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

1 Introduction

1.1 The goal and outline of the project

The first goal of the project is to approximate $\mathrm{E}[f(X(t))]$, using multi-index stochastic collocation(MISC) method, proposed in [14], where

- The payoff $f : \mathbb{R}^d \to \mathbb{R}$ has either jumps or kinks. Possible choices of f that we wanted to test are:
  - hockey-stick function, i.e., put or call payoff functions;
  - indicator functions (both relevant in finance (binary option,...) and in other applications of estimation of probabilities of certain events);
  - delta-functions for density estimation (and derivatives thereof for estimation of derivatives of the density).

  More specifically, f should be the composition of one of the above with a smooth function. (For instance, the basket option payoff as a function of the log-prices of the underlying.)
- The process X is simulated via a time-stepping scheme. Possible choices that we wanted to test are
  - The one/multi dimensional discretized Black-Scholes(BS) process where we compare different ways to identify the location of the kink, such as:
    - Exact location of the continuous problem
    - Exact location of the discrete problem by root finding of a polynomial in y.
    - Newton iteration.
  - A relative simple interest rate model or stochastic volatility model, for instance CIR or Heston models: In fact, the impact of the Brownian bridge will disappear in the limit, which may make the effect of the smoothing, but also of the errors in the kink location difficult to identify. For this reason, we suggest to study a more complicated 1-dimensional problem next. We suggest to use a CIR process. To avoid complications at the boundary, we suggest "nice" parameter choices, such that the discretized process is very unlikely to hit the boundary (Feller condition).

– The multi dimensional discretized Black-Scholes(BS) process: Here, we suggest to return to the Black-Scholes model, but in multi-dimensional case. In this case, linearizing the exponential, suggest that a good variable to use for smoothing might be the sum of the final values of the Brownian motion. In general, though, one should probably eventually identify the optimal direction(s) for smoothing via the duals algorithmic differentiation.

The desired outcome is a paper including

- Theoretical results including: i) an analiticity proof for the integrand in the time stepping setting, ii) a numerical analysis of the schemes involved, such as Newton iteration, etc.
- Applications that tests the examples above.

What has beed achieved so far:

1. Numerical outputs:
   - **Example 1**: Tests for the basket option with the smoothing trick as in [4] (see Section 5.5): in that example we checked the performance of MISC without time stepping scheme and also compare the results with reference [4]. (Done).
   - **Example 2**: The one dimensional binary option under discretized BS model (see Section 5.3): We tested a simple case which is given by the binary option. We tested the weak rates (with and without using the Richardson extrapolation). The results are promising (Done).
   - **Example 3**: The one dimensional call option under discretized BS model (see Section 5.4). The results are promising (Done).
2. Theoretical outputs:
   - Heuristic proof of analiticity (See attached file.)

1.2 Literature review

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 coreesponds 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 [6] 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 [11, 4, 12, 13, 17] adressed similar kind of problems, characterized by the presence of kinks and jumps, but with much more emphasis on Quasi Monte Carlo (QMC). In [11, 12, 13], an analysis of the performance of Quasi Monte Carlo (QMC) and SG methods has been conducted, in the presence of kinks and jumps. In [11, 12], 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, [13] extended the work in [11, 12] from kinks to jumps for the case of an arithmetic average digital Asian option with the principal component analysis (PCA). The main findings in [11, 12] 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 [11, 12, 13] 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 [17] adresses similar kind of problems using QMC. Being very much related to [4], 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 assumptions, which is why we need numerical methods, more precisely root finding and the quadrature in the first direction.

1.3 Notation

In the following, we clarify some notations that we will be using in this paper:

- Given $\mathbf{x} \in \mathbb{R}^N$, $\mid \mathbf{x} \mid_0$ denotes the number of non-zero components of $\mathbf{x}$.
- $\mathcal{L}_+$ denotes the set of sequences with positive components with only finitely many elements larger than 1, *i.e.*, $\mathcal{L}_+ = \{\boldsymbol{\beta} \in \mathbb{N}_+^{\mathbb{N}} :\mid \boldsymbol{\beta} - 1 \mid_0 < \infty\}$.

2 Problem formulation and Setting

In the context of option pricing, we aim at approximating the option price, $E[g(\mathbf{X}(t))]$, where $g : \mathbb{R}^d \to \mathbb{R}$ is the payoff function and where the process of the asset prices $\mathbf{X} \in \mathbb{R}^d$ solves

$$\mathbf{X}(t) = \mathbf{X}(0) + \int_0^t a(s, \mathbf{X}(s))ds + \sum_{\ell=1}^{\ell_0} \int_0^t b^\ell(s, \mathbf{X}(s))dW^\ell(s) \tag{1}$$

Let us denote by $\Phi : (\mathbf{z}_1, \ldots, \mathbf{z}_N) \to \mathbf{X}_T$, the mapping consisting of the time-stepping scheme, where $\{\mathbf{z}_i\}_{i=1}^N$ are independent d-dimensional Gaussian random vectors, and N is the number of time steps. Without loss of Generality, we assume that Φ may include pre-processing transformations to reduce the effective dimension. We also assume that $d = 1$ and the extension to higher dimension is trivial.

In this setting, we are interetsed in the basic problem of approximating

$$E[g(\mathbf{X}(t))] = I_N(g \circ \Phi) := \int_{\mathbb{R}^N} g \circ \Phi(\mathbf{z}) d\mathbf{z} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g \circ \Phi(z_1, \ldots, z_N) \rho_d(\mathbf{z}) dz_1, \ldots, dz_N, \tag{2}$$

with

$$\rho_N(\mathbf{z}) = \frac{1}{(2\pi)^{N/2}} e^{-\frac{1}{2}\mathbf{z}^T\mathbf{z}}. \tag{3}$$

where ρ is a continuous and strictly positive probability density function on $\mathbb{R}$ and g is a real-valued function integrable with respect to ρ_N.

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

$$g(\mathbf{x}) = \max(\phi(\mathbf{x}), 0). \tag{4}$$

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

$$g(\mathbf{x}) = \mathbf{1}_{(\phi(\mathbf{x}) \geq 0)}. \tag{5}$$

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. Then, if we assume for some $j \in \{1, \ldots, d\}$

$$\frac{\partial \phi}{\partial x_j}(\mathbf{x}) > 0, \ \forall \mathbf{x} \in \mathbb{R}^d \ \textbf{(Monotonicity condition)} \tag{6}$$

$$\lim_{x \to +\infty} \phi(\mathbf{x}) = \lim_{x \to +\infty} \phi(x_j, \mathbf{x}_{-j}) = +\infty, \text{ or } \frac{\partial^2 \phi}{\partial x_j^2}(\mathbf{x}) \ \textbf{(Growth condition)}, \tag{7}$$

then, using Fubini's theorem, we can rewrite (2) as

$$\begin{aligned} I_N(g \circ \Phi) &= \int_{\mathbb{R}^{d-1}} \left(\int_{-\infty}^{\infty} g \circ \Phi(z_j, \mathbf{z}_{-j}) \rho(z_j) dz_j \right) \rho_{z-1}(\mathbf{z}_{-j}) d\mathbf{z}_{-j}, \\ &= \mathrm{E}\left[E\left[g \circ \Phi(z_j, \mathbf{z}_{-j}) \mid z_j \right] \right] \end{aligned} \tag{8}$$

where we evaluate the inner integral for each $\mathbf{z}_{-j}$ and which results in a smooth integrand for the outer $(N-1)$-dimensional integral.

We note that conditions ((6) and (7)) imply that for each $\mathbf{z}_{-j}$, the function $\phi \circ \Phi(z_j, \mathbf{z}_{-j})$ either has a simple root z_j or is positive for all $z_j \in \mathbb{R}$.

We generally do not have a closed form for the inside integral in 8. Therefore, the pre-integration (conditional sampling) step should be performed numerically.

3 Details of our approach

In the following, we describe our approach which basically can be seen as a two stage method. In the first step, we use root finding procedure to get the inside integral in 8, then in a second stage we employ adaptive quadraure, multi-index stochastic collocation (MISC), to compute the obtained smooth integrand.

For illustration purposes, let us focus on the one dimensional case, where under the risk-neutral measure, the undelying asset follows the geometric Brownian motion (GBM)

$$dX_t = rX_t dt + \sigma X_t dB_t, \tag{9}$$

where r is the risk-free rate, σ is the volatility and B_t is the standard Brownian motion. The analytical solution to (9) is

$$X_t = X_0 \exp((r - \sigma^2)t + \sigma B_t). \tag{10}$$

The Brownian motion B_t can be constructed either sequentially using a standard random walk construction or hierarchically using Brownian bridge (BB) construction. To make an effective use of MISC, which is badly affected by isotropy, we use the BB construction since it produces dimensions with different importance for MISC (creates anisotropy), contrary to random walk procedure for which all the dimension of the stochastic space have equal importance (isotropic). We explain the BB construction in Section 3.3. This transformation plays a role of reducing the effective dimension of the problem and as a consquence accelerating the MISC procedure by reducing the compotational cost.

Another way to reduce the dimension of the problem is by using Richardson extrapolation, explained in Section 3.4. In fact, Richardson extrapolation acts on both the bias (by reducing it) and MISC procedure by redcing the number of needed time steps, N, neeeded to achive a certain tolerance, resulting in a lower dimensional problem.

Let us denote by $\psi : (z_1, \ldots, z_N) \to (B_1, \ldots, B_N)$ the mapping of BB construction and by $\Phi : (B_1, \ldots, B_N) \to X_T$, the mapping consisting of the time-stepping scheme. Then, we can express the option price as

$$\begin{aligned} \mathrm{E}\left[g(X(T))\right] &= \mathrm{E}\left[g\left(\Phi \circ \psi\right)(z_1, \ldots, z_N)\right] \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} G(z_1, \ldots, z_N)\rho_N(\mathbf{z}) dz_1, \ldots, dz_N, \end{aligned} \tag{11}$$

where $G = g \circ \Phi \circ \psi$ and

$$\rho_N(\mathbf{z}) = \frac{1}{(2\pi)^{N/2}} e^{-\frac{1}{2}\mathbf{z}^T\mathbf{z}}. \tag{12}$$

Now, we can easily apply the procedure of pre-integration of section 2, where we can assume that the payoff function g can be either the maximum or indicator function and $\phi = \Phi \circ \psi$. The remaining ingredient is to determine with respect to which variable z_j we will integrate.

Claiming that pre-integrating with respect to z_1 is the optimal option then from (11), we have

$$
\begin{aligned}
\mathrm{E}\left[g(X(T))\right] &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} G(z_1,\ldots,z_N)\rho_N(\mathbf{z})dz_1,\ldots,dz_N \\
&= \int_{\mathbb{R}^{N-1}}\left(\int_{-\infty}^{\infty} G(z_1,\mathbf{z}_{-1})\rho(z_1)dz_1\right)\rho_{N-1}(\mathbf{z}_{-1})d\mathbf{z}_{-1} \\
&= \int_{\mathbb{R}^{N-1}} h(\mathbf{z}_{-1})\rho_{N-1}(\mathbf{z}_{-1})d\mathbf{z}_{-1}, \qquad (13)\\
&= \mathrm{E}\left[h(\mathbf{z}_1)\right]
\end{aligned}
$$

where $h(\mathbf{z}_{-1}) = \int_{-\infty}^{\infty} G(z_1,\mathbf{z}_{-1})\rho(z_1)dz_1 = E\left[G\left(z_1,\ldots,z_N\right) \mid z_1\right]$.

Since g can have a kink or jump. Computing $h(\mathbf{z}_{-1})$ in the pre-integration step 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. In Sections (3.5,3.6), we explain those points.

Once we perform stage 1 procedure, we use multi-index stochastic collocation (MISC) procedure, suggested in [14], to compute the expectation $\mathrm{E}\left[h(\mathbf{z}_1)\right]$. We describe the general strategy for the multi-index construction in Section 3.2.

We have a natural error decomposition for the total error of computing the the expectation in (13), namely, $\mathcal{E}$

$$
\mathcal{E} \leq \mathcal{E}_Q(TOL_{\mathrm{MISC}}, N) + \mathcal{E}_B(N), \qquad (14)
$$

where $\mathcal{E}_Q$ is the quadrature error, function of MISC tolerance TOL_{MISC} and N (the number of time steps) and $\mathcal{E}_B$ is the bias, function of N (the number of time steps) or $\Delta_t = \frac{T}{N}$ (size of the time grid). We provide a discussion about the different errors in Section 4.

3.1 Extension to the High dimensional case

In the high dimensional case, we denote by $(z_1^i,\ldots,z_N^i)$ the N Gaussian independent rdv that will be used to construct the path of the i-th asset X_i, where $1 \leq i \leq d$ (d denotes the number of underlyings considered in the basket). We keep the same notations by denoting $\psi : (z_1^{(i)},\ldots,z_N^{(i)}) \to (B_1,\ldots,B_N)$ the mapping of BB construction and by $\Phi : (B_1^{(i)},\ldots,B_N^{(i)}) \to X_T^i$, the mapping consisting of the time-stepping scheme. Then, we can express the option price as

$$
\begin{aligned}
\mathrm{E}\left[g(\mathbf{X}(T))\right] &= \mathrm{E}\left[g\left(\Phi\circ\psi\right)\left(z_1^{(1)},\ldots,z_N^{(1)},\ldots,z_1^{(d)},\ldots,z_N^{(d)}\right)\right] \\
&= \int_{\mathbb{R}^{d\times N}} G(z_1^{(1)},\ldots,z_N^{(1)},\ldots,z_1^{(d)},\ldots,z_N^{(d)}))\rho_{d\times N}(\mathbf{z})dz_1^{(1)}\ldots dz_N^{(1)}\ldots z_1^{(d)}\ldots dz_N^{(d)}, \qquad (15)
\end{aligned}
$$

where $G = g\circ\Phi\circ\psi$ and

$$
\rho_{d\times N}(\mathbf{z}) = \frac{1}{(2\pi)^{d\times N/2}} e^{-\frac{1}{2}\mathbf{z}^T\mathbf{z}}. \qquad (16)
$$

3.2 MISC details

We focus on solving the problem of approximating the expected value of $\mathrm{E}[f(y)]$ on a tensorization of quadrature formulae over the stochastic domain, Γ. Assuming that $f(y)$ is a continuous function (analytic) over Γ. A quadrature approach is very adequate.

Let us define $\beta \leq 1$ be an integer positive value referred to as a "stochastic discretization level", and $m : \mathbb{N} \to \mathbb{N}$ be a strictly increasing function with $m(0) = 0$ and $m(1) = 1$, that we call a "level-to-nodes function". At level β, we consider a set of $m(\beta)$ distinct quadrature points in $(-\infty; \infty)$, $\mathcal{H}^{m(\beta)} = \{y_\beta^1, y_\beta^2, \ldots, y_\beta^{m(\beta)}\} \subset [-\infty, \infty]$, and a set of quadrature weights, $\mathcal{W}^{m(\beta)} = \{\omega_\beta^1, \omega_\beta^2, \ldots, \omega_\beta^{m(\beta)}\}$. We also let $C^0((-\infty, \infty))$ be the set of real-valued continuous functions over $(-\infty, \infty)$. We then define the quadrature operator as

$$Q(m(\beta)) : C^0((-\infty, \infty)) \to \mathbb{R}, \quad Q(m(\beta))[f] = \sum_{j=1}^{m(\beta)} f(y_\beta^j)\omega_\beta^j. \tag{17}$$

In the multi-variate case Γ is defined as a countable tensor product of intervals. Therefore, we define, for any definitely supported multi-index $\boldsymbol{\beta} \in \mathcal{L}_+$

$$Q^{m(\boldsymbol{\beta})} : \Gamma \to \mathbb{R}, \quad Q^{m(\boldsymbol{\beta})} = \bigotimes_{n \geq 1} Q^{m(\beta_n)}$$

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 $\mathcal{T}^{m(\boldsymbol{\beta})} = \times_{n \geq 1} \mathcal{H}^{m(\beta_n)}$ with cardinality $\#\mathcal{T}^{m(\boldsymbol{\beta})} = \prod_{n \geq 1} m(\beta_n)$ and computing

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

where $\widehat{y}_j \in \mathcal{T}^{m(\boldsymbol{\beta})}$ amd $\overline{\omega}_j$ are (infinite) products of weights of the univariate quadrature rules. We Note that it is essential in this construction that $m(1) = 1$ so that the cardinality of $\mathcal{T}^{m(\boldsymbol{\beta})}$ is finite for any $\boldsymbol{\beta} \in \mathcal{L}_+$ and $\omega_{\beta_n}^1 = 1$ whenever $n = 1$, so that all weights, $\overline{\omega}_j$, are bounded.

We mention that the quadrature points are chosen to optimize the convergence properties of the quadrature error.

A direct approximation $\mathrm{E}[f] \approx Q^{m(\boldsymbol{\beta})}[f]$ is not an appropriate option due to the well-known "curse of dimensionality" effect. We use multi-index stochastic collocation (MISC) as it was suggested in [14]. MISC as a hierarchical adaptive quadrature strategy that uses stochastic discretizations and classic sparsification approach to obtain an effective approximation scheme for $\mathrm{E}[f]$.

In our setting, we are left with a N-dimensional Gaussian random inputs, which are chosen independently, and which we use as the basis of the multi-index construction.

For a multi-index $\ell = (l_i)_{i=1}^N \in \mathbb{N}^N$, we denote by $Q_\ell^N := Q(m_\ell)$ the result of a discretized integral, using N time steps , with parameters $m_\ell := (m_{l_i})_{i=1}^N$. We further define the set of differences ΔQ_ℓ^N as follows: for a single index $1 \leq i \leq N$, let

$$\Delta_i Q_\ell^N := \begin{cases} Q^N(m_\ell) - Q^N(m'_\ell) \text{ with } m'_\ell = m_{\ell - e_i}, \text{ if } \ell_i > 0 \\ Q^N(m_\ell) \text{ otherwise} \end{cases} \tag{18}$$

where e_i denotes the ith N-dimensional unit vector. Then, ΔQ_ℓ^N is defined as

$$\Delta Q_\ell^N \coloneqq \left(\prod_{i=1}^{N} \Delta_i\right) Q_\ell^N. \tag{19}$$

Note that $Q^N(m)$ converges to the biased option price (denoted by $Q^N(\infty)$ as $m \to \infty$. Hence, we have the telescoping property

$$Q^N(\infty) = \sum_{l_1=0}^{\infty} \cdots \sum_{l_N=0}^{\infty} \Delta Q^N_{(l_1,\ldots,l_N)} = \sum_{\ell \in \mathbb{N}^N} \Delta Q_\ell^N, \tag{20}$$

provided that $m_{l_1} \xrightarrow{l_1 \to \infty} \infty$, ..., $m_{l_N} \xrightarrow{l_N \to \infty} \infty$. The telescoping property is accompanied by a corresponding error factorization, i.e., the size of the increment ΔQ_ℓ^N can be bounded by a product of error terms depending on m_i.

We denote the computational work at level $\ell = (l_1, \ldots, l_N)$ for adding an increment ΔQ_ℓ^N in the telescoping sum by W_ℓ^N, and define the actual estimator for the quantity of interest $Q^N(\infty)$: given a set of multi-indices $\mathcal{I} \subset \mathbb{N}^N$, let

$$Q^N(\mathcal{I}) \coloneqq \sum_{\ell \in \mathcal{I}} \Delta Q_\ell^N.$$

Then the error is given by

$$\left|Q^N(\infty) - Q^N(\mathcal{I})\right| \leq \sum_{\ell \in \mathbb{N}^N \setminus \mathcal{I}} \left|\Delta Q_\ell^N\right|,$$

The construction of $\mathcal{I}$ will be done by profit thresholding, i.e., for a certain threshold value T, we add a multi-index ℓ to $\mathcal{I}$ provided that

$$\log\left(\frac{\left|\Delta Q_\ell^N\right|}{W_\ell^N}\right) \leq T.$$

(Actually, we take the error estimate instead of the true error.)

3.3 Brownian bridge construction

Let us denote $\{t_i\}_{i=0}^N$ the grid of time steps, then the BB construction [10] 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 the formula:

$$B_{t_j} = (1-\rho)B_{t_i} + \rho B_{t_k} + \sqrt{\rho(1-\rho)(k-i)\Delta t}z,\ z \sim \mathcal{N}(0,1), \tag{21}$$

where $\rho = \frac{j-i}{k-i}$. In particular, if N is a power of 2, then given $B_0 = 0$, BB generates the Brownian motion at times $T, T/2, T/4, 3T/4, \ldots$ according

$$\begin{aligned} B_T &= \sqrt{T} z_1 \\ B_{T/2} &= \frac{1}{2}(B_0 + B_T) + \sqrt{T/4} z_2 = \frac{\sqrt{T}}{2} z_1 + \frac{\sqrt{T}}{2} z_2 \\ B_{T/4} &= \frac{1}{2}(B_0 + B_{T/2}) + \sqrt{T/8} z_3 = \frac{\sqrt{T}}{4} z_1 + \frac{\sqrt{T}}{4} z_2 + \sqrt{T/8} z_3 \\ &\vdots \end{aligned}$$

(22)

where $\{z_j\}_{j=1}^N$ are independent standard normal variables. In BB construction given by (22), the most important values that determine the large scale structure of Brownian motion are the first components of $\mathbf{z} = (z_1, \ldots, z_N)$.

3.4 Richardson extrapolation

We recall that the Euler (often) scheme has weak order 1 so that

$$\left| \mathrm{E}\left[f(\widehat{X}_T^h) \right] - \mathrm{E}\left[f(X_T) \right] \right| \leq Ch \tag{23}$$

for some constant C, all sufficiently small h and suitably smooth f. It was shown that 23 can be improved to

$$\mathrm{E}\left[f(\widehat{X}_T^h) \right] = \mathrm{E}\left[f(X_T) \right] + ch + \mathcal{O}\left(h^2\right), \tag{24}$$

where c depends on f.

Applying 24 with discretization step $2h$, we obtain

$$\mathrm{E}\left[f(\widehat{X}_T^{2h}) \right] = \mathrm{E}\left[f(X_T) \right] + 2ch + \mathcal{O}\left(h^2\right), \tag{25}$$

implying

$$2\mathrm{E}\left[f(\widehat{X}_T^{2h}) \right] - \mathrm{E}\left[f(\widehat{X}_T^h) \right] = \mathrm{E}\left[f(X_T) \right] + \mathcal{O}\left(h^2\right), \tag{26}$$

For higher levels 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 the level of the Richardson extrapolation, and by $I(J,K)$ the approximation of $\mathrm{E}\left[f(\widehat{X}_T^{h_J}) \right]$ by terms up to level K (leading to a weak error of order K), then we have

$$I(J,K) = \frac{2^K \left[I(J,K-1) - I(J-1,K-1) \right]}{2^K - 1} + \mathcal{O}\left(h^{K+1}\right), \quad J = 1, 2, \ldots, K = 1, 2, \ldots \tag{27}$$

3.5 Root Finding

Without loss of generality, we can assume that the integration domain can be divided into two parts Ω_1, and Ω_2 such that the integrand f is smooth and positive in Ω_1 whereas $f(\mathbf{x}) = 0$ in Ω_2. Therefore,

$$If := \int_{\Omega_1} f(\mathbf{x}) d\mathbf{x} \tag{28}$$

This situation may arise when the integrand is non-differentiable or noncontinuous along the boundary between Ω_1 and Ω_2. For these problems, kinks and jumps can efficiently be identified by a one-dimensional root finding. Then, the kinks and jumps can be transformed to the boundary of integration domain such that they no longer deteriorate the performance of the numerical methods. In fact, we compute the zeros of the integrand with respect to the last dimension. In this dimension, then, e.g., Newton's method or bisection can be used to identify the point which separates Ω_1 and Ω_2. In our project, we use Newton 's iteration solver.

Let us call y the mapping such that: $y : \mathbf{z}_1 \to z^{\text{kink}}$, where z^{kink} is the "location of irregularity", *i.e.*, g is not smooth at the point $\phi \circ \Phi \circ \Psi(z^{\text{kink}}, \mathbf{z}_{-1})$. Generally, there might be (for given $\mathbf{z}_{-1}$

- no solution, i.e., the integrand in the definition of $h(\mathbf{z}_{-1})$ 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

$$\mathbf{z}_{-1} \mapsto h(\mathbf{z}_{-1}) \text{ and } \mathbf{z}_{-1} \mapsto \widehat{h}(\mathbf{z}_{-1})$$

are smooth, where $\widehat{h}$ denotes the numerical approximation of h based on a grid containing $y(\mathbf{z}_{-1})$. In particular, y itself should be smooth in $\mathbf{z}_{-1}$. This would already be challenging in practice in the third case. Moreover, in the general situation we expect the number of solutions y 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 (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 (and $\mathbf{z}_{-1}$ representing the Brownian bridge), 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.[1]

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

[1] 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.

linear combination z_1 of the terminal values of the driving Bm, such that the basket is a monotone function of z_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.

3.6 Description of the Domain Decomposition and Suitable Transformation

The payoff function is not smooth due to the nature of the option. In fact, the holder would not exercise the option if a purchase or sale of the underlying asset would lead to a loss. As a result, the discontinuity of the payoff function carries over to the integrand. In this case, The integrand shows a kink or even a jump with respect to a manifold. Since some (mixed) derivatives are not bounded at these manifolds, the smoothness requirements for the sparse grid method are clearly not fulfilled any more.

The first step consists of identifying the areas of discontinuity or non-differentiability. Then, we decompose the total integration domain Ω into sub-domains Ω_i, $i = 1, \ldots, n$ such that the integrand is smooth in the interior of Ω_i and such that all kinks and jumps are located along the boundary of these areas. This procedure results in integrating several smooth functions, instead of one discontinuous function. The total integral is then given as the sum of the separate integrals, *i.e.*

$$If := \int_\Omega f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{n} \int_{\Omega_i} f(\mathbf{x}) d\mathbf{x} \tag{29}$$

In this way, the fast convergence of SG can be regained whereas the costs only increase by a constant (the number of terms in the sum), provided the cost required for the decomposition is sufficiently small such that it can be neglected.

In general, such a decomposition is even more expensive than to integrate the function. Nevertheless, for some problem classes, the areas of discontinuity have a particular simple form, which allows to decompose the integration domain with costs that are much smaller than the benefit which results from the decomposition. In this work, we consider those cases.

In the literature, there two classes that have been tackled. In the first one, we have the information that the kinks are part of the integration domain where the integrand is zero and can thus be identified by root finding as proposed in [8].

In the second class, we have the information that the discontinuities are located on hyperplanes, which allows a decomposition first into polyhedrons and then into orthants as discussed in [9]. In this work, we start by the first class of problems.

4 Error discussion

4.1 Errors in smoothing

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

$$\widehat{h}(\mathbf{z}_{-1}) = \sum_{k=-K}^{K} \eta_k g\left(\phi \circ \Phi \circ \Psi\left(\zeta_k(y(\mathbf{z}_{-1})), \mathbf{z}_{-1}\right)\right),$$

for points $\zeta_k \in \mathbb{R}$ with $\zeta_0 = y$ and corresponding weights η_k.[2] However, in reality we have to numerical approximate y by $\overline{y}$ with error $|y - \overline{y}| \leq \delta$. Now, the actual integrand in $\mathbf{z}_{-1}$ becomes

$$\overline{h}(\mathbf{z}_{-1}) \coloneqq \sum_{k=-K}^{K} \eta_k g\left(\phi \circ \Phi \circ \Psi\left(\zeta_k(\overline{y}(\mathbf{z}_{-1})), \mathbf{z}_{-1}\right)\right),$$

which we cannot assume to be smooth anymore. On the other hand, if $\zeta_k(y)$ is a continuous function of y and y and $\overline{y}$ are continuous in $\mathbf{z}_{-1}$, 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 4.1. 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.

Remark 4.2. In some important cases, f may be trivial (e.g., $\equiv 0$). In these cases, we may be able to make sure that $\overline{y}$ never crosses the "location of non-smoothness". Then even $\overline{h}$ is smooth.

Remark 4.3. We expect that the global error of our procedure will be bounded by the weak error which is in our case of order $O(\Delta t)$. In this case, the overall complexity of our procedure will be of order $O(TOL^{-1})$. We note that this rate can be improved up to $O(TOL^{-\frac{1}{2}})$ if we use **Richardson extrapolation**. Another way that can improve the complexity could be based on **Cubature on Wiener Space** (This is left for a future work). The aimed complexity rate illustrates the contribution of our procedure which outperforms Monte Carlo forward Euler (MC-FE) and multi-level MC-FE, having complexity rates of order $O(TOL^{-3})$ and $O(TOL^{-2}log(TOL)^2)$ respectively.

Remark 4.4.

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.

4.2 Discussion about the Bias

4.3 Discussion about the Quadrature error

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

5 Numerical examples

5.1 Summary of numerical results

We conduct our experiments for 2 different examples under discretized BS model: Binary and Call options.

In Sections 5.3.1 and 5.4.1, we estimate the weak error (Bias) of MC combined with root finding, for the different examples, for 2 scenarios involving with/without Richardson extrapolation. The conclusions of this section are:

- Without Richardson extrapolation: For all cases, we get a weak error of order Δt, with different constants. Interestingly, for the call example, we needed to increase the number of quadrature points sufficiently, used for Laguerre quadrature, to observe the right behavior of the weak error (See tables (2, 12) for the corresponding Bias values as well the statistical errors.).

- With Richardson extrapolation: For the case of binary option, we get a weak error of order almost Δt^2. For the case of Call, we get a weak error of order higher than Δt^2 (See tables (7,17) for the corresponding Bias values as well the statistical errors.).

In Sections 5.3.2 and 5.4.2, we show tables and plots reporting the different errors involved in MC method (Bias and Statistical error) and in MISC (Quadrature error). We do this for each example. The quadrature error is computed by subtracting the MISC solution from the biased solution with huge number of samples (to kill the statistical error). I explain in each section, what each table and plot refers to. The conclusions of this section are:

- The Quadrature error of MISC is very stable with respect to $TOL_{\mathbf{MISC}}$, for all cases. (See figures (12,3,14,5)).

- MISC is outperforming significantly MC and MC+root finding methods. The ratio of gain is more significant for the Call option compared to Binary option (See tables (5,10,15)). See also figures (4,6,13) for the comparison of complexity for different methods.

- Richardson extrapolation has a great advantage over the rate as well the constant for complexity of MISC:

  - For the binary case: we may see from figure 7 and tables (4,5,9,10), that without using Richardson extrapolation, with 16 time steps, we achieve 1.4% of total relative error, with a cost of 1090 of CPU time. This cost is dramatically decreased when using Richardson extrapolation. In fact, with 16 steps in the finer level, we only pay 42 of CPU time to get a total relative error of 0.3%.
  - For the Call case: we may see from figure 15 and tables (14,15,19,20), that without using Richardson extrapolation, with 16 time steps, we achieve 0.46% of total relative error, with a cost of 656 of CPU time. This cost is dramatically decreased when using Richardson extrapolation. In fact, with 8 steps in the finer level, we only pay 135 of CPU time to get a total relative error of 0.25%.

5.2 The discretized 1D Black-Scholes

The second example that we test is the binary and call options under BS model where the process X is the discretized 1D Black-Scholes model and the payoff function g is the indicator or maximum function, and which has a kink. Precisely, we are interested in the 1-D lognormal example where the dynamics of the stock are given by

$$dX_t = \sigma X_t dW_t. \tag{30}$$

Assume that $\{W_t, 0 \leq t \leq T\}$ is a standard one-dimensional Brownian motion, and $(z_1, \ldots, z_N)$ are standard gaussian random variables, then, in the discrete case, we have

$$\begin{aligned} \Delta W_i &= (B_{t_{i+1}} - B_{t_i}) + \Delta t \frac{z_1}{\sqrt{T}} \\ &= \Delta B_i + \Delta t \frac{z_1}{\sqrt{T}}, \end{aligned} \tag{31}$$

implying that the numerical approximation of $X(T)$ satisfies

$$\begin{aligned} \overline{X}_T &= \Phi(\Delta t, z_1, \Delta B_0, \ldots, \Delta B_{N-1}), \\ &= \Phi(\Delta t, \Psi(z_1, \ldots, z_N)) \end{aligned} \tag{32}$$

for some path function Φ and Brwonian bridge map Ψ as described in Section 3.3.

As explained in Section 3, 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.

5.2.1 Determining the kink location

Exact location of the kink for the continuous problem

Let us denote y_* an invertible function that satisfies

$$X(T; y_*(x), B) = x. \tag{33}$$

We can easily prove that the expression of y_* for model given by (30) is given by

$$y_*(x) = \left(\log(x/x_0) + T\sigma^2/2\right) \frac{1}{\sqrt{T}\sigma}, \tag{34}$$

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

$$y_*(K) = \left(\log(K/x_0) + T\sigma^2/2\right) \frac{1}{\sqrt{T}\sigma}. \tag{35}$$

Exact location of the kink for the discrete problem

The discrete problem of model (30) is solved by simulating

$$\begin{aligned} \Delta X_{t_i} &= \sigma X_{t_i} \Delta W_i,\ 0 \le i \le N-1 \\ X_{t_{i+1}} - X_{t_i} &= \sigma X_{t_i} \left(W_{t_{i+1}} - W_{t_i} \right),\ 0 < i < N \end{aligned} \tag{36}$$

where $X(T_0) = X_0$ and $X(t_N) = X(T)$.

Using Brownian bridge construction given by (31), we have

$$\begin{aligned} X_{t_1} &= X_{t_0} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_0 \right] = X_{t_0} \left[1 + \sigma \Delta W_0 \right] \\ X_{t_2} &= X_{t_1} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_1 \right] = X_{t_1} \left[1 + \sigma \Delta W_1 \right] \\ &\vdots = \vdots = \vdots \\ X_{t_N} &= X_{t_{N-1}} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_{N-1} \right] = X_{t_{N-1}} \left[1 + \sigma \Delta W_{N-1} \right], \end{aligned} \tag{37}$$

implying that

$$\overline{X}(T) = X_0 \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} z_1 \Delta t + \sigma \Delta B_i \right]. \tag{38}$$

Therefore, in order to determine y_*, we need to solve

$$x = \overline{X}(T; y_*, B) = X_0 \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} y_*(x) \Delta t + \sigma \Delta B_i \right], \tag{39}$$

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

$$P(y_*(K)) = \prod_{i=0}^{N-1} \left[1 + \frac{\sigma}{\sqrt{T}} y_*(K) \Delta t + \sigma \Delta B_i \right] - \frac{K}{X_0}. \tag{40}$$

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

Approximate location of the discrete problem

Here, we try to find the roots of polynomial $P(y_*(K))$, given by (40), by using **Newton iteration method**. In this case, we need the expression $P' = \frac{dP}{dy_*}$. If we denote $f_i(y) = 1 + \frac{\sigma}{\sqrt{T}} y \Delta t + \sigma \Delta B_i$, then we can easily show that

$$P'(y) = \frac{\sigma \Delta t}{\sqrt{T}} \left(\prod_{i=0}^{N-1} f_i(y) \right) \left[\sum_{i=0}^{N-1} \frac{1}{f_i(y)} \right] \tag{41}$$

5.3 Results for the binary option example

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

$$h(\mathbf{z}_{-1}) = \int \mathbf{1}_{\Phi\circ\Psi(T;z_1,\mathbf{z}_{-1})>K} \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2)dy$$
$$= P(Y > y_*(K)), \tag{42}$$

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

$$\Phi \circ \Psi(T; y_*(x), \mathbf{z}_{-1}) = x \tag{43}$$

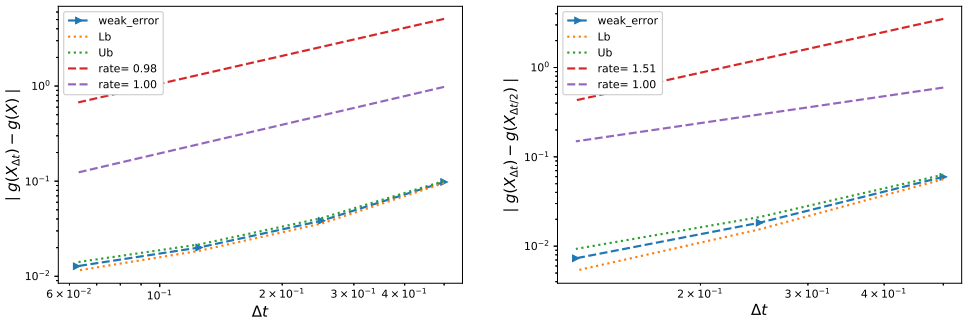
We get the kink point by running Newton iteration with a precision of 10^{-10}.

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.

5.3.1 Weak error plots

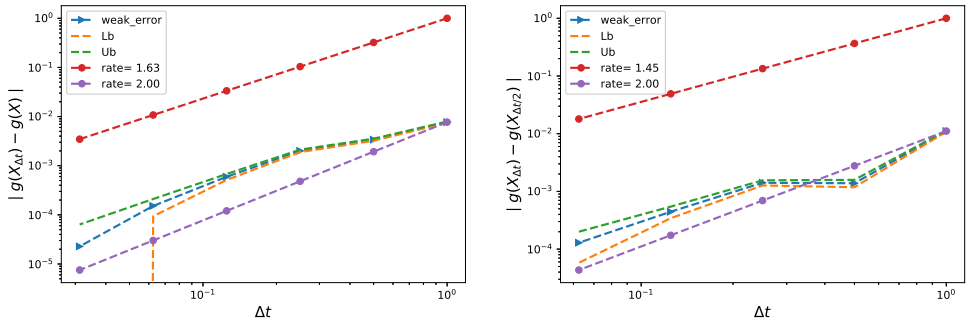
In this section, we include the results of weak error rates, for the binary option example, for 2 scenarios, without/with Richardson extrapolation (level 1). We note that the weak errors plotted here correspond to relative errors. The upper and lower bounds are 95% confidence interval.

From figure 1, we see that we get a weak error of order Δt. From figure 2, we observe an improvement in the rate and the constant when using level 1 of Richardson extrapolation. The corresponding values of the Bias are reported in tables (2,7).



(a) (b)

Figure 1: The rate of convergence of the weak error for the binary option, without Richardson extraploation, using MC with $M = 10^4$: a) $|\mathrm{E}\left[g(X_{\Delta t})\right] - g(X)|$ b) $\left|\mathrm{E}\left[g(X_{\Delta t}) - g(X_{\Delta t/2})\right]\right|$



(a) (b)

Figure 2: The rate of convergence of the weak error for the binary option, with Richardson extraploation (level 1), using MC with $M = 5.10^6$: a) $\left|\mathrm{E}\left[2g(X_{\Delta t/2}) - g(X_{\Delta t})\right] - g(X)\right|$ b) $\left|\mathrm{E}\left[3g(X_{\Delta t/2}) - g(X_{\Delta t}) - 2g(X_{\Delta t/4})\right]\right|$

5.3.2 Comparing relative errors

Without Richardson extrapolation

In this Section, we report the results for the binary option, using the different Methods: MISC, MC + root finding and MC, without Richardson extrapolation . We mention that for MISC we used a very small tolerance for the Newton solver, when solving the Kink point problem ($TOL_{\text{Newton}} = 10^{-10}$). We start by reporting the observed approximated values using different methods (See table 1. The biased values for MC method were computed using the values of Bias, reported in table 2. In table 3, we report the behavior of quadrature error with respect to MISC tolerance. We precise that the quadrature error is computed by substracting the MISC approximated value from the biased MC value. We report in red the values where MISC becomes stable (see also figure 3). Those values where used to compute the needed number of samples for MC (with and without root finding), to achieve similar magnitude for statistical error. Later, in table 4, we report the total relative error for all methods (Quadrature error + Bias for MISC and Statistical error + Bias for MC). We also report in table 5, the computational time needed for all different methods. We finally provide in figure 4, the complexity rates for the different involved methods.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	0.4620	0.4404	0.4299	0.4250
MISC ($TOl = 10^{-1}$)	0.4620	0.4404	0.4301	0.4250
MISC ($TOl = 5.10^{-2}$)	0.4620	0.4403	0.4300	0.4250
MISC ($TOl = 10^{-2}$)	0.4620	0.4406	0.4300	0.4250
MISC ($TOl = 10^{-3}$)	0.4620	0.4406	0.4301	0.4254
MISC ($TOl = 10^{-4}$)	0.4620	0.4406	0.4301	–
MC method ($M = 10^4$)	0.4620	0.4369	0.4292	0.4261

Table 1: Binary option price of the different methods for different number of time steps, without Richardson extrapolation.

Method \Steps	2	4	8	16
MC Bias ($M = 10^4$)	**0.0980** (0.0412)	**0.0384** (0.0162)	**0.0201** (0.0085)	**0.0127** (0.0053)
MC Statistical error ($M = 10^4$)	**1.2e − 03** (5.0e−04)	**1.2e − 03** (5.0e−04)	**8.0e − 04** (3.4e−04)	**6.5e − 04** (2.7e−04)

Table 2: Bias and Statistical errors of MC for computing Binary option price for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	**1.0e − 05** (2.4e−05)	**0.0083** (0.0035)	**0.0017** (0.0007)	**0.0026** (0.0011)
MISC ($TOl = 10^{-1}$)	**1.0e − 05** (2.4e−05)	**0.0083** (0.0035)	**0.0021** (0.0009)	**0.0026** (0.0011)
MISC ($TOl = 5.10^{-2}$)	**1.0e − 05** (2.4e−05)	**0.0081** (0.0034)	**0.0019** (0.0008)	**0.0026** (0.0011)
MISC ($TOl = 10^{-2}$)	**1.0e − 05** (2.4e−05)	**0.0088** (0.0037)	**0.0019** (0.0008)	**0.0026** (0.0011)
MISC ($TOl = 10^{-3}$)	**1.0e − 05** (2.4e−05)	**0.0088** (0.0037)	**0.0021** (0.0009)	**0.0017** (0.0007)
MISC ($TOl = 10^{-4}$)	**1.0e − 05** (2.4e−05)	**0.0088** (0.0037)	**0.0021** (0.0009)	– (–)

Table 3: Quadrature error of MISC to compute Binary option price of the different tolerances for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	**0.0980**	**0.0467**	**0.0218**	**0.0153**
MISC ($TOl = 10^{-1}$)	**0.0980**	**0.0467**	**0.0222**	**0.0153**
MISC ($TOl = 5.10^{-2}$)	**0.0980**	**0.0465**	**0.0220**	**0.0153**
MISC ($TOl = 10^{-2}$)	**0.0980**	**0.0472**	**0.0220**	**0.0153**
MISC ($TOl = 10^{-3}$)	**0.0980**	**0.0472**	**0.0222**	**0.0144**
MISC ($TOl = 10^{-4}$)	**0.0980**	**0.0472**	**0.0222**	–
MC+root finding	–	**0.0471**	**0.0221**	**0.0141**
MC	–	**0.0467**	**0.0222**	**0.0146**

Table 4: Total error of MISC and MC to compute Binary option price of the different tolerances for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors. We will not include later the first point, corresponding to number of time steps $N = 2$, since as from table 3, MISC is killing the quadrature error for that case.

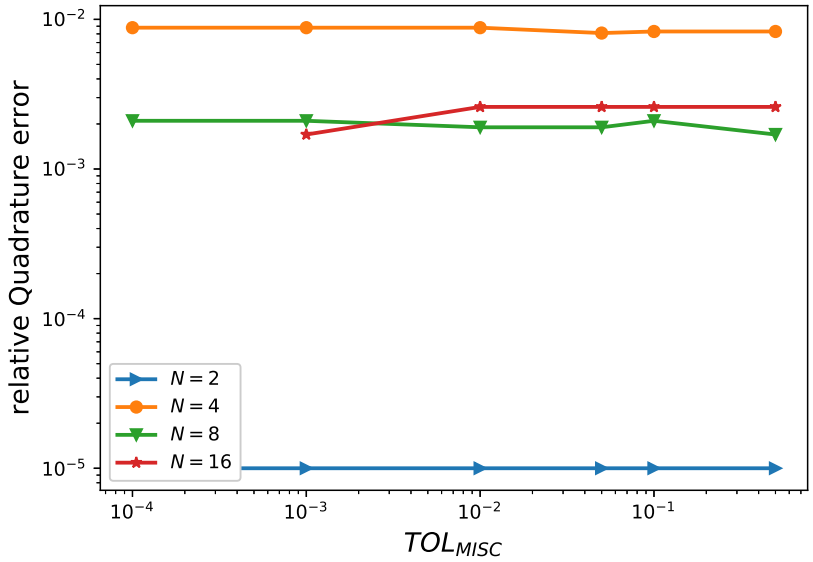


Figure 3: Relative quadrature error of MISC to compute Binary option price of the different tolerances for different number of time steps, without Richardson extrapolation.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	0.3	0.8	2	9
MISC ($TOl = 10^{-1}$)	0.3	0.8	9	49
MISC ($TOl = 5.10^{-2}$)	0.3	1.3	12	59
MISC ($TOl = 10^{-2}$)	0.3	2	14	63
MISC ($TOl = 10^{-3}$)	0.3	2	34	1090
MISC ($TOl = 10^{-4}$)	0.3	16	193	–
MC+root finding method	–	9	96	141
MC method	–	7	95	152
Ratio of (MC+root finding)/(MISC)	–	4.5	11	0.13
Ratio of (MC)/(MISC)	–	3.5	11	0.14

Table 5: Comparsion 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.

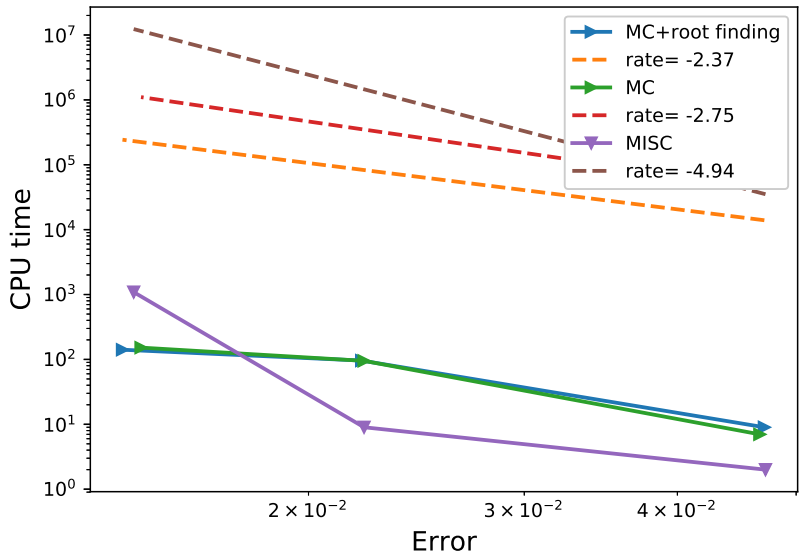


Figure 4: Complexity plot for MC and MISC for the case without Richardson extrapolation.

With Richardson extrapolation (level 1)

In this Section, we report the results for the binary option, using the different Methods: MISC, MC + root finding and MC, with Richardson extrapolation . We mention that for MISC we used a very small tolerance for the Newton solver, when solving the Kink point problem ($TOL_{\text{Newton}} = 10^{-10}$). We start by reporting the observed approximated values using different methods (See table 6. The biased values for MC method were computed using the values of Bias, reported in table 7. In table 8, we report the behavior of quadrature error with respect to MISC tolerance. We precise that the quadrature error is computed by substracting the MISC approximated value from the biased MC value. We report in red the values where MISC becomes stable (see also figure 5). Those values where used to compute the needed number of samples for MC (with and without root finding), to achieve similar magnitude for statistical error. Later, in table 9, we report the total relative error for all methods (Quadrature error + Bias for MISC and Statistical error + Bias for MC). We also report in table10, the computational time needed for all different methods. We finally provide in figure 6, the complexity rates for the different involved methods, as well a comparison between the two versions of MISC (without/with Richardson extrapolation) in figure 7.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	0.4239	0.4188	0.4191	0.4200
MISC ($TOl = 10^{-1}$)	0.4239	0.4188	0.4191	0.4199
MISC ($TOl = 5.10^{-2}$)	0.4239	0.4188	0.4190	0.4199
MISC ($TOl = 10^{-2}$)	0.4239	0.4192	0.4194	0.4199
MISC ($TOl = 10^{-3}$)	0.4239	0.4192	0.4199	0.4205
MISC ($TOl = 10^{-4}$)	0.4239	0.4193	0.4199	–
MC method ($M = 5.10^6$)	0.4240	0.4224	0.4216	0.4210

Table 6: Binary option price of the different methods for different number of time steps, with Richardson extrapolation (level 1).

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MC Bias ($M = 5.10^6$)	**0.0077** (0.0032)	**0.0039** (0.0016)	**0.0020** (0.0008)	**0.0006** (0.0003)
MC Statistical error ($M = 5.10^6$)	**1.1e − 04** (4.6e−05)	**8.4e − 05** (3.5e−05)	**6.0e − 05** (2.5e−05)	**4.2e − 05** (1.8e−05)

Table 7: Bias and Statistical errors of MC for computing Binary option price for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	**2.4e − 04** (1.0e−04)	**8.6e − 03** (3.6e−03)	**5.9e − 03** (2.5e−03)	**2.4e − 03** (1.0e−03)
MISC ($TOl = 10^{-1}$)	**2.4e − 04** (1.0e−04)	**8.6e − 03** (3.6e−03)	**5.9e − 03** (2.5e−03)	**2.6e − 03** (1.1e−04)
MISC ($TOl = 5.10^{-2}$)	**2.4e − 04** (1.0e−04)	**8.6e − 03** (3.6e−03)	**6.2e − 03** (2.6e−03)	**2.6e − 03** (1.1e−04)
MISC ($TOl = 10^{-2}$)	**2.4e − 04** (1.0e−04)	**7.6e − 03** (3.2e−03)	**5.2e − 03** (2.2e−03)	**2.6e − 03** (1.1e−04)
MISC ($TOl = 10^{-3}$)	**2.4e − 04** (1.0e−04)	**7.6e − 03** (3.2e−03)	**4.0e − 03** (1.7e−03)	**1.2e − 03** (5.e−04)
MISC ($TOl = 10^{-4}$)	**2.4e − 04** (1.0e−04)	**7.4e − 03** (3.1e−03)	**4.0e − 03** (1.7e−03)	()

Table 8: Quadrature error of MISC to compute Binary option price of the different tolerances for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors.

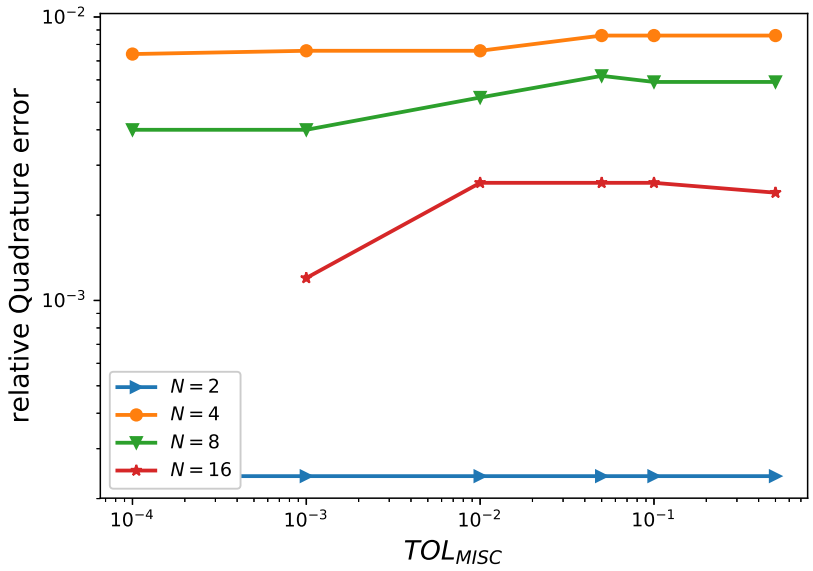


Figure 5: Relative quadrature error of MISC to compute Binary option price of the different tolerances for different number of time steps, with Richardson extrapolation.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	**0.0079**	**0.0125**	**0.0079**	**0.0030**
MISC ($TOl = 10^{-1}$)	**0.0079**	**0.0125**	**0.0079**	**0.0032**
MISC ($TOl = 5.10^{-2}$)	**0.0079**	**0.0125**	**0.0082**	**0.0032**
MISC ($TOl = 10^{-2}$)	**0.0079**	**0.0115**	**0.0072**	**0.0032**
MISC ($TOl = 10^{-3}$)	**0.0079**	**0.0115**	**0.0060**	**0.0018**
MISC ($TOl = 10^{-4}$)	**0.0079**	**0.0113**	**0.0060**	–
MC+root finding	–	**0.0111**	**0.0070**	**0.0035**
MC	–	**0.0116**	**0.0073**	**0.0029**

Table 9: Total error of MISC and MC to compute Binary option price of the different tolerances for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors. We will not include later the first point, corresponding to number of time steps $N = 2$, since as from table 8, MISC is killing the quadrature error for that case.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	0.3	1	4	9
MISC ($TOl = 10^{-1}$)	0.3	1	8	42
MISC ($TOl = 5.10^{-2}$)	0.3	1	10	72
MISC ($TOl = 10^{-2}$)	0.3	4	15	78
MISC ($TOl = 10^{-3}$)	0.3	4	100	2253
MISC ($TOl = 10^{-4}$)	0.3	68	392	
MC+root finding method	–	48	62	122
MC	–	12	23	147
Ratio of (MC+root finding)/(MISC)	–	12	4.1	2.9
Ratio of (MC)/(MISC)	–	3	1.5	3.5

Table 10: Comparsion of the computational time of MC and MISC, used to compute Binary option price for different number of time steps, with Richardson extrapolation (level 1). The average computational time of MC is computed over 10 runs.

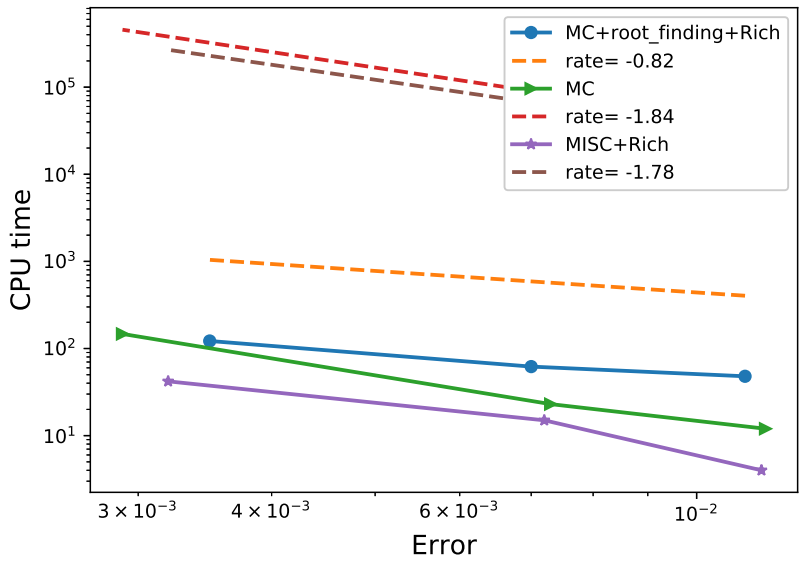


Figure 6: Complexity plot for MC and MISC for the case with Richardson extrapolation.

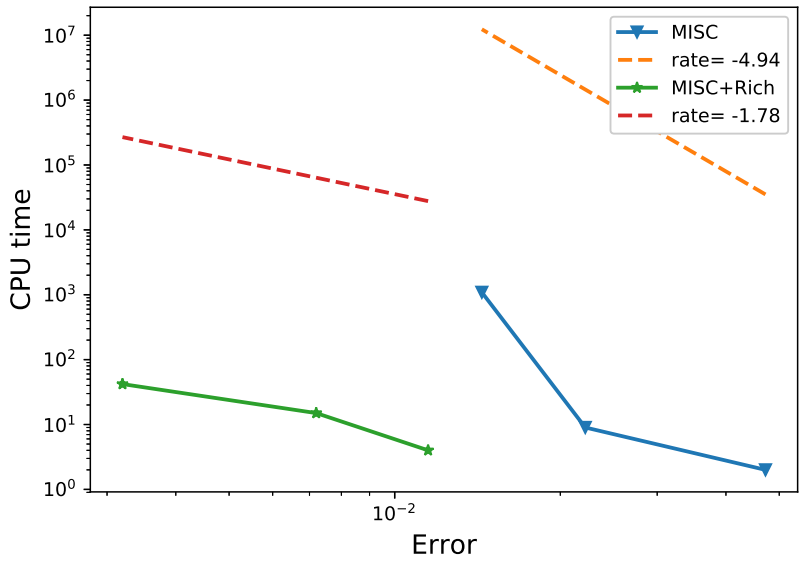


Figure 7: Complexity plot for MISC without and with Richardson extrapolation, for the Binary option.

5.4 Results for the Call option example

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

$$h(\mathbf{z}_{-1}) = \int_{\Omega} \max\left(\Phi \circ \Psi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dy \tag{44}$$

We get the kink point by running Newton iteration with a precision of 10^{-10}. We decompose the total integration domain Ω into sub-domains Ω_i, $i = 1, 2$ such that the integrand is smooth in the interior of Ω_i 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.*

$$\begin{aligned} h(\mathbf{z}_{-1}) &:= \int_{\Omega} \max\left(\Phi \circ \Psi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dy \\ &= \sum_{i=1}^{2} \int_{\Omega_i} \max\left(\Phi \circ \Psi(T; z_1, \mathbf{z}_{-1}) - K, 0\right) \frac{1}{\sqrt{2\pi}} \exp(-z_1^2/2) dy, \end{aligned} \tag{45}$$

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

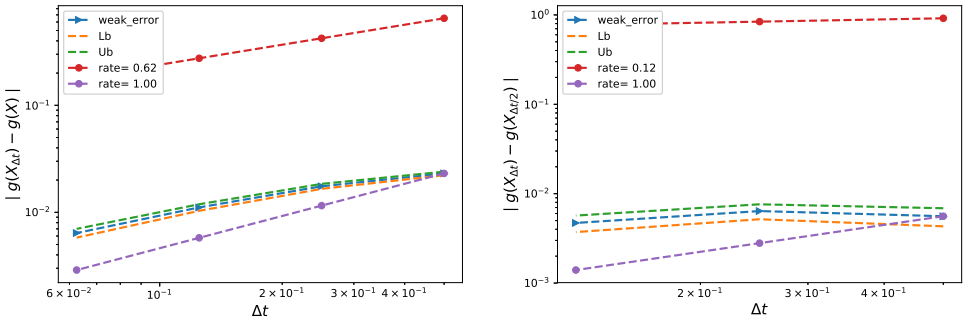
The paramters that we used in our numerical experiments are: $T = 1$, $\sigma = 0.4$ and $S_0 = K = 100$. The exact value of this case is 15.85193755.

5.4.1 Weak error plots

In this section, we include the results of weak error rates for the call option for 2 scenarios, without/with Richardson extrapolation (level 1). We note that the weak errors plotted here correspond to relative errors. We note that in order to get accurate estimates of the bias, we needed to kill the laguerre quadrature error by increasing β (β: the number of quadrature points used for Laguere to get $h(\mathbf{z}_{-1})$). For illustration, we show to cases $\beta = 10$ and $\beta = 32$. We mention that from our numerical experiences that the bias did not change as we increase β greater than 32 points.

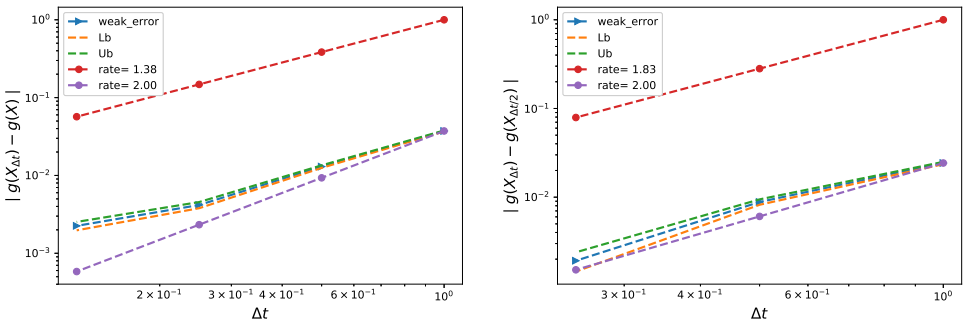
Focusing on the results produced by seeting $\beta = 32$ (which gives the right behavior to be observed for the bias), we can see from figure 10 that we get a weak error of order Δt for the case without Richardson extrapolation. From figure 11, we observe an improvement in the rate and the constant when using level 1 of Richardson extrapolation, approximately of order Δt^2. For all the plots below, the upper and lower bounds are 95% confidence interval.

$\beta = 10$



(a) (b)

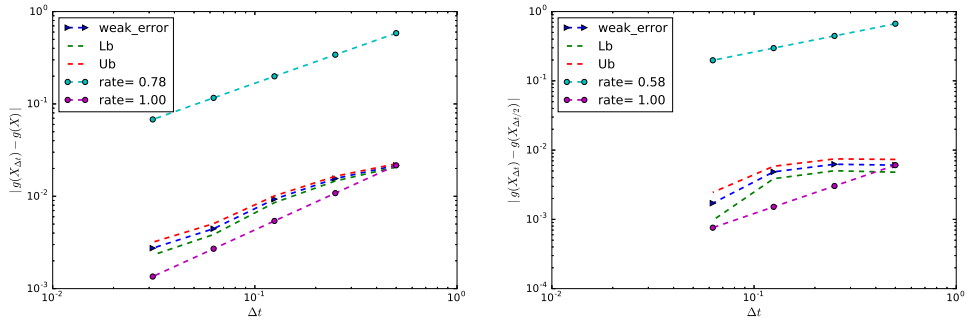
Figure 8: The rate of convergence of the weak error for the call option, without Richardson extraploation, using MC with $M = 10^5$: a) $|\mathrm{E}\left[g(X_{\Delta t})\right] - g(X)|$ b) $\left|\mathrm{E}\left[g(X_{\Delta t}) - g(X_{\Delta t/2})\right]\right|$



(a) (b)

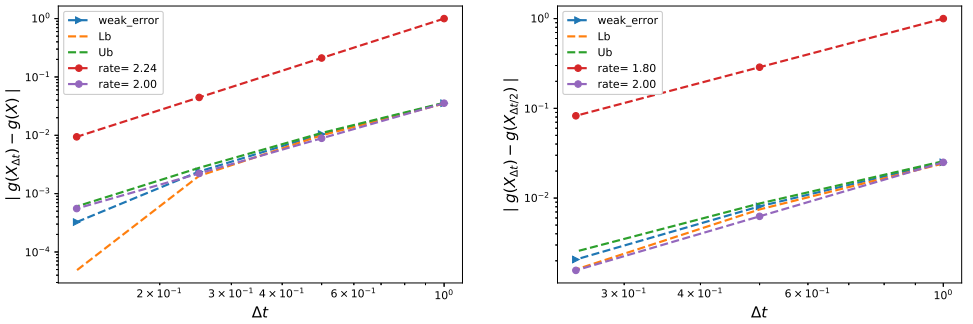
Figure 9: The rate of convergence of the weak error for the call option with Richardson extraploation, using MC with $M = 10^6$: a) $\left|\mathrm{E}\left[2g(X_{\Delta t/2}) - g(X_{\Delta t})\right] - g(X)\right|$ b) $\left|\mathrm{E}\left[3g(X_{\Delta t/2}) - g(X_{\Delta t}) - 2g(X_{\Delta t/4})\right]\right|$

$\beta = 32$



(a) (b)

Figure 10: The rate of convergence of the weak error for the call option, without Richardson extraploation, using MC with $M = 10^5$: a) $|\mathrm{E}\left[g(X_{\Delta t})\right] - g(X)|$ b) $\left|\mathrm{E}\left[g(X_{\Delta t}) - g(X_{\Delta t/2})\right]\right|$



(a) (b)

Figure 11: The rate of convergence of the weak error for the call option with Richardson extraploation (level 1), using MC with $M = 5.10^6$: a) $\left|\mathrm{E}\left[2g(X_{\Delta t/2}) - g(X_{\Delta t})\right] - g(X)\right|$ b) $\left|\mathrm{E}\left[3g(X_{\Delta t/2}) - g(X_{\Delta t}) - 2g(X_{\Delta t/4})\right]\right|$

5.4.2 Comparing relative errors

Without Richardson extrapolation

In this Section, we report the results for the Call option, using the different Methods: MISC, MC + root finding and MC, without Richardson extrapolation . We mention that for MISC we used a very small tolerance for the Newton solver, when solving the Kink point problem ($TOL_{\text{Newton}} = 10^{-10}$), we also used $\beta = 32$ (number of Laguerre quadrature points). We start by reporting the observed approximated values using different methods (See table 11). The biased values for MC method were computed using the values of Bias, reported in table 12. In table 13, we report the behavior of quadrature error with respect to MISC tolerance. We precise that the quadrature error is computed by substracting the MISC approximated value from the biased MC value. We report in red the values where MISC becomes stable (see also figure 12). Those values where used to compute the needed number of samples for MC (with and without root finding), to achieve similar

magnitude for statistical error. Later, in table 14, we report the total relative error for all methods (Quadrature error + Bias for MISC and Statistical error + Bias for MC). We also report in table 15, the computational time needed for all different methods. We finally provide in figure 13, the complexity rates for the different involved methods.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}, \beta = 32$)	16.184	16.070	15.998	15.930
MISC ($TOl = 10^{-1}, \beta = 32$)	16.184	16.070	15.996	15.928
MISC ($TOl = 5.10^{-2}, \beta = 32$)	16.184	16.070	15.996	15.928
MISC ($TOl = 10^{-2}, \beta = 32$)	16.184	16.103	15.996	15.928
MISC ($TOl = 10^{-3}, \beta = 32$)	16.184	16.103	15.996	–
MC method ($M = 10^5$)	16.194	16.099	15.999	15.923

Table 11: Call option price of the different methods for different number of time steps, without Richardson extrapolation.

Method \Steps	2	4	8	16
MC Bias ($M = 10^5$)	**0.0216** (0.3424)	**0.0156** (0.2473)	**0.0093** (0.1474)	**0.0045** (0.0713)
MC Statistical error ($M = 10^5$)	**4.4e − 04** (7.5e−03)	**4.7e − 04** (7.5e−03)	**4.0e − 04** (6.3e−03)	**3.1e − 04** (4.9e−03)

Table 12: Bias and Statistical errors of MC for computing Call option price for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	**0.0007** (0.0103)	**0.0018** (0.0290)	**6.3e − 05** (0.0010)	**0.0004** (0.0070)
MISC ($TOl = 10^{-1}$)	**0.0007** (0.0103)	**0.0018** (0.0290)	**0.0002** (0.0030)	**0.0001** (0.0020)
MISC ($TOl = 5.10^{-2}$)	**0.0007** (0.0103)	**0.0018** (0.0290)	**0.0002** (0.0030)	**0.0001** (0.0020)
MISC ($TOl = 10^{-2}$)	**0.0007** (0.0103)	**0.0003** (0.0040)	**0.0002** (0.0030)	**0.0001** (0.0020)
MISC ($TOl = 10^{-3}$)	**0.0007** (0.0103)	**0.0003** (0.0040)	**0.0002** (0.0030)	– (–)

Table 13: Quadrature error of MISC to compute Call option price of the different tolerances for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors.

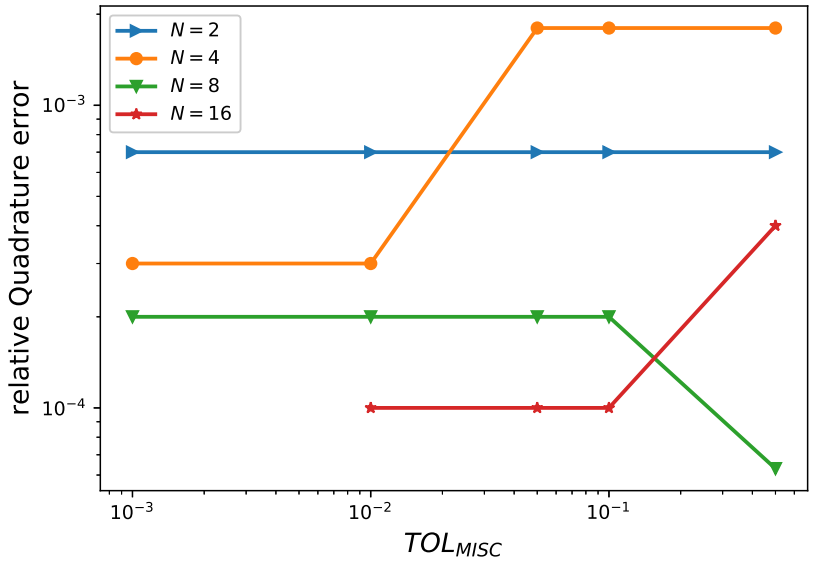


Figure 12: Relative quadrature error of MISC to compute Call option price of the different tolerances for different number of time steps, without Richardson extrapolation.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	**0.0223**	**0.0174**	**0.0094**	**0.0049**
MISC ($TOl = 10^{-1}$)	**0.0223**	**0.0174**	**0.0095**	**0.0046**
MISC ($TOl = 5.10^{-2}$)	**0.0223**	**0.0174**	**0.0095**	**0.0046**
MISC ($TOl = 10^{-2}$)	**0.0223**	**0.0159**	**0.0095**	**0.0046**
MISC ($TOl = 10^{-3}$)	**0.0223**	**0.0159**	**0.0095**	
MC +root finding	**0.0223**	**0.0159**	**0.0095**	**0.0046**
MC	**0.0223**	**0.0159**	**0.0095**	**0.0046**

Table 14: Total error of MISC and MC to compute Call option price of the different tolerances for different number of time steps, without Richardson extrapolation. The numbers between parentheses are the corresponding absolute errors.

Method \Steps	2	4	8	16
MISC ($TOl = 5.10^{-1}$)	0.3	3	17	473
MISC ($TOl = 10^{-1}$)	0.3	3	58	656
MISC ($TOl = 5.10^{-2}$)	0.3	3	73	731
MISC ($TOl = 10^{-2}$)	0.3	6	108	1972
MISC ($TOl = 10^{-3}$)	0.3	28	264	–
MC method +root finding	1328	8140	21400	70200
MC method	1450	9990	32790	158108
Ratio of (MC+root finding)/(MISC)	$4.4e+03$	$1.4e+03$	369	107
Ratio of (MC)/(MISC)	$4.8e+03$	1665	565	241

Table 15: Comparison of the computational time of MC and MISC, used to compute Call option price for different number of time steps, without Richardson extrapolation

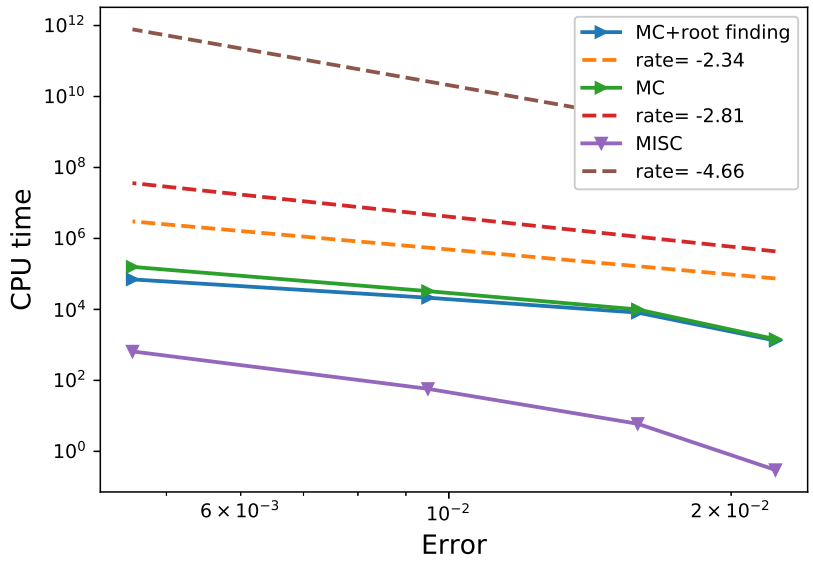


Figure 13: Complexity plot for MC and MISC for the case without Richardson extrapolation.

With Richardson extrapolation (level 1)

In this Section, we report the results for the Call option, using the different Methods: MISC, MC + root finding and MC, with Richardson extrapolation . We mention that for MISC we used a very small tolerance for the Newton solver, when solving the Kink point problem ($TOL_{\text{Newton}} = 10^{-10}$), we also used $\beta = 32$ (number of Laguerre quadrature points). We start by reporting the observed approximated values using different methods (See table 16. The biased values for MC method were computed using the values of Bias, reported in table 17. In table 18, we report the behavior of quadrature error with respect to MISC tolerance. We precise that the quadrature error is computed by substracting the MISC approximated value from the biased MC value. We report in red the values where MISC becomes stable (see also figure 14). Those values where used to compute the needed number of samples for MC (with and without root finding), to achieve similar magnitude for statistical error. Later, in table 19, we report the total relative error for all methods (Quadrature error + Bias for MISC and Statistical error + Bias for MC). We also report in table20, the computational time needed for all different methods. We finally provide in figure 15, the comparison between the two versions of MISC (without/with Richardson extrapolation).

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	16.4108	16.0254	15.8912	15.8621
MISC ($TOl = 10^{-1}$)	16.4108	16.0254	15.8883	15.8603
MISC ($TOl = 5.10^{-2}$)	16.4108	16.0218	15.8885	15.8600
MISC ($TOl = 10^{-2}$)	16.4108	16.0218	15.8888	15.8595
MISC ($TOl = 10^{-3}$)	16.4108	16.0207	15.8885	–
MC method ($M = 5.10^6$)	16.4147	16.0184	15.8900	15.8567

Table 16: Call option price of the different methods for different number of time steps, with Richardson extrapolation (level 1).

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MC Bias ($M = 5.10^6$)	**0.0355** (0.5627)	**0.0105** (0.1664)	**0.0024** (0.0380)	**0.0003** (0.0048)
MC Statistical error ($M = 5.10^6$)	**2.8e − 04** (4.4e−03)	**2.4e − 04** (3.8e−03)	**1.9e − 04** (3.0e−03)	**1.4e − 04** (2.2e−03)

Table 17: Bias and Statistical errors of MC for computing Call option price for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	**2.5e − 04** (0.0039)	**4.4e − 04** (0.0070)	**7.6e − 05** (0.0012)	**3.4e − 04** (0.0054)
MISC ($TOl = 10^{-1}$)	**2.5e − 04** (0.0039)	**4.4e − 04** (0.0070)	**1.1e − 04** (0.0017)	**2.3e − 04** (0.0036)
MISC ($TOl = 5.10^{-2}$)	**2.5e − 04** (0.0039)	**2.1e − 04** (0.0034)	**9.5e − 05** (0.0015)	**2.1e − 04** (0.0033)
MISC ($TOl = 10^{-2}$)	**2.5e − 04** (0.0039)	**2.1e − 04** (0.0034)	**7.6e − 05** (0.0012)	**1.7e − 04** (0.0028)
MISC ($TOl = 10^{-3}$)	**2.5e − 04** (0.0039)	**1.5e − 04** (0.0023)	**9.5e − 05** (0.0015)	()

Table 18: Quadrature error of MISC to compute Call option price of the different tolerances for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors.

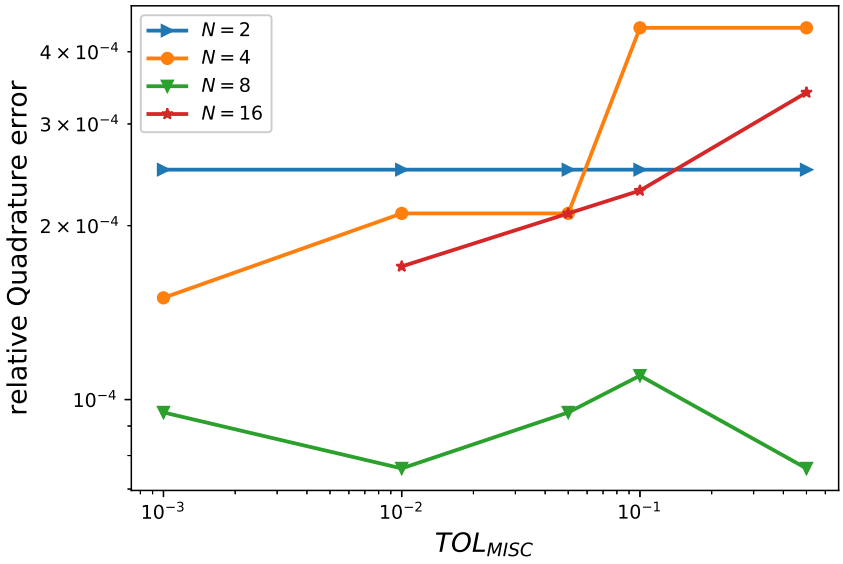


Figure 14: Relative quadrature error of MISC to compute Call option price of the different tolerances for different number of time steps, with Richardson extrapolation.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	**0.0358**	**0.0109**	**0.0025**	**0.0006**
MISC ($TOl = 10^{-1}$)	**0.0358**	**0.0109**	**0.0025**	**0.0005**
MISC ($TOl = 5.10^{-2}$)	**0.0358**	**0.0107**	**0.0025**	**0.0005**
MISC ($TOl = 10^{-2}$)	**0.0358**	**0.0107**	**0.0025**	**0.0005**
MISC ($TOl = 10^{-3}$)	**0.0358**	**0.0107**	**0.0025**	

Table 19: Total error of MISC to compute Call option price of the different tolerances for different number of time steps, with Richardson extrapolation (level 1). The numbers between parentheses are the corresponding absolute errors.

Method \Steps	$1-2$	$2-4$	$4-8$	$8-16$
MISC ($TOl = 5.10^{-1}$)	0.3	4	56	713
MISC ($TOl = 10^{-1}$)	0.3	4	107	1126
MISC ($TOl = 5.10^{-2}$)	0.3	9	135	1253
MISC ($TOl = 10^{-2}$)	0.3	9	186	3540
MISC ($TOl = 10^{-3}$)	0.3	63	836	

Table 20: Comparsion of the computational time of MISC, used to compute Call option price for different number of time steps, with Richardson extrapolation (level 1)

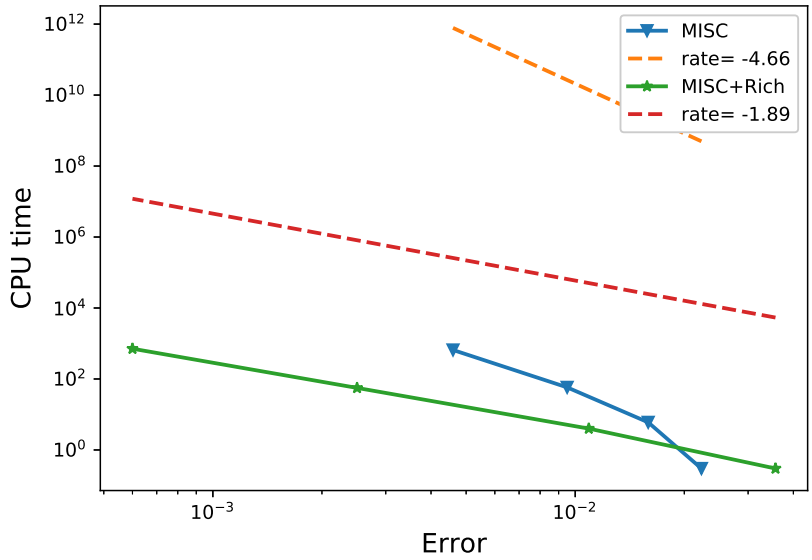


Figure 15: Complexity plot for MISC without and with Richardson extrapolation, for the Call option.

5.5 The basket option with the smoothing trick as in [4]

The first experiment that we consider is the pricing of a European basket call option in a Black-Scholes model. The basket is composed of d assets ($d = 3, 8, 25$) and we use the same trick of smoothing the integrand that was proposed in [4]. In this case, the dimension of the parameter space $N = d - 1$. The interpolation over the parameter space is based on the tensorized Lagrangian interpolation technique with Gaussian points.

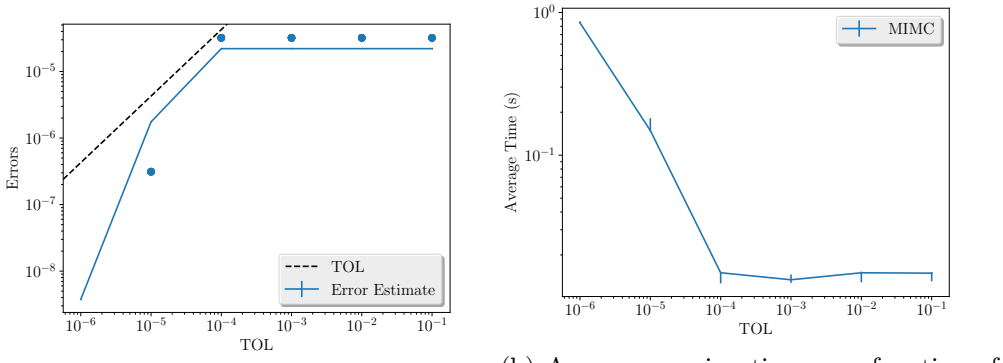
5.5.1 Results using MISC

In table 21, we summarize the observed complexity rates for different tested settings for the basket example. From this table, we can check that even with the 25 dimensional case, the complexity rate in terms of the elapsed time is at least order 1, which is better than MC, which is 2. Detailed plots for each case are given by figures (16, 17) for $d = 3$, figures (18, 19) for $d = 8$ and figures (20, 21) for $d = 25$. Mainly, from the plots, we checked that we achive the prescribed tolerance using MISC, the convergence rates of mixed differences which is a basic assumption for using MISC (we observe exponential decay of error rates wrt to the number of quadrature points) and finally the complexity rates. In the next Section, we try to extend these results to the time stepping framework.

# assets \	3	8	25
rate	$-1/3$	$-9/20$	$-16/25$

Table 21: Complexity rates of the different experiments for the basket option using BS model

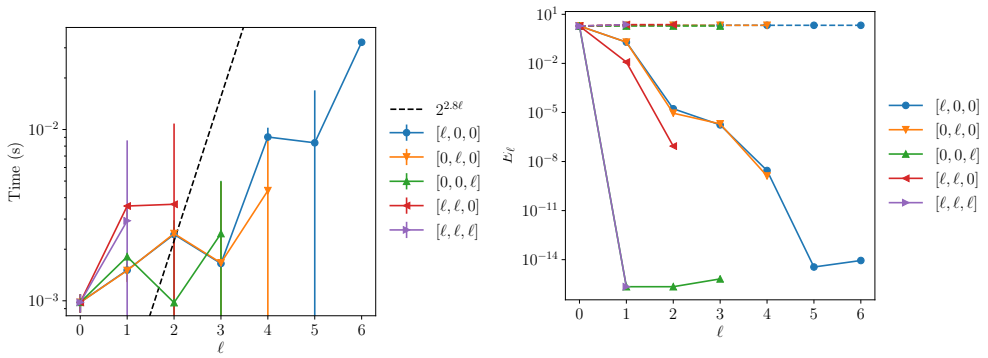
Case of 3-dimensional Basket



(a) Error estimate

(b) Average running time as a function of TOL

Figure 16: Convergence and complexity results for the 3-dimensional basket option using BS model.

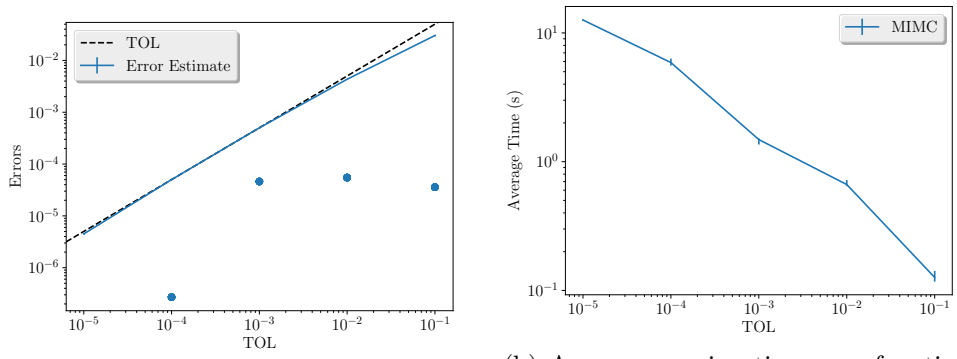


(a) Average Computational time per level.

(b) The convergence rate of mixed differences per level.

Figure 17: Convergence and work rates for discretization levels for the 3-dimensional basket option using BS model.

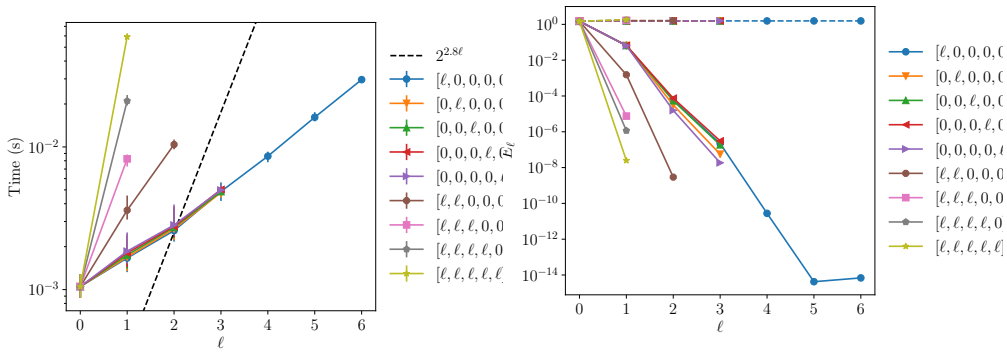
Case of 8-dimensional Basket



(a) Error estimate

(b) Average running time as a function of TOL

Figure 18: Convergence and complexity results for the 8-dimensional basket option using BS model.

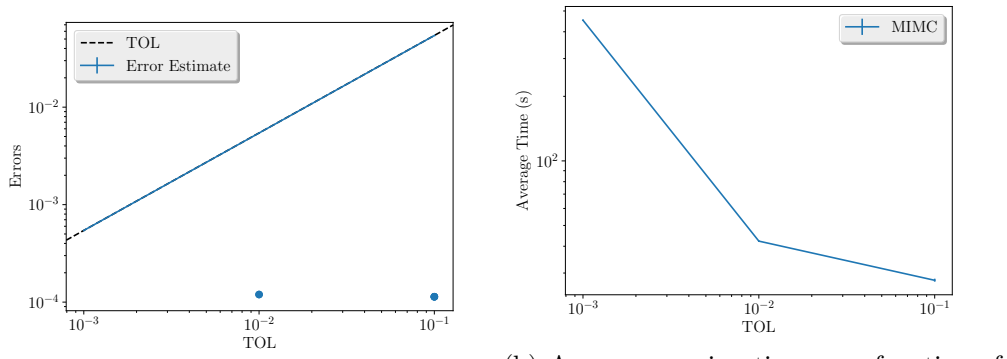


(a) Average Computational time per level.

(b) The convergence rate of mixed differences per level.

Figure 19: Convergence and work rates for discretization levels for the 8-dimensional basket option using BS model.

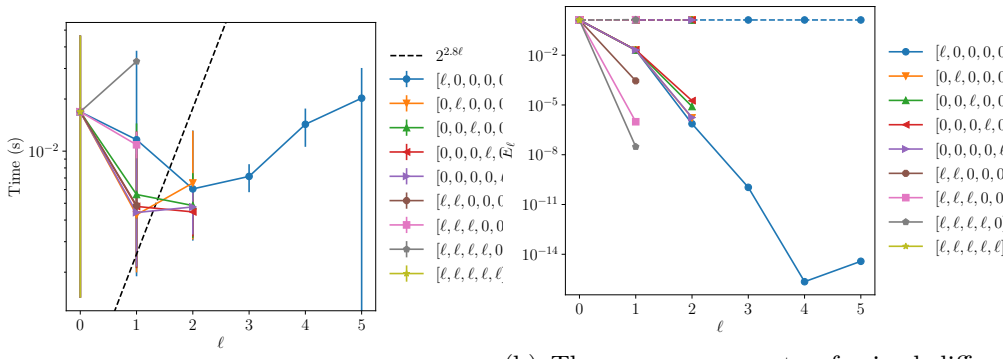
Case of 25-dimensional Basket



(a) Error estimate

(b) Average running time as a function of TOL

Figure 20: Convergence and complexity results for the 25-dimensional basket option using BS model.



(a) Average Computational time per level.

(b) The convergence rate of mixed differences per level.

Figure 21: Convergence and work rates for discretization levels for the 25-dimensional basket option using BS model.

5.6 The basket option under time stepping framework

5.7 Results for the CIR process

The impact of the Brownian bridge will disappear in the limit, which may make the effect of the smoothing, but also of the errors in the kink location difficult to identify . For this reason, we study a more complicated 1-dimensional problem. In fact, we use a CIR process. To avoid complications at the boundary, we use nice parameter choices, such that the discretized process is very unlikely to hit the boundary (Feller condition).

The CIR model specifies that the instantaneous interest rate follows the SDE given by

$$dX_t = a(b - X_t)dt + \sigma\sqrt{X_t}dW_t,\ X(0) = X_0 > 0 \tag{46}$$

where and $a > 0$, $b > 0$ and $\sigma > 0$, are the parameters. The parameter a corresponds to the speed of adjustment, b to the mean and σ to volatility.

If the parameters obey the following condition (known as the Feller condition) then the process X_t is strictly positive

$$2ab \geq \sigma^2. \tag{47}$$

The SDE (46) is not explicitly solvable. In general there are two ways to do it, namely, exact simulation methods and approximation schemes. Exact simulation in general requires more time than a simulation with approximation schemes (Up to a factor 10). Therefore, it is used to compute expectations that depend on the values of the process at just a few fixed times. However, for expectations that depends on all the path (such as integrals) discretization schemes should be preferred.

The main problem when discretizing a CIR process using Euler scheme is that it can lead to negative values for which the square root is not defined. In fact, if we consider the following straightforward Euler scheme on the time interval $[0, T]$ for a CIR process $X(t)$:

$$\overline{X}(t_{i+1}) = \overline{X}(t_i) + a(b - \overline{X}(t_i))\Delta t_i + \sigma\sqrt{\overline{X}(t_i)}\Delta W_i, \tag{48}$$

then it can lead to negative values since the Gaussian increment is not bounded from below. Thus, this scheme is not well defined.

Many modified Euler schemes were proposed to overcome this issue. For instance, Deelstra and Delbaen [7] have propose the full truncation scheme given by

$$\overline{X}(t_{i+1}) = \overline{X}(t_i) + a(b - \overline{X}(t_i))\Delta t_i + \sigma\sqrt{\overline{X}(t_i)^+}\Delta W_i. \tag{49}$$

Higham and Mao [16] proposed the following scheme

$$\overline{X}(t_{i+1}) = \overline{X}(t_i) + a(b - \overline{X}(t_i))\Delta t_i + \sigma\sqrt{\left|\overline{X}(t_i)\right|}\Delta W_i. \tag{50}$$

Lord et al [15] proposed the following scheme

$$\overline{X}(t_{i+1}) = \overline{X}(t_i) + a(b - \overline{X}(t_i)^+)\Delta t_i + \sigma\sqrt{\overline{X}(t_i)^+}\Delta W_i. \tag{51}$$

Diop proposed in [5] proposed the reflection scheme, given by

$$\overline{X}(t_{i+1}) = \mid \overline{X}(t_i) + a(b - \overline{X}(t_i))\Delta t_i + \sigma\sqrt{\overline{X}(t_i)}\Delta W_i \mid . \tag{52}$$

Also, implicit and higher order schemes were proposed by Alfonsi [1, 2, 3].

Those modified Euler schemes were studied numerically in [1]. It is observed that when σ is small enough, typically $\sigma^2 \leq 2a$, these schemes have a weak error of order one and a strong error of order $1/2$. However, when σ is getting large, say $\sigma^2 \gg 4a$, the convergence of all these schemes is degraded. As observed by Lord et al. [15], the schemes (49) and (51) behave better than the schemes (50) and (52). In fact, when σ gets large, the CIR process spends more time close to zero and get stuck in the neighbourhood of zero when σ is really large. When the scheme takes a negative value, the absolute value in both schemes ((50),(52)) produces a noise that pushes the scheme away from zero. On the other hand, the positive part in truncation schemes cancels the noise when the scheme gets negative, which better reproduces the behaviour of the CIR process.

In the following, we use the scheme given by (49) to simulate the discrete CIR process.

5.7.1 Location of the kink for the discrete problem: Using full truncation scheme

The full truncation scheme simulating the CIR process is given by (49). Here we are interested in finding the location of the kink for hockey-stick function.

Using Brownian bridge construction given by (**??**), we have

$$\begin{aligned}
X_{t_1} &= X_{t_0}\left[1 - a\Delta t\right] + \sigma\sqrt{X_{t_0}^+}\left[Y\frac{\Delta t}{\sqrt{T}} + \Delta B_0\right] + ab\Delta t \\
X_{t_2} &= X_{t_1}\left[1 - a\Delta t\right] + \sigma\sqrt{X_{t_1}^+}\left[Y\frac{\Delta t}{\sqrt{T}} + \Delta B_1\right] + ab\Delta t \\
&\vdots = \vdots = \vdots \\
X_{t_N} &= X_{t_{N-1}}\left[1 - a\Delta t\right] + \sigma\sqrt{X_{t_{N-1}}^+}\left[Y\frac{\Delta t}{\sqrt{T}} + \Delta B_{N-1}\right] + ab\Delta t,
\end{aligned} \tag{53}$$

to simplify the notation we set $f_i(y) := X_{t_i}$. Then the location of the kink point for the approximate problem is equivalent to finding the roots of the polynomial $P(y_*(K))$, given by

$$P(y_*(K)) = f_N(y_*(K)) - \frac{K}{X_0}, \tag{54}$$

where $f_N(y)$ is computed using recursion (53).

To apply the **Newton iteration method**, we need the derivative $P' = \frac{dP}{dy_*} = f'_N$, which is deduced from recursion (53), and given by the the following relation

$$f'_i(y) = f'_{i-1}(y)\left[1 - a\Delta t\right] + \sigma\sqrt{f_{i-1}(y)^+}\left[\frac{\Delta t}{\sqrt{T}} + \left[y\frac{\Delta t}{\sqrt{T}} + \Delta B_{i-1}\right]\frac{f'_{i-1}(y)}{2}\right],\ 1 \le i \le N$$

$$f'_0 = 0. \tag{55}$$

5.7.2 Results

The code of this section is found in the script discretized_CIR.py, which compares the different ways of determining the kink location for 1D CIR model.

References Cited

[1] Aurélien Alfonsi. On the discretization schemes for the cir (and bessel squared) processes. *Monte Carlo Methods and Applications mcma*, 11(4):355–384, 2005.

[2] Aurélien Alfonsi. A second-order discretization scheme for the cir process: application to the heston model. *Preprint CERMICS hal-00143723*, 14, 2008.

[3] 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.

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

[5] Abdel Berkaoui, Mireille Bossy, and Awa Diop. Euler scheme for sdes with non-lipschitz diffusion coefficient: strong convergence. *ESAIM: Probability and Statistics*, 12:1–11, 2008.

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

[7] G DEELSTRA and F DELBAEN. Convergence of discretized stochastic (interest rate) processes with stochastic drift term. *Applied stochastic models and data analysis*, 14(1):77–84, 1998.

[8] Thomas Gerstner. Sparse grid quadrature methods for computational finance.

[9] Thomas Gerstner and Markus Holtz. Valuation of performance-dependent options. *Applied Mathematical Finance*, 15(1):1–20, 2008.

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

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

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

[13] 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.

[14] 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.

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

[16] Xuerong Mao. *Stochastic differential equations and applications*. Elsevier, 2007.

[17] 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.