Better Pricing and Risk Management with High Dimensional Quasi Monte Carlo

Marco Bianchetti, Intesa Sanpaolo, Market Risk Management
Sergei Kucherenko, Imperial College London
Stefano Scoleri, Iason Ltd.

WBS 10th Fixed Income Conference, Barcelona, Sept. 26th, 2014
1. Introduction
2. Monte Carlo and Quasi Monte Carlo
   o History
   o Multi-dimensional multi-step MC simulation
   o Numbers: true random, pseudo random, and low discrepancy
   o Lattice integration vs Pseudo vs quasi Monte Carlo
   o Global Sensitivity Analysis
   o Effective dimensions
   o Error analysis
3. Prices and sensitivities for selected derivatives
   o Selected payoffs
   o Numerical results for test cases
4. Market and Counterparty risk of derivatives’ portfolios
   o MC simulation for risk measures computation
   o Numerical results for real cases
5. Conclusions
6. References
1: Introduction

- Monte Carlo simulation in finance has been traditionally focused on pricing derivatives. Actually nowadays market and counterparty risk measures, based on multi-dimensional multi-step Monte Carlo simulation, are very important tools for managing risk, both on the front office side (sensitivities, CVA) and on the risk management side (estimating risk and capital allocation). Furthermore, they are typically required for internal models and validated by regulators.

- The daily production of prices and risk measures for large portfolios with multiple counterparties is a computationally intensive task, which requires a complex framework and an industrial approach. It is a typical high budget, high effort project in banks.

- We will focus on the Monte Carlo simulation, showing that, despite some common wisdom, Quasi Monte Carlo techniques can be applied, under appropriate conditions, to successfully improve price and risk figures and to reduce the computational effort.
2: Monte Carlo and Quasi Monte Carlo

History

The Monte Carlo method was coined in the 1940s by John von Neumann, Stanislaw Ulam and Nicholas Metropolis, working on nuclear weapons (Manhattan Project) at Los Alamos National Laboratory [JVN51]. Metropolis suggested the name Monte Carlo, referring to the Monte Carlo Casino, where Ulam's uncle often gambled away his money [Met87, Wiki]. Enrico Fermi is suspected to have used some “manual simulation” in the 1930s, working in Rome on nuclear reactions induced by slow neutrons (without publication) [Los66, Met87].
2: Monte Carlo and Quasi Monte Carlo

Other «quasi»

- **Quasi crystal**
  a structure that is ordered but not periodic. A quasi crystalline pattern can continuously fill all available space, but it lacks translational symmetry (http://en.wikipedia.org/wiki/Quasicrystal).

- **Quasi particle**
  phenomena that occur when a microscopically system such as a solid behaves as if it contained different weakly interacting particles in free space. For example, the aggregate motion of electrons in the valence band of a semiconductor is the same as if the semiconductor contained instead positively charged quasi particles called “holes”.

- **Quasi satellite**
  A quasi satellite's orbit around the Sun takes exactly the same time as the planet's, but has a different eccentricity (usually greater).
The risk factors dynamics are described by stochastic differential equations (SDE)

$$dr_i(t) = \mu_i(t, \mathbf{r}) dt + \Sigma(t, \mathbf{r}) \cdot d\mathbf{W}(t), \quad i = 1, \ldots, N_{rf}$$

where $\mu$ is the drift, $\Sigma$ is the $N_{rf} \times N_{rf}$ variance-covariance matrix, and $d\mathbf{W}$ is a $N_{rf}$-dimensional brownian motion. Model parameters can be calibrated to market quotes (risk neutral world measure) or to historical series (real world measure).

We discretize the future time axis by choosing a time simulation grid $t = [t_1, \ldots, t_{N\text{step}}]$.

The Monte Carlo scenario $s_{jk} := s_k(t_j)$ at (discrete) time simulation step $t_j$ is a $N_{MC}$-dimensional (random) draw of standard brownian motions across the time step $[t_{j-1}, t_j]$

$$s_{jk} := \Delta \mathbf{W}_k(t_j) = \sqrt{t_j - t_{j-1}} \xi_k,$$

$$\xi_k \sim \mathcal{N}^{N_{rf}}(0, 1), \quad k = 1, \ldots, N_{MC}.$$

The risk factor scenario $r_{ijk} := r_{ik}(t_j, s_{jk})$ is the value of the risk factor $r_i$ at time step $t_j$ on Monte Carlo scenario $s_{ik}$

$$r_{ijk} := r_i(t_j, \mathbf{r}(t_j), s_{jk}) \approx r_i[t_{j-1}, \mathbf{r}(t_{j-1})] + \mu_i[t_{j-1}, \mathbf{r}(t_{j-1})] \Delta t_j + \sqrt{\Delta t_j \Sigma[t_{j-1}, \mathbf{r}(t_{j-1})]} \cdot \xi_k.$$
The MC computation of financial quantities is based, in general, on a multi-dimensional multi-step simulation, proceeding as follows.

- for each Monte Carlo scenario $s_{jk}$
  - for each time simulation step $t_j$
    - for each risk factor $r_i$
      - simulate the risk factor values $r_{ijk}$
      - for each trade $l$ in the portfolio
        - compute the mark to future value $v_{jkl}$
      - loop over portfolio trades $l = 1, \ldots, N_{ptf}$
    - loop over risk factors $i = 1, \ldots, N_{rf}$
  - loop over time simulation steps $j = 1, \ldots, N_{step}$
- loop over Monte Carlo scenarios $k = 1, \ldots, N_{MC}$
Better Pricing and Risk Management

with High Dimensional Quasi Monte Carlo

2: Monte Carlo and Quasi Monte Carlo

Multi-dimensional multi-step MC simulation [3]

- The mark to future is the future value $v_{jkl}$ of the individual trades survived at time step $t_j$.

$$v_{jkl} := v_l \left[ t_j, r(t_j, s_{jk}) \right].$$

- The portfolio mark to future $V_{jk}$ is the sum of the future values of all trades in the portfolio (we assume linear combination of trades). The aggregation of trades can follow different rules, for example to accommodate different counterparties with different netting sets. The netting set mark to future $V_{jkh}$ is the sum of all the trades in the netting set $n_h$ subject to the same netting agreement with a given counterparty.

$$V_{jkh} = \sum_{l \in n_h} v_{jkl}, \quad V_{jk} = \sum_{h=1}^{N_h} V_{jkh}.$$

- Notice that if the $l$-th trade maturity $T_l$ is smaller than the time simulation step $t_j$, the $l$-th trade is dead and there is nothing to compute,

$$v_l \left[ t_j, r(t_j, s_{jk}) \right] \begin{cases} \neq 0, & \text{if } t_j \leq T_l \text{ (alive)}, \\ = 0, & \text{if } t_j > T_l \text{ (dead)}. \end{cases}$$
2: Monte Carlo and Quasi Monte Carlo

**Multi-dimensional multi-step MC simulation [4]**

- The dimension of the Monte Carlo simulation is (see e.g. [Jac03])

\[
D^{MC} = n^\circ \text{ of risk factors} \times n^\circ \text{ of time simulation steps} = N_{rf} \times N_{step}
\]

- The performance of the Monte Carlo simulation depends on
  - the number of risk factors \(N_{rf}\),
  - the number of time simulation steps \(N_{step}\),
  - the number of MC scenarios \(N_{MC}\),
  - the properties of (random) numbers \(\xi\) generator,
  - the speed of convergence of the MC simulation,
  - the stability of the MC simulation,
  - the number of trades \(N\) and of netting sets \(N_h\) in the portfolio,
  - the computational cost to price each trade with analytical formulas, PDE, or MC simulations,
  - the dependence of trades on risk factors,
  - ...

In particular, notice that the (netting set or portfolio) mark to future $V_{jk}$ at time step $t_j$ depends, in principle, on all the risk factors

$$V_{jkh} = \sum_{l \in n_h} v_l \left[ t_j, r(t_j, s_{jk}) \right].$$

In practice, each trade and netting set will show a lower effective dimension, due to the trade expiration and to hierarchical dependency on risk factors as follows:
- higher sensitivity to primary risk factors
- smaller sensitivity to secondary risk factors
- negligible or null sensitivity to negligible risk factors

$\nabla V_{jkh} := \left\{ \frac{\partial V_{jkh}}{\partial r_1}, \ldots, \frac{\partial V_{jkh}}{\partial r_\alpha}, \frac{\partial V_{jkh}}{\partial r_{\alpha+1}}, \ldots, \frac{\partial V_{jkh}}{\partial r_\beta}, \ldots, \frac{\partial V_{jkh}}{\partial r_{N rf}} \right\}$

Primary risk factors  Secondary risk factors  Negligible risk factors

How to take advantage of these features of the problem? Global Sensitivity Analysis!
Random vs deterministic events
- Random events are intrinsically unpredictable.
- Deterministic events are, in principle, predictable. In practice, it depends on our knowledge.
  Determinism is the hypothesis that actually there is no randomness in the universe, only unpredictability (that is, our ignorance).
- According to the Bayesian interpretation of probability, probability can be used to represent a lack of complete knowledge of events.
- Random events are very common in our universe. Some examples:
  - Radioactive decay of a single unstable nucleus or particle is intrinsically random. Its average lifetime is perfectly deterministic. Think to radiocarbon dating.
  - Atomic motion in a gas is random. Newton’s equations of motion are deterministic.
  - Genetic mutations are random. DNA reproduction is deterministic.

True random numbers generators (TRNGs)
- Random numbers can be produced by appropriate hardware, called True Random Number Generators (TRNGs), based on statistically random physical processes, such as quantum mechanical effects (e.g. radioactive decay), or thermal noise.
- Random numbers cannot be produced by a computer executing deterministic instructions. “Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.” (John von Neumann, 1951 [JVN51]).
Better Pricing and Risk Management with High Dimensional Quasi Monte Carlo

2: Monte Carlo and Quasi Monte Carlo

*Numbers: pseudo random*

- **Pseudo Random Numbers Generators (PRNGs)**
  Pseudo random numbers are generated by algorithms called Pseudo Random Number Generators (PRNGs). PRNGs produce deterministic sequences of numbers that approximates the properties of a random sequences. Such sequences are completely determined by a set of initial values, called the PRNG’s state. Thus, sequences produced by PRNGs are reproducible, using the same set of state variables.

- **Main characteristics of PRNGs:**
  - **Seed**: the number used to initialize the PRNG. It must be a random number.
  - **Periodicity**: the maximum length, over all possible state variables, of the sequence without repetition.
  - **Distribution**: distribution of the random numbers generated, generally uniform \([0,1)\).

- **Most common PRNGs:**
  - **Pioneer PRNGs**: Mid Square Method by John Von Neumann in 1946 (see [Jac02]).
  - **Classical PRNGs**: see e.g. [NR02] and [Jac02].
  - **Mersenne Twister**: the best at the moment, with the longest period of 106000 iterations. See [MT97,JAC02].

- **Main lessons:**
  - **all PRNGs are flawed by definition.**
  - **know your PRNG**: seed, periodicity, limits, etc., never use it as a black box.
Low Discrepancy Numbers or Quasi Random Numbers (QRN), are such that any sequence of these numbers has low discrepancy. Formally, a sequence of d-dimensional numbers in \([0,1]^d\), has low discrepancy if the first \(N\) points \(\{u_1,\ldots,u_N\}\) in the sequence satisfy

\[
D_N^d(u_1,\ldots,u_N) \leq c(d) \frac{(\ln N)^d}{N}, \quad \forall N > 1,
\]

where \(D\) is the discrepancy and \(c(d)\) is some constant depending only on \(d\). The key point is that low discrepancy is required for any subsequence with \(N > 1\), not for some fixed \(N\).

The discrepancy of a sequence \(\{u_1,\ldots,u_N\}\) is a measure of how inhomogeneously the sequence is distributed inside the unit hypercube \(I^d = [0,1]^d\). Formally

\[
D_N^d(u_1,\ldots,u_N) = \sup_{\mathbf{x} \in I^d} \left| \frac{n \left[ S^d(\mathbf{x}), u_1, \ldots, u_N \right]}{N} - V^d(\mathbf{x}) \right|,
\]

where

\[
S^d(\mathbf{x}) = [0, x_1) \times \cdots \times [0, x_d) \subset I^d,
\]

\[
V^d(\mathbf{x}) = \prod_{j=1}^{d} x_j,
\]

\[
n \left[ S^d(\mathbf{x}, u_1, \ldots, u_N) \right] = \sum_{i=1}^{N} 1[u_i \in S^d(\mathbf{x})] = \sum_{i=1}^{N} \prod_{j=1}^{d} 1[u_{j,i} \leq x_j],
\]
2: Monte Carlo and Quasi Monte Carlo

Numbers: low discrepancy [2]

Notice how the discrepancy of the sequence of N QRNs is minimized for each sub-sequence n=1,…,N, with respect to the PRNs.

The first 256 points from a PRNG (top) compared with the first 256 points from the 2,3 Sobol sequence (bottom). The Sobol sequence covers the space more evenly (red=1,…,10, blue=11,…,100, green=101,…,256). Source: [Wikipedia].

The first 1024 points of two-dimensional Sobol sequence. At each stage, the new points regularly fill the gaps in the distribution generated at the previous stage. Source: Numerical Recipes [NR02]
### Low Discrepancy Numbers Generators (LDNGs):
- as for PRNGs, there are many algorithms to produce low discrepancy numbers, the most important being (see e.g. [Jac02], [Gla03]):
  - Van der Corput numbers
  - Halton numbers
  - Faure numbers
  - Sobol\` numbers

#### Sobol\` Numbers
- use a base of 2 to form successively finer uniform partitions of the unit interval, and then reorder the coordinates in each dimension.
- The number of iterations should be taken as $N=2^n$ for $n$ integer.
- Once well-initialised, they provide:
  - the lowest discrepancy in lower dimensions
  - comparable discrepancy with pseudo random numbers in higher dimensions

$L_2$ norm discrepancy (y-axis) as a function of the number of iterations (x-axis) for various PRNGs and LDNGs, in dimension $d=2$ (up) and $d=100$ (down). Source: [Jac02].
2: Monte Carlo and Quasi Monte Carlo

**Numbers: distribution**

- **Uniform NGs**
  Standard NGs produce numbers with a continuous uniform distribution $U[a,b]$, such that the probability density function $f$ is constant over the interval $[a,b]$,

  \[
  f(x) = \frac{1}{b-a} \mathbf{1}_{[a,b]}(x) = \begin{cases} 
  1 & \text{for } a \leq x \leq b, \\
  0 & \text{for } x < a \text{ or } x > b.
  \end{cases}
  \]

- **Non uniform RNGs**
  Numbers with any distribution with probability density $f(x)$ can be produced, in principle, starting from the uniform distribution $U[0,1]$ by inverting the following relation

  \[
  F(x) := \int_{-\infty}^{x} f(u)du \sim U[0,1],
  \]

  because the cumulative probability function $F(x)$ is a probability measure, which is uniform on $[0,1]$ by definition. Given a number $y$ with uniform distribution $U(0,1)$ we obtain

  \[
  F(x) = y \sim U[0,1],
  \]

  \[
  x = F^{-1}(y) \sim f.
  \]

  See e.g. [NR02], [Jac02] and [Gla03] for (a lot of) details.
2: Monte Carlo and Quasi Monte Carlo

*Lattice integration vs pseudo Monte Carlo vs quasi Monte Carlo* [1]

- **Lattice integration**
  The numerical integration of a given d-dimensional function on a regular grid with $N_{\text{lattice}}$ points in the hypercube domain of volume $L^d$ has a relative error

$$
\epsilon(N_{\text{lattice}}, d) = \frac{c}{(N_{\text{lattice}})^d}
$$

- **Pseudo Monte Carlo**
  The PMC standard error associated to PMC, by the central limit theorem, is

$$
\epsilon(N_{\text{MC}}) = \frac{\sigma_{N_{\text{MC}}^2}}{N_{\text{MC}}^{0.5}}
$$

where $\sigma_{N_{\text{MC}}^2}$ is the estimated variance of the simulation. Variance reduction techniques, e.g. antithetic variables, affect only the numerator [Gla03].
Quasi Monte Carlo

The discrepancy of QMC simulation is

\[ C(d, N_{MC}) = c(d) \frac{(\ln N_{MC})^d}{N_{MC}} \]

Notice that the discrepancy depends on the dimension. Since there is no statistics behind low discrepancy sequences, there is no variance estimation. A variance estimation can be achieved by multiple simulations with scrambled low discrepancy sequences (randomised QMC, see [Gla03]).

Following [Jac02] “[...] is a misunderstanding in the literature that they begin to fail as and when you start using dimensionalities above a few dozens”.

Recently, efficient high dimensional low discrepancy generators have been made available. In particular [Broda] proposes Sobol’ generators with dimension up to \( 2^{15} = 32,768 \).

The recent financial literature on the subject is focused on pricing analysis (see e.g. [Kuc07,Kuc12]), while risk management applications are, to our knowledge, much less recent and focused on market risk only (see e.g. [Pap98], [Kre98a], [Kre98b], [Mon99]).
## 2: Monte Carlo and Quasi Monte Carlo

*Global sensitivity analysis [1]*

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Model function</th>
<th>Output variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ (x_1, \ldots, x_D) \in V_D $</td>
<td>$ f(x_1, \ldots, x_D) $</td>
<td>$ y = f(x_1, \ldots, x_D) $</td>
</tr>
</tbody>
</table>

Without loss of generality we can take $ V_D $ to be the $ D $-dimensional hypercube, so that

$$ x_i \sim i.i.d. \ U[0, 1] \quad \forall i = 1, \ldots, D $$

- **Option pricing:**
  - $ D = N_{\text{step}} $ (single asset)
  - $ D = N_{\text{rf}} \times N_{\text{step}} $ (multiple assets)
  - $ y = \text{Price, Greeks} $

- **Risk measures:**
  - $ D = N_{\text{rf}} \times N_{\text{step}} $ (can be very high!)
  - $ y = \text{portfolio P&L (VaR, ES), portfolio value (EPE/ENE, PFE)} $  

### Advantages of GSA w.r.t. other SA approaches:

- It quantifies the effect of varying a given input (or set of inputs) while all other inputs are varied as well.
- It provides a measure of interaction among variables.
- It can be applied also to non-linear models.
2: Monte Carlo and Quasi Monte Carlo

Global sensitivity analysis [2]

- ANOVA decomposition

\[ f(x) = f_0 + \sum_{i} f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \ldots + f_{1,2\ldots D}(x_1, \ldots, x_D) \]

First order terms \hspace{0.5cm} Second order terms \hspace{0.5cm} D-order term

The decomposition is unique if

\[ \int_{0}^{1} f_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s}) dx_{i_k} = 0, \quad \forall k = 1, \ldots, s \]

In this case terms are orthogonal and are given by integrals involving only \( f(x) \).

For square integrable functions, the total variance of \( f \) decomposes as

\[ \sigma^2 = \sum_{i} \sigma_i^2 + \sum_{i<j} \sigma_{ij}^2 + \ldots + \sigma_{12\ldots D}^2, \]

\[ \sigma_{i_1,\ldots,i_s}^2 = \int_{0}^{1} f_{i_1,\ldots,i_s}^2(x_{i_1}, \ldots, x_{i_s}) dx_{i_1} \cdots dx_{i_s}. \]
Global sensitivity analysis [3]

- **Sobol’ sensitivity indices**

\[ S_{i_1,\ldots,i_s} = \frac{\sigma^2_{i_1,\ldots,i_s}}{\sigma^2} \]

- They measure the fraction of variance accounted by \( f_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s}) \)
- They sum up to 1
- For \( s>1 \) they measure interactions among \( x_1, \ldots, x_{i_s} \)
- One can introduce Sobol’ indices for subsets of variables \( S_y \) in order to measure the importance of a subset \( y \) of \( x \) w.r.t. the complementary subset \( z \). Moreover, one can introduce total effect indices \( S_y^{tot} \) in order to measure the total contribution of a subset \( y \) to the total variance [Sob05b].

They can be used to:
- Rank variables in order of importance
- Fix unessential variables
- Reduce model complexity
- Analyze/predict the efficiency of various numerical schemes
2: Monte Carlo and Quasi Monte Carlo

Global sensitivity analysis [4]

- The most used Sobol’ sensitivity indices in practice are

\[ S_i = \frac{\sigma_i^2}{\sigma^2}, \]

\[ S_{i_{\text{tot}}} = \sum_{i \in \{i_1, \ldots, i_s\}} \frac{\sigma_{i_1, \ldots, i_s}^2}{\sigma^2}, \]

\[ 0 \leq S_i \leq S_{i_{\text{tot}}} \leq 1. \]

- Special cases are:
  - \( S_{i_{\text{tot}}} = 0 \): the output function doesn’t depend on \( x_i \)
  - \( S_i = 1 \): the output function depends only on \( x_i \)
  - \( S_i = S_{i_{\text{tot}}} \): interactions between \( x_i \) and other variables are absent
Sobol’ indices are expressed as high-dimensional integrals, so they are usually evaluated via MC/QMC techniques. Efficient formulas have been developed, which allow to compute Sobol’ indices for subsets of variables directly from $f(x)$, thus avoiding the knowledge of ANOVA components [Sob05b]. Furthermore, the computation of $S_i$ and $S_{i}^{tot}$ can be reduced to $N_{MC}(D + 2)$ function evaluations [Kuc11],

$$S_y = \frac{1}{\sigma^2} \int_0^1 [f(y', z') - f_0][f(y', z) - f(y, z)]dydzdy'dz'$$

$$S_{y}^{tot} = \frac{1}{2\sigma^2} \int_0^1 [f(y, z) - f(y', z)]^2dydzdy'$$

$$\sigma^2 = \int_0^1 f^2(y, z)dydz - f_0^2$$

$$f_0 = \int_0^1 f(y, z)dydz$$

where $y$ and $z$ (or $y'$ and $z'$) are two complementary subsets of $x$ (or $x'$).
2: Monte Carlo and Quasi Monte Carlo

*Effective dimensions [1]*

- Many financial problems are high-dimensional, but often not all the variables are equally important: *effective dimensions* were introduced to quantitatively measure the number of most important variables of a model function [Caf97].
- The superior efficiency of QMC methods for some high-dimensional problems can be ascribed to a reduced effective dimension w.r.t. nominal dimension of the model function $f(x)$.
- GSA can be used to compute effective dimensions and, thus, to predict the performance of QMC method w.r.t. standard MC, for a given $f(x)$.

- Different notions of *effective dimensions* can be introduced:
  - The effective dimension in the superposition sense is the smallest $d_S$ such that
    \[ \sum_{0 < |y| < d_S} S_y \geq 1 - \varepsilon, \quad \text{for some threshold } \varepsilon \]
  - The effective dimension in the truncation sense is the smallest $d_T$ such that
    \[ \sum_{y \subseteq \{1, 2, \ldots, d_T\}} S_y \geq 1 - \varepsilon, \quad \text{for some threshold } \varepsilon \]
  - The average dimension $d_A$ is defined as
    \[ d_A = \sum_{0 < |y| < D} |y| S_y \]
2: Monte Carlo and Quasi Monte Carlo

**Effective dimensions [2]**

- The effective dimension $d_S$ doesn’t depend on the order of sampling of variables, while the effective dimension $d_T$ does.

- The following inequality holds: $d_S \leq d_T$.

- All high-dimensional numerical experiments found in the literature show that QMC is more efficient than MC when the effective dimension (in one or more senses) is small. In particular [Sob14] suggests that QMC could be more efficient than MC when $d_A \lesssim 3$.

- Functions $f(x)$ can be classified according to their dependence on variables:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Relationship between SI</th>
<th>Eff. dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Few important variables</td>
<td>$S_y^{tot}/n_y \gg S_z^{tot}/n_z$</td>
<td>$d_T \ll D$</td>
</tr>
<tr>
<td>B</td>
<td>Low-order interactions</td>
<td>$S_i \approx S_j \forall i, j$, $S_i \approx S_i^{tot} \forall i$</td>
<td>$d_S \ll D$</td>
</tr>
<tr>
<td>C</td>
<td>High-order interactions</td>
<td>$S_i \approx S_j \forall i, j$, $S_i \ll S_i^{tot} \forall i$</td>
<td>$d_S \approx D$</td>
</tr>
</tbody>
</table>

For Type A and Type B functions, QMC is more efficient than MC, for Type C functions QMC cannot achieve higher accuracy than MC. Type A and Type B functions are very common in financial problems, possibly after effective dimension reduction.
2: Monte Carlo and Quasi Monte Carlo

Effective dimensions [3]

- Contrary to MC, QMC efficiency is sensitive to the order in which variables \( x \) are sampled from the LDS. The common explanation is that, since lower coordinates in each \( D \)-dimensional LDS draw are particularly well distributed, one should use them to simulate the most important variables and concentrate there most of the variance; moreover, since lower dimensional projections of LDS (especially for well initialized Sobol’ sequences) are much better distributed, in the case of Type B functions, higher-order interaction terms should be relegated to later dimension, whose projections are not so well distributed.

- The optimal sampling order is not \textit{a priori} known for a given \( f(x) \). In path-dependent option pricing, a popular choice is to apply Brownian Bridge to the discretization of the underlying stochastic process [Caf97]. Some authors report cases where Brownian Bridge is not superior to standard discretization [Pap01].

\[
W(0) = 0, \quad W(t_D) = t_D \Phi^{-1}(x_1), \quad \text{Brownian Bridge discretization of a Brownian motion}
\]

\[
W(t_j) = \frac{t_k - t_j}{t_k - t_i} W(t_i) + \frac{t_j - t_i}{t_k - t_i} W(t_k) + \sqrt{\frac{(t_k - t_j)(t_j - t_i)}{t_k - t_i}} \Phi^{-1}(x_j), \quad t_i < t_j < t_k, \quad j = 2, \ldots, D
\]

- GSA explains why a given discretization scheme improves QMC convergence and tells if it will be superior to MC, at a reasonable computational cost, without actually computing the convergence rate via direct simulations (which would be very time-consuming).
2: Monte Carlo and Quasi Monte Carlo

Error Analysis [1]

Many problems in financial engineering can be reduced to high-dimensional integration of a given function $f$, hence MC/QMC simulation is usually the election method to solve them.

- In order to measure the efficiency of the two techniques, integration error $\varepsilon$ as a function of the number of simulated paths $N_{MC}$ is analysed.

- If $V$ is the exact value of the integral and $V_N$ is the simulated value using $N$ paths, the root mean square error, averaged on $L$ independent runs, is defined as

$$
\varepsilon = \sqrt{\frac{1}{L} \sum_{i=1}^{L} (V - V_N^{(i)})}
$$

In the QMC case, for each run, a different part of the LDS is used.

- In order to compare the performances of the two techniques we conduct the three following analyses:
  A. Convergence analysis
  B. Monotonicity and stability analysis
  C. Speed-up analysis
A. Convergence analysis

- **MC convergence rate** is known to be $\frac{\sigma(f)}{\sqrt{N_{MC}}}$ irrespective of the dimensionality.

- An upper bound to **QMC convergence rate** is $\log^D(N_{MC})/N_{MC}$. It is asymptotically faster than MC but it depends on dimensionality and, for feasible $N$, it can be too slow. However, it is not observed in practice: integrands with low effective dimension show faster convergence:

$$\varepsilon = c \, N^{-\alpha}$$

with $\alpha$ approaching 1.

- The root mean square error can be computed for different numbers of simulated paths and constants $c$ and $\alpha$ can be estimated from linear regression.
B. Monotonicity and stability analysis

- Numerical tests present in the literature show that QMC convergence is often smoother than MC: such monotonicity and stability guarantee higher accuracy for a given $N_{MC}$.

- In order to quantify monotonicity and stability we adopt the following strategy:
  1. divide the range of path simulations in $n$ windows,
  2. compute sample mean and sample variance for each window,
  3. we propose:
     - “log-returns” as a measure of monotonicity, and
     - “volatility” as a measure of stability.

- In this way:
  - Monotonic convergence will show non oscillating log-return converging to zero
  - Stable convergence will show low and almost flat volatility
C. Speed-up analysis

- Such measure is sometimes considered in the literature [Kre98a]. It allows to quantify the computational gain of a method $i$ with respect to a method $j$.

- It is defined as

$$S_{i,j}^*(a) = \frac{N_{i}^{(j)}(a)}{N_{i}^{(i)}(a)}$$

where $N_{i}^{*}(a)$ is the threshold number of paths needed to reach and maintain a given accuracy $a$. Speed-Up can be either computed by direct simulation (but it is extremely computationally expensive) or extrapolated by the estimated law of the convergence rate (but it wouldn’t capture unexpected fluctuations due to possible instability).

- We choose to identify the threshold $N_{i}^{*}$ as the first $N$ such that

$$V - a < V_{N} \pm 3\sigma < V + a, \quad \forall N > N_{i}^{*}$$
The aim of this analysis is to compare the efficiency of QMC w.r.t. PMC in computing prices and greeks (Delta, Gamma, Vega) for selected payoffs $P$ with increasing degrees of complexity, using simple dynamical models. In particular we rely on Black-Scholes model, where the underlying process $S$ follows a geometric Brownian motion.

- **European Call:**

  $$ P = \max(S_D - K, 0), $$

- **Asian Call:**

  $$ P = \max(\bar{S} - K, 0), \quad \bar{S} = \left( \prod_{i=1}^{D} S_i \right)^{1/D}, $$

- **Double Knock-Out:**

  $$ P = \max(S_D - K, 0) 1_{\{B_l < S_i < B_u\}}, \quad \forall i = 1, \ldots, D, $$

- **Cliquet:**

  $$ P = \max \left\{ \sum_{i=1}^{D} \max \left[ 0, \min(C, \frac{S_i - S_{i-1}}{S_i-1}) \right], F \right\}. $$

where $K$ is the strike price, $B_l$ and $B_u$ the values of the lower and upper barrier, respectively, $C$ a local cap and $F$ a global floor. The underlying process is discretized across $D$ simulation dates (remember that in the single asset case the number of time simulations steps is equal to the dimension of the MC simulation.)
3: Prices and sensitivities for selected derivatives

Simulation details

- Scenarios for the underlying process $S$ are simulated using:
  - MC + antithetic variables + Mersenne Twister generator [MT98]
  - QMC + standard discretisation + SobolSeq generator [Broda]
  - QMC + brownian bridge discretisation + SobolSeq generator [Broda]
  - $N_{MC}$ scenarios as a power of two, to maximise the efficiency of Sobol’ Low Discrepancy Sequence (LDS)

- Prices are computed as averages of the payoff value over MC scenarios.

- Greeks are computed via finite differencing using
  - Common PRN/LDS for both base and bumped scenarios
  - Central differences

- Root mean square error is computed as an average over $L$ independent runs (using non-overlapping LDS) w.r.t. a reference value: the latter is given by analytical formulas for European and Asian options, or by a simulated value using a sufficiently large ($2^{23}$) number of MC scenarios in other cases.

- Codes are implemented in MATLAB.
3: Prices and sensitivities for selected derivatives

*European Call: set up*

The first test case regards an ATM European call option.

**Test #1 specs**
- Risk-free rate: \( r = 0.03 \)
- Spot price: \( S_0 = 100 \)
- Volatility: \( \sigma = 0.3 \)
- Maturity: \( T = 1 \)
- Strike price: \( K = 100 \)
- Number of time steps: \( N_{step} = 32 \)
- Output values: Price, Delta, Gamma, Vega
- Number of MC/QMC trials for the computation of SI: \( N = 10^5 \)
- Number of simulated paths: \( N_{MC} = 2^p, p = 9, \ldots, 18 \)
- Number of independent runs: \( L = 30 \)
- Increment on finite differences:
  - \( \delta S_0 = \epsilon S_0, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
  - \( \delta \sigma = \epsilon, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
3: Prices and sensitivities for selected derivatives

*European Call: Global Sensitivity Analysis [1]*

**GSA for Standard DisCRETIZATION**

Price, Delta and Vega are Type-B functions: in these cases we expect that QMC will outperform MC.

Gamma is a Type-C function: in this case we expect that QMC won’t outperform MC.

<table>
<thead>
<tr>
<th></th>
<th>$S_i / S_i^{tot}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>0.49</td>
<td>0.68</td>
<td>32</td>
<td>&lt; 32</td>
<td>1.40</td>
<td>-</td>
</tr>
<tr>
<td>Delta</td>
<td>0.26 → 0.23</td>
<td>0.77</td>
<td>32</td>
<td>&lt; 32</td>
<td>3.2</td>
<td>small</td>
</tr>
<tr>
<td>Gamma</td>
<td>$10^{-4} \rightarrow 10^{-2}$</td>
<td>$10^{-4} \rightarrow 10^{-2}$</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>yes</td>
</tr>
<tr>
<td>Vega</td>
<td>0.33</td>
<td>0.543</td>
<td>32</td>
<td>&lt; 32</td>
<td>1.64</td>
<td>no</td>
</tr>
</tbody>
</table>
3: Prices and sensitivities for selected derivatives

European Call: Global Sensitivity Analysis [2]

Price, Delta, Gamma and Vega are Type-A functions: we expect that QMC will always outperform MC.

Effective dimensions equal 1: this is a trivial result, since the payoff explicitly depends only on one variable (spot at maturity)

<table>
<thead>
<tr>
<th></th>
<th>$S_i/S_i^{tot}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Delta</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>no</td>
</tr>
<tr>
<td>Gamma</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>no</td>
</tr>
<tr>
<td>Vega</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>no</td>
</tr>
</tbody>
</table>
3: Prices and sensitivities for selected derivatives

**European Call: Error Analysis**

From log-log plots of error vs number of simulated paths, one can see that QMC always outperforms MC: in particular, for BB discretization, the root mean square error has lower intercept and slope. It scales with the number of paths approximately as $N_{MC}^{-1}$.

**Speed-up analysis**

QMC + BB yields speed-ups up to hundreds, especially if higher accuracy is needed.

<table>
<thead>
<tr>
<th>$S_*(a)$</th>
<th>QMC Std / MC</th>
<th>QMC BB / MC</th>
<th>QMC BB / QMC Std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a = 1%$</td>
<td>$a = 0.1%$</td>
<td>$a = 1%$</td>
</tr>
<tr>
<td>Price</td>
<td>3</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>Delta</td>
<td>0.3</td>
<td>0.5</td>
<td>20</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.5</td>
<td>0.5</td>
<td>200</td>
</tr>
<tr>
<td>Vega</td>
<td>5</td>
<td>6</td>
<td>50</td>
</tr>
</tbody>
</table>
3: Prices and sensitivities for selected derivatives

**European Call: Convergence Analysis**

For Price, Delta and Vega, QMC + BB converges faster than MC and is much more stable.
3: Prices and sensitivities for selected derivatives

Asian Call: set up

The second test case regards an ATM geometric average Asian call option.

Test #2 specs

- Risk-free rate: \( r = 0.03 \)
- Spot price: \( S_0 = 100 \)
- Volatility: \( \sigma = 0.3 \)
- Maturity: \( T = 1 \)
- Strike price: \( K = 100 \)
- Number of time steps: \( N_{\text{step}} = 32 \)
- Output values: Price, Delta, Gamma, Vega
- Number of MC/QMC trials for the computation of SI: \( N = 10^5 \)
- Number of simulated paths: \( N_{\text{MC}} = 2^p, p = 9, \ldots, 18 \)
- Number of independent runs: \( L = 30 \)
- Increment on finite differences:
  - \( \delta S_0 = \epsilon S_0, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
  - \( \delta \sigma = \epsilon, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
3: Prices and sensitivities for selected derivatives

Asian Call: Global Sensitivity Analysis [1]

GSA for Standard Discretization

Price and Vega are Type-A functions. In the case of Delta, some higher-order interactions among variables are present, since $S_i \ll S_i^{tot}$ for some $i$. In these cases we expect that QMC will outperform MC (perhaps not too much for Delta).

Gamma is a Type-C function: in this case we expect that QMC won’t outperform MC.
3: Prices and sensitivities for selected derivatives

Asian Call: Global Sensitivity Analysis [2]

Price, Delta and Vega are Type-A functions. Variables are not equally important, so that a significant reduction in effective dimension in the truncation sense can be achieved. In these cases we expect that QMC will outperform MC.

Gamma is still a Type-C function: in this case we expect that QMC won’t outperform MC.

<table>
<thead>
<tr>
<th></th>
<th>$S_i / S_{i^{tot}}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>0.853 → 0.4</td>
<td>0.875</td>
<td>2</td>
<td>$\leq 2$</td>
<td>1.13</td>
<td>-</td>
</tr>
<tr>
<td>Delta</td>
<td>0.733 → 0.01</td>
<td>0.778</td>
<td>4</td>
<td>$\leq 4$</td>
<td>1.68 → 1.43</td>
<td>small</td>
</tr>
<tr>
<td>Gamma</td>
<td>$10^{-2}$ → $10^{-4}$</td>
<td>0.022 → $10^{-4}$</td>
<td>32</td>
<td>32</td>
<td>31 → 8</td>
<td>yes</td>
</tr>
<tr>
<td>Vega</td>
<td>0.802 → 0.03</td>
<td>0.827</td>
<td>2</td>
<td>$\leq 2$</td>
<td>1.20</td>
<td>no</td>
</tr>
</tbody>
</table>
Speed-up analysis

QMC + BB yields speed-ups from 10 to 100 for Price and Vega, much less for Delta

From log-log plots of error vs number of simulated paths, one can see that QMC outperforms MC, even though this is less evident for Delta. For BB discretization, the root mean square error scaling law approaches $N_{MC}^{-1}$ for Price and Vega, while it is $N_{MC}^{-0.6}$ for Delta and $N_{MC}^{-0.5}$ for Gamma.
3: Prices and sensitivities for selected derivatives

**Asian Call: Convergence Analysis**

For Price, (Delta) and Vega, QMC + BB converges faster than MC and is more stable.
3: Prices and sensitivities for selected derivatives

**Asian Call: Monotonicity and Stability analysis [1]**

Log-returns (up) and volatility (down) for Price, Delta, Gamma and Vega.

\[ D = 32, \text{ BB discretization:} \]

MC (blue) vs pure QMC (red) vs randomized QMC (green)

QMC is highly monotonic for Price and Vega. Pure QMC is more stable than randomized QMC for Delta and Gamma: higher order interactions decrease monotonicity/stability?
3: Prices and sensitivities for selected derivatives

Asian Call: Monotonicity and Stability analysis [2]

Log-returns (up) and volatility (down) for Price, Delta, Gamma and Vega.

$D = 252$, Std vs BB discretization:
pure QMC (Std: blue, BB: red) vs randomized QMC (Std: green, BB: pink)

When effective dimension is low, i.e. with Brownian Bridge discretization, pure QMC with BRODA generator is always more monotonic and more stable.
3: Prices and sensitivities for selected derivatives

*Double knock-out barrier option: set up*

The third test case regards an ATM double knock-out call option.

**Test #3 specs**

- Risk-free rate: \( r = 0.03 \)
- Spot price: \( S_0 = 100 \)
- Volatility: \( \sigma = 0.3 \)
- Maturity: \( T = 1 \)
- Strike price: \( K = 100 \)
- Number of time steps: \( N_{\text{step}} = 32 \)
- Lower barrier: \( B_l = 0.5 \, S_0 \)
- Upper barrier: \( B_u = 1.5 \, S_0 \)
- Output values: Price, Delta, Gamma, Vega
- Number of MC/QMC trials for the computation of SI: \( N = 10^5 \)
- Number of simulated paths: \( N_{MC} = 2^p, p = 9, ..., 18 \)
- Number of independent runs: \( L = 30 \)
- Increment on finite differences:
  - \( \delta S_0 = \epsilon S_0, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
  - \( \delta \sigma = \epsilon, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
3: Prices and sensitivities for selected derivatives

Double knock-out barrier option: Global Sensitivity Analysis [1]

<table>
<thead>
<tr>
<th></th>
<th>$S_i/S_i^{tot}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$0.01 \rightarrow 0.15$</td>
<td>0.22</td>
<td>32</td>
<td>$&lt; 32$</td>
<td>8.5</td>
<td>-</td>
</tr>
<tr>
<td>Delta</td>
<td>$0.01 \rightarrow 0.12$</td>
<td>0.22</td>
<td>32</td>
<td>$&lt; 32$</td>
<td>7.6</td>
<td>no</td>
</tr>
<tr>
<td>Gamma</td>
<td>$10^{-5} \rightarrow 10^{-7}$</td>
<td>$10^{-4} \rightarrow 10^{-2}$</td>
<td>32</td>
<td>32</td>
<td>31.2 $\rightarrow$ 29.8</td>
<td>yes</td>
</tr>
<tr>
<td>Vega</td>
<td>$10^{-5} \rightarrow 10^{-8}$</td>
<td>$10^{-4} \rightarrow 10^{-2}$</td>
<td>32</td>
<td>32</td>
<td>28</td>
<td>yes</td>
</tr>
</tbody>
</table>

GSA for Standard Discretization

Some higher-order interactions among variables are present in any case, even though the effective dimension in the superposition sense could be slightly reduced for Price and Delta. Variables appear to lose importance while approaching maturity, for Price, Delta and Vega.

Gamma and Vega are Type-C function: in these cases QMC won’t outperform MC.
3: Prices and sensitivities for selected derivatives

*Double knock-out barrier option: Global Sensitivity Analysis [2]*

GSA for BB Discretization

Price, Delta and Gamma are Type-A functions: they show highly reduced effective dimensions since few variables are important. QMC is expected to outperform MC in these cases.

Vega is a Type-C function, even though the average dimension is not too high: in this case QMC is not expected to outperform MC.

<table>
<thead>
<tr>
<th></th>
<th>$S_i/S_i^{\text{tot}}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>0.70 → 0.01</td>
<td>0.70</td>
<td>$\sim$ 2</td>
<td>$\leq$ 2</td>
<td>1.63</td>
<td>-</td>
</tr>
<tr>
<td>Delta</td>
<td>0.83 → 0.01</td>
<td>0.83</td>
<td>2</td>
<td>$\leq$ 2</td>
<td>1.37</td>
<td>no</td>
</tr>
<tr>
<td>Gamma</td>
<td>1</td>
<td>$1 \rightarrow 0.95$</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td>small</td>
</tr>
<tr>
<td>Vega</td>
<td>$10^{-4} \rightarrow 0.2$</td>
<td>$10^{-6} \rightarrow 10^{-4}$</td>
<td>32</td>
<td>32</td>
<td>4.8 → 3.9</td>
<td>yes</td>
</tr>
</tbody>
</table>
3: Prices and sensitivities for selected derivatives

Double knock-out barrier option: Error Analysis

From log-log plots of error vs number of simulated paths, one can see that QMC + BB outperforms MC for Price, Delta and Gamma: remarkably, it is not due to the scaling law of the root mean square error, which is about $N_{MC}^{-0.6}$ in all cases, but to lower intercepts (QMC + BB “starts good”).

Vega cannot achieve higher efficiency with QMC than MC.

Speed-up analysis

QMC + BB yields speed-ups from 10 to >500 for Price, Delta and Gamma.
3: Prices and sensitivities for selected derivatives

Double knock-out barrier option: Convergence Analysis

For Price, Delta and Gamma, QMC + BB is much more stable and accurate than MC
3: Prices and sensitivities for selected derivatives

*Cliquet option: set up*

The fourth test case regards a **Cliquet option**, which is known to be highly path-dependent.

**Test #4 specs**

- Risk-free rate: \( r = 0.03 \)
- Spot price: \( S_0 = 100 \)
- Volatility: \( \sigma = 0.3 \)
- Maturity: \( T = 1 \)
- Local cap: \( C = 0.08 \)
- Global floor: \( F = 0.16 \)
- Number of time steps: \( N_{\text{step}} = 32 \)
- Output values: Price, Vega
- Number of MC/QMC trials for the computation of SI: \( N = 10^5 \)
- Number of simulated paths: \( N_{MC} = 2^p, p = 9, \ldots, 18 \)
- Number of independent runs: \( L = 30 \)
- Increment on finite differences:
  - \( \delta S_0 = \epsilon S_0, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
  - \( \delta \sigma = \epsilon, \epsilon = 10^{-4}, 10^{-3}, 10^{-2} \)
3: Prices and sensitivities for selected derivatives

**Cliquet option: Global Sensitivity Analysis**

<table>
<thead>
<tr>
<th>$S_i/S_i^{tot}$</th>
<th>$\sum_i S_i$</th>
<th>$d_T$</th>
<th>$d_S$</th>
<th>$d_A$</th>
<th>effect of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Vega</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td>1</td>
<td>no</td>
</tr>
</tbody>
</table>

**GSA for Standard Discretization**

Price and Vega are Type-B functions: variables equally important but interactions among variables are absent (effective dimension in superposition sense is 1): QMC will outperform MC.

**GSA for BB Discretization**

Price is a Type-A function: variables are not equally important and QMC would outperform MC. Vega shows some higher order interactions but effective dimension is slightly reduced.
3: Prices and sensitivities for selected derivatives

**Cliquet option: Error and Convergence Analysis**

From log-log plots of error vs number of simulated paths, one can see that, in this case, QMC + Std discretization outperforms MC and behaves better than QMC + BB for both Price and Vega: the scaling law of the root mean square error approaches $N_{MC}^{-1}$. Moreover, QMC Std is very stable (it already “starts good”).

### Speed-up analysis

QMC Std yields speed-ups up to 100 in both cases.

<table>
<thead>
<tr>
<th>$S_{\sigma}(a)$</th>
<th>QMC Std / MC</th>
<th>QMC BB / MC</th>
<th>QMC BB / QMC Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 1%$</td>
<td>$a = 0.1%$</td>
<td>$a = 1%$</td>
<td>$a = 0.1%$</td>
</tr>
<tr>
<td>Price</td>
<td>10</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>Vega</td>
<td>20</td>
<td>0.5</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Bianchetti, Kucherenko, Scoleri  
WBS 10th Fixed Income Conference  

Better Pricing and Risk Management with High Dimensional Quasi Monte Carlo  

p. 51
3: Prices and sensitivities for selected derivatives

Conclusions

- We have applied QMC method with high dimensional LDS (BRODA implementation of Sobol’ sequences) to pricing problems, extending the analyses found in the literature in several ways:
  - pricing of various payoffs with increasing complexity
  - computation of greeks
  - standard vs brownian bridge discretisation
  - global sensitivity, error, convergence, monotonicity and stability analysis

- QMC achieves significant improvements in efficiency, w.r.t. standard MC, provided that the effective dimension of the problem is low. This is due to a faster convergence rate (in most cases $\sim N^{-1}$) and higher stability.

- QMC also provides speed-up over MC up to several hundreds, especially when higher accuracy is desired: this means that, to reach a given accuracy, QMC needs up to hundreds of scenarios less than MC. In the worst cases, QMC efficiency is comparable to MC.

- In most cases, Brownian Bridge discretization provides better convergence, but this is not a general rule: e.g. Cliquet options behave better with standard discretization. Unfortunately, the optimal choice is not known in advance and empirical studies are necessary. Global Sensitivity Analysis helps a lot in this task, since it allows to easily compute the effective dimensions, which are powerful indicators of QMC performance.
4: Market and Counterparty risk of derivatives’ portfolios

Counterparty risk measures [1]

“Counterparty credit exposure is the amount a company could potentially lose in the event of one of its counterparties defaulting. At a general level, computing credit exposure entails simulating in different scenarios and at different times in the future, prices of transactions, and then using one of several statistical quantities to characterise the price distributions that has been generated. Typical statistics used in practice are (i) the mean, (ii) a high-level quantile such as the 97.5% or 99%, usually called Potential Future Exposure (PFE), and (iii) the mean of the positive part of the distribution, usually referred to as Expected Positive Exposure (EPE).” [Ces09].

In more detail, we will consider the following counterparty risk measures:

- Mark to Future (MtF)
- Expected Positive/Negative Exposure (EPE, or EE, ENE)
- Effective Positive/Negative Exposure (Eff. EPE)
- Potential Future Exposure $\alpha$° percentile (PFE $\alpha$)
- Effective Potential Future Exposure $\alpha$° percentile (Eff. PFE $\alpha$)
4: Market and Counterparty risk of derivatives’ portfolios

**Counterparty risk measures [2]**

Within the multi-dimensional multi-step MC simulation, the counterparty risk measures can be computed starting from the mark to future value $V_{jkh}$

- EPE/ENE for netting set $h$ at time simulation step $j$

  $EPE_{jh}(t_0) = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} [V_{jkh}(t_0) - C_{jkh}(t_0)]^+$,

  $ENE_{jh}(t_0) = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} [V_{jkh}(t_0) - C_{jkh}(t_0)]^-$,

  where $C_{jkh}(t_0)$ is the **collateral** available for netting set $h$ at time $t_j$ on MC scenario $s_{jk}$.

- Collateral simulation is discussed e.g. in [Pyk09].
4: Market and Counterparty risk of derivatives’ portfolios

*Numerical test #1: set up*

We run a first test on a selected counterparty, focused to check the convergence behaviour of the Monte Carlo simulation with respect to increasing scenario number.

**Test #1 specs**

- **Test date:** 7 Aug. 2013.
- **Two runs:**
  - 32,768 ($2^{15}$) Pseudo Monte Carlo scenarios, Mersenne Twister generator [MT98].
  - 32,768 ($2^{15}$) Quasi Monte Carlo scenarios, SobolSeq16384 generator [Broda].
- **Risk measure:** expected Mark to Future (MtF).
- **Portfolio:** one single netting set, primary bank under collateral, ~1,000 trades up to very long maturities (35Y).
- **Simulation dimension:**
  - nº of risk factors: $N_{rf} = 150$
  - nº of time simulation steps = $N_{step} = 53$
  - total = $N_{rf} \times N_{step} = 150 \times 53 = \text{7,950}$
4: Market and Counterparty risk of derivatives’ portfolios

**Numerical test #1: results [1]**

Counterparty primary bank under CSA, PMC vs QMC simulation, mark to future term structure with randomised standard deviation (16 runs 2,048 scenarios each, right hand axis). The PMC/QMC term structures with different scenarios are indistinguishable. The randomised standard deviation is smaller and more stable for QMC.
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #1: results [2]

Counterparty primary bank under CSA, PMC vs QMC simulation, mark to future, time step 28d (1M), ± 3 std dev PMC. Convergence as a function of increasing number of MC scenarios.
Numerical test #1: results [3]

Counterparty primary bank under CSA, PMC vs QMC simulation, mark to future, time step 365d (1Y), ± 3 std dev PMC. Convergence as a function of increasing number of MC scenarios.
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #1: results [4]

Counterparty primary bank under CSA, PMC vs QMC simulation, mark to future, time step 10,950d (30Y), ± 3 std dev PMC. Convergence as a function of increasing number of MC scenarios.
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #1: results [5]

Counterparty primary bank, PMC vs QMC simulation, mark to future distribution, time step 3650d (10Y). 2,048 vs 8,192 scenarios (top left), 2,048 vs 32,768 scenarios (bottom right).

The distributions are always very similar to each other. Minor differences may appear locally, where convexity is large, e.g. close to the tip.

Conclusion: 2,048 scenarios are not bad.
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #1: conclusions

- The mark to future profiles are already very similar with 2,000 scenarios. The QMC randomised standard deviation is much smaller than the corresponding PMC value.

- For each time simulation step, the mark to future difference PMC vs QMC with 2000 scenarios is always smaller than 3 Monte Carlo standard error (confidence level 99.7%). In many cases it is smaller than 1 MC standard error (confidence level 68.3%). The mark to future distribution is already stable with 2000 scenarios.

- The mark to future profile as a function of the number of MC scenarios is always much smoother for QMC than for PMC. The PMC «initial values» (with just a few scenarios) may be very different from the «final values» (with 32,768 scenarios), while the QMC simulation always «starts good».

- The facts above may be explained in terms of the properties of the PMC and QMC simulations:
  - Statistical properties of the PMC simulation
  - Low discrepancy of Sobol’ sequences, even with high dimensions
  - Reduced effective dimensionality of the mark to future MC simulation [Kuch05]: the mark to future depends, at each time simulation step, on a limited number of primary risk factors.

- In conclusion, at any finite number of MC scenarios, we may consider the QMC simulated values (e.g. the mark to future, or other risk measures), as reference values.
4: Market and Counterparty risk of derivatives’ portfolios

**Numerical test #2: set up [1]**

**Test #2 specs**

- **Test date:** 30 Apr. 2013
- **Four runs:**
  - 2,000 \(\approx 2^{11}\) PMC scenarios, Mersenne Twister generator [MT98]
  - 8,192 \(2^{13}\) PMC scenarios, Mersenne Twister generator [MT98]
  - 2,048 \(2^{11}\) QMC scenarios, SobolSeq16384 generator [Broda]
  - 8,192 \(2^{15}\) QMC scenarios, SobolSeq16384 generator [Broda]
- We analysed all the **counterparty risk measures:** MtF, EPE, Eff. EPE, PFE 95, Eff. PFE 95.
- Portfolio with \(\sim 500\) counterparties/netting sets, including major banks, under CSA or not, around \(10^5\) trades up to very long maturities (40Y).
- Simulation dimension: \(N_{rf} \times N_{step} = 150 \times 53 = 7,950\).
- In order to compare risk measures with a term structure, we used the distance (norm \(L^2\)),

\[
d_f(x, y) := \sqrt{\sum_{j=1}^{N_{step}} [f_x(T_j) - f_y(T_j)]^2}.
\]

where \(f_x(T_j)\) is the risk measure (e.g. the EPE) of type \(x\) (e.g. \(x = PMC 2,000\)).
### 4: Market and Counterparty risk of derivatives’ portfolios

**Numerical test #2: set up [2]**

<table>
<thead>
<tr>
<th>Expected mark to future (MtF)</th>
<th>Expected Positive Exposure and Effective Expected Positive Exposure (EPE and eff.EPE)</th>
</tr>
</thead>
</table>
| Potential Future Exposure 95% and Effective Potential Future Exposure 95% (PFE95 and eff.PFE95) | Relative Monte Carlo performance: PMC 2,000 vs QMC 8,192  
                               PMC 8,192 vs QMC 8,192  
                               QMC 2,048 vs QMC 8,192 |
Netting set: corporate (a few swaps, no CSA). The risk measures are already stable with PMC 2,000. QMC 2,048 similar to PMC 8,192. Minor differences in term structures.
Netting set: primary bank (thousands of trades, CSA). The risk measures are already stable with PMC 2,000. Minor differences. QMC better than PMC for mark to future, not for other risk measures.
4: Market and Counterparty risk of derivatives’ portfolios

**Numerical test #2: results [3]**

Netting set: primary bank (thousands of trades, CSA). The long term structure of EPE and PFE95 is sensible to the simulation, with visible consequences on the Effective PFE95. Only the MtF is improved with QMC 2,048.
Overview of relative performances in the mark to future computation. First 50 counterparties, ordered by effective EPE (PMC 2,000). Overall, the relative performance is strongly improved moving from PMC 2,000 to PMC 8,192 (red vs blue). Moving further to QMC 2048 allows a further improvement, with just a few exceptions (6-7, yellow higher than red).
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #2: results [5]

Overview of the relative performance to compute the Effective EPE.

First 50 counterparties, ordered by effective EPE (PMC 2,000).

The effective EPE depends on the local shape of the exposure in the 1Y time step. Hence it is less sensitive to the simulation when the short term term structure is smooth, while it may change a lot when the risk term structure shows large variations.
4: Market and Counterparty risk of derivatives’ portfolios

Numerical test #2: results [6]

Overview of the relative performance to compute the Effective PFE95.

First 50 counterparties, ordered by effective PFE95 (PMC 2,000).

The effective PFE depends on the local shape of the PFE. Hence it is less sensitive to the simulation when the short term term structure is smooth, while it may change a lot when the risk term structure shows large variations.
4: Market and Counterparty risk of derivatives’ portfolios

Conclusions

- We have tested our counterparty risk framework against the scenarios used in the multi-dimensional multi-step Monte Carlo simulation of counterparty risk measures.

- We found evidences that:
  - for many counterparties/netting sets 2,000 Pseudo Monte Carlo scenarios guarantee a sufficient convergence of the simulation, an increase of MC scenarios does not improve very much the counterparty risk figures;
  - for some counterparties/netting sets 2,048 Quasi Monte Carlo scenarios improve the simulation;
  - switching from 2,000 PMC to 2,048 QMC scenarios is a good compromise allowing a better convergence of the simulation without additional computational effort.

- The number of (independent) risk factors must be contained such that the dimension of the MC simulation does not exceed the dimension of the available low discrepancy sequences generator.

Quasi Monte Carlo is a candidate tool to make happy your regulators, IT department, and budget.
5: Conclusions

Directions of future work

- Apply to multi-asset options’ price and greeks using correlated Global Sensitivity Analysis [Kuc14].
- Apply to CVA using Global Sensitivity Analysis.
- Apply to risk measures using Global Sensitivity Analysis.

Work is in progress
6: References [1]

6: References [2]

6: References [3]

Disclaimer and acknowledgments

Disclaimer
The views and the opinions expressed here are those of the author and do not represent the opinions of his employer. They are not responsible for any use that may be made of these contents. No part of this presentation is intended to influence investment decisions or promote any product or service.

Acknowledgments
The results presented here are based on the joint effort of many colleagues at Intesa Sanpaolo and Banca IMI.