}\left(\widehat{X}_T \geq K\right) \\
&= \mathbb{P}\left(\frac{\widehat{X}_T - \mu_T(\mathbf{v})}{\sqrt{\sigma_T(\mathbf{v})}} \geq \frac{K - \mu_T(\mathbf{v})}{\sqrt{\sigma_T(\mathbf{v})}}\right) = 1 - \mathbb{P}\left(\frac{\widehat{X}_T - \mu_T(\mathbf{v})}{\sqrt{\sigma_T(\mathbf{v})}} < \frac{K - \mu_T(\mathbf{v})}{\sqrt{\sigma_T(\mathbf{v})}}\right) \\
&= 1 - \Phi\left(\frac{K - \mu_T(\mathbf{v})}{\sqrt{\sigma_T(\mathbf{v})}}\right),
\end{aligned}
$$

where $\Phi(.)$ is the standard cumulative distribution function, and μ_T and σ_T are computed using (2.11) and (2.12).

Then the next step to determine $\mathbf{v}^*$ is to maximize p over $\mathbf{v}$.

The last statement maybe changed if the true quantity to maximize over v is not 2.5.

Example 2.2 (Call option). We consider the case of one dimensional call option, g is given by (2.14) and $\phi(x) = x - K$, where K is the strike price.

Add details here for the optimization problem (2.5) for the call option example.

2.2 Discrete time formulation

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

given by

$$(2.18) \qquad g(\mathbf{X}(T)) = \max\left(\sum_{j=1}^{d} c_j X^{(j)}(T) - K, 0\right).$$

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

$$(2.19) \qquad dX_t^{(j)} = \sigma^{(j)} X_t^{(j)} dW_t^{(j)},$$

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

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

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

where

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

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

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

2.2.1 Step 1: Numerical smoothing

The first step of our idea is to smoothen the problem by solving the root finding problem in one dimension after using a sub-optimal linear mapping for the coarsest factors of the Brownian

increments $\mathbf{Z}_1 = (Z_1^{(1)}, \ldots, Z_1^{(d)})$. In fact, let us define for a certain $d \times d$ matrix $\mathcal{A}$, the linear mapping

$$(2.22) \qquad\qquad \mathbf{Y} = \mathcal{A}\mathbf{Z}_1.$$

Then from (2.21), we have

$$\overline{X}^{(j)}(T) = X_0^{(j)} \prod_{i=0}^{N-1} f_i^{(j)}(\mathcal{A}^{-1}\mathbf{Y})_j, \quad 1 \le j \le d,$$

$$(2.23) \qquad\qquad = X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1, \mathbf{Y}_{-1}) \quad 1 \le j \le d$$

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

$$g_i^{(j)}(Y_1, \mathbf{Y}_{-1}) = \left[1 + \frac{\sigma^{(j)}}{\sqrt{T}} \left(\sum_{i=1}^{d} A_{ji}^{\mathrm{inv}} Y_i \right) \Delta t + \sigma^{(j)} \Delta \widetilde{B}_i^{(j)} \right]$$

$$(2.24) \qquad = \left[1 + \frac{\sigma^{(j)}\Delta t}{\sqrt{T}} A_{j1}^{\mathrm{inv}} Y_1 + \frac{\sigma^{(j)}}{\sqrt{T}} \left(\sum_{i=2}^{d} A_{ji}^{\mathrm{inv}} Y_i \right) \Delta t + \sigma^{(j)} \Delta \widetilde{B}_i^{(j)} \right].$$

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

$$(2.25) \qquad\qquad x = \sum_{j=1}^{d} c_j X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*(x), \mathbf{Y}_{-1}),$$

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

$$(2.26) \qquad\qquad P(Y_1^*(K)) = \left(\sum_{j=1}^{d} c_j X_0^{(j)} \prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*) \right) - K.$$

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

$$(2.27) \qquad P'(Y_1^*) = \sum_{j=1}^{d} c_j X_0^{(j)} \frac{\sigma^{(j)}\Delta t A_{j1}^{\mathrm{inv}}}{\sqrt{T}} \left(\prod_{i=0}^{N-1} g_i^{(j)}(Y_1^*) \right) \left[\sum_{i=0}^{N-1} \frac{1}{g_i^{(j)}(Y_1^*)} \right].$$

Remark 2.3. For our purposes, we suggest already that the coarsest factors of the Brownian increments are the most important ones, compared to the remaining factors. Furthermore, one may expect that in case we want to optimize over the choice of the linear mapping $\mathcal{A}$, and which direction is the most important for the kink location, one needs then to solve

$$(2.28) \qquad\qquad \sup_{\substack{\mathcal{A} \in \mathbb{R}^{d \times d} \\ \mathcal{A} \text{ is a rotation}}} \left(\mathrm{Var} \left[g \left(\overline{\mathbf{X}}(T) \right) \right] \right),$$

which becomes hard to solve when d increases. We note that the first row of $\mathcal{A}$ is expected to be the most important one and we can determine it using (2.5).

8

Remark 2.4. A general choice for $\mathcal{A}$ should be in the family of rotations. However, we think that a sufficiently good matrix $\mathcal{A}$ would be the one leading to $Y_1 = \sum_{i=1}^{d} Z_1^{(i)}$ up to re-scaling. Therefore, the first attempt will be to set $\mathcal{A}$ to be a rotation matrix, with first row leading to $Y_1 = \sum_{i=1}^{d} Z_1^{(i)}$ up to re-scaling, and with no constraints for the remaining rows. In practice, we construct $\mathcal{A}$ by fixing the first row to be $\frac{1}{\sqrt{d}}\mathbf{1}_{1 \times d}$ and the remaining rows are obtained by Gram-Schmidt procedure.

2.2.2 Step 2: Integration

At this stage, we want to perform the pre-integrating step with respect to y_1^*. In fact, using Fubini's theorem, we have from (2.20)

$$
\mathrm{E}\left[g(\mathbf{X}(T))\right] \approx \int_{\mathbb{R}^{d \times N}} G(z_1^{(1)}, \ldots, z_N^{(1)}, \ldots, z_1^{(d)}, \ldots, z_N^{(d)}))\rho_{d \times N}(\mathbf{z})dz_1^{(1)} \ldots dz_N^{(1)} \ldots z_1^{(d)} \ldots dz_N^{(d)}
$$

$$
= \int_{\mathbb{R}^{dN-1}} \left(\int_{\mathbb{R}} G(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1 \right) \rho_{d-1}(\mathbf{y}_{-1})d\mathbf{y}_{-1}\rho_{d \times (N-1)}(\mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})d\mathbf{z}_{-1}^{(1)} \ldots d\mathbf{z}_{-1}^{(d)}
$$

(2.29)

$$
= \int_{\mathbb{R}^{dN-1}} h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\rho_{d-1}(\mathbf{y}_{-1})d\mathbf{y}_{-1}\rho_{d \times (N-1)}(\mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})d\mathbf{z}_{-1}^{(1)} \ldots d\mathbf{z}_{-1}^{(d)},
$$

$$
= \mathrm{E}\left[h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)}) \right],
$$

where

$$
h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)}) = \int_{\mathbb{R}} G(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1
$$

$$
= \int_{-\infty}^{y_1^*} G(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1
$$

(2.30)

$$
+ \int_{y_1^*}^{+\infty} G(y_1, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\rho_{y_1}(y_1)dy_1.
$$

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

Remark 2.5. We note that conditions ((2.16) and (2.17)) imply that for each $\mathbf{Y}_{-1}$, the function G either has a simple root y_1^* or is positive for all $y_1 \in \mathbb{R}$.

3 Analiticity Analysis

3.1 Haar construction of Brownian motion revisited

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

With the Haar mother wavelet

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

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

$$(3.2a) \qquad \psi_{-1}(t) := \mathbb{1}_{[0,1]}(t),$$

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

We note that $\operatorname{supp} \psi_{n,k} = [2^{-n}k, 2^{-n}(k+1)]$. 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

$$(3.3a) \qquad \Psi_{-1}(t) := \int_0^t \psi_{-1}(s)ds,$$

$$(3.3b) \qquad \Psi_{n,k}(t) := \int_0^t \psi_{n,k}(s)ds.$$

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

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

and the truncated version

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

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

$$(3.6) \qquad \Delta_\ell^N F := F(t_{\ell+1}^N) - F(t_\ell^N).$$

3.2 Stochastic differential equations

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

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

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

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

10

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

$$(3.8) \qquad X_{\ell+1}^N := X_\ell^N + b\left(X_\ell^N\right)\Delta_\ell^N W.$$

For convenience, we also define $X_T^N := X_{2^N}^N$.

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

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

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

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

Define

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

We claim that H^N is analytic.

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

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

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

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

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

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

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

---

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

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

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

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

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

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

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

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

Lemma 3.4. *We have*

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

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

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

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

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

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

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

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

Noting that

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

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

$\square$

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

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

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

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

Further,

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

Note that

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

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

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

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

Proof. We have

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

$$= -\int_{\mathbb{R}} g\left(X_T^N(y, z^N)\right) \left[-\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-2} \frac{\partial^2 X_T^N}{\partial y^2}(y, z^N) \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N)+ \right.$$

$$\left. + \left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial^2 X_T^N}{\partial z_{n,k}\partial y}(y, z^N) - y\left(\frac{\partial X_T^N}{\partial y}(y, z^N)\right)^{-1} \frac{\partial X_T^N}{\partial z_{n,k}}(y, z^N)\right] \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy.$$

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

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

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

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

with constants independent of n and k. $\square$

For the general case we need

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

We may need to clearly state the goal of this section. Do we proof analiticity ot just provide a motivation.

4 Details of our hierarchical method

In the following, we describe our approach which can be seen as a two stage method. In the first stage, we use root finding procedure to perform the numerical smoothing described in Section 2.2.1, then in a second stage we perform the numerical integration to compute (2.29), described in Section 2.2.2, by employing hierarchical adaptive sparse grids quadrature, using the same construction as in MISC, proposed in [9]. Therefore, the initial integration problem that we are solving lives in $dN - 1$-dimensional space, which becomes very large as either the number of time steps N, used in the discretization scheme, increases, or the number the assets increase.

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

Since g can have a kink or jump. Computing h in (2.30) should be carried carefully to not deteriorate the smoothness of h. This can be done by applying a root finding procedure and then computing the uni-variate integral by summing the terms coming from integrating in each region where g is smooth. We provide details about the root finding procedure in Section 4.4.

If we denote by $\mathcal{E}_{\text{tot}}$ the total error of approximating the expectation in (2.29) using the MISC estimator, Q_N, then we have a natural error decomposition

$$\mathcal{E}_{\text{tot}} \leq \left| \mathrm{E}\left[g(\mathbf{X}(T))\right] - \mathrm{E}\left[h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\right] \right| + \left| \mathrm{E}\left[h(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})\right] - Q_N \right|$$

(4.1) $$\leq \mathcal{E}_B(N) + \mathcal{E}_Q(\text{TOL}_{\text{MISC}}, N),$$

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

4.1 The MISC method

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

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

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

In our case, we have in (2.29) a multi-variate integration problem with, $f := h$, $\mathbf{Y} = (\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$, and $\Gamma = \mathbb{R}^{dN-1}$, in the previous notations. Furthermore, since we are dealing with Gaussian densities, using Gauss-Hermite quadrature points is the appropriate choice.

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

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

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

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

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

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

To be concrete, in our setting, we are left with a $dN - 1$-dimensional Gaussian random input, which is chosen independently, resulting in $dN - 1$ numerical parameters for MISC, which we use as the basis of the multi-index construction. For a multi-index $\boldsymbol{\beta} = (\beta_n)_{n=1}^{dN-1} \in \mathbb{N}^{dN-1}$, we denote by $Q_N^{\boldsymbol{\beta}}$, the result of approximating (2.29) with a number of quadrature points in the i-th dimension equal to $m(\beta_i)$. We further define the set of differences $\Delta Q_N^{\boldsymbol{\beta}}$ as follows: for a single index $1 \le i \le dN - 1$, let

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

---

[2]More details about sparse grids can be found in [3].

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

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

The MISC estimator used for approximating (2.29), and using a set of multi-indices $\mathcal{I} \subset \mathbb{N}^{dN-1}$ is given by

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

The quadrature error in this case is given by

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

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

$$\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \text{Work}[Q_N^{\mathcal{I}}]$$
$$(4.4) \qquad \Delta E_{\boldsymbol{\beta}} = \left| Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - Q_N^{\mathcal{I}} \right|.$$

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

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

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

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

4.2 Brownian bridge (Bb) construction

In the literature of adaptive sparse grids and QMC, several hierarchical path generation methods (PGMs) or transformation methods have been proposed to reduce the effective dimension. Among these transformations, we cite the Brownian bridge (Bb) construction [11, 12, 4], the principal component analysis (PCA) [1] and the linear transformation (LT) [10].

In our context, the Brownian motion on a time discretization can be constructed either sequentially using a standard random walk construction, or hierarchically using other PGMs as listed above. For our purposes, to make an effective use of MISC, which benefits from anisotropy, we use the Bb construction since it produces dimensions with different importance for MISC, contrary to a

random walk procedure for which all the dimensions of the stochastic space have equal importance. In fact, Bb uses the first several coordinates of the low-discrepancy points to determine the general shape of the Brownian path, and the last few coordinates influence only the fine detail of the path. Consequently, this transformation reduces the effective dimension of the problem, which results in accelerating the MISC method by reducing the computational cost.

Let us denote $\{t_i\}_{i=0}^N$ the grid of time steps. Then the Bb construction [5] consists of the following: given a past value B_{t_i} and a future value B_{t_k}, the value B_{t_j} (with $t_i < t_j < t_k$) can be generated according to

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

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

4.3   Richardson extrapolation

Another transformation that we couple with MISC is Richardson extrapolation [13]. In fact, applying level K_R (level of extrapolation) of Richardson extrapolation dramatically reduces the bias, and as a consequence reduces the number of time steps N needed in the coarsest level to achieve a certain error tolerance. As a consequence, Richardson extrapolation directly reduces the total dimension of the integration problem for achieving some error tolerance.

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

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

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

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

implying

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

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

$$I(J, K_R) = \frac{2^{K_R}\left[I(J, K_R - 1) - I(J-1, K_R - 1)\right]}{2^{K_R} - 1}, \quad J = 1, 2, \ldots, K_R = 1, 2, \ldots$$

4.4   Root Finding

From Section 2.2, we denoted the location of irregularity (the kink) by y_1^*, that is G, defined in (2.29), is not smooth at the point $(y_1^*, \mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$. Let us call R the mapping such that: $R : (\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)}) \to y_1^*$. Generally, there might be, for given $(\mathbf{y}_{-1}, \mathbf{z}_{-1}^{(1)}, \ldots, \mathbf{z}_{-1}^{(d)})$

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

- a unique solution;

- multiple solutions.

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

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

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

In many situations, case 2 (which is thought to include case 1) can be guaranteed by monotonicity (see assumption (2.16)) (I think we need to add also the growth condition). For instance, in the case of one-dimensional SDEs with z_1 representing the terminal value of the underlying Brownian motion, this can often be seen from the SDE itself. Specifically, if each increment "dX" is increasing in z_1, no matter the value of X, then the solution X_T must be increasing in z_1. This is easily seen to be true in examples such as the Black-Scholes model and the CIR process. (Strictly speaking, we have to distinguish between the continuous and discrete time solutions. In these examples, it does not matter.) On the other hand, it is also quite simple to construct counter examples, where monotonicity fails, for instance SDEs for which the "volatility" changes sign, such as a trigonometric function.[3]

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

5 Error discussion

5.1 Errors in smoothing

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

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

---

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

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

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

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

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

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

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

We need to check the impact of the error caused by the Newton iteration on the integration error. In the worst case, we expect that if the error in the Newton iteration is of order $O(\epsilon)$ than the integration error will be of order $\log(\epsilon)$. But we need to check that too.

6 Numerical experiments

In this section, we conduct our experiments for four different examples: i) single binary option under discretized GBM model, ii) single call option under discretized GBM model, iii) basket call option under discretized GBM model and iv) best of call option under discretized Heston model.

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

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

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

---

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

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

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

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

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

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

We show the summary of our numerical findings in Table 6.1, which highlights the computational gains achieved by MISC over MC method to meet a certain error tolerance, which we set approximately to 1%. More detailed results for each case are provided in Sections 6.1.2, 6.1.3, 6.2 and 6.5.

Example	Level of Richardson extrapolation	Total relative error	Ratio of CPU time (MC/MISC)
Single binary option (GBM)	without	2%	4
Single call option (GBM)	without	0.9%	22
$2d$-Basket call option (GBM)	without	0.9%	21
Best of call option (Heston)	without		

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

6.1 Options under the discretized one dimensional GBM model

The first two examples that we will test are the single binary and call options under GBM model where the process X is the discretized one dimensional GBM model and the payoff function g is the indicator or maximum function as given by (2.15) and (2.14) respectively. Precisely, we are interested in the one dimensional lognormal example where the dynamics of the stock are given by

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

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

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

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

6.1.1   Determining the kink location

Exact location of the kink for the continuous problem

Let us denote y^* an invertible function that satisfies

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

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

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

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

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

Location of the kink for the discrete problem

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

$$\overline{X}_{t_1} = \overline{X}_{t_0}\left[1 + \frac{\sigma}{\sqrt{T}}z_1\Delta t + \sigma\Delta B_0\right]$$

$$\overline{X}_{t_2} = \overline{X}_{t_1}\left[1 + \frac{\sigma}{\sqrt{T}}z_1\Delta t + \sigma\Delta B_1\right]$$

$$\vdots = \vdots$$

$$(6.7) \qquad\qquad \overline{X}_{t_N} = \overline{X}_{t_{N-1}}\left[1 + \frac{\sigma}{\sqrt{T}}z_1\Delta t + \sigma\Delta B_{N-1}\right]$$

implying that

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

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

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

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

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

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

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

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

Therefore, in this case, the integrand $h(\mathbf{z}_{-1})$, as expressed in (2.30), is given by

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

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

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

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

6.1.2 Results for the single binary option under discretized GBM model

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

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

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

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

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

Figure 6.1 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and MISC in Tables 6.2 and 6.3, and Figure 6.2. Our numerical experiments show that MISC requires approximately 25% of the work of MC to achieve a total relative error of around 2%.
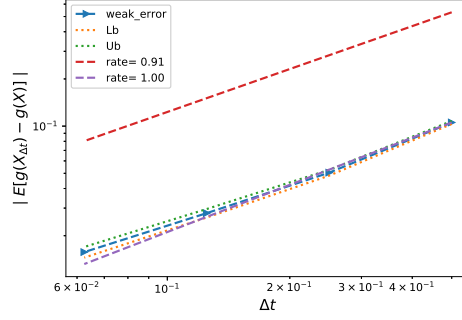
Figure 6.1: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, using MC with $M = 10^4$ samples for the binary option example. The upper and lower bounds are 95% confidence intervals.

Method	Steps		
	4	8	16
MISC ($TOL_{\mathrm{MISC}} = 5.10^{-1}$)	**0.05** (0.04,0.01)	**0.02** (0.02,0.002)	**0.01** (0.01,0.002)
MC+root finding	**0.08** (0.04,0.04)	**0.04** (0.02,0.02)	**0.02** (0.01,0.01)
M(# MC samples)	10^1	4×10^1	10^2
MC	**0.1** (0.05,0.05)	**0.05** (0.025,0.025)	**0.02** (0.01,0.01)
M(# MC samples)	10^3	8×10^3	5×10^4

Table 6.2: Total relative error of MISC, with different tolerances, and MC to compute binary option price for different number of time steps, without Richardson extrapolation. The values marked in red, for MISC method, correspond to the total relative errors associated with stable quadrature errors for MISC, and will be used for complexity comparison against MC.

Method	Steps		
	4	8	16
MISC ($TOL_{\mathrm{MISC}} = 5.10^{-1}$)	0.8	2	9
MC+root finding method	0.2	1	9
MC method	0.3	3	29

Table 6.3: Comparison of the computational time of MC and MISC, used to compute binary option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.
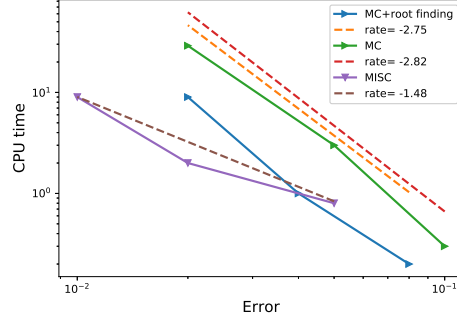
Figure 6.2: Computational work comparison for MISC and MC methods, for the case of binary option. This plot shows that to achieve a relative error below 1%, MISC outperforms MC method in terms of computational time

6.1.3 Results for the single call option example

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

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

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

$$(6.17) \qquad h(\mathbf{z}_{-1}) := \int_{\mathbb{R}} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dy$$

$$= \int_{-\infty}^{y^*} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1$$

$$+ \int_{y^*}^{\infty} \max\left(\Psi \circ \Phi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dz_1,$$

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

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

Figure 6.3 shows the estimated weak error for the case without Richardson extrapolation, and we report the results for comparing MC and MISC in Tables 6.4 and 6.5, and Figure 6.4. Our numerical experiments show that MISC requires approximately 5% of the work of MC to achieve a total relative error of around 0.9%.
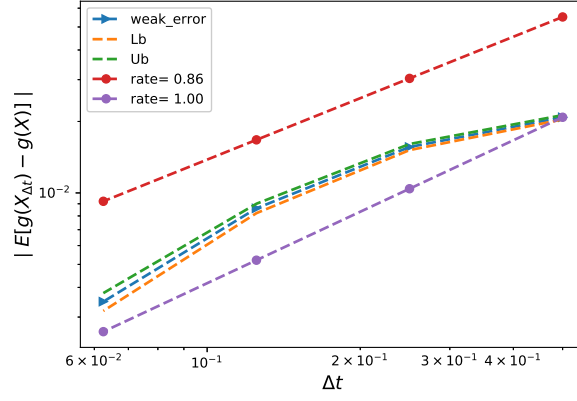
25

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

Method	Steps			
	2	4	8	16
MISC ($TOL_{\mathrm{MISC}} = 5.10^{-1}$, $\beta = 32$)	**0.022** (0.021,0.001)	**0.018** (0.016,0.002)	**0.009** (0.009,0.0001)	**0.004** (0.004,0.0004)
MC +root finding ($\beta = 32$)	**0.041** (0.021,0.02)	**0.032** (0.016,0.016)	**0.018** (0.009,0.009)	**0.008** (0.004,0.004)
M(# MC samples)	10^2	3×10^2	10^3	4×10^3
MC	**0.04** (0.02,0.02)	**0.032** (0.016,0.016)	0.02 (0.01,0.01)	**0.008** (0.004,0.004)
M(# MC samples)	3×10^4	5×10^4	2×10^5	8×10^5

Table 6.4: Total relative error of MISC, with different tolerances, and MC to compute call option price for different number of time steps, without Richardson extrapolation. The values marked in red, for MISC method, correspond to the total relative errors associated with stable quadrature errors for MISC, and will be used for complexity comparison against MC.

Method	Steps			
	2	4	8	16
MISC ($TOL_{\mathrm{MISC}} = 5.10^{-1}$, $\beta = 32$)	0.3	3	17	473
MC method +root finding ($\beta = 32$)	3	16	70	408
MC method	3	13	76	380

Table 6.5: Comparison of the computational time of MC and MISC, used to compute call option price for different number of time steps, without Richardson extrapolation. The average computational time of MC is computed over 10 runs.
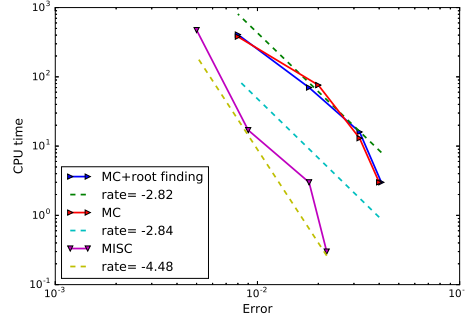
Figure 6.4: Computational work comparison for MISC and MC methods, for the case of call option. This plot shows that to achieve a relative error below 1%, MISC outperforms significantly MC method in terms of computational time.

6.2 The basket call option under GBM model

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

6.3 $d = 2$

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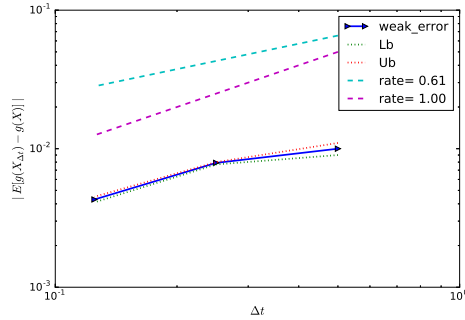

Figure 6.5: The convergence of the relative weak error $\mathcal{E}_B(N)$ defined in 4.1, for the two dimensional basket call option with a number of Laguerre quadrature points $\beta = 128$ and number of samples for MC $M = 10^7$. The upper and lower bounds are 95% confidence intervals.

27

Method	Steps		
	2	4	8
MISC	**0.016** (0.01,0.006)	**0.0134** (0.0079,0.0055)	**0.0085** (0.0043,0.0042)
MC +root finding	**0.02** (0.01,0.01)	**0.0155** (0.0079,0.0076)	**0.0081** (0.0043,0.0038)
M(# MC samples)	2×10^3	5×10^3	2×10^4
MC	**0.02** (0.01,0.01)	**0.0155** (0.0079,0.0076)	**0.0083** (0.0043,0.004)
M(# MC samples)	10^5	$2 \times 0.0161 0^5$	8×10^5

Table 6.6: Total relative error of MISC, with different tolerances, and MC to compute two dimensional basket call option price for different number of time steps, without Richardson extrapolation. The values marked in red, for MISC method, correspond to the total relative errors associated with stable quadrature errors for MISC, and will be used for complexity comparison against MC.

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

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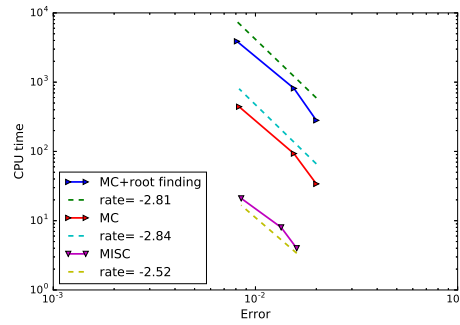

Figure 6.6: Computational work comparison for MISC and MC methods, for the case of two dimensional basket call option. This plot shows that to achieve a relative error below 1%, MISC outperforms significantly MC method in terms of computational time.

6.4 $d = 4$

We consider now the four dimensional basket call option with parameters: $S_0^{1,2,3,4} = K = 100$, $\sigma_{1,2,3,4} = 0.4$, $\rho = 0.3$, $T = 1$ $r = 0$ and $c_{1,2,3,4} = 1/4$, the reference value for those parameters is 11.0.

6.5 The best call option under GBM and Heston model

The fourth example that we consider is the multi-dimensional best call option under GBM and Heston model. I am in process of formulating this but somehow we agreed that the first potential directions of smoothing for the Heston model will be the first factors related to the asset prices.

References Cited

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

[2] CHRISTIAN BAYER, MARKUS SIEBENMORGEN, and RAUL TEMPONE. Smoothing the payoff for efficient computation of basket option pricing.

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

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

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

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

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

[8] 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.

[9] 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.

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

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

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

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

[14] 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.