Application of Quasi Monte Carlo and Global Sensitivity Analysis to Option Pricing and Greeks

Stefano Scoleri[*], Marco Bianchetti[†], and Sergei Kucherenko[‡]

February 5, 2017

Abstract

Quasi Monte Carlo (QMC) and Global Sensitivity Analysis (GSA) techniques are applied for pricing and hedging representative financial instruments of increasing complexity. We compare standard Monte Carlo (MC) vs QMC results using Sobol' low discrepancy sequences, different sampling strategies, and various analyses of performance.

We find that QMC outperforms MC in most cases, including the highest-dimensional simulations, showing faster and more stable convergence. Regarding greeks computation, we compare standard approaches, based on finite differences (FD) approximations, with adjoint methods (AAD) providing evidences that, when the number of greeks is small, the FD approach combined with QMC can lead to the same accuracy as AAD, thanks to increased convergence rate and stability, thus saving a lot of implementation effort while keeping low computational cost. Using GSA, we are able to fully explain our findings in terms of reduced effective dimension of QMC simulation, allowed in most cases, but not always, by Brownian bridge discretization or PCA construction.

We conclude that, beyond pricing, QMC is a very efficient technique also for computing risk measures, greeks in particular, as it allows to reduce the computational effort of high-dimensional Monte Carlo simulations typical of modern risk management.

[*]Iason Ltd, Italy, `stefano.scoleri@gmail.com`

[†]Market Risk Management, Banca Intesa Sanpaolo, Piazza G. Ferrari 10, 20121, Milan, Italy, `marco@bianchetti.org`, corresponding author.

[‡]Imperial College, London, UK, `s.kucherenko@imperial.ac.uk`

Contents

JEL classifications: C63, G12, G13.

Keywords: derivative, option, European, Asian, barrier, knock-out, cliquet, greeks, Monte Carlo, Quasi Monte Carlo, random, pseudo random, quasi random, low discrepancy, Sobol', convergence, speed-up, Brownian bridge, global sensitivity analysis, principal component analysis, adjoint algorithmic differentiation.

Acknowledgements: M.B. acknowledges fruitful discussions with many colleagues at international conferences, in Risk Management of Intesa Sanpaolo, and in Financial Engineering and Trading Desks of Banca IMI.

1 Introduction

Monte Carlo simulation is a common way to tackle problems in finance. However, this method is rather time consuming and the convergence rate is slow, since the root mean square error decays as $N^{-1/2}$, where N is the number of sampled points. Various "variance reduction" techniques exist, which can improve the efficiency of the simulation, but they don't modify the convergence rate [Jac01, Gla03].

Quasi Monte Carlo (QMC) represents a very efficient alternative to standard MC, capable to achieve, in many cases, a faster convergence rate and, hence, higher accuracy [Jac01, Gla03, MF99, SK05b, SK05a, Wan09, KFSM11, SAKK12]. QMC methods are based on using instead of pseudo-random numbers (PRN), low discrepancy sequences (LDS), also known as quasi-random numbers for sampling points. LDS are designed in such a way that the integration domain is covered as uniformly as possible, while PRN are known to statistically form clusters and holes of points. On the contrary, LDS "know" about the positions of previously sampled points and fill the gaps between them. Among several known LDS, Sobol' sequences have been proven to show better performance than others and for this reason they are widely used in Finance [Jac01, Gla03]. However, it is also known that construction of efficient high-dimensional Sobol' sequences heavily depends on the so-called initial numbers and therefore very few Sobol' sequence generators show good efficiency in practical tests (see e.g. [SAKK12]).

Compared to MC, QMC techniques also have some disadvantages. Firstly, there is no "in sample" estimation of errors: since LDS are deterministic, there is not a notion of probabilistic error. There are some techniques, known under the name of randomized QMC, which introduce appropriate randomizations in the construction of LDS, opening up the possibility of measuring errors through a confidence interval while preserving the convergence rate of Quasi Monte Carlo (see e.g. [Gla03]). Secondly, effectiveness of QMC depends on the integrand function, and, most importantly, the convergence rate can depend on the dimensionality of the problem. However, it's been shown that in many financial applications QMC outperforms standard Monte Carlo even in the presence of very high dimensions [SAKK12, PT95, PP99, CMO97, KMRZ98a, KMRZ98b, KS07, BKS15]. This fact is usually explained by a reduced effective dimension of the problem, with respect to its nominal dimension. The concept of effective dimensions was introduced in [CMO97]. It was suggested that QMC is superior to MC if the effective dimension of an integrand is not too large. The notion is based on the ANalysis Of VAriances (ANOVA). In [LO00] it was shown how the ANOVA components are linked to the effectiveness of QMC integration methods. It is important to measure the effective dimension in order to predict the efficiency of a Quasi Monte Carlo algorithm. Moreover, various techniques can be used to reduce effective dimension and, thus, improve efficiency: this is possible because the effective dimension[1] can vary by changing the order in which the variables are sampled. The optimal way to achieve this can be a hard task, it could depend on the specific model and a general solution is not known at present. One popular choice in the financial literature on path-dependent option pricing [CMO97, KS07] is to apply the Brownian bridge discretization to the simulation of the underlying stochastic process, which is based on the use of conditional distributions. Unlike the standard discretization, which generates values of the Brownian motion sequentially along the time horizon, the Brownian bridge discretization first generates the Brownian motion value at the terminal point, then it fills a midpoint using the value already found at the terminal point and then subsequent values at the successive midpoints using points already simulated at previous steps. In terms of QMC sampling, this simulation scheme means that the first coordinate of the QMC vector is used to simulate the terminal value of the Brownian motion, while subsequent coordinates are used to generate intermediate points. There are many studies which show

[1]Actually, the effective dimension in the truncation sense can be reduced in this way. See Section 3 for the formal definition of effective dimensions.

that superior performance of the QMC approach with the Brownian bridge discretization in comparison with the standard discretization using MC or QMC sampling, in application e.g. to Asian options [CMO97, KS07]. However, it was pointed in [Pap01] that, in some cases, the Brownian bridge can perform worse than the standard discretization in QMC simulation. The big question is how to know with certainty which numerical scheme will provide superior efficiency in QMC simulation. Global Sensitivity Analysis (GSA) can provide this answer.

GSA is a very powerful tool in the analysis of complex models as it offers a comprehensive approach to model analysis. Traditional sensitivity analysis, within the present context, is "local" since, given a function $f(x_0)$ and a point x_0 in the function domain, computes the derivative $\frac{\partial f}{\partial x}$ at $x = x_0$. GSA instead does not require to specify a particular point x_0 in the domain, since it explores the whole domain (hence the name *global*). It also quantifies the effect of varying a given input (or set of inputs) while all other inputs are varied as well, providing a measure of interactions among variables. GSA is used to identify key parameters whose uncertainty most affects the output. This information can be used to rank variables, fix unessential variables and decrease problem dimensionality. Reviews of GSA can be found in [SK05a] and [SAA$^+$10]. The variance-based method of global sensitivity indices developed by [Sob01] became very popular among practitioners due to its efficiency and easiness of interpretation. There are two types of Sobol' sensitivity indices: the main effect indices, which estimate the individual contribution of each input parameter to the output variance, and the total sensitivity indices, which measure the total contribution of a single input factor or a group of inputs.

For modelling and complexity reduction purposes, it is important to distinguish between the model *nominal* dimension and its *effective* dimension. The notions of effective dimension in the truncation and superposition sense were introduced by [CMO97]. Further, [LO06] added the notion of "average dimension" which is more practical from the computational point of view. Definitions and evaluations of effective dimensions are based on the knowledge of Sobol' sensitivity indices. Quite often complex mathematical models have effective dimensions much lower than their nominal dimensions. In the context of quantitative finance, GSA can be used to estimate effective dimensions of a given problem. In particular, it can assess the efficiency of a particular numerical scheme (such as the Brownian bridge, Principal Component Analysis or standard discretizations along the time direction or Cholesky factorization or Principal Component Analysis in the risk factor space).

"Greeks" represent the sensitivity of the price of a financial instrument with respect to specific risk factors and are formally defined as partial derivatives of the price function. They are very important quantities which need to be computed besides price both on the Front Office side (for hedging purposes) and on Risk Management (to monitor the risk of a portfolio w.r.t. individual risk factors). If a simulation approach is used to price the instrument, standard techniques (Finite Differences, FD) require bumping each risk factor and re-pricing the instrument on each MC path. The computational cost of computing all the greeks, therefore, increases linearly with the number of underlying risk factors and becomes particularly expensive e.g. for options on multiple underlyings. One popular and faster alternative to finite differences is Adjoint Algorithmic Differentiation (AAD) [GW08, Nau12, GG06, LLS09, CG10, CG12, Cap11, CLP11]. It is based on the "Pathwise Derivative" method: unbiased estimators of the Greeks are obtained by differentiating the discounted payoff along each MC path, see [BG96, Gla03]. If we want to compute the gradient of a single output w.r.t. many variables (as in the case of Greeks of multi-asset options), the adjoint mode of algorithmic differentiation can be employed to dramatically increase the efficiency of pathwise differentiation. In particular, it can be proven that the computational cost of evaluating a function and its gradient with AAD is less then approximately four times the cost of evaluating the function alone, independently of the number of derivatives to compute. So, the relative computational cost of computing all the greeks with this approach is constant (and this constant is a small number, say ≤ 4) making AAD favourable in presence of many risk factors, such as in the case of multi-asset options. In the present work we apply adjoints to simple test cases in multi-asset option pricing with both MC and QMC, and we measure its efficiency w.r.t. finite differences, taking into account the accuracy of the computation.

The paper is organized as follows: Section 2 contains a brief review on QMC methodology and on Low Discrepancy Sequences, with particular emphasis on financial applications. Section 3 introduces GSA and the notions of effective dimensions, establishing a link with QMC efficiency. In Section 4 we present the results of prices and sensitivities (greeks) computation for selected payoffs: both GSA and convergence analysis are performed, with the purpose to compare MC and QMC efficiencies via a thorough error analysis. Finally, conclusions and directions of future work are given in Section 5. In particular we propose to apply our methodology to risk management issues, where a faster and smoother convergence would represent a great advantage in terms of both computational effort and budget. A brief review of the AAD methodology is contained in the Appendix.

2 Monte Carlo and quasi Monte Carlo methods in finance

2.1 General motivation

Let's consider a generic financial instrument written on N_{rf} assets $\boldsymbol{S} = (S_1, \ldots, S_{N_{rf}})$ with a single payment date T. We denote the instrument's payoff at time T as $\mathcal{P}(\boldsymbol{S}(t), \boldsymbol{\theta})$, where $\boldsymbol{S}(t)$ is the underlying assets' value at time $t \in [0, T]$, and $\boldsymbol{\theta}$ is a set of relevant parameters, including *instrument* parameters, such as strikes, barriers, fixing dates of the underlyings, callable dates, payment dates, etc., described in the contract, and *pricing* parameters, such as interest rates, volatilities, correlations, etc., associated with the pricing model.

Using standard no-arbitrage pricing theory, see e.g. [Duf01], the price of the instrument at time $t = 0$ is given by

$$V_0(\boldsymbol{\theta}) = \mathbb{E}^Q[D(0,T)\mathcal{P}(\boldsymbol{S}(t), \boldsymbol{\theta})|\mathcal{F}_0], \tag{2.1}$$

where $(\Omega, \mathcal{F}, Q)$ is a probability space with risk-neutral probability measure Q and filtration $\mathcal{F}_t$ at time t, $\mathbb{E}^Q[\,\cdot\,]$ is the expectation with respect to Q, $D(0,T) = \exp\left(-\int_0^T r(t)dt\right)$ is the stochastic discount factor, and $r(t)$ is the risk-neutral short spot interest rate. Notice that the values of $\boldsymbol{S}$ at intermediate times t before final payment date T may enter into the definition of the payoff $\mathcal{P}$. "Greeks" are derivatives of the price $V_0(\boldsymbol{\theta})$ w.r.t. specific parameters $\boldsymbol{\theta}$. In the present work, we will consider in particular the following component greeks:

$$\Delta_i = \frac{\partial V_0}{\partial S_i(0)}, \quad \mathcal{V}_i = \frac{\partial V_0}{\partial \sigma_i}, \quad \Gamma_{ij} = \frac{\partial^2 V_0}{\partial S_i(0) \partial S_j(0)}, \tag{2.2}$$

called delta, vega and gamma respectively. Notice that *model* parameters $\boldsymbol{\sigma}$ denote the volatilities of assets $\boldsymbol{S}$ and are assumed to be constant in a Black-Scholes framework.

The underlying assets dynamics is described by a set of stochastic differential equations (SDE). A generic Wiener diffusion model in N_{rf} dimensions is generally considered and it is characterized by the following dynamics:

$$d\boldsymbol{S}(t) = \boldsymbol{\mu}(t, \boldsymbol{S})dt + \Sigma(t, \boldsymbol{S})\, d\boldsymbol{X}^P(t), \tag{2.3}$$

with initial conditions $\boldsymbol{S}_0$, where P is the real-world probability measure, $\boldsymbol{\mu}$ is the real-world drift, Σ is the $N_{rf} \times N_{rf}$ volatility matrix, and $\boldsymbol{X}^P(t)$ is a N_{rf}-dimensional Brownian motion under P with correlation matrix R[2]. In particular, in the Black-Scholes model the underlying assets $\boldsymbol{S}(t)$ follow a N_{rf}-dimensional geometric Brownian motion, i.e. in (2.3) $\mu_i(t, \boldsymbol{S}) = \mu_i S_i(t)$ and $[\Sigma(t, \boldsymbol{S})\, d\boldsymbol{X}^P(t)]_i = \sigma_i S_i(t)\, d\boldsymbol{X}_i^P(t)$ for $i = 1, \ldots, N_{rf}$, with constant drift and volatility parameters, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{N_{rf}})$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{Nrf})$ respectively. The covariance matrix is defined as

$$\Sigma = \tilde{D} R \tilde{D} \tag{2.4}$$

where $\tilde{D} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{N_{rf}})$. Geometric Brownian motion dynamics can be reformulated in terms of independent Brownian motions $\boldsymbol{W}^P(t)$:

$$dS_i(t) = \mu_i\, S_i(t)\, dt + S_i(t) \sum_{k=1}^{N_{rf}} A_{ik}\, dW_k^P(t), \tag{2.5}$$

where A is a square root of Σ, i.e. any matrix such that $AA^T = \Sigma$. The solution to equation (2.5) in a risk-neutral world (under the risk-neutral probability measure Q) is given by[3]

$$S_i(t) = S_i(0)\, \exp\left[\left(r - \frac{1}{2}\sigma_i^2\right) t + \sum_{k=1}^{N_{rf}} A_{ik}\, W_k^Q(t)\right]. \tag{2.6}$$

We notice that $\boldsymbol{Y}(t) = \tilde{D}\boldsymbol{X}(t) = A\boldsymbol{W}(t)$, appearing in (2.5) and (2.6), is a N_{rf}-dimensional Brownian motion with covariance Σ. In this work, we use two different methods in order to find such matrix A, for any fixed $t \in [0, T]$. The first one is the Cholesky method, which yields a triangular matrix thus reducing the number of elementary operations subsequently needed to compute the Brownian motion. The second one is the Principal Component Analysis (PCA) construction, which requires a diagonalization of Σ. Let λ_i and $\boldsymbol{v}_i$ be the N_{rf} eigenvalues and an orthonormal set of corresponding eigenvectors of the covariance matrix, respectively[4]. Then, the covariance matrix can be written as

$$\Sigma = V\,\Lambda\,V^T\,, \tag{2.7}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{N_{rf}})$ and $V = (\boldsymbol{v}_1|\cdots|\boldsymbol{v}_{N_{rf}})$. It follows that

$$A = V\Lambda^{1/2} \tag{2.8}$$

is a square root of Σ. Even though the PCA factorization isn't faster than the Cholesky method, it is optimal in the sense that, if the eigenvalues are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_{rf}}$, most of the variance of the Brownian motion $\boldsymbol{Y}$ is explained by the first few principal components: formally, if $Z_1, \ldots, Z_K$ ($K \leq N_{rf}$) are independent standard normals, then the error $\mathbb{E}\left[\|\boldsymbol{Y} - \sum_{i=1}^K \boldsymbol{a}_i Z_i\|^2\right]$ is minimized taking $\boldsymbol{a}_i$ as the columns of A as given in (2.8) and $Z_i = \boldsymbol{v}_i^T\boldsymbol{Y}/\sqrt{\lambda_i}$, called the ith principal component of $\boldsymbol{Y}$. This optimality turns out to be relevant in QMC applications.

The solution to the pricing equation (2.1) requires the knowledge of the values of the underlying assets $\boldsymbol{S}$ at the relevant contract dates $\{T_1, \ldots, T_n\}$. Such values may be obtained by solving the SDE (2.3). If the SDE cannot be solved explicitly, we must resort to a discretization

[2]A correlated N_{rf}-dimensional Brownian motion with correlation matrix R satisfies $\mathbb{E}[dX_i(t)dX_k(t)] = \rho_{ik}dt$, where $i, k = 1, \ldots, N_{rf}$ and ρ_{ik} are the entries of R, which is a symmetric, positive (semi)definite matrix with diagonal terms equal to 1.

[3]We assume a constant interest rate r for simplicity. See e.g. [BM06], appendix B, for a generalization to stochastic interest rates.

[4]Since Σ is symmetric and positive semidefinite, it has N_{rf} real non-negative eigenvalues and, by the spectral theorem, an orthonormal set of eigenvectors.

scheme, computing the values of $\boldsymbol{S}$ on a time grid $\{t_1, \dots, t_{N_{ts}}\}$, where $t_1 < t_2 < \dots < t_{N_{ts}}$, and N_{ts} is the number of time steps. Notice that the contract dates must be included in the time grid, $\{T_1, \dots, T_n\} \subset \{t_1, \dots, t_{N_{ts}}\}$. For example, the Euler discretization scheme consists of approximating the SDE (2.3) by

$$S_i(t_j) = S_i(t_{j-1}) + \mu_i\left(t_{j-1}, S_i(t_{j-1})\right)\Delta t_j + \left[\Sigma\left(t_{j-1}, \boldsymbol{S}(t_{j-1})\right)\Delta \boldsymbol{X}(t_j)\right]_i, \quad j = 1, \dots, N_{ts}, \tag{2.9}$$

where $\Delta t_j = t_j - t_{j-1}$, $\Delta X_i(t_j) = X_i(t_j) - X_i(t_{j-1})$ and $t_0 = 0$. In particular, the discretization of Black-Scholes solution (2.6) leads to

$$S_i(t_j) = S_i(t_{j-1}) \exp\left[\left(r - \frac{\sigma_i^2}{2}\right)\Delta t_j + \sum_{k=1}^{N_{rf}} A_{ik}\, \Delta W_k(t_j)\right], \quad i = 1, \dots, N_{rf}\,,\ j = 1, \dots, N_{ts}. \tag{2.10}$$

where $\Delta W_i(t_j) = W_i(t_j) - W_i(t_{j-1})$.

Clearly, the price in eq. (2.1) will depend on the discretization scheme adopted. We consider three discretization schemes in eq. (2.10): standard discretization (SD), Brownian bridge discretization (BBD) and Principal Component Analysis (PCA). In the SD scheme the Brownian motion is discretized as follows:

$$\Delta W_i(t_j) = \sqrt{\Delta t_j} Z_{ij}, \quad i = 1, \dots, N_{rf}\,,\ j = 1, \dots, N_{ts}, \tag{2.11}$$

where Z_{ij} are $N_{ts}N_{rf}$ independent standard normal variates[5].

In the BBD scheme the first variate is used to generate the terminal value of the Brownian motion, while subsequent variates are used to generate intermediate points, conditioned to points already simulated at earlier and later time steps, according to the following formula,

$$\begin{aligned} &W_i(t_0) = 0, \\ &W_i(t_{N_{ts}}) = \sqrt{\Delta t_{N_{ts}0}} Z_{i1}, \\ &W_i(t_j) = \frac{\Delta t_{kj}}{\Delta t_{kh}} W_i(t_h) + \frac{\Delta t_{jh}}{\Delta t_{kh}} W_i(t_k) + \sqrt{\frac{\Delta t_{kj}\Delta t_{jh}}{\Delta t_{kh}}} Z_{il}, \quad t_h < t_j < t_k\,, \quad l = 2, \dots, N_{ts}, \end{aligned} \tag{2.12}$$

where $\Delta t_{ab} = t_a - t_b$. Unlike the SD scheme, which generates the Brownian motion sequentially across time steps, the BBD scheme uses different orderings: as a result, the variance in the stochastic part of (2.12) is smaller than that in (2.11) for the same time steps, so that the first few points contain most of the variance. It follows that, with the BBD, much of the shape of the Brownian motions are determined by the first few coordinates of $\boldsymbol{Z}$. However, in this way, the points of the Brownian motion where to concentrate the variance are somewhat a priori determined.

The PCA discretization scheme optimally samples from the gaussian vector so that most of the variance of the Brownian path is explained by the first few coordinates of $\boldsymbol{Z}$. It is based on the PCA factorization of the covariance matrix C of the (discretized) Brownian motion vector $(W_i(t_1), \dots, W_i(t_{N_{ts}}))$. In the case of a multi-dimensional Brownian motion, when the covariance matrix Σ of the underlying assets is also factorized by PCA, the optimality of principal components would be reduced if an independent PCA time discretization were applied to each component. Therefore, one has to work with the full covariance matrix $C \otimes \Sigma$ of the (discretized) $N_{ts}N_{rf}$-dimensional Brownian motion

$$\left(Y_1(t_1), \dots, Y_{N_{rf}}(t_1), \dots, Y_1(t_{N_{ts}}), \dots, Y_{N_{rf}}(t_{N_{ts}})\right) \tag{2.13}$$

[5]Gaussian numbers Z_{ij} are usually sampled from a $N_{ts}N_{rf}$-dimensional gaussian vector using PRN or QRN generators. As will be discussed in the following sections, the sampling order chosen to fill $\boldsymbol{Z}$ has a relevant impact when QRN such as Sobol' sequences are used.

and then apply a single PCA directly to it. This also reduces the computational effort of the diagonalization, [Gla03]. As we will discuss in the following sections, QMC sampling shows different efficiencies for SD, BBD and PCA.

Throughout this work, we will focus on the *relative* effects of the dimension D and of the discretization schemes on the MC and QMC simulations. Thus, we will assume a simple Black-Scholes underlying dynamics for simplicity.

2.2 Pseudo Random Numbers and Low Discrepancy Sequences

The nominal dimension of the computational problem of finding option prices and greeks is $D = N_{ts}N_{rf}$, i.e. the product of the number of time steps required in the discretization of the SDE (2.9) and the number of risk factors (the underlying assets): indeed, the expectation value in (2.1) is formally an integral of the payoff, regarded as a function of D standard normal[6] variables $Z_1, \ldots, Z_D$. Hence the pricing problem (2.1) can be reduced to the evaluation of integrals of the following generic form

$$V = \int_{H^D} f(\boldsymbol{x}) d^D \boldsymbol{x}, \tag{2.14}$$

where $H^D = [0,1]^D$ is the $D-$dimensional unit hypercube. This motivates the use of Monte Carlo techniques. The standard Monte Carlo estimator of (2.14) has the form

$$V_N \simeq \frac{1}{N} \sum_{k=1}^{N} f(\boldsymbol{x}_k), \tag{2.15}$$

where $\{\boldsymbol{x}_k\}_{k=1}^N$ is a sequence of N random points in H^D. Consider an integration error

$$\varepsilon = |V - V_N|. \tag{2.16}$$

By the Central Limit Theorem the root mean square error of the Monte Carlo method is

$$\varepsilon_{MC} = \left[\mathbb{E}(\varepsilon^2)\right]^{1/2} = \frac{\sigma_f}{\sqrt{N}}, \tag{2.17}$$

where σ_f is the standard deviation of $f(x)$. Although ε_{MC} does not depend on the dimension D, as in the case of lattice integration on a regular grid, it decreases slowly with increasing N. Variance reduction techniques, such as antithetic variables [Jac01, Gla03], only affect the numerator in (2.17).

In order to increase the rate of convergence one has to resort to Low Discrepancy Sequences (LDS), also called Quasi Random Numbers (QRNs), instead of PRNs. The discrepancy of a sequence $\{\boldsymbol{x}_k\}_{k=1}^N$ is a measure of how inhomogeneously the sequence is distributed inside the unit hypercube H^D and is denoted by $\mathcal{D}_N^D(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$. A Low Discrepancy Sequence is a sequence $\{\boldsymbol{x}_k\}_{k=1}^N$ in H^D such that, for any $N > 1$, the first N points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ satisfy inequality

$$\mathcal{D}_N^D(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \leq c(D) \frac{\ln^D N}{N}, \tag{2.18}$$

for some constant $c(D)$ depending only on D, see [Nie88] for details.

[6]Independent standard gaussian numbers Z_j are computed using transformation of uniform variates $x_j \sim$ *i.i.d.* $U(0,1)$, $Z_j = \Phi^{-1}(x_j)$, $j = 1, \ldots, D$ where Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution.

A QMC estimator of the integral (2.14) is of the form (2.15) with the only difference that the sequence $\{\boldsymbol{x}_k\}_{k=1}^N$ is sampled using LDS instead of PRNs. An upper bound for the QMC integration error is given by the Koksma-Hlawka inequality

$$\varepsilon_{QMC} \leq V(f)\mathcal{D}_N^D = \mathcal{O}\left(\frac{\ln^D N}{N}\right), \tag{2.19}$$

where $V(f)$ is the variation of the integrand function in the sense of Hardy and Krause ([Nie88]), which is finite for functions of bounded variation, see [KFSM11]. The convergence rate of (2.19) is asymptotically faster than (2.17), but it depends on the dimensionality D. However, eq. (2.19) is just an upper bound: what is observed in most numerical tests [KFSM11, CMO97] is a power law

$$\varepsilon_{QMC} \sim \frac{c}{N^\alpha}, \tag{2.20}$$

where the value of α depends on the model function. When $\alpha > 0.5$ the QMC method outperforms standard MC: this situation turns out to be quite common in financial problems. We will measure α for some representative financial instruments, showing that its value can be very close to 1 when the *effective* dimension of f is low, irrespective of the nominal dimension D. The concept of effective dimension, and the methodology to compute it, will be introduced in the following sections.

Since LDS are deterministic, there are no statistical measures like variances associated with them. Hence, the constant c in (2.20) is not a variance and (2.20) does not have a probabilistic interpretation as for standard MC. To overcome this limitation, [Owe93] suggested to introduce randomization into LDS at the same time preserving their superiority to PRN uniformity properties. Such LDS became known as *scrambled* (see also [Gla03]). In practice, the root mean square error (RMSE) for both MC and QMC methods for any fixed N can be estimated by computing the following error averaged over L independent runs:

$$\varepsilon_N = \sqrt{\frac{1}{L}\sum_{\ell=1}^{L}\left(V - V_N^{(\ell)}\right)^2}, \tag{2.21}$$

where V is the exact, or estimated at a very large extreme value of $N \to \infty$, value of the integral and $V_N^{(\ell)}$ is the simulated value for the ℓth run, performed using N PRNs, LDS, or scrambled LDS. For MC and QMC based on scrambled LDS, runs based on different seed points are statistically independent. In the case of QMC, different runs are obtained using non overlapping sections of the LDS. Actually, scrambling LDSs weakens the smoothness and stability properties of the Monte Carlo convergence, as we will see in Section 4.5. Hence, in this paper we will use the approach based on non-overlapping LDSs, as in [SK05b].

The most known LDS are Halton, Faure, Niederreiter and Sobol' sequences. Sobol' sequences, also called $LP\tau$ sequences or (t,s) sequences in base 2, see [Nie88], became the most known and widely used LDS in finance due to their efficiency [Jac01, Gla03]. As explained e.g. in [Sob67], Sobol' sequences were constructed under the following requirements:

1. Best uniformity of distribution as $N \to \infty$.
2. Good distribution for fairly small initial sets.
3. A very fast computational algorithm.

The efficiency of Sobol' LDS depend on the so-called initialisation numbers. In this work we used `SobolSeq8192` generator provided by [BRO]. `SobolSeq` is an implementation of the 8192-dimensional Sobol' sequences with modified initialisation numbers. Sobol' sequences produced by `SobolSeq8192` can be up to and including dimension 2^{13}, and satisfy additional uniformity properties: Property A for all dimensions and Property A' for adjacent dimensions (see [SAKK12] for details[7]). It has been found in [SAKK12] that `SobolSeq` generator outperforms all other known LDS generators both in speed and accuracy.

3 Global sensitivity analysis and effective dimensions

Effective dimension is the key to explain the superior efficiency of QMC w.r.t. MC. Hence, it is crucial to develop techniques to estimate the effective dimension and to find the most important variables in a MC simulation.

The variance-based method of global sensitivity indices developed by Sobol' became very popular among practitioners due to its efficiency and easiness of interpretation [SK05a, SAA$^+$10]. There are two types of Sobol' sensitivity indices: the main effect indices, which estimate the individual contribution of each input parameter to the output variance, and the total sensitivity indices, which measure the total contribution of a single input factor or a group of inputs. Sobol' indices can be used to rank variables in order of importance, to identify non-important variables, which can then be fixed at their nominal values to reduce model complexity, and to analyze the efficiency of various numerical schemes.

Consider a mathematical model described by an integrable function $f(x)$, where the input $x = (x_1, \ldots, x_D)$ is taken in a D-dimensional domain Ω and the output is a scalar. Without loss of generality, we choose Ω to be the unit hypercube H^D. The input variables $x_1, \ldots, x_D$ can, then, be regarded as independent uniform random variables each defined in the unit interval $[0, 1]$. The starting point of global sensitivity analysis (GSA) is the analysis of variance (ANOVA) decomposition of the model function,

$$f(x) = f_0 + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \ldots + f_{1\,2\cdots D}(x_1, \ldots, x_D)\,. \tag{3.1}$$

The expansion (3.1) is unique, provided that

$$\int_0^1 f_{i_1\cdots i_s}(x_{i_1}, \ldots, x_{i_s}) dx_{i_k} = 0\,, \quad \forall k = 1, \ldots, s\,. \tag{3.2}$$

The ANOVA decomposition expands the function f into a sum of terms, each depending on an increasing number of variables: a generic component $f_{i_1\cdots i_s}(x_{i_1}, \ldots, x_{i_s})$, depending on s variables, is called an s-order term. It follows from (3.2) that the ANOVA decomposition is orthogonal and that its terms can be explicitly found as follows,

$$\begin{aligned} f_0 &= \int_{H^D} f(x) d^D x, \\ f_i(x_i) &= \int_{H^{D-1}} f(x) \prod_{k\neq i} dx_k \, - f_0, \\ f_{ij}(x_i, x_j) &= \int_{H^{D-2}} f(x) \prod_{k\neq i,j} dx_k \, - f_0 - f_i(x_i) - f_j(x_j), \end{aligned} \tag{3.3}$$

and so on. If f is square-integrable, its variance decomposes into a sum of partial variances:

$$\sigma^2 = \sum_i \sigma_i^2 + \sum_{i<j} \sigma_{ij}^2 + \ldots + \sigma_{12\cdots D}^2, \tag{3.4}$$

[7]The most recent generator released by BRODA is `SobolSeq65536` which, not only has the highest dimensionality available and employs the super fast generation algorithm, but also the generated Sobol' sequences satisfy Property A in all dimensions and property A' for the adjacent dimensions.

where

$$\sigma^2_{i_1\cdots i_s} = \int_0^1 f^2_{i_1\cdots i_s}(x_{i_1},\ldots,x_{i_s})dx_{i_1}\cdots dx_{i_s}. \tag{3.5}$$

Sobol' sensitivity indices are defined as

$$S_{i_1\cdots i_s} = \frac{\sigma^2_{i_1\cdots i_s}}{\sigma^2} \tag{3.6}$$

and measure the fraction of total variance accounted by each $f_{i_1\cdots i_s}$ term of the ANOVA decomposition. From (3.4) it follows that all Sobol' indices are non negative and normalized to 1. First order Sobol' indices S_i measure the effect of single variables x_i on the output function; second order Sobol' indices S_{ij} measure the interactions between pairs of variables, i.e. the fraction of total variance due to variables x_i and x_j which cannot be explained by a sum of effects of single variables; higher order Sobol' indices $S_{i_1\cdots i_s}$, with $s > 2$, measure the interactions among multiple variables, i.e. the fraction of total variance due to variables $x_{i_1},\ldots,x_{i_s}$ which cannot be explained by a sum of effects of single variables or lower order interactions.

The calculation of Sobol' sensitivity indices in eq. (3.6) requires, in principle, 2^D valuations of the multi-dimensional integrals in eq. (3.5), which is unfeasible if D is large. However, for practical purposes it is not actually necessary to know all the possible Sobol' indexes, but just an appropriate selection of them. It is very useful to introduce Sobol' indices for subsets of variables and total Sobol' indices. Let $y = \{x_{i_1},\ldots,x_{i_m}\} \subseteq x, 1 \le i_1 \le \ldots, \le i_m \le D$, be a subset of x, and $z = y^\complement \subseteq x$ its complementary subset, and define

$$\begin{aligned} S_y &= \sum_{s=1}^{D} \sum_{(i_1<\cdots<i_s)\in K} S_{i_1\cdots i_s}, \\ S_y^{tot} &= 1 - S_z\,, \end{aligned} \tag{3.7}$$

where $K = \{i_1,\ldots,i_m\}$. Notice that $0 \le S_y \le S_y^{tot} \le 1$. The quantity $S_y^{tot} - S_y$ accounts for all the interactions between the variables in subsets y and z. It turns out that there exist efficient formulas which allow to avoid the knowledge of ANOVA components and to compute Sobol' indices directly from the values of function f [Sob01]. These formulas are based on the following integrals,

$$\begin{aligned} S_y &= \frac{1}{\sigma^2}\int_0^1 [f(y',z')-f_0][f(y',z)-f(y,z)]dy\,dz\,dy'dz'\,, \\ S_y^{tot} &= \frac{1}{2\sigma^2}\int_0^1 [f(y,z)-f(y',z)]^2 dy\,dz\,dy'\,, \\ \sigma^2 &= \int_0^1 f^2(y,z)dy\,dz\, - f_0^2\,, \\ f_0 &= \int_0^1 f(y,z)dy\,dz\,, \end{aligned} \tag{3.8}$$

where the integration variables are the components of the vectors y, z, y', z', such that $x = y \cup z$, and the first two integrals depend on the choice of y. Such integrals can be evaluated, in general, via MC/QMC techniques [KFSM11, Sal02].

Furthermore, usually enough information is already given by the first order indices S_i and by corresponding total effect indices S_i^{tot}, linked to a single variable $y = \{x_i\}$. For these Sobol' indices, it's easy to see that

- $S_i^{tot} = 0$: the output function does not depend on x_i,

- $S_i = 1$: the output function depends only on x_i,
- $S_i = S_i^{tot}$: there is no interaction between x_i and other variables.

Notice that just $D+2$ function evaluations for each MC trial are necessary to compute all S_i and S_i^{tot} indices in eqs. (3.8): one function evaluation at point $x = \{y, z\}$, one at point $x' = \{y', z'\}$, and D evaluations at points $x'' = \{y', z\}, \forall\, y' = \{x_i\},\ i = 1, \ldots, D$.

Approach presented above is applicable only to the case of independent input variables, which admits a unique ANOVA decomposition. In the case of dependent (correlated) input variables, the computation of variance-based global sensitivity indices is more involved. A generalization of GSA to dependent variables can be found in [KTA12].

We finally come to the notion of effective dimensions, firstly introduced in [CMO97]. Let $|y|$ be the cardinality of a subset of variables y. The effective dimension in the *superposition sense*, for a function f of D variables, is the smallest integer d_S such that

$$\sum_{0<|y|<d_S} S_y \geq 1 - \varepsilon \tag{3.9}$$

for some threshold ε (arbitrary and usually chosen to be less than 5%). If a function has an effective dimension d_S in the superposition sense, it can be approximated by a sum of d_S-dimensional terms, with an approximation error below ε.

The effective dimension in the *truncation sense* is the smallest integer d_T such that

$$\sum_{y \subseteq \{1,2,\ldots,d_T\}} S_y \geq 1 - \varepsilon. \tag{3.10}$$

The effective dimension d_S does not depend on the order of sampling of variables, while d_T does. In general, the following inequality holds,

$$d_S \leq d_T \leq D. \tag{3.11}$$

Effective dimensions can be estimated solely from indices S_i and S_i^{tot} using eqs. (3.8) with $y = i$, as described in [KFSM11], where relationships among such indices are used to classify functions in three categories according to their dependence on variables. For the so-called type A functions, variables are not all equally important and the effective dimension in the truncation sense d_T is small, such that $d_S \leq d_T \ll D$. They are characterized by the following relationship

$$\frac{S_y^{tot}}{|y|} \gg \frac{S_z^{tot}}{|z|}, \tag{3.12}$$

where $y \subseteq x$ is a leading subset of variables, $z = y^{\complement} \subseteq x$ its complementary subset. Functions with equally important variables have $d_T \simeq D$ and they can be further divided in two groups: type B and C functions. Type B functions have dominant low-order interactions and small effective dimension in the superposition sense d_S, such that $d_S \ll d_T \simeq D$. For such functions, Sobol' indices satisfy the following relationships:

$$S_i \simeq S_i^{tot}, \quad \forall\, i = 1, \ldots, D, \quad \sum_{i=1}^{D} S_i \simeq 1. \tag{3.13}$$

Type C functions have dominant higher-order interactions

$$S_i \ll S_i^{tot}\,, \sum_{i=1}^{D} S_i \ll 1 \tag{3.14}$$

Type	Description	Relationship between SI	Eff. dimensions
A	Few important variables	$S_y^{tot}/\|y\| \gg S_z^{tot}/\|z\|$	$d_S \le d_T \ll D$
B	Low-order interactions	$S_i \simeq S_j, S_i \simeq S_i^{tot}, \forall\, i,j$	$d_S \ll d_T \simeq D$
C	High-order interactions	$S_i \simeq S_j, S_i \ll S_i^{tot}, \forall\, i,j$	$d_S \simeq d_T \simeq D$

Table 1: Classification of functions w.r.t. their dependence on variables, based on GSA.

and effective dimensions $d_S \simeq d_T \simeq D$. This classification is summarized in Table 1.

[Owe03] introduced the notion of the average dimension d_A, which can assume fractional values, defined as

$$d_A := \sum_{0<|y|<D} |y|\, S_y\,, \tag{3.15}$$

and showed that it can be rather straightforwardly computed as

$$d_A = \sum_{i=1}^{D} S_i^{tot}\,. \tag{3.16}$$

It has been suggested in [SS14] that QMC should outperform MC when $d_A \lesssim 3$. This is confirmed in our findings, see Section 4.2.

It has been proved in many works [KFSM11, CMO97, Owe03] that QMC outperforms MC regardless of the nominal dimension whenever the effective dimension is low in one or more senses. Hence, in the case of type A and type B functions (we assume that functions are sufficiently smooth), QMC always outperforms MC, while for type C functions the two methods are expected to have similar efficiency. Actually, type A and B functions are very common in financial problems. We also note that the performance of the QMC method for Type A functions sometimes, but not always, can be greatly improved by using effective dimension reduction techniques, such as Brownian bridge, which will be demonstrated in the following section.

4 Numerical results

In this Section we apply MC and QMC techniques to high-dimensional pricing problems. Our aim is to test the efficiency of QMC with respect to standard MC in computing prices and greeks (delta, gamma, vega) for selected option payoffs $\mathcal{P}$ with increasing degree of complexity and path-dependency. We will consider both single asset and multi-asset test cases.

4.1 Selected payoffs and test set-up

We selected the following instruments as test cases: the first three payoffs depend on a single underlying S, while the remaining two payoffs depend on multiple underlyings $\boldsymbol{S}$.

1. **Asian call**:

$$\mathcal{P} = \max(\bar{S} - K, 0), \quad \bar{S} = \frac{1}{N_{ts}} \sum_{j=1}^{N_{ts}} S(t_j). \tag{4.1}$$

2. **Double knock-out call**:

$$\mathcal{P} = \max(S(T) - K, 0)\, \mathbb{1}_{\{\min_{j=1,\dots,N_{ts}} S(t_j) > B_l,\, \max_{j=1,\dots,N_{ts}} S(t_j) < B_u\}}. \tag{4.2}$$

3. **Cliquet**:
$$\mathcal{P} = \max\left\{\sum_{j=1}^{N_{ts}} \max\left[0, \min\left(C, \frac{S(t_j) - S(t_{j-1})}{S(t_{j-1})}\right)\right], F\right\}. \tag{4.3}$$

4. **European basket call**:
$$\mathcal{P} = \max(B(T) - K, 0), \quad B(t) = \sum_{i=1}^{N_{rf}} w_i \, S_i(t). \tag{4.4}$$

5. **Asian basket call**:
$$\mathcal{P} = \max(\bar{B} - K, 0), \quad \bar{B} = \frac{1}{N_{ts}} \sum_{j=1}^{N_{ts}} B(t_j). \tag{4.5}$$

In the above definitions, K denotes the strike price, T is the maturity, B_l and B_u are the values of the lower and upper barrier, respectively, C is a local cap, F is a global floor and w_i are the weights for a basket B composed of $N_{rf} > 1$ underlyings. Furthermore, payoffs (4.1, 4.2, 4.3, 4.5) are defined on a schedule $(t_1, \ldots, t_{N_{ts}})$ of fixing dates.

In all test cases we use the following payoff parameters:

- maturity: $T = 1$,
- strike: $K = 100$,
- lower barrier: $B_l = 0.5\, S_0$,
- upper barrier: $B_u = 1.5\, S_0$,
- global floor: $F = 0.16$,
- local cap: $C = 0.08$,
- number of basket components: $N_{rf} = 5$,
- basket weights: $w_i = 0.2 \quad \forall i = 1, \ldots, N_{rf}$.

Such selection guarantees an increasing level of complexity and path-dependency. The Asian call with arithmetic average is the simplest and most diffused non-European payoff. The double barrier is another very diffused payoff with stronger path-dependency. The cliquet option is a typical strongly path-dependent payoff based on the performance of the underlying stock [Wil06]. Finally, basket options allow to test, in addition, the contribution of correlation among different stocks. Clearly, many other possible payoffs could be added to the test (e.g. autocallable), but we think that such selection should be complete enough to cover most of the path-dependency and correlation characteristics relevant in the Monte Carlo simulation.

We assume that the stochastic processes $\boldsymbol{S}(t)$, which govern the evolution of the underlying assets, follow a N_{rf}-dimensional geometric Brownian motion as described in Section 2.1, with the following model parameters:

1. Payoffs (4.1, 4.2, 4.3):
   - spot price: $S_0 = 100$,
   - volatility: $\sigma = 0.3$.

2. Payoffs (4.4, 4.5)
    - spot prices: $\boldsymbol{S}_0 = (80, 90, 100, 110, 120)$,
    - volatilities: $\boldsymbol{\sigma} = (0.5, 0.4, 0.2, 0.3, 0.6)$,
    - correlations: $\rho_{ij} = \rho,\ i \neq j\ ,\ \rho = 0, 0.3, 0.6, 0.9$.

Therefore, we assume for simplicity that all assets have the same (constant) correlation ρ and we let ρ vary from 0 to 1. We notice that the choice of the values of $\boldsymbol{S}_0$ and $\boldsymbol{\sigma}$ ensures that the variables associated to different assets have different importance, which is a realistic case and is crucial in the application of QMC techniques, as will be discussed. The processes $\boldsymbol{S}(t)$ are discretized across N_{ts} time steps $\{t_1 < \cdots < t_j < \cdots < t_{N_{ts}}\}$, where $N_{ts} = 32$ for the single asset cases and $N_{ts} = 16$ for the multi-asset cases (also for the European basket in order to have the same dimensionality as for the Asian one). The dimension of the simulation is therefore $D = N_{ts} = 32$ for the single asset cases and $D = N_{ts} N_{rf} = 80$ for the multi-asset cases.

The numerical computations are performed in `Matlab` using different combinations of sampling techniques[8], both in time dimension and in risk-factors space (in parenthesis the factorization method of the covariance matrix of the underlying assets, relevant only for the multi-asset case, is indicated):

- MC + SD (+ CHOL),
- QMC + SD (+ CHOL),
- QMC + SD (+ PCA),
- QMC + BBD (+ CHOL),
- QMC + BBD (+ PCA),
- QMC + PCA (+ CHOL),
- QMC + PCA (+ PCA).

Regarding simulation parameters, we denote by N the number of simulated paths for the underlyings and by L the number of independent runs. Following the specifics of Sobol' sequences, we take $N = 2^p$, where p is an integer, since this guarantees the lowest discrepancy properties. For the MC case, we use Mersenne Twister generator, while in the case of QMC we use `SobolSeq8192` generator.

Simulation errors ε_N are analyzed by computing the root mean square error (RMSE) as defined by (2.21), where V is a reference value of prices or greeks simulated with a large number of MC scenarios ($N \approx 10^8 \div 10^9$ depending on the payoff, according to the necessary accuracy). To assess and compare performance of MC and QMC methods with different schemes, we compute the scaling of the RMSE as a function of N by fitting the function ε_N with a power law $c\,N^{-\alpha}$ (2.20). RMSE is obtained with $L = 100$ independent runs as discussed in Section 2.2.

Finally, greeks for the payoffs above are computed with the following alternative approaches:

- FD: central finite differences with path recycling and with shift parameter $\epsilon = 10^{-3}$ (see appendix A for details).
- AAD: hand coded adjoints as described in appendix A, only for multi-asset options and for first order sensitivities.

Notice that the analysis of the RMSE for FD greeks is, in general, more complex than that for prices, since the variance of the MC simulation mixes with the bias due to the approximation of derivatives with finite differences. This issue is discussed in Appendix A.

[8]Notice that, in contrast to [BKS15], here we use crude MC simulations instead of MC + antithetic variables.

Payoff	**Function**	$\mathbf{S_i/S_i^{tot}}$	$\sum_\mathbf{i}\mathbf{S_i}$	$\mathbf{d_T}$	$\mathbf{d_S}$	$\mathbf{d_A}$	**Effect of** ϵ
Asian	Price	0.54→0.43	0.714	< 32	< 32	1.38	-
	Delta	$0.32\rightarrow 10^{-2}$	0.71→0.74	32	< 32	3.5	small
	Gamma	$10^{-4} \rightarrow 10^{-2}$	$10^{-4} \rightarrow 10^{-2}$	32	32	$31 \rightarrow 25$	high
	Vega	0.42→0.01	0.611	< 32	< 32	1.57	no
Double KO	Price	0.01→0.15	0.22	32	< 32	8.5	-
	Delta	0.01→ 0.12	0.22	32	< 32	7.6	no
	Gamma	$10^{-5} \rightarrow 10^{-7}$	$10^{-4} \rightarrow 10^{-2}$	32	32	$31.2 \rightarrow 29.8$	high
	Vega	$10^{-5} \rightarrow 10^{-8}$	$10^{-4} \rightarrow 10^{-2}$	32	32	28	high
Cliquet	Price	1	1	32	1	1	-
	Vega	1	1	32	1	1	no

Table 2: Summary of GSA metrics and effective dimensions of prices and greeks for SD scheme. Arrow "→" in the column for S_i/S_i^{tot} denotes the change in the value with the increase of index i and/or with the increase of shift parameter $\epsilon = 10^{-4}, 10^{-3}, 10^{-2}$ (see appendix A); in the column for $\sum_i S_i$ it denotes the change in the value with the increase of shift parameter ϵ. The numerical computations were obtained evaluating the $D+2$ integrals (3.8) on 2^{17} QMC scenarios for each function (price or greeks). We show significant digits only.

4.2 Global sensitivity analysis for prices and greeks

Sobol' indices S_i and S_i^{tot} are computed for different sampling techniques using eqs. (3.8), where f is the relevant model function (the instrument payoff or a greek with finite differences[9]) and $y = \{x_i\}$, $y' = \{x'_i\}$, $z = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_D\}$, $z' = \{x'_1, \ldots, x'_{i-1}, x'_{i+1}, \ldots, x'_D\}$. Here x_i are the uniform variates used to construct gaussian numbers (see Section 2.2). The integrals in eqs. (3.8) are computed using QMC simulation.

Effective dimensions are estimated in the following way:

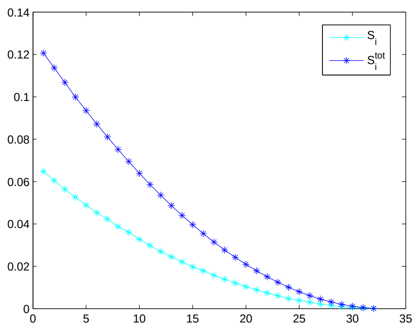
- The effective dimension in the truncation sense d_T is computed using inequality (3.12), looking for a minimal set of variables $y = \{x_1, \ldots, x_{d_T}\}$ such that the quantity $S_z^{tot}|y|/S_y^{tot}|z|$ is smaller than 1%. Since the calculation of d_T depends on the order of sampling variables, the result depends on the sampling techniques.
- The effective dimension in the superposition sense d_S is estimated using dimension d_T as an upper bound according to inequality (3.11). In order to distinguish between Type B and Type C functions, we look at ratios S_i/S_i^{tot} and $\sum_i S_i$ according to eqs. (3.13), (3.14).
- The effective average dimension d_A is computed using eq. (3.16).

Let's consider the single-asset payoffs (4.1), (4.2), (4.3) first. The results of GSA for the SD are shown in Figs. 1-3. Measures based on Sobol' indices are provided in Table 2. These measures are used to compute effective dimensions and to classify integrands in (2.14) corresponding to prices and greeks according to Table 1.

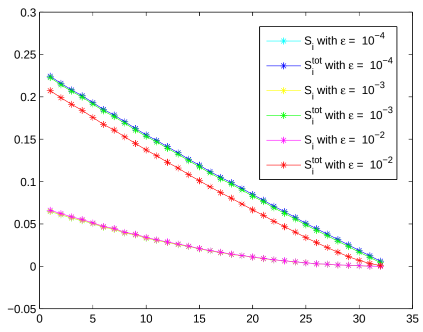
From these results we draw the following conclusions.

1. Asian option (Figure 1): price, and vega are type B functions, while delta and gamma are type C function.
2. Double KO option (Figure 2): price and all greeks are type C functions.

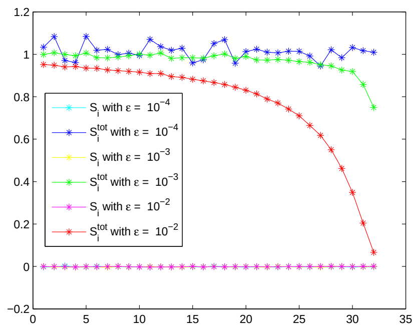
[9]Notice that only FD estimators of greeks are considered, since AAD is discussed, here, only with standard MC, for which GSA is not necessary.



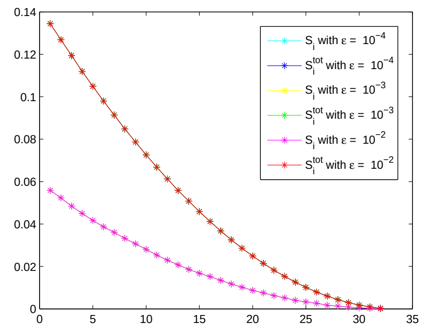
(a) Price



(b) Delta



(c) Gamma

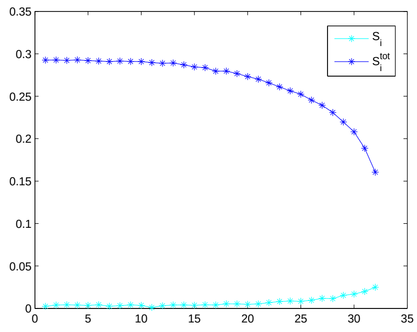


(d) Vega

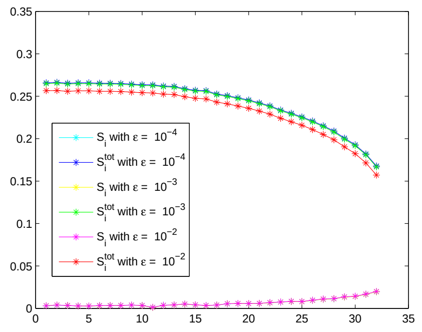
Figure 1: Asian call option price (a) and greeks $(b), (c), (d)$, SD, $D = 32$. First order Sobol' indices S_i and total sensitivity indices S_i^{tot} versus time step i.

3. Cliquet option (Figure 3): price and vega are type B functions with $d_S = 1$ (delta and gamma for a Cliquet option are null). We recall that $d_S = 1$ means that there are no interactions among variables.

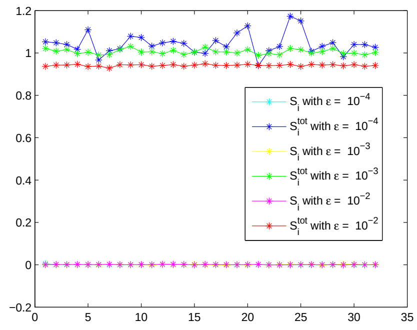
The analogous results of GSA for BBD are shown in Fig.s 4-6 and in Table 3.



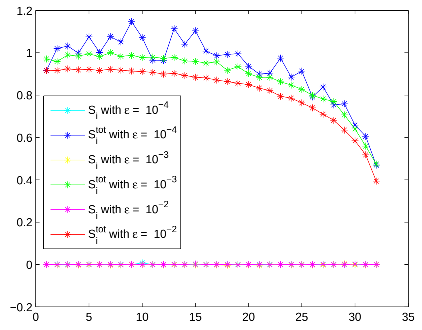
(a) Price



(b) Delta

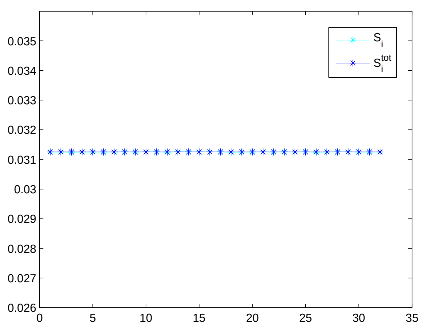


(c) Gamma

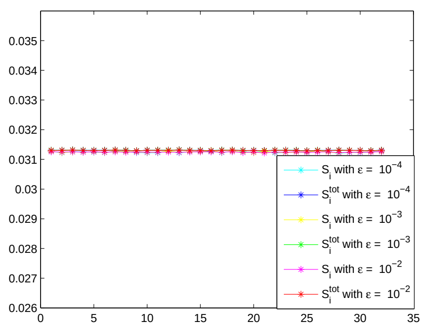


(d) Vega

Figure 2: Double Knock-out call option. Parameters as in Fig. 1.



(a) Price

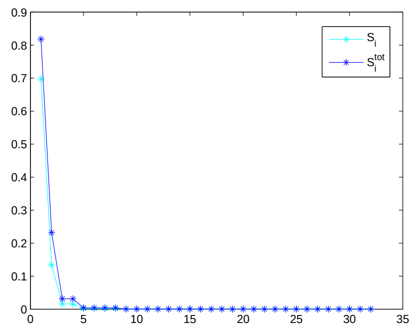


(b) Vega

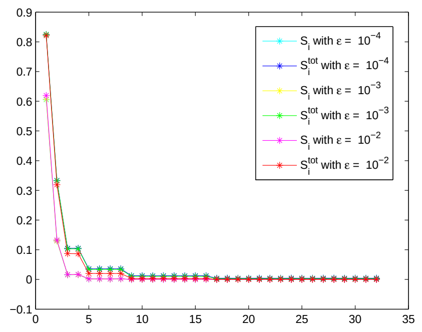
Figure 3: Cliquet option. Parameters as in Fig. 1. Delta and gamma are null for Cliquet options.

Payoff	**Function**	$\mathbf{S_i/S_i^{tot}}$	$\sum_\mathbf{i}\mathbf{S_i}$	$\mathbf{d_T}$	$\mathbf{d_S}$	$\mathbf{d_A}$	**Effect of** ϵ
Asian	Price	0.853→0.4	0.875	2	$\leq$2	1.13	-
	Delta	0.733→0.01	0.778	4	$\leq$4	1.68 → 1.43	small
	Gamma	$10^{-2} \to 10^{-4}$	$0.022 \to 10^{-4}$	32	32	31 → 8	high
	Vega	0.802→0.03	0.827	2	$\leq$2	1.20	no
Double KO	Price	0.70→0.01	0.70	$\simeq$ 2	$\leq$2	1.63	-
	Delta	0.83→0.01	0.83	2	$\leq$2	1.37	no
	Gamma	1	1 → 0.95	1	1	1.0	small
	Vega	$10^{-4} \to 0.2$	$10^{-6} \to 10^{-4}$	32	32	4.8 → 3.9	high
Cliquet	Price	0.978→0.2	0.892	$\simeq$ 2	$\leq$ 2	1.19	-
	Vega	0.595→0.001	0.32	$\simeq$ 32	$\leq$ 32	2.6	no

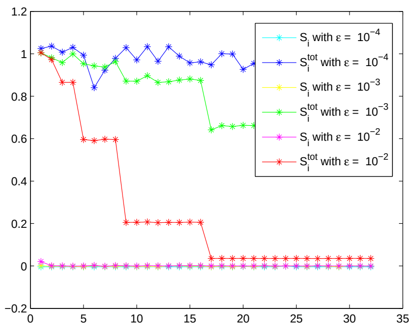
Table 3: Summary of GSA metrics and effective dimensions of prices and greeks for BBD scheme. Details as in Table 2.



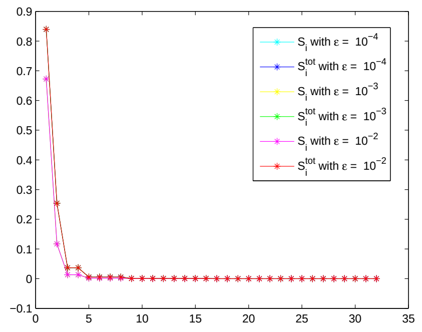
(a) Price



(b) Delta



(c) Gamma



(d) Vega

Figure 4: Asian call option price (a) and greeks $(b), (c), (d)$, BBD, $D = 32$. First order Sobol' indices S_i and total sensitivity indices S_i^{tot} versus time step i.

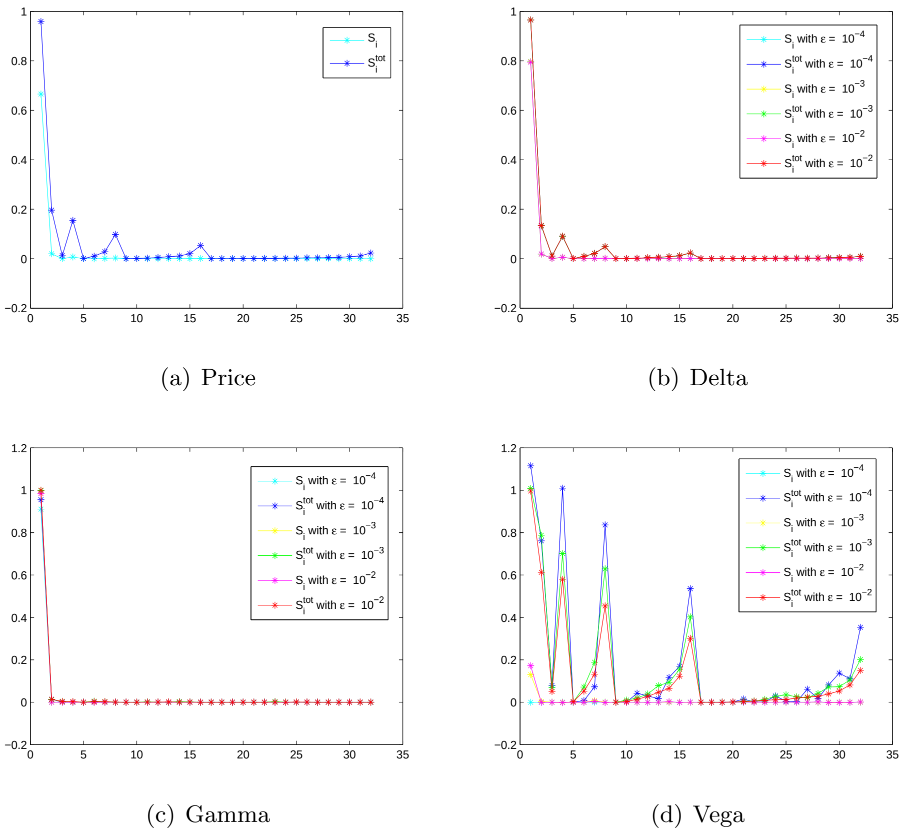


Figure 5: Double Knock-out call option. Details as in Fig. 4.

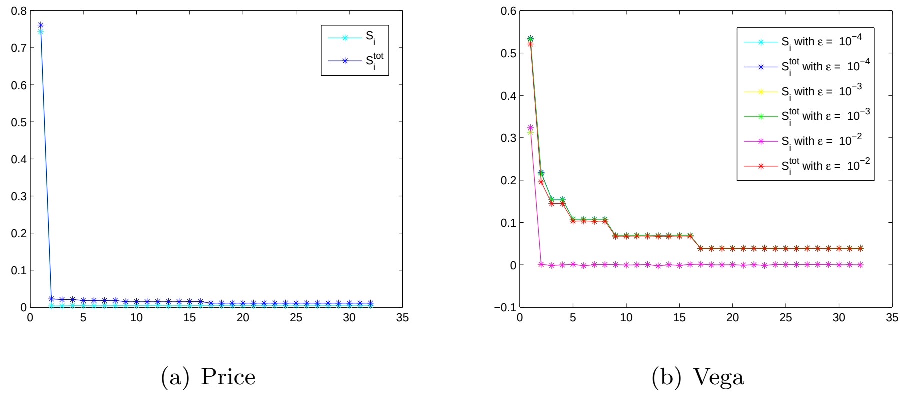


Figure 6: Cliquet option. Details as in Fig. 4.

From these results we draw the following conclusions.

1. Asian option (Figure 4): price, delta and vega are type A functions with $d_S = 1$. The value of sensitivity indexes for the first input, corresponding to the terminal value $t = T$, is $\simeq 1$, while the following variables have sensitivity indexes $\simeq 0$. Gamma remains a type C function as for SD.

2. Double KO option (Figure 5): price, delta and gamma are type A functions. Comments as for the Asian option above. Vega remains a type C function as for SD.

3. Cliquet option (Figure 6): price is a type A function. Similarly to the European option, the value of sensitivity indexes for the first input, corresponding to the terminal value $t = T$, is $\simeq 1$, while the following values of S_i are $\simeq 0$. Vega is a type C function, since the ratio S_i/S_i^{tot} reaches small values revealing interacting variables. Thus in this case BBD is much less efficient than SD.

Finally, the results of GSA for PCA are shown in Figs. 7-9 and in Table 4.

Payoff	**Function**	$\mathbf{S_i/S_i^{tot}}$	$\mathbf{\sum_i S_i}$	$\mathbf{d_T}$	$\mathbf{d_S}$	$\mathbf{d_A}$	**Effect of** ϵ
Asian	Price	0.99→0.3	0.993	1	1	1.01	-
	Delta	0.93→0.001	0.925	1	1	1.12	small
	Gamma	$10^{-2} \to 10^{-4}$	$0.1 \to 10^{-3}$	32	32	$20 \to 3$	high
	Vega	0.99→0.05	0.992	1	1	1.01	no
Double KO	Price	$0.44 \to 10^{-3}$	0.49	32	< 32	2.76	-
	Delta	$0.52 \to 10^{-3}$	0.56	32	< 32	2.6	small
	Gamma	$10^{-3} \to 10^{-4}$	$0.04 \to 10^{-3}$	32	32	27	high
	Vega	$0.02 \to 10^{-4}$	$0.08 \to 10^{-3}$	32	32	$36 \to 12$	high
Cliquet	Price	0.98→0.2	0.888	$\simeq 2$	≤ 2	1.43	-
	Vega	0.565→0.001	0.33	32	≤ 32	6.5	small

Table 4: Summary of GSA metrics and effective dimensions of prices and greeks for PCA scheme. Details as in Table 2.

From these results we draw the following conclusions.

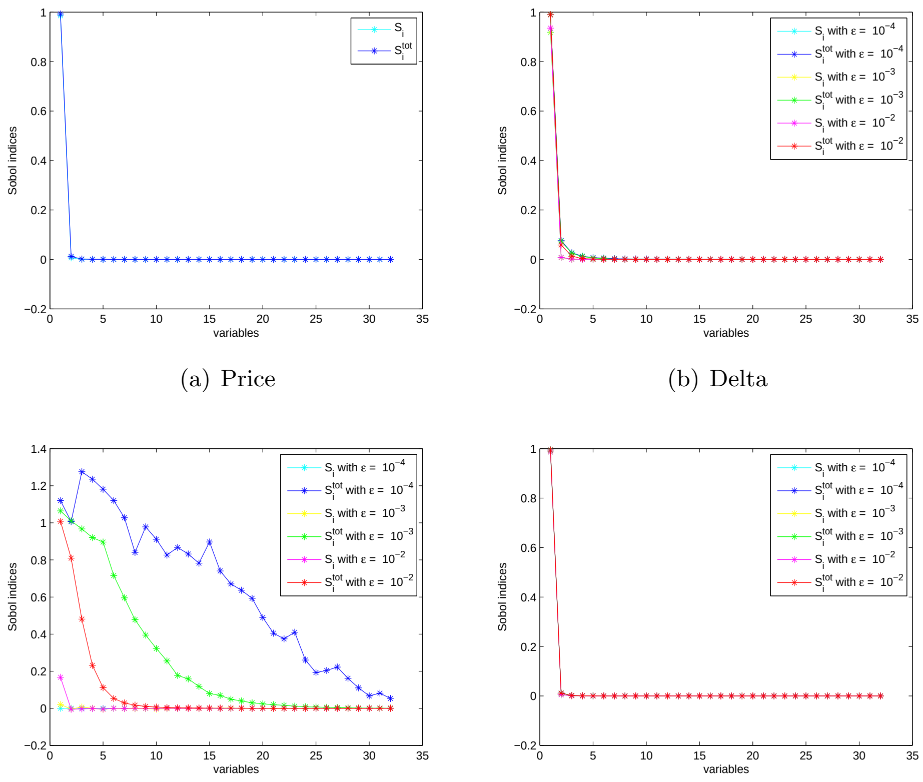
1. Asian option (Figure 7): price, delta and vega are type A functions. Comments as for the BBD case (actually, PCA results to be more efficient than BBD in this case). Gamma remains a type C function as for SD.

2. Double KO option (Figure 8): price and delta are type B functions. Gamma and Vega are type C functions.

3. Cliquet option (Figure 9): price is a type A function. The value of sensitivity indexes for the first input, corresponding to the terminal value $t = T$, is $\simeq 1$, while the following values of S_i are $\simeq 0$. Vega is a type C function, since the ratio S_i/S_i^{tot} reaches small values revealing interacting variables. Thus in this case, PCA is much less efficient than SD (as for BBD).

Let's now move to the multi-asset payoffs (4.4) and (4.5). We have performed the same analysis as for the single-asset case, computing Sobol' indices for all possible combinations of the sampling strategies described in Section 2.1 and with zero correlation[10]: we observed that, when

[10]This is due to the fact that the GSA approach described in this work assumes that the input variables are independent. However, based on the numerical results of Section 4.3, we argue that the conclusions should be similar in the case of non zero correlation. In [KTA12] a possible extension of GSA to models with dependent variables was discussed, but the interpretation of Sobol' indexes is less transparent in that approach.



(a) Price

(b) Delta

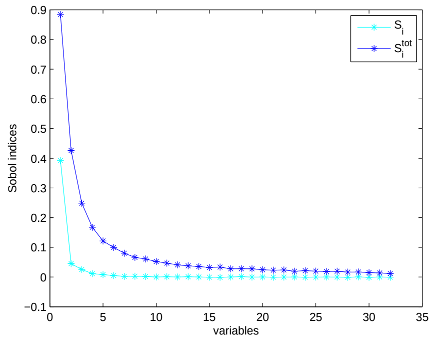
(c) Gamma

(d) Vega

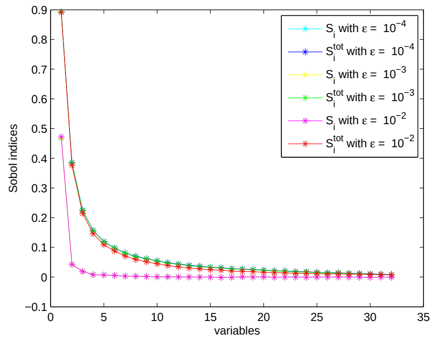
Figure 7: Asian call option price (a) and greeks $(b),(c),(d)$, PCA, $D = 32$. First order Sobol' indices S_i and total sensitivity indices S_i^{tot} versus variate i.

BBD or PCA time discretizations and PCA factorization of the covariance matrix are used and input variables are appropriately re-arranged, a significant reduction in effective dimensions is obtained. In Fig. 10 and Table 5 we compare the results for the price of an Asian basket option with SD+CHOL, BB+PCA and PCA+PCA sampling strategies. In order to highlight the sensitivity of Sobol' indexes to the sampling order of the variables, we also re-arranged the inputs in the following way:

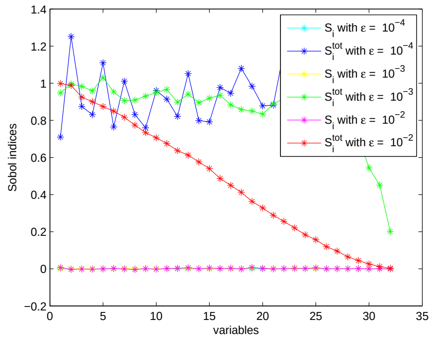
- SD+CHOL: the underlying assets are not sorted, so that spots and volatilities are read in the original (and, thus, not optimal) order. The input variables x_i are used to construct the Brownian motion in the following order (by risk factors): $(W_1(t_1), \ldots, W_1(t_{Nts}), \ldots, W_{Nrf}(t_1), \ldots, W_{Nrf}($
- BB+PCA: the input variables x_i are used to construct the Brownian motion in the following order (by time steps): $(W_{Nrf}(t_{Nts}), \ldots, W_1(t_{Nts}), \ldots)$.
- PCA+PCA: no particular ordering is applied, since the algorithm does the job by itself.



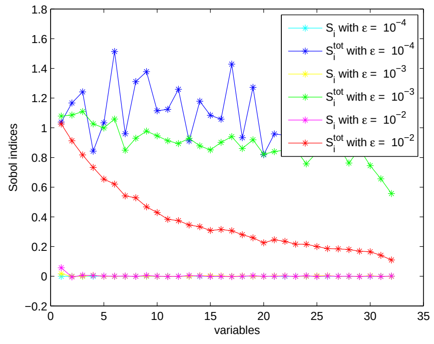
(a) Price



(b) Delta

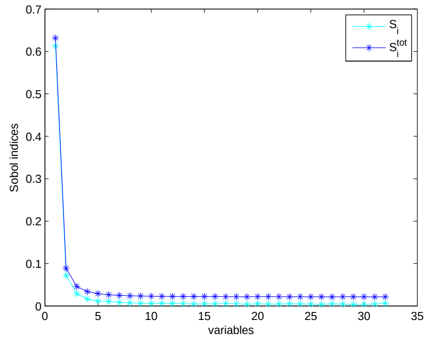


(c) Gamma

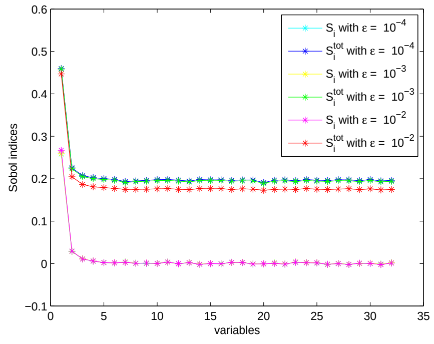


(d) Vega

Figure 8: Double Knock-out call option. Details as in Fig. 7.



(a) Price

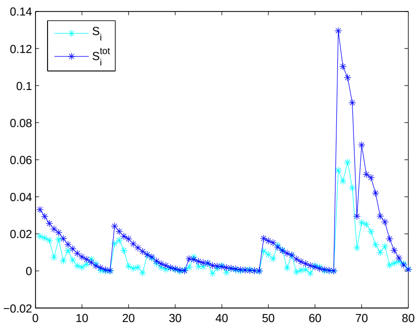


(b) Vega

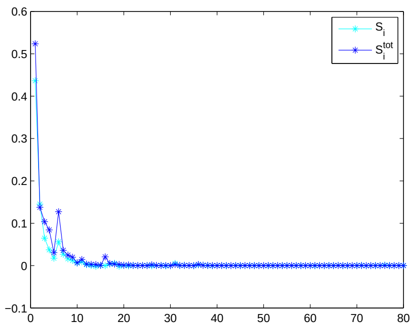
Figure 9: Cliquet option. Details as in Fig. 7.

Payoff	Function	Method	S_i/S_i^{tot}	$\sum_i S_i$	d_T	d_S	d_A
Asian basket	Price	SD+CHOL	0.7→0.1	0.62	$\simeq 80$	< 80	1.3
	Price	BB+PCA	0.83→0.1	0.85	2	≤ 2	1.2
	Price	PCA+PCA	0.99→0.34	0.93	1	1	1.1

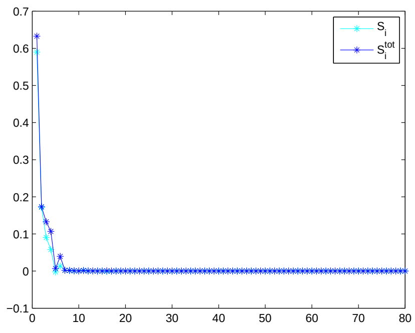
Table 5: Summary of GSA metrics and effective dimensions for the price of the Asian basket option with SD+CHOL, BB+PCA and PCA+PCA schemes. In the first case the variables are ordered by risk factors, while in the second case by time steps. Nominal dimension is $D = 80$. The numerical computations were obtained evaluating the $D + 2$ integrals (3.8) on 2^{13} QMC scenarios. We show significant digits only.



(a) SD+CHOL



(b) BB+PCA



(c) PCA+PCA

Figure 10: Asian basket call option price with different sampling strategies along time and risk fator directions: standard discretization and Cholesky factorization (a), Brownian bridge and PCA (b), PCA on both directions (c). Nominal dimension is $D = 80$. First order Sobol' indices S_i and total sensitivity indices S_i^{tot} versus variate i.

From the results it is clear that, with BB+PCA or PCA+PCA, the price of the Asian basket option is a Type A function, while with SD+CHOL it is a Type B function. In the first case, the most important variables are in the first places, while in the latter they are placed in the end of the sampling vector.

In conclusion, prices and FD greeks are predominantly Type A or Type B functions when BBD or PCA are employed, so that the effective dimension in the truncation sense d_T is reduced.

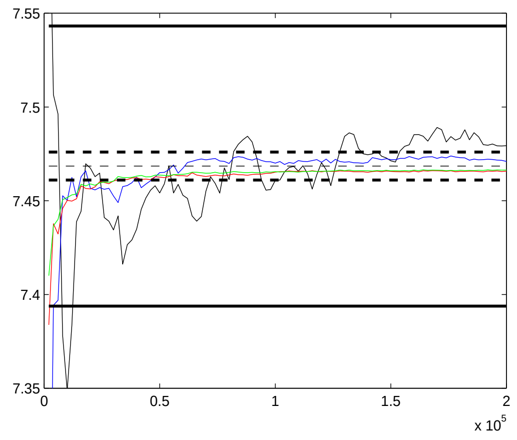
The different efficiency of QMC with BBD or PCA is completely explained by the properties of Sobol' low discrepancy sequences. The initial coordinates of Sobol' LDS are much better distributed than the later high dimensional coordinates [Gla03, CMO97]. BBD and PCA change the order in which inputs (linked with time steps) are sampled. As follows from GSA, in most cases for BBD and PCA the low index variables are much more important than higher index variables. The BBD uses lower index, well distributed coordinates from each D-dimensional LDS vector to determine most of the structure of a path, and reserves other coordinates to fill in finer details. That is, well distributed coordinates are used for important variables and other not so well distributed coordinates are used for far less important variables. Similar considerations hold for the PCA construction. This results in a significantly improved accuracy of QMC integration. However, this technique does not always improve the efficiency of the QMC method as e.g. for Cliquet options: in this case GSA reveals that for SD all inputs are equally important and, moreover, there are no interactions among them, which is an ideal case for application of Sobol' low discrepancy sequences; BBD and PCA, on the other hand, favoring higher index variables destroys independence of inputs introducing interactions, which leads to higher values of d_S and d_A: as a result, we observe degradation in performance of the corresponding QMC methods.

4.3 Convergence analysis

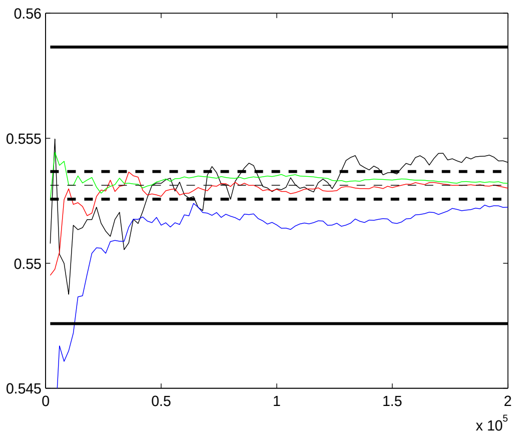
In this Section we compare the relative performances of MC and QMC techniques. This analysis is crucial to establish if QMC outperforms MC, and in what sense.

Firstly, following the suggestion of [Jac01], Section 14.4, we analyze convergence diagrams for prices and greeks, showing the dependence of the MC simulation error upon the number of MC paths. The results for the five payoffs are shown in Figs. 11-15. In the case of multi-asset options, we plot just delta and vega w.r.t. the fifth asset, for the case of correlation $\rho = 0.6$. Similar results hold for other cases.

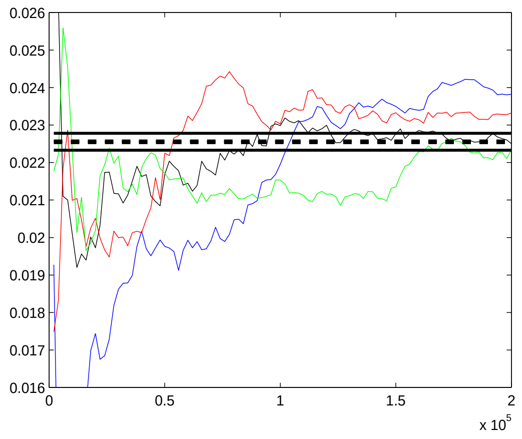
Next, we analyze the relative performance of QMC vs MC in terms of convergence rate. We plot in Figs. 16-20 the root mean square error, eq. (2.21), versus the number of MC scenarios N in Log-Log scale. In all our tests we have chosen an appropriate range for N such that, in the computation of greeks, the bias term is negligible with respect to the variance term (see Appendix A for details). Hence, the observed relations are, with good accuracy, linear, therefore the power law (2.20) is confirmed, and the convergence rates α can be extracted as the slopes of the regression lines. Furthermore, also the intercepts of regression lines provide useful information about the efficiency of the QMC and MC methods: in fact, lower intercepts mean that the simulated value starts closer to the exact value. The resulting slopes and intercepts from linear regression are presented in tabs. 6-9 for all test cases.



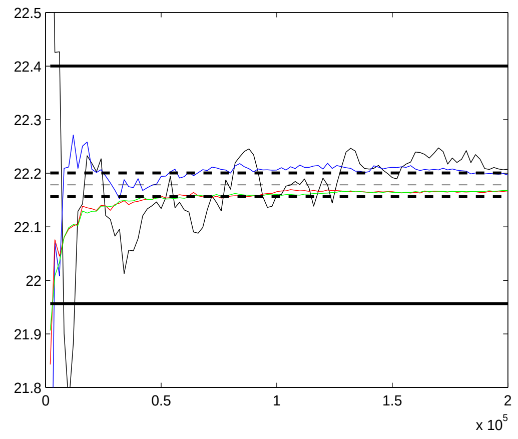
(a) Price



(b) Delta



(c) Gamma

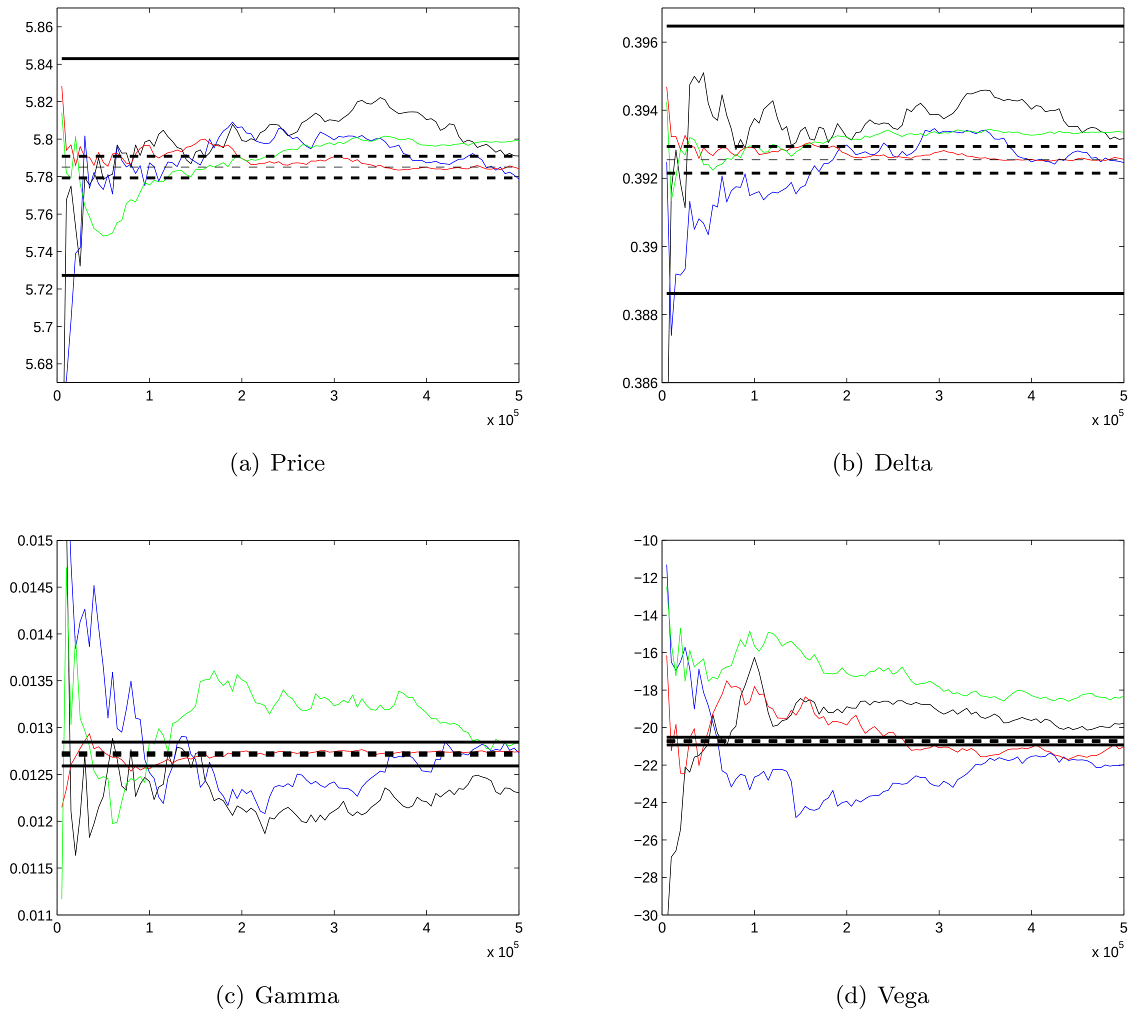


(d) Vega

Figure 11: Asian call option price (a) and greeks $(b), (c), (d)$ convergence diagrams versus number of simulated paths for MC+SD (solid black line), QMC+SD (solid blue line), QMC+BBD (solid red line) and QMC+PCA (solid green line). 1% and 0.1% accuracy regions are marked by horizontal black solid and dashed lines, respectively. Number of dimensions is $D = 32$. Shift parameter is $\epsilon = 10^{-3}$.

Payoff	**Function**	**MC+SD**	**QMC+SD**	**QMC+BBD**	**QMC+PCA**
Asian	Price	-0.3	-0.9	-1.1	-1.1
	Delta	-1.6	-1.8	-2.1	-2.4
	Gamma	-1.8	-1.7	-1.8	-1.8
	Vega	0.2	-0.2	-0.4	-0.5
Double KO	Price	-0.3	-0.4	-0.7	-0.7
	Delta	-1.6	-1.7	-2.1	-2.0
	Gamma	-1.8	-1.9	-2.7	-1.8
	Vega	1.6	1.6	1.6	1.6
Cliquet	Price	-2.2	-3.3	-2.6	-2.6
	Vega	-1.8	-3.1	-1.9	-1.8

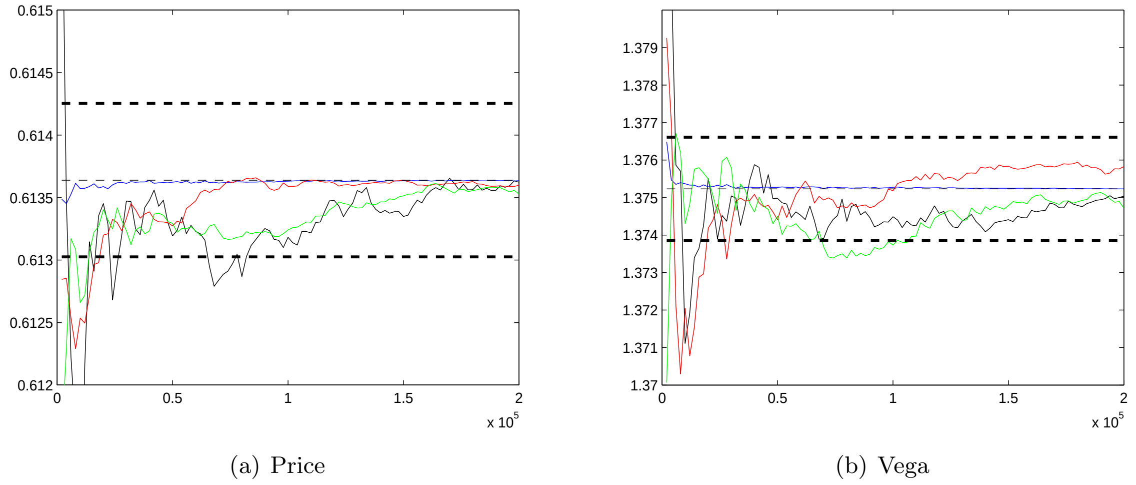
Table 6: Intercepts from linear regression, for single-asset options with MC+SD, QMC+SD, QMC+BBD and QMC+PCA, $L = 100$ runs. Only significant digits are shown, standard error on the estimates is around 10%. Results are shown for $N = 512$ paths.



(a) Price (b) Delta

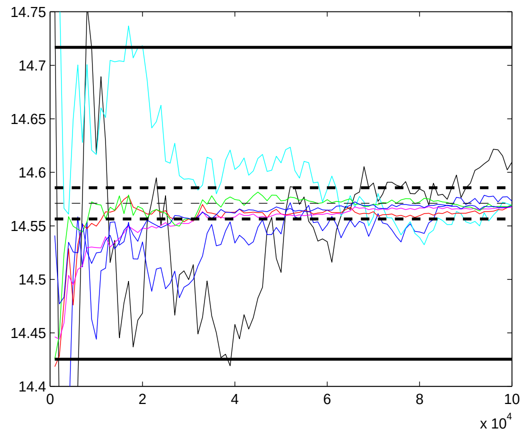
(c) Gamma (d) Vega

Figure 12: Double Knock-out call option. Details as in Fig. 11.

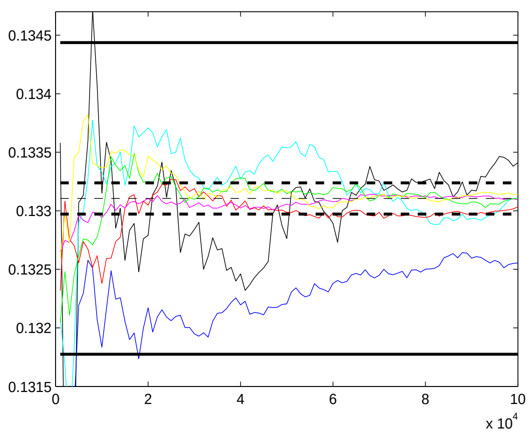


(a) Price (b) Vega

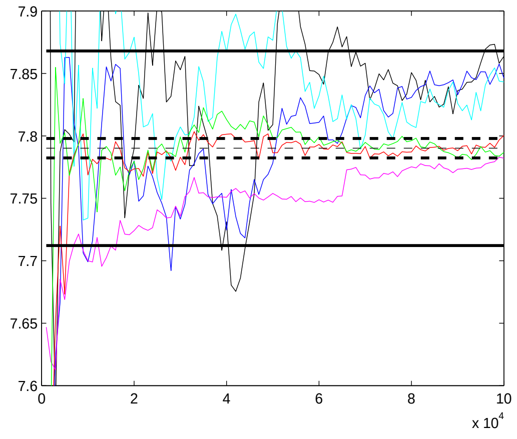
Figure 13: Cliquet option. Details as in Fig. 11.



(a) Price



(b) Delta w.r.t. 5th asset

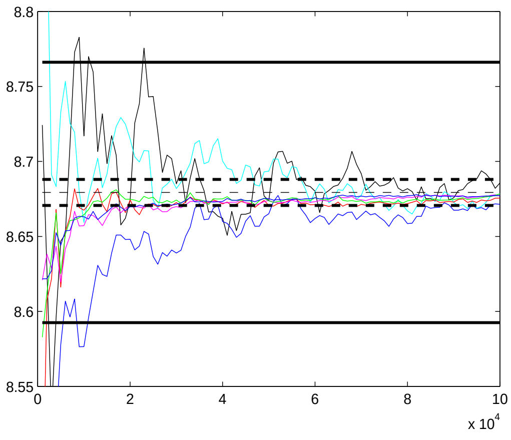


(c) Vega w.r.t. 5th asset

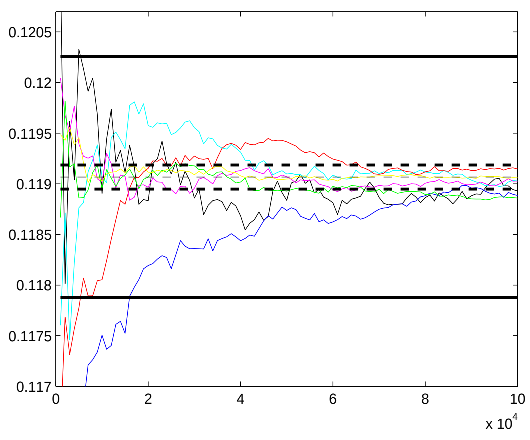
Figure 14: European basket call option price (a) and selected greeks (b), (c) convergence diagrams versus number of simulated paths for different combinations of sampling techniques: MC+SD+CHOL (black), QMC+SD+CHOL (blue), QMC+SD+PCA (cyan), QMC+BBD+CHOL (red), QMC+BBD+PCA (magenta), QMC+PCA+CHOL (green), QMC+PCA+PCA (yellow). 1% and 0.1% accuracy regions are marked by horizontal black solid and dashed lines, respectively. Number of dimensions is $D = 80$, correlation is $\rho = 0.6$ and shift parameter is $\epsilon = 10^{-3}$.

Payoff	**Function**	**MC+SD**	**QMC+SD**	**QMC+BBD**	**QMC+PCA**
Asian	Price	-0.49	-0.65	-0.76	-0.74
	Delta	-0.48	-0.55	-0.61	-0.64
	Gamma	-0.48	-0.53	-0.51	-0.50
	Vega	-0.49	-0.64	-0.72	-0.74
Double KO	Price	-0.51	-0.50	-0.60	-0.56
	Delta	-0.50	-0.51	-0.59	-0.56
	Gamma	-0.52	-0.48	-0.58	-0.53
	Vega	-0.52	-0.52	-0.55	-0.53
Cliquet	Price	-0.50	-1.00	-0.80	-0.55
	Vega	-0.51	-0.76	-0.61	-0.55

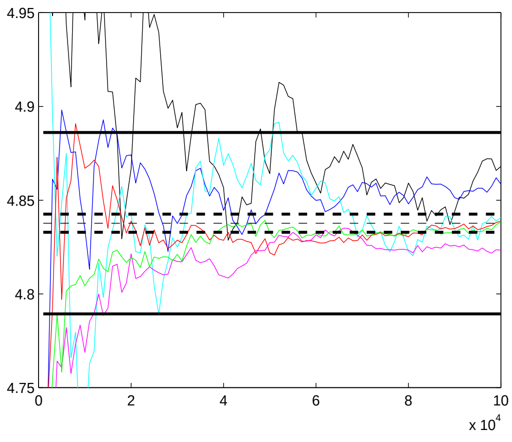
Table 7: Slopes from linear regression for single-asset options, as in Table 6. Standard error on the estimates is around 5% (20% for the price of the Cliquet with QMC+BBD).



(a) Price



(b) Delta w.r.t. 5th asset



(c) Vega w.r.t. 5th asset

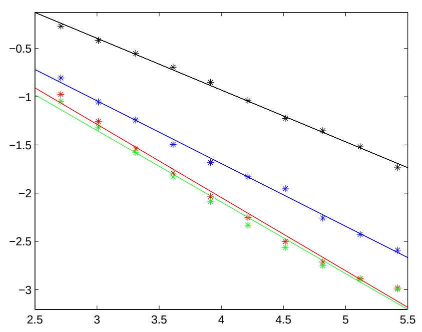
Figure 15: Asian basket call option. Details as in Fig. 14.

Correlation	Method	Slopes			Intercepts		
		Price	Delta	Vega	Price	Delta	Vega
$\rho = 0$	MC+STD+CHOL	-0.49	-0.50	-0.51	-0.04	-2.00	0.28
	QMC+STD+CHOL	-0.71	-0.54	-0.67	-0.19	-2.33	0.13
	QMC+STD+PCA	-0.75	-0.64	-0.74	-0.07	-2.14	0.32
	QMC+BB+CHOL	-0.91	-0.63	-0.76	-0.47	-2.49	-0.17
	QMC+BB+PCA	-0.92	-0.65	-0.78	-0.42	-2.51	-0.13
	QMC+PCA+CHOL	-0.80	-0.58	-0.74	-0.55	-2.45	-0.15
	QMC+PCA+PCA	-0.75	-0.56	-0.61	-0.51	-2.47	0.07
$\rho = 0.3$	MC+STD+CHOL	-0.50	-0.48	-0.51	0.12	-2.00	0.28
	QMC+STD+CHOL	-0.71	-0.55	-0.65	-0.01	-2.33	0.21
	QMC+STD+PCA	-0.75	-0.60	-0.62	0.02	-2.22	0.30
	QMC+BB+CHOL	-0.92	-0.63	-0.80	-0.36	-2.49	-0.09
	QMC+BB+PCA	-0.95	-0.65	-0.65	-0.45	-2.57	-0.17
	QMC+PCA+CHOL	-0.78	-0.54	-0.76	-0.46	-2.49	-0.11
	QMC+PCA+PCA	-0.82	-0.59	-0.64	-0.53	-2.50	-0.01
$\rho = 0.6$	MC+STD+CHOL	-0.49	-0.48	-0.50	0.21	-2.00	0.27
	QMC+STD+CHOL	-0.72	-0.60	-0.62	0.12	-2.21	0.25
	QMC+STD+PCA	-0.79	-0.65	-0.67	0.16	-2.13	0.24
	QMC+BB+CHOL	-0.93	-0.64	-0.81	-0.28	-2.54	-0.10
	QMC+BB+PCA	-0.97	-0.67	-0.76	-0.32	-2.67	-0.23
	QMC+PCA+CHOL	-0.78	-0.60	-0.75	-0.40	-2.45	-0.13
	QMC+PCA+PCA	-0.80	-0.62	-0.73	-0.51	-2.52	0.06
$\rho = 0.9$	MC+STD+CHOL	-0.49	-0.47	-0.48	0.27	-2.01	0.26
	QMC+STD+CHOL	-0.78	-0.64	-0.64	0.23	-2.13	0.25
	QMC+STD+PCA	-0.83	-0.64	-0.74	0.28	-2.16	0.30
	QMC+BB+CHOL	-0.97	-0.64	-0.83	-0.22	-2.64	-0.09
	QMC+BB+PCA	-0.95	-0.72	-0.85	-0.25	-2.64	-0.12
	QMC+PCA+CHOL	-0.86	-0.56	-0.78	-0.29	-2.57	-0.10
	QMC+PCA+PCA	-0.83	-0.58	-0.74	-0.46	-2.61	-0.02

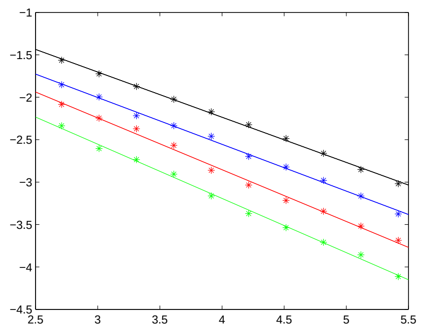
Table 8: Slopes and intercepts from linear regression, for European basket option with different sampling techniques and correlation levels, $L = 100$ runs. Only significant digits are shown, standard error on the estimates is around 10% for intercepts and 5% for slopes. Results of intercepts are shown for $N = 512$ paths.

Correlation	Method	Slopes			Intercepts		
		Price	Delta	Vega	Price	Delta	Vega
$\rho = 0$	MC+STD+CHOL	-0.46	-0.47	-0.46	-0.31	-2.13	-0.10
	QMC+STD+CHOL	-0.70	-0.56	-0.64	-0.57	-2.37	-0.34
	QMC+STD+PCA	-0.76	-0.56	-0.80	-0.40	-2.31	-0.02
	QMC+BB+CHOL	-0.76	-0.58	-0.69	-0.75	-2.44	-0.46
	QMC+BB+PCA	-0.79	-0.58	-0.69	-0.72	-2.47	-0.45
	QMC+PCA+CHOL	-0.87	-0.65	-0.72	-0.79	-2.43	-0.53
	QMC+PCA+PCA	-0.84	-0.57	-0.64	-0.74	-2.62	-0.39
$\rho = 0.3$	MC+STD+CHOL	-0.49	-0.48	-0.45	-0.19	-2.14	-0.08
	QMC+STD+CHOL	-0.71	-0.55	-0.68	-0.37	-2.33	-0.24
	QMC+STD+PCA	-0.82	-0.59	-0.64	-0.23	-2.32	-0.01
	QMC+BB+CHOL	-0.76	-0.56	-0.70	-0.66	-2.52	-0.48
	QMC+BB+PCA	-0.89	-0.54	-0.57	-0.69	-2.57	-0.33
	QMC+PCA+CHOL	-0.86	-0.57	-0.75	-0.68	-2.53	-0.46
	QMC+PCA+PCA	-0.93	-0.59	-0.62	-0.81	-2.69	-0.60
$\rho = 0.6$	MC+STD+CHOL	-0.48	-0.46	-0.48	-0.05	-2.14	-0.05
	QMC+STD+CHOL	-0.74	-0.55	-0.67	-0.21	-2.33	-0.18
	QMC+STD+PCA	-0.83	-0.57	-0.70	-0.13	-2.35	-0.12
	QMC+BB+CHOL	-0.81	-0.55	-0.73	-0.56	-2.54	-0.46
	QMC+BB+PCA	-0.95	-0.61	-0.61	-0.56	-2.49	-0.52
	QMC+PCA+CHOL	-0.89	-0.59	-0.67	-0.57	-2.60	-0.55
	QMC+PCA+PCA	-0.97	-0.55	-0.73	-0.65	-2.81	-0.46
$\rho = 0.9$	MC+STD+CHOL	-0.49	-0.48	-0.48	0.01	-2.13	-0.06
	QMC+STD+CHOL	-0.79	-0.54	-0.76	-0.09	-2.39	-0.09
	QMC+STD+PCA	-0.84	-0.56	-0.82	-0.05	-2.37	-0.05
	QMC+BB+CHOL	-0.89	-0.58	-0.77	-0.48	-2.52	-0.50
	QMC+BB+PCA	-0.94	-0.64	-0.74	-0.49	-2.47	-0.55
	QMC+PCA+CHOL	-0.92	-0.53	-0.82	-0.52	-2.71	-0.49
	QMC+PCA+PCA	-0.95	-0.63	-0.74	-0.54	-2.79	-0.53

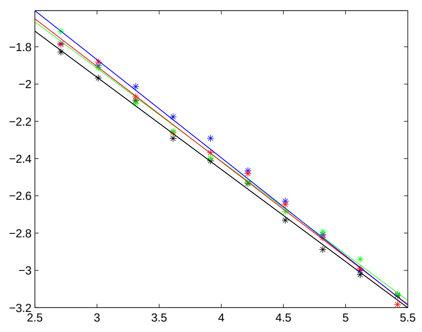
Table 9: Slopes and intercepts for Asian basket option. Details as in Table 8.



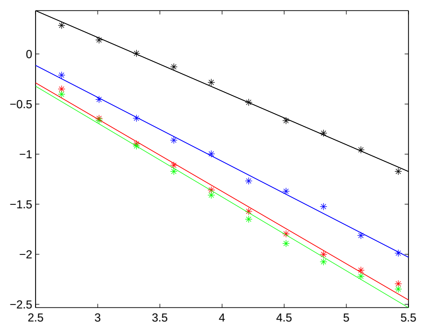
(a) Price



(b) Delta

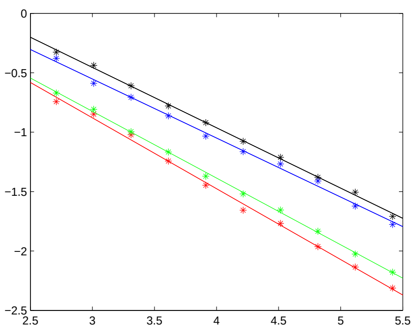


(c) Gamma

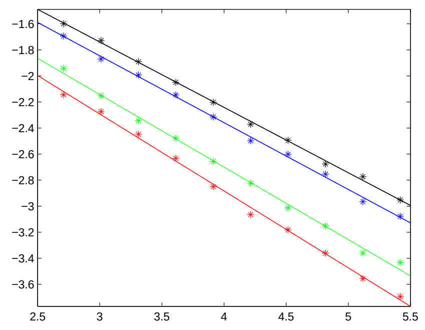


(d) Vega

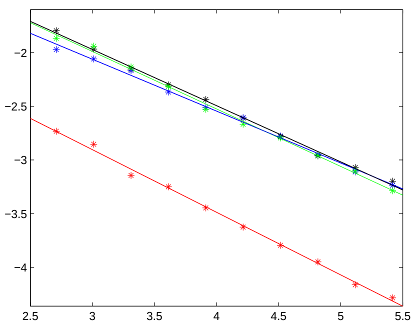
Figure 16: Asian call option price (a) and greeks $(b), (c), (d)$, Log-Log plots of error ε_N versus number of simulated paths $N = 2^p$, $p = 9, \dots, 18$, $D = 32$, $\epsilon = 10^{-3}$, $L = 100$ runs: MC+SD (black), QMC+SD (blue), QMC+BBD (red), QMC+PCA (green). Linear regression lines are also shown.



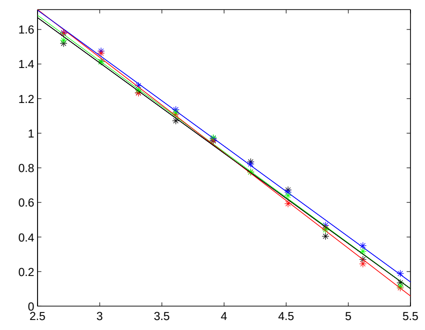
(a) Price



(b) Delta

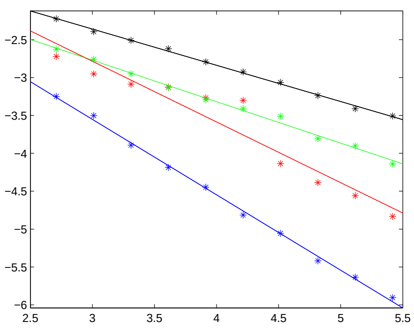


(c) Gamma

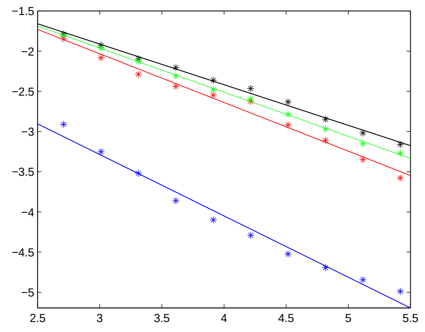


(d) Vega

Figure 17: Double Knock-out call option. Details as in Fig. 16.

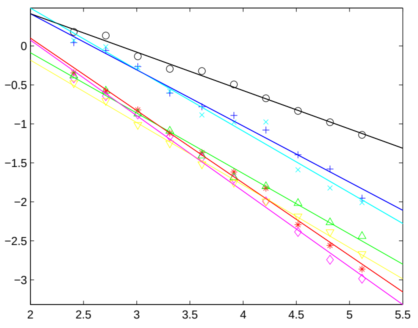


(a) Price

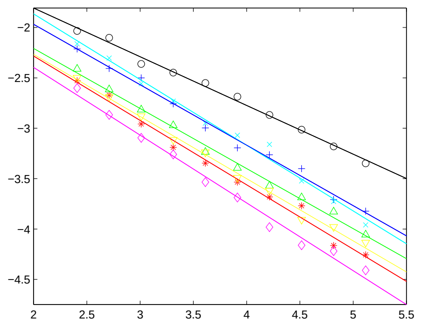


(b) Vega

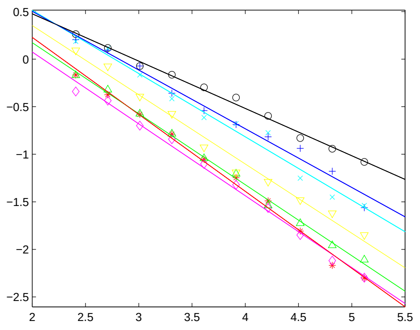
Figure 18: Cliquet option. Details as in Fig. 16.



(a) Price

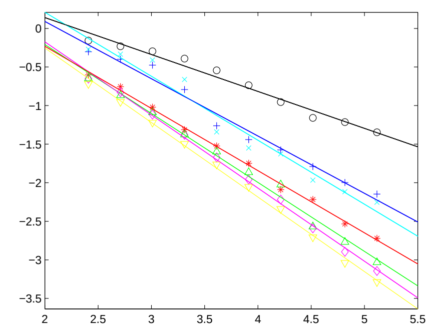


(b) Delta asset 5

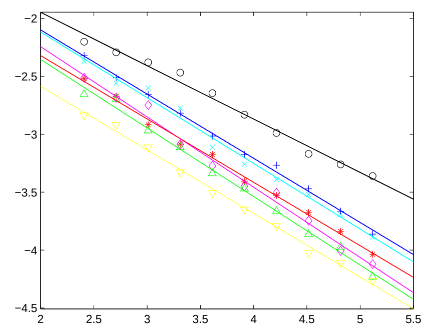


(c) Vega asset 5

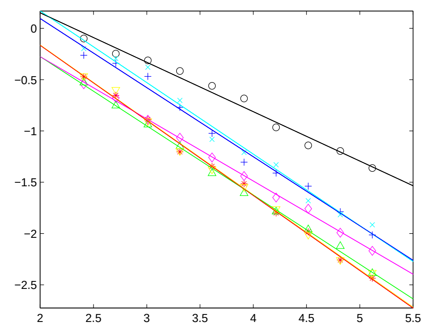
Figure 19: European basket call option price (a) and selected greeks $(b), (c)$, Log-Log plots of error ε_N versus number of simulated paths, $N = 2^p,\ p = 8, \ldots, 17,\ D = 80,\ \epsilon = 10^{-3},\ L = 100$ runs:: MC+SD+CHOL (black), QMC+SD+CHOL (blue), QMC+SD+PCA (cyan), QMC+BBD+CHOL (red), QMC+BBD+PCA (magenta), QMC+PCA+CHOL (green), QMC+PCA+PCA (yellow). Linear regression lines are also shown. Correlation is $\rho = 0.6$.



(a) Price



(b) Delta asset 5



(c) Vega asset 5

Figure 20: Asian basket call option. Details as in Fig. 19.

We observe that, for the Asian call, QMC+PCA outperforms all other methods for price, delta and vega (for gamma all methods show similar convergence), with QMC+BBD being only marginally less efficient. Higher α and lower intercepts allow for faster convergence[11]. Moreover, convergence appears to be much smoother and more stable. Similar considerations hold for the double barrier option with QMC+BBD, the Cliquet with QMC+SD, the European basket option with QMC+BBD+PCA and the basket Asian option with QMC+PCA+PCA. These results can be explained by the fact that the above mentioned sampling strategies are optimal, in the sense that they are intrinsically designed to extract initial coordinates of the gaussian vector $\boldsymbol{Z} = (Z_1, \ldots, Z_D)$ to construct the most important coordinates of the underlying assets' vector $\boldsymbol{S} = \big(S_1(t_1), \ldots, S_{N_{rf}}(t_1), \ldots, S_1(t_{N_{ts}}), \ldots, S_{N_{rf}}(t_{N_{ts}})\big)$.

4.4 Speed-Up analysis

A typical question with Monte Carlo simulation is "how many scenarios are necessary to achieve a given precision?". When comparing two numerical simulation methods, the typical question becomes "how many scenarios may I save using method B instead of method A, preserving the same precision?".

A useful measure of the relative computational performance of two numerical methods is the so called speed-up $S_*(a)$ [KMRZ98a, PT96]. It is defined as

$$S_*^{(A,B)}(a) = \frac{N_*^{(A)}(a)}{N_*^{(B)}(a)}\,, \tag{4.6}$$

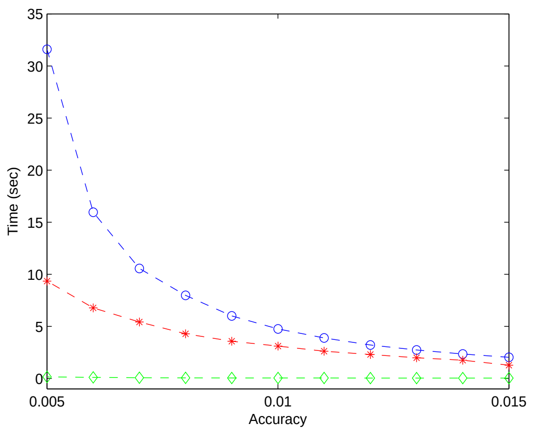
where $N_*^{(A)}(a)$ is the number of scenarios needed to computational scheme A (which can be any of the different combinations of random number generator, time discretization and covariance factorization algorithms) to reach and maintain a given accuracy a w.r.t. exact or almost exact results[12]. Thus, the speed-up $S_*(a)$ quantifies the computational gain of method B w.r.t. method A.

We show in table 10 the Speed-Up computation results for QMC method with optimal sampling strategies over standard MC, when accuracies of 1% and 0.1% are to be reached. The simulation methods chosen for the computation of QMC Speed-Up are those which achieved the highest performance for our test cases, as concluded from Section 4.3.

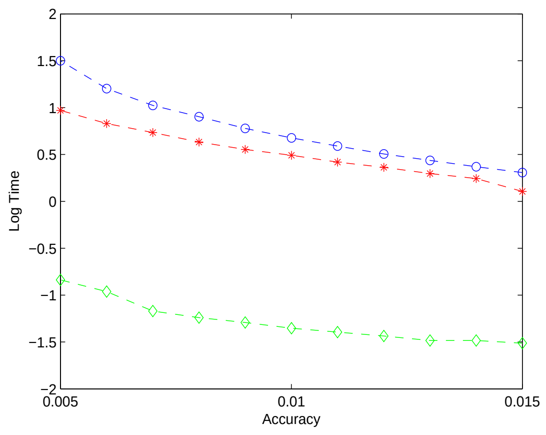
In general, QMC with BBD or PCA largely outperforms the other methods, with a speed-up factor close to 10^3 in some cases. QMC with SD is the best method for Cliquet. We notice in particular that, in most cases, a ten-fold increase of the accuracy a results in a two-fold increase of speed-up $S_*(a)$. However, in a few cases (Cliquet and basket options), such an increase can result in up to ten-folds increase of $S_*(a)$. We notice that the Speed-Up measure actually makes no reference to the computational time but is rather defined as a ratio of number of simulations. This, in turn, is proportional to CPU time. Therefore, it is interesting to fix a given accuracy and compute the CPU time needed to reach it with various combinations of methods, including FD and AAD. Indeed, even though adjoints allow for big savings, in terms of computational time, w.r.t. finite differences, the accuracy of the computation is rather given by the simulation method: fixing a target accuracy a, QMC will reach it with much fewer scenarios

[11]We stress that slopes and intercepts shown in the previous Figs. 16-20 do not depend on the details of the simulations, in particular the MC seed or the LDS starting point, since we are averaging over $L = 100$ runs.

[12]The thresholds N_* could be evaluated through direct simulation, but this would be extremely computationally expensive. Thus we resort to a simpler algorithm: we identify the number of scenarios $N_*^{(A)}(a)$ in eq. (4.6) as the first number of simulated paths such that, for any $N > N_*$, $V - a \leq V_N \pm 3\,\varepsilon \leq V + a$, where V and V_N are respectively the exact and simulated values of prices or greeks and ε is the standard error. Then, $N_*^{(A)}(a)$ can be estimated through linear regression results of Section 4.3. Some concern should be given to extrapolation for finite differences, as discussed in [BKS15].

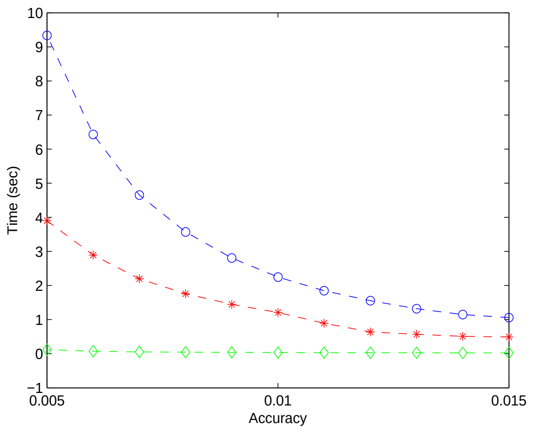


(a) Abslolute CPU time

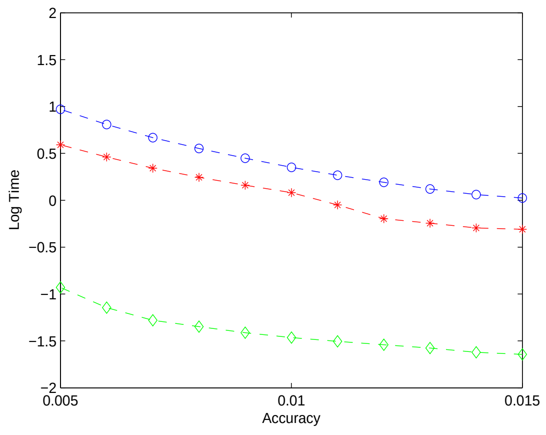


(b) Log CPU time

Figure 21: European basket option. Absolute CPU time (a) and logarithmic CPU time (b) needed to compute price and all greeks (deltas and vegas), for different target accuracies: MC+SD+CHOL with AAD (blue), QMC+BBD+PCA with FD (red), QMC+BBD+PCA with AAD (green). The number of underlyings is $N_{rf} = 5$ and the number of time steps is $N_{ts} = 16$. Correlation is $\rho = 0.3$, the other parameters are as described in Section 4.1

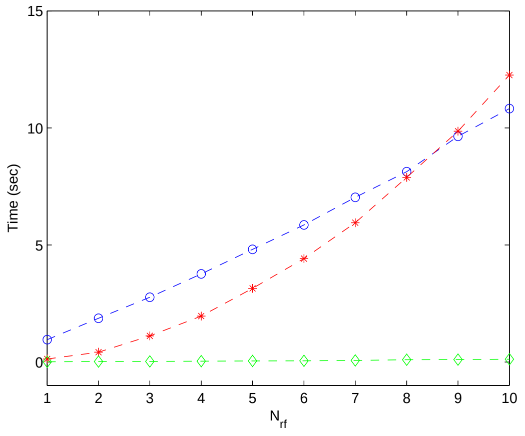


(a) Abslolute CPU time

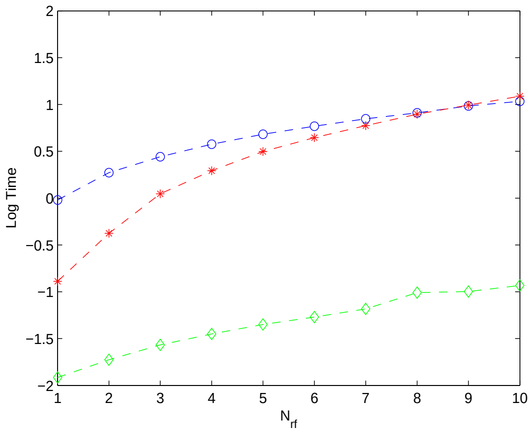


(b) Log CPU time

Figure 22: Asian basket option. Absolute CPU time (a) and logarithmic CPU time (b) needed to compute price and all greeks (deltas and vegas), for different target accuracies: MC+SD+CHOL with AAD (blue), QMC+PCA+PCA with FD (red), QMC+PCA+PCA with AAD (green). The number of underlyings is $N_{rf} = 5$ and the number of time steps is $N_{ts} = 16$. Correlation is $\rho = 0.3$, the other parameters are as described in Section 4.1

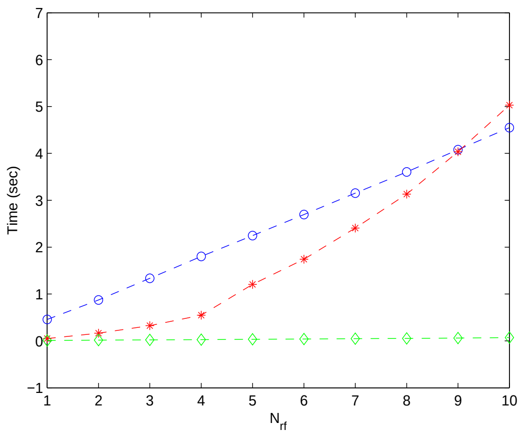


(a) Abslolute CPU time

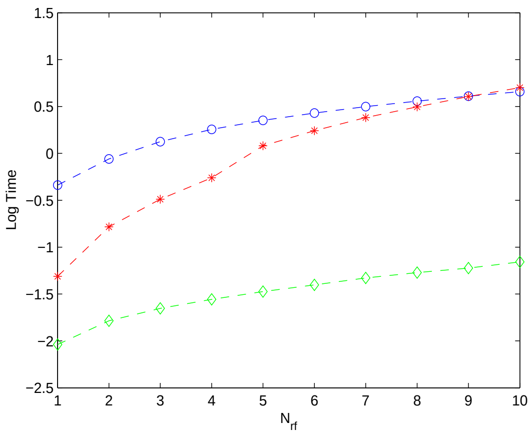


(b) Log CPU time

Figure 23: European basket option. Absolute CPU time (a) and logarithmic CPU time (b) needed to compute price and all greeks (deltas and vegas), for increasing number of underlying assets N_{rf}: MC+SD+CHOL with AAD (blue), QMC+BBD+PCA with FD (red), QMC+BBD+PCA with AAD (green). The target accuracy is fixed to 1% and the number of time steps is always $N_{ts} = 16$. Correlation is $\rho = 0.3$, the other parameters are as described in Section 4.1



(a) Abslolute CPU time



(b) Log CPU time

Figure 24: Asian basket option. Absolute CPU time (a) and logarithmic CPU time (b) needed to compute price and all greeks (deltas and vegas), for increasing number of underlying assets N_{rf}: MC+SD+CHOL with AAD (blue), QMC+BBD+PCA with FD (red), QMC+BBD+PCA with AAD (green). The target accuracy is fixed to 1% and the number of time steps is always $N_{ts} = 16$. Correlation is $\rho = 0.3$, the other parameters are as described in Section 4.1

Payoff	Function	QMCvsMC Speed-Up	
		$a = 1\%$	$a = 0.1\%$
Asian	Price	80	300
	Delta	50	150
	Gamma	1	-
	Vega	60	300
Double KO	Price	10	30
	Delta	20	30
	Gamma	150	300
	Vega	3	5
Cliquet	Price	30	400
	Vega	100	500
European basket	Price	110	800
	Delta	40	150
	Vega	30	80
Asian basket	Price	80	700
	Delta	20	50
	Vega	30	150

Table 10: Speed-Up $S_*(a)$ of optimal QMC vs MC techniques, in order to achieve 1% and 0.1% accuracy in the computation of prices and FD greeks. The shift ϵ for finite differences and the correlations are the same as used in the previous sections. Missing values of S_* mean that the required accuracy cannot be reached since it is smaller than the bias. Regarding the QMC simulation, the sampling strategies considered are: PCA for Asian call, BBD for double knock-out, SD for cliquet, BBD+PCA for European basket, PCA+PCA for Asian basket.

than MC, as measured by $S_*(a)$. In Figs. 21 and 22 we show absolute CPU times necessary to evaluate price and all greeks at a given accuracy for MC and optimized QMC, with AAD or FD, for our test cases of European and Asian basket options with 5 correlated underlyings and 16 time steps, which is quite typical case in real financial applications.

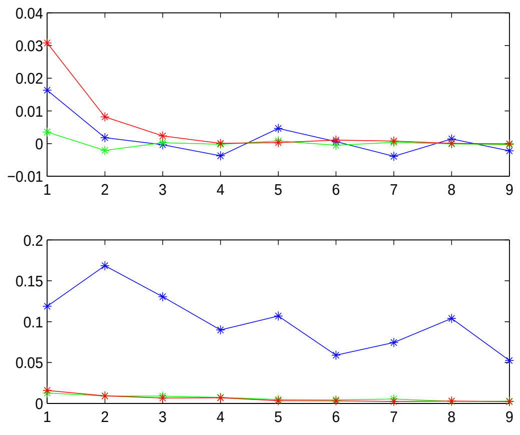
We observe that, while QMC with AAD is of course the best choice, QMC with FD runs in comparable times as MC with AAD for accuracies up to few percent and is actually faster for higher accuracies. When the number of underlyings is increased, AAD will become favourable w.r.t. FD in terms of computational time when the same accuracy is to be reached. Fixing target accuracy to 1% and increasing the number of underlyings[13], we observe from Figs. 23 and 24 that AAD becomes faster than FD starting from $N_{rf} \simeq 10$. We recall that, since we are using central differences, FD computation of all deltas and vegas requires $4 \cdot N_{rf}$ re-pricings.

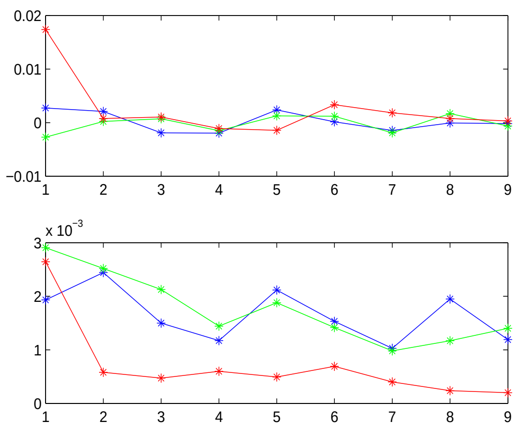
It follows from these simple experiments that AAD without QMC is not guaranteed to be faster than FD if accuracy is concerned. We further comment on this in the Conclusions.

4.5 Stability analysis

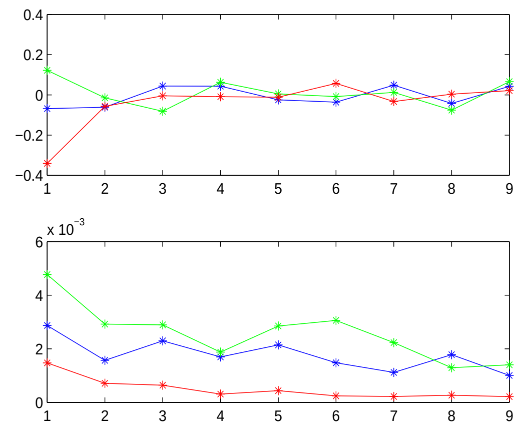
We have already observed that QMC convergence is often smoother than MC (see Figs. 11-15): such monotonicity and stability guarantee better convergence for a given number of paths N. In order to quantify monotonicity and stability of the various numerical techniques, the following strategy is used: we divide the range of path simulations N in 10 windows of equal length, and we compute the sample mean m_i and the sample standard deviation ("volatility") s_i for each window i. Then, log-returns $\log(m_i/m_{i-1})$ and volatilities s_i, for $i = 2, \ldots, 10$, are used as measures of, respectively, monotonicity and stability: "monotonic" convergence will show non oscillating log-returns converging to zero, "stable" convergence will show low and almost

[13]For simplicity we assume that $S_*(a)$ is almost constant in the range $N_{rf} = 1, \ldots, 10$.


(a) Price



(b) Delta



(c) Gamma

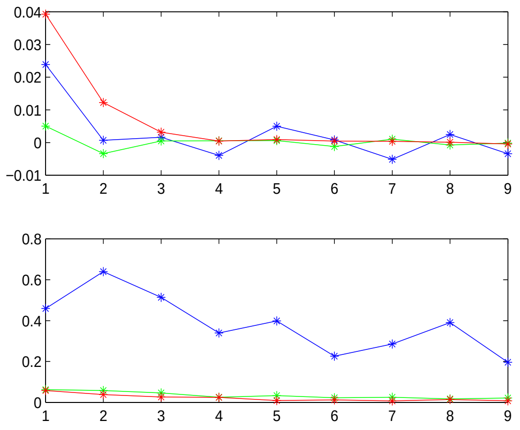

(d) Vega

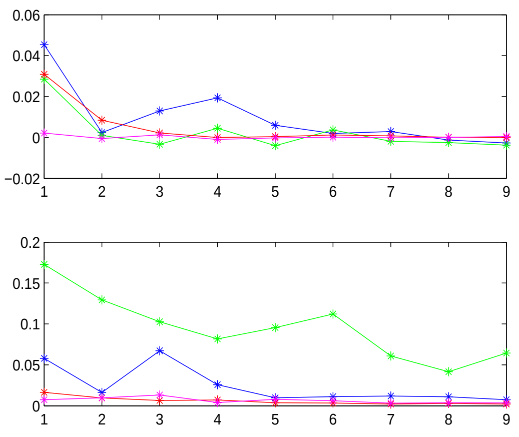
Figure 25: Log-returns (upper plots) and volatilities (lower plots) of Asian call option price (a) and greeks $(b), (c), (d)$, for $D = 32$, $\epsilon = 10^{-3}$, MC+SD (blue), rQMC+BBD (green) and pure QMC+BBD (red). The number of simulation paths ranges from 100 to 10,000 grouped in 10 windows each containing 10 samples (x-axis).

flat volatility. We performed stability analysis for MC and QMC methods. For QMC we used two different generators: pure QMC with `BRODA` generator and randomized Quasi Monte Carlo (rQMC) with `Matlab` generator[14]. As an example, we plot in Fig. 25 the results for the Asian option comparing the stability of QMC+BBD with MC+SD.

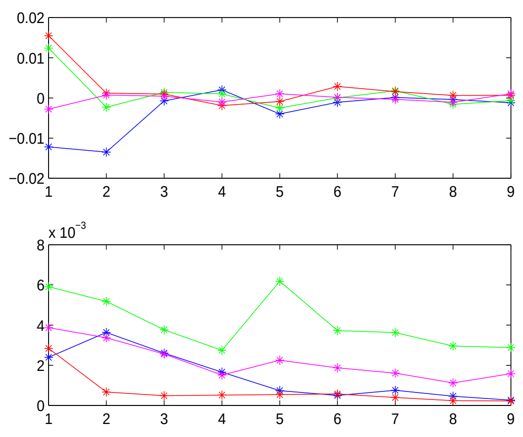
We observe that, in general, QMC+`BRODA` and rQMC+`Matlab` are more monotonic and stable than MC+SD. However, this fact is less evident for Asian delta and gamma, where QMC lacks monotonicity and stability w.r.t. MC, with QMC+BRODA being slightly more stable than rQMC+`Matlab`. As we know from the results of GSA for this case that higher order interactions are present and the effective dimensions are large (see Table 3).

In order to understand also the effect of dimension D on monotonicity and stability, we run a similar experiment for an Asian option with $D = 252$ fixing dates using both QMC and rQMC with SD and BBD. The results are shown in Fig. 26. We observe that pure QMC with

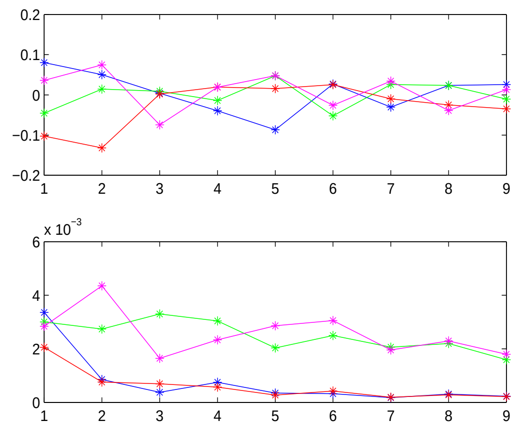
[14]Matlab Function `sobolset` with the `MatousekAffineOwen` scrambling method was used.



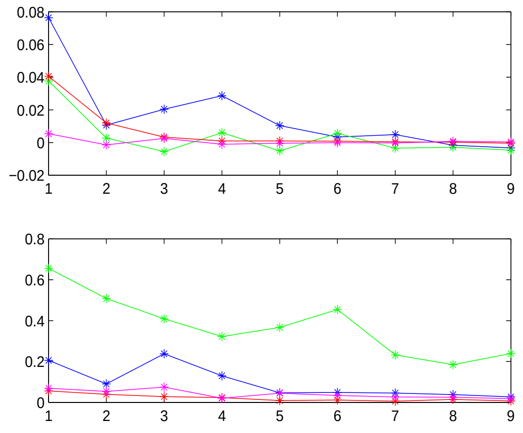
(a) Price



(b) Delta



(c) Gamma



(d) Vega

Figure 26: Asian call option with $D = 252$, $\epsilon = 10^{-3}$. Results are shown for rQMC+SD+`Matlab` (green) and rQMC+BBD+`Matlab` (magenta), and QMC+SD+`BRODA` (blue) and QMC+BBD+`Broda` (red).

`BRODA` generator preserves monotonicity and stability much more than randomized QMC based on `Matlab` generator for all cases including delta and gamma, with QMC+BBD+`BRODA` showing the best stability. It is also interesting to note that the increase in dimension resulted in the decrease in the effective dimensions for the case of the BBD but not for the SD.

We conclude that good high-dimensional LDS generators are crucial to obtain a smooth monotonic and stable convergence of the MC Simulation in high effective dimensional problems.

5 Conclusions

In this work we presented an updated overview of the application of QMC and GSA methods in finance, w.r.t. standard MC methods. In particular, we considered prices and greeks (delta, gamma, vega) for selected payoffs with increasing degree of complexity and path-dependency (Arithmetic Asian Call, Double Knock-Out Barrier, Cliquet, European and Asian Basket options). We compared standard discretization (SD) vs Brownian bridge discretization (BBD) and Principal Component Analysis (PCA) schemes of the underlying stochastic diffusion process, as well as Cholesky vs PCA factorization of the covariance matrix, and different sampling of the underlying distribution using pseudo random vs high dimensional Sobol' low discrepancy sequences. We applied GSA and performed detailed and systematic analysis of convergence diagrams, error estimation, performance, speed-up and stability of the different MC and QMC simulations.

The GSA results in Section 4.2 revealed that effective dimensions associated with BBD and PCA simulations are generally lower than those associated with SD and Cholesky simulations, and showed how much such dimension reduction acts for different payoffs and greeks (Figures 1-10 and Tables 2-5). Effective dimensions, being linked with the structure of ANOVA decompositions (the number of important inputs, importance of high order interactions) fully explain the superior efficiency of QMC with BBD and PCA due to the specifics of Sobol' sequences. An exception is represented by Cliquet options.

The performance analysis results presented in Section 4.3 confirmed that QMC with BBD or PCA outperforms MC in most cases, showing faster and more stable convergence to exact or almost exact results (Figs. 11-15, 16-20, and Tables 6-9).

The speed-up analysis results in Section 4.4 shows that QMC with optimal sampling strategies allows significative reduction of the number of scenarios to achieve a given accuracy, leading to significative reduction of computational efforts (Table 10). The size of the reduction costs scales up to almost 10^3. This fact is very important when the computation of a large number of price sensitivities (greeks) has to be performed, as in the case of multi-asset options, because the computational time increases linearly with the number of underlyings if standard (finite differences, or FD) techniques are employed. We compared the computational efforts needed by FD and AAD in computing price and all first order greeks with and without QMC, fixing the desired accuracy. Remarkably, we obtained that QMC with FD runs in comparable times as MC with AAD for medium sized baskets, while the best choice is clearly QMC with AAD, which allows for very fast and efficient results as shown in Figs. 21-24. It means that, taking into account the accuracy of the computation, AAD is not guaranteed to be faster than FD if it is implemented with standard MC rather than QMC (at least for a modest number of derivatives to be computed). Since, as discussed in appendix A, AAD requires a considerable implementation effort, especially in industrial applications, our results suggest that, if a financial institution doesn't have AAD implemented, the use of FD coupled with QMC (which is much easier to implement) remains competitive in many realistic applications. Moreover, if a financial institution already has AAD, it should use QMC instead of MC: this allows for huge savings in computational time and achieves high accuracy, in contrast to standard MC.

Finally, the stability analysis results presented in Section 4.5 confirmed that QMC simulations with optimal sampling strategies are generally more stable and monotonic than MC (Figures 25, 26).

The methodology presented in this paper can be used for more complex problems in finance, in particular, fair value adjustments (XVAs) and market and counterparty risk measures[15], based on multi-dimensional, multi-step Monte Carlo simulations of large portfolios of trades. Such simulations can require, in typical real cases, $\sim 10^2$ time simulation steps, $\sim 10^3$ (possibly correlated) risk factors, $\sim 10^3 - 10^4$ MC scenarios, $\sim 10^4 - 10^5$ trades, 60 years maturity, leading to a nominal dimensionality of the order $D \sim 10^5$, and to a total of $10^9 - 10^{11}$ evaluations. Moreover, a fraction $\sim 1\%$ of exotic trades may require distinct MC simulations for their evaluation, nesting another set of $\sim 10^3 - 10^5$ MC scenarios, thus leading up to 10^{14} evaluations. Finally, hedging CVA/DVA/FVA/KVA/MVA valuation adjustments w.r.t. to their underlying risk factors (typically credit/funding curves) also requires the computation of their corresponding greeks w.r.t. each term structure node, adding another $\sim 10^2$ simulations. This is the reason

[15]Some of these metrics, such as EPE/ENE or expected shortfall, are defined as means or conditional means, while some other metrics, such as VaR or PFE, are defined as quantiles of appropriate distributions.

why the industry is continuously looking for advanced techniques to reduce computational times: grid computing, GPU computing, AAD, etc. (see e.g. [She15]).

We argue that, using QMC sampling (instead of MC) to generate the scenarios of the underlying risk factors and to price exotic trades may significantly improve the accuracy, the performance and the stability of such monster-simulations, as shown by preliminary results on real portfolios in [BKS14]. Furthermore, GSA should suggest how to order the risk factors according to their relative importance, thus reducing the effective dimensionality. Such applications will need further research.

Appendices

A FD vs AAD greeks

Complex financial instruments can be priced only resorting to Monte Carlo simulation. The main drawback of this approach is that it is generally computationally expensive to reach an acceptable degree of accuracy. This problem becomes even more striking when the computation of Greeks is concerned: indeed, due to its simplicity, the most widely used technique is to form finite difference (FD) approximations and then re-price the instrument on bumped scenarios (i.e. with the relevant parameter shifted by a predefined finite amount). The FD estimator of a generic greek (say, the sensitivity w.r.t. to parameter θ_i) is defined as[16]

$$\frac{\partial V_0}{\partial \theta_i} \doteq \frac{Y(\theta_i + h) - Y(\theta_i)}{h} \tag{A.1}$$

where Y is the discounted payoff and h the increment on θ_i, which is chosen to be $h = \epsilon S_i(0)$, for deltas and gammas, and $h = \epsilon$, for vegas, for a given[17] "shift parameter" ϵ. Clearly, the value of the greek is obtained by averaging (A.1) on many Monte Carlo paths: all that is required for the computation of the price is thus sufficient for the computation of FD greeks as well, no additional implementation effort is required. However, this approach has two disadvantages. The first one concerns with the accuracy of the computation: finite differences are subject to truncation errors, which can be mitigated by the use of central differences so that the bias of the greek estimator (A.1) is of second order in the increment. Bias can be decreased by choosing a small increment, however this would also increase the variance of the estimator, even if path recycling is adopted, so that a fine tuning is needed. It is often hard to find the optimal increment and, in concrete applications, the same increment usually has to be applied to many different situations[18].The

[16]For simplicity, here the one-sided forward difference is shown. As it will be discussed, generally central (two-sided) differences are preferable [Gla03].

[17]Of course, the shift parameters need not to be the same for delta and vega. They can also be different for each component greek.

[18]The choice of the appropriate increment is guided by the following considerations. The MC/QMC root mean square error estimate of finite differences is given by [Gla03]

$$\varepsilon = \sqrt{\frac{c}{N^{2\alpha} h^{\beta}} + b^2 h^4} \,. \tag{A.2}$$

The first term in the square root is a "statistical" error related to the variance c. It depends on N as well as on ϵ. $\alpha = 0.5$ for MC and, usually, $0.5 < \alpha < 1$ for QMC, while $\beta = 1$ for first derivatives and $\beta = 3$ for second derivatives. The second term is the systematic error due to the bias of finite differences: it is independent of N but it depends on ϵ. The constant b is given by $b = \frac{1}{6}\frac{\partial^3 V}{\partial \theta^3}(\theta)$ for central differences of the first order (delta and vega) and $b = \frac{1}{12}\frac{\partial^4 V}{\partial \theta^4}(\theta)$ for central differences of the second order (gamma). The optimal value of h doesn't vary too much for reasonable ranges of N and it can be chosen in such a way that the bias term remains negligible so that (A.2) follows approximately a power law.

second disadvantage of FD techniques is concerned with computational efforts: indeed, the instrument has to be re-priced as many times as the number of derivatives to compute (actually twice as many, if central differences are used). Therefore, the computational cost of evaluating the price and all greeks, i.e. the gradient of the price function w.r.t. all relevant parameters, increases linearly with the number of required sensitivities. This becomes particularly expensive e.g. in the case of options on multiple underlyings, where at least deltas and vegas w.r.t. each underlying are usually needed.

Both disadvantages of FD techniques can be eliminated by applying Adjoint Algorithmic Differentiation (AAD), which we briefly review in the following. This method was introduced in finance in [GG06] and further developed in [LLS09, CG10, CG12, Cap11, CLP11]. Instead of the FD estimator, let's introduce the "Pathwise Derivative" estimator of the greeks [BG96, Gla03],

$$\frac{\partial V_0}{\partial \theta_i} \doteq \frac{\partial Y}{\partial \theta_i}\,, \tag{A.3}$$

which is simply the derivative of the discounted payoff. If the pathwise derivative (A.3) exists with probability 1 and if the payoff function is regular enough (e.g. it is Lipschitz continuous, see [Gla03] for other sufficient conditions), then (A.3) provides an unbiased estimator of the greeks since expectation and differentiation can be safely interchanged:

$$\frac{\partial V_0}{\partial \theta_i} = \frac{\partial \mathbb{E}[Y]}{\partial \theta_i} = \mathbb{E}\left[\frac{\partial Y}{\partial \theta_i}\right]. \tag{A.4}$$

In other words, we just need to differentiate the discounted payoff path by path and the value of the greek is then recovered by a Monte Carlo average as usual. Notice that the pathwise derivatives have to be computed explicitly in order to compute greeks according to this approach, so that extra implementation effort is required. This can be tedious for complex payoffs. Moreover, in the limit $h \to 0$, both the FD and Pathwise Derivative estimators provide the same estimates with the same Monte Carlo variance, so that the implementation effort required by the latter seems to be hardly justifiable at first sight. However, the major benefit comes from the fact that, if one has to compute the gradient of a single output w.r.t. many variables (as in the case of greeks of multi-asset options), the adjoint mode of algorithmic differentiation dramatically increases the efficiency of pathwise differentiation.

We now describe the basics ideas behind this methodology. Algorithmic differentiation (AD) is a set of programming techniques aimed at calculating *exact* (i.e. free of truncation errors) derivatives of functions given as computer codes [GW08, Nau12]. Let $f : \mathbb{R}^n \to \mathbb{R}\,, f(X) = Y$ be a scalar function of n variables $X = (x_1, \ldots, x_n)$, such as the discounted payoff regarded as a function of parameters $\boldsymbol{\theta}$. No matter how complicated f might be, it is always given as the composition of elementary functions and/or basic arithmetic operations. AD exploits the information on the structure of the code and on the dependencies between its various parts, in order to optimize the calculation of the derivatives. The main tool which makes AD work is the chain rule, which is repeatedly used on the arcs connecting the nodes of the computational graph representing the instructions needed to execute f. Here we discuss only the adjoint mode of algorithmic differentiation (AAD), which is the most efficient when the derivatives of few outputs w.r.t. many inputs are needed[19]. First of all, a *forward sweep* is performed where, starting from the values of the inputs, the value of the output is computed recording all necessary information in intermediate steps. After that, a *backward sweep* is performed, where the derivatives w.r.t all the intermediate variables, i.e. the adjoints, are propagated from the output to the inputs until

[19]The discussion of the tangent mode, useful when the derivatives of many outputs w.r.t. few inputs have to be computed, is left to the references [GW08, Nau12, Cap11].

the whole gradient is obtained in a single run.

$$\begin{tikzcd}
X \arrow[r] & \cdots \arrow[r] & U_i \arrow[r] & V_j \arrow[r] & \cdots \arrow[r] & Y \arrow[d] \\
\bar{X} & \cdots \arrow[l] & \bar{U}_i \arrow[l] & \bar{V}_j \arrow[l] & \cdots \arrow[l] & \bar{Y} \arrow[l]
\end{tikzcd}$$

In more detail: any intermediate instruction is an intrinsic (unary or binary) operation of the form

$$V_j = V_j(\{U_i\}_{i \prec j}) \tag{A.5}$$

whose derivative is known from calculus. Here, the notation means that U_i are the variables on which V_j explicitly depends. The adjoints are then defined as

$$\bar{U}_i = \sum_{j \succ i} \frac{\partial V_j}{\partial U_i} \bar{V}_j \, . \tag{A.6}$$

Initializing $\bar{Y} = 1$ and propagating the adjoints backward through each intermediate step, at the end of the computation the adjoints of the inputs (i.e. the gradient) are obtained.

It is easy to appreciate that the cost for the propagation of the chain rule back to the inputs is of the same order as the cost of evaluating the function f itself. Indeed, there's a precise result stating that, in the adjoint mode, AD provides the full gradient of f at a cost which is up to 4 times the computational cost of evaluating the function f itself, independently of the number of variables. This explains the power of AAD, enabling an extremely fast computation of an arbitrary number of greeks[20]. In Fig. 27 the relative cost of evaluating the price and all deltas and vegas of an Asian basket option w.r.t the cost of evaluating just the price is shown for an increasing number of underlyings and for both FD and AAD techniques. It is easily observed that, while for FD the relative cost increases linearly with the number of underlyings (or greeks to be computed), for AAD it remains constant. However, AAD also has some disadvantages w.r.t FD: first of all, it is harder to implement than finite differences and the implementation effort within large architectures would be challenging without the availability of automatic tools. Secondly, it is not always applicable: in particular it cannot handle discontinuous payoffs. One must regularize the payoff by explicitly smoothing the discontinuity, e.g. approximating a digital call with a call spread or something smoother, or using conditional expectations, e.g. smoothing payoffs with barriers [Gla03]. However this introduces a bias and the use of automatic tools is not straightforward, so that extra effort is needed. Finally, second order Greeks do not have the benefits of the adjoints for multi-asset options. One usually is forced to use a mixed approach AAD+FD. Recently, a mixed approach AAD + Likelihood Ratio Method has been proposed [Cap15]. Furthermore, we want to stress that the results such as those shown in Fig. 27, typically presented in many works on adjoint methods in finance, solely refer to the speed of the computation and by no means are indicative of the accuracy of the computation. The latter is rather given by the details of the simulation method, such as the number of scenarios, the random number generator, the use of variance reduction techniques, and so forth. Of course, an optimal use of the simulation technique will increase accuracy. In other words, the same accuracy can be reached with less scenarios if the simulation details are accurately chosen (e.g. QMC with BBD): this is, therefore, another source of speed-up of the whole computation, besides the mere

[20]AAD, being mechanical in nature, can be automated. Indeed, several AD tools have been developed which automatically implement the adjoint counterpart of a given computer code. These tools typically make use of source code transformations or operator overloading. The latter are well suited to the object oriented programming paradigm. The drawback is that the computation of derivatives is slower, since a preliminary step is needed where all the information of the original code is recorded in a kind of tape, which is a representation of the computational graph necessary to run the forward and backward sweeps. A lot of memory is also necessary since all intermediate variables cannot be overwritten. We refer to the literature for further details on AD tools [GW08, Nau12].

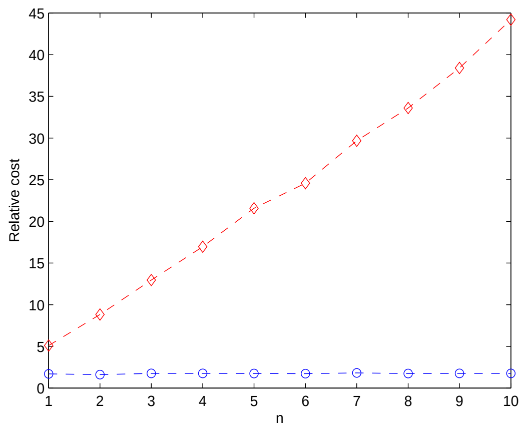


Figure 27: CPU time required to compute price and all greeks (deltas and vegas) divided by CPU time required to compute only price of an Asian option on a basket of n underlyings, for increasing n. Monte Carlo simulation with 200,000 scenarios is used. The red curve refers to central finite differences computation, while the blue curve refers to AAD computation.

use of adjoints instead of finite differences in a MC simulation. These two sources of speed-up are compared in the present work.

References

[BG96] Mark Broadie and Paul Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42:269–285, 1996.

[BKS14] Marco Bianchetti, Sergei Kucherenko, and Stefano Scoleri. Better Pricing and Risk Management with High Dimensional Quasi Monte Carlo. WBS 10th Fixed Income Conference, September 2014.

[BKS15] Marco Bianchetti, Sergei Kucherenko, and Stefano Scoleri. Pricing and Risk Management with High Dimensional Quasi Monte Carlo and Global Sensitivity Analysis. *Wilmott Magazine*, July 2015.

[BM06] Damiano Brigo and Fabio Mercurio. *Interest-Rate Models - Theory and Practice*. Springer, 2nd edition, 2006.

[BRO] BRODA Ltd., High-dimensional Sobol' sequence generators.

[Cap11] Luca Capriotti. Fast greeks by algorithmic differentiation. *The Journal of Computational Finance*, pages 3–35, Spring 2011.

[Cap15] Luca Capriotti. Likelihood ratio method and algorithmic differentiation: Fast second order greeks. *Algorithmic Finance*, pages 81–87, May 2015.

[CG10] Luca Capriotti and Mike Giles. Fast correlation greeks by adjoint algorithmic differentiation. *Risk Magazine*, (29):77–83, 2010.

[CG12] Luca Capriotti and Mike Giles. Algorithmic differentiation: Adjoint greeks made easy. *Risk Magazine*, September 2012.

[CLP11] Luca Capriotti, Jacky Lee, and Matthew Peacock. Real-time counterparty credit risk management in monte carlo. *Risk Magazine*, pages 86–90, June 2011.

[CMO97] R. E. Caflish, W. Morokoff, and A. Owen. Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. *The Journal of Computational Finance*, 1(1):27–46, 1997.

[Duf01] Darrel Duffie. *Dynamic Asset Pricing Theory.* Princeton University Press, 3rd edition, 2001.

[GG06] Mike Giles and Paul Glasserman. Smoking adjoints: Fast monte carlo greeks. *Risk*, (19):88–92, 2006.

[Gla03] Paul Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2003.

[GW08] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiatoin.* Society for Industrial and Applied Mathematics, Philadelphia, PA., 2008.

[Jac01] Peter Jackel. *Monte Carlo Methods in Finance.* Wiley, 2001.

[KFSM11] Sergei Kucherenko, Balazs Feil, Nilay Shah, and Wolfgang Mauntz. The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering and System Safety*, 96:440–449, 2011.

[KMRZ98a] Alexander Kreinin, Leonid Merkoulovitch, Dan Rosen, and Michael Zerbs. Measuring Portfolio Risk Using Quasi Monte Carlo Methods. *Algo Research Quarterly*, 1(1), September 1998.

[KMRZ98b] Alexander Kreinin, Leonid Merkoulovitch, Dan Rosen, and Michael Zerbs. Principal Component Analysis in Quasi Monte Carlo Simulation. *Algo Research Quarterly*, 1(2), December 1998.

[KS07] Sergei Kucherenko and Nilay Shah. The Importance of being Global. Application of Global Sensitivity Analysis in Monte Carlo option Pricing. *Wilmott Magazine*, 4, 2007.

[KTA12] S. Kucherenko, S. Tarantola, and P. Annoni. Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, 183:937–946, 2012.

[LLS09] M. Leclerc, Q. Liang, and I. Schneider. Fast monte carlo bermudan greeks. *Risk magazine*, 22:84–88, 2009.

[LO00] C. Lemieux and A. Owen. Quasi-regression and the relative importance of the anova component of a function. *In: Fang K-T, Hickernell FJ, Niederreiter H, editors. Monte Carlo and quasi-Monte Carlo.*, 2000.

[LO06] R. Liu and A.B. Owen. Estimating mean dimensionality of analysis of variance decompositions. *Journal of the American Statistical Association*, 101(474):712–721, 2006.

[MF99] Maurizio Mondello and Maurizio Ferconi. Quasi Monte Carlo Methods in Financial Risk Management. Tech Hackers, Inc., 1999.

[Nau12] Uwe Naumann. *The Art of Differentiating Coputer Programs. An Introduction to Algorithmic Differentiation.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012.

[Nie88] H. Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30:51?70, 1988.

[Owe93] A.B. Owen. Variance and discrepancy with alternative scramblings. *ACM Transactions on Modeling and Computer Simulation*, 13:363–378, 1993.

[Owe03] A. Owen. The dimension distribution and quadrature test functions. *Stat Sinica*, 13:1–17, 2003.

[Pap01] A. Papageorgiou. The Brownian Brisge does not offer a Consistent Advantage in Quasi-Monte Carlo Integration. *Journal of complexity*, 2001.

[PP99] A. Papageorgiou and S. Paskov. Deterministic Simulation for Risk Management. *Journal of Portfolio Management*, pages 122–127, May 1999.

[PT95] S. H. Paskov and J. F. Traub. Faster Valuation of Financial Derivatives. *The Journal of Portfolio Management*, pages 113–120, Fall 1995.

[PT96] A. Papageorgiou and J. F. Traub. New Results on Deterministic Pricing of Financial Derivatives. *presented at "Mathematical Problems in Finance", Institute for Advanced Study, Princeton, New Jersey*, April 1996.

[SAA$^+$10] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communication*, 181:259–270, 2010.

[SAKK12] Ilya M. Sobol', Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. Construction and Comparison of High-Dimensional Sobol' Generators. *Wilmott Magazine*, Nov:64–79, 2012.

[Sal02] A. Saltelli. Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.*, 145:280–297, 2002.

[She15] Nazneen Sherif. AAD vs GPUs: banks turn to maths trick as chips lose appeal. *Risk*, January 2015.

[SK05a] Ilya M. Sobol' and Sergei Kucherenko. Global Sensitivity Indices for Nonlinear Mathematical Models. Review. *Wilmott Magazine*, 1:56–61, 2005.

[SK05b] Ilya M. Sobol' and Sergei Kucherenko. On the Global Sensitivity Analysis of Quasi Monte Carlo Algorithms. *Monte Carlo Methods and Applications*, 11(1):1–9, 2005.

[Sob67] Ilya M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Comp Math Math Phys*, 7:86–112, 1967.

[Sob01] Ilya M. Sobol'. Global Sensitivity Indices for Nonlinear Mathematical Models and their Monte Carlo Estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.

[SS14] Ilya M. Sobol' and Boris V. Shukhman. Quasi-Monte Carlo: A high-dimensional experiment. *Monte Carlo Methods and Applications*, May:167–171, 2014.

[Wan09] Xiaoqun Wang. Dimension Reduction Techniques in Quasi-Monte Carlo Methods for Option Pricing. *INFORMS Journal on Computing*, 21(3):488–504, Summer 2009.

[Wil06] Paul Wilmott. *Paul Wilmott on Quantitative Finance.* John Wiley & Sons, Ltd, 2 edition, 2006.