WORCESTER POLYTECHNIC INSTITUTE

Valuation of European Call Basket options by using a Quasi-Monte Carlo method

Author:
Weixi LIU

Advisor:
Professor Stephan Sturm

A capstone report submitted to the
Department of Mathematical Science
in partial fulfillment of
the requirements for the degree of

Master of Financial Mathematics

April 25, 2018

Abstract

This thesis briefly outlines the theoretical background to Quasi-Monte Carlo methods, and takes an European call basket option as an example to compare the prices, convergence rate and the root mean square errors between Monte Carlo methods and Quasi-Monte Carlo methods. The thesis is built on the assumption that each asset in basket can be represented by a geometric Brownian motion (GBM) and thus has log-normally distributed returns. The implied volatility can be calibrated from market data, and the assets correlation matrix can be calculated by Exponentially Weighted Moving Average (EWMA) (Hull, 2012, Chapter 10). For Quasi-Monte Carlo, we use Halton sequences to generate normally distributed sequences due to Beasley, Springer & Moro (Beasley and Springer, 1977), and by using random shift to improve accuracy and to produce a family of related quasi random sequences for applying Monte Carlo integration.

Keywords: Monte Carlo methods, Quasi-Monte Carlo methods, Implied Volatility, Assets Correlation, Exponentially Weighted Moving Average, Low-discrepancy sequences

Acknowledgements

I would like to express my gratitude to my advisor Professor Stephan Sturm, Department of Mathematics-University of WPI, for finding a relavant topic reflecting my academic interest, but also for his constructive suggestions and always having an answer to my questions of this project.

I would also like to thank all the students of Financial Math for their guidance and friendship.

I would also like to thank my beloved parents, who have encouraged me to make my dreams come true and have provided me with love and spiritual support. Finally, many thanks are due to everyone who helped me during my time at Worcester Polytechnic Institute.

Contents

List of Figures

List of Tables

List of Abbreviations

ECO	**E**uropean **C**all **O**ption
BS	**B**lack-**S**choles
QMC	**Q**uasi-**M**onte **C**arlo
MC	**M**onte **C**arlo
SDE	**S**tochastic **D**ifferential **E**quatiion
CEV	**C**onstant **E**lasticity of **V**ariance model
VDC	**V**an **d**er **C**orput
IBM	**I**nternational **B**usiness **M**achines
CDF	**C**umulative **D**istribution **F**unction
EWMA	**E**xponentially **W**eighted **M**oving **A**verage
ODE	**P**artial **D**ifferential **E**quation

List of Symbols

r	risk-free interest rate
$\mathbb{Q}$	the risk-neutral measure
B_i	standard Brownian motions
S	underlying price processes
W_i	Bownian motions under risk-free measure
Φ	CDF of standard normal distribution

Chapter 1

Introduction

1.1 Background

A basket option is an asset similar to the Asian option, a multidimentional derivative whose payoff depends on the average price of the underlying assets. But instead of taking the time average of one asset, the value of the basket option depends on the weighted sum of a number of underlying assets. These types of path-dependent derivatives are one of the more complicated contracts to value and price.

When pricing averaging options there is a challenge that traditional methods as finding numerical approximations for the partial differential equation are not efficient since the number of underlying assets might be large. Also the fact that the correlation among assets needs to be taken into account. The simple Black-Scholes model is built on the crude assumption that asset returns are log-normal distributed. It can be shown that a finite sum of log-normal distributed random variables is not log-normal distributed anymore, thus there does not exist a closed form solution as in the Black-Scholes model. Hence, we have to use standard Monte Carlo methods and Quasi-Monte Carlo methods which both are computational methods to price basket options.

In general, on the one hand, pricing financial derivatives using standard Monte Carlo methods has the limiting factor, that the clumping occurs in the points of a random or a pseudo-random sequence. The reason for this clumping is that the points are independent. On the contrary, Quasi-Monte Carlo methods use quasi-random sequences which are deterministic with correlations between the points to eliminate clumping they are also called low-discrepancy sequences. On the other hand, the simulation rate of standard Monte Carlo methods is rather slow, whereas Quasi-Monte Carlo methods is relatively fast.

1.2 Purpose

The goal of the present report is the introduction to basic knowledge about Quasi-Monte Carlo methods. We give an example by pricing specific basket options to compare the option prices for different strike prices, convergence rates and the root mean square errors between standard Monte Carlo methods and Quasi-Monte Carlo methods.

1.3 Outline

In the second chapter we will present some basic knowledge about Quasi-Monte Carlo methods, as Quasi-Random sequences, the definition of discrepancy and integration errors. Then we will present two specific low-discrepancy sequences, the Van der Corput Sequence and the Halton sequence. After that, we will introduce what is the root mean square error and randomizing QMC technics which include random shift, random permutation of digits, scrambled nets and linear permutation of digits.

In the third chapter we will present the market model and how to calculate the price of European call basket options. Then we will present an algorithm created by Beasley, Springer & Moro (Beasley and Springer, 1977) to generate normally distributed sequences, and introduce how to calculate implied volatility, and how to calculate asset correlation using historical data.

In chapter 4 we will present the data we will use for simulation and the numerical results.

All calculations and simulations in this report are coded and performed in Python, and the report was written in LaTeX.

Chapter 2

Quasi-Monte Carlo Methods

The Quasi-Monte Carlo (QMC) method is a method for the problem of numerical integration over the unit hypercube. It approximates the integral of a function f as the average of the function evaluated at a set of points $x_1, \cdots, x_n$ with carefully (and deterministically) chosen in the unit hypercube $[0,1)^d$:

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i). \tag{2.1}$$

2.1 Quasi-random sequences and Discrepancy

Quasi-random sequences were invented by number theorists who were interested in the uniformity properties of numerical sequences (Kuipers and Niederreiter, 1974, Hua and Wang, 1981), they are deterministic alternatives to random or pseudo-random sequences (Niederreiter, 1992). In contrast to pseudo-random sequences, which try to mimic the properties of random sequences, quasi-random sequences are designed to provide better uniformity than a random sequence, and hence faster convergence for quadrature formulas. Uniformity of a sequence of points is measured in terms of its *discrepancy*, and for this reason quasi-random sequences are also called low-discrepancy sequences.

Given a collection $\mathcal{A}$ of (Lebesgue measurable) subsets of $[0,1)^d$, define

$$R_n(\mathcal{A}) = \frac{\#\{x_i \in A\}}{n} - \text{vol}(A),$$

then under L^∞ norm on $\mathcal{A}$, the *discrepancy* of the point set $\{x_1, \cdots, x_n\}$ relative to $\mathcal{A}$ is

$$D_n(\mathcal{A}) = D(x_1, \ldots, x_n; \mathcal{A}) = \sup_{A\in\mathcal{A}} |R_n(\mathcal{A})|. \tag{2.2}$$

Here, $\#\{x_i \in A\}$ denotes the number of x_i contained in A and vol(A) denotes the volume (measure) of A. Thus, the discrepancy is the supremum over errors in integrating the indicator function of A using the points $x_1, \cdots, x_n$.

If we restrict $\mathcal{A}$ be a rectangular set, since a rectangular set can be defined by two antipodal vertices, a rectangle $\mathcal{A}$ can be defined as $\mathcal{A} = \mathcal{A}(x, y)$ in which the points x and y are antiposal vertices. So, under L^2 norm on $\mathcal{A}$, the *discrepancy* of the point

set $\{x_1, \cdots, x_n\}$ relative to $\mathcal{A}$ can be also defined as

$$T_n = \left[\int_{(x,y), x<y} R_n(\mathcal{A}(x,y))^2 dxdy \right]^{\frac{1}{2}}. \tag{2.3}$$

Taking $\mathcal{A}$ to be the collection of all rectangles in $[0,1)^d$ of the form

$$\prod_{j=1}^{d} [u_j, v_j), 0 \leq u_j < v_j \leq 1,$$

yields the ordinary (or *extreme*) discrepancy $D(x_i, \ldots, x_n)$. Restricting $\mathcal{A}$ to rectangles of the form

$$\prod_{j=1}^{d} [0, u_j) \tag{2.4}$$

defines the *star discrepancy* $D^*(x_i, \cdots, x_n)$. The star discrepancy is obviously not larger than the ordinary discrepancy; Niederreiter (Niederreiter, 1992), Proposition 2.4, shows that

$$D^*(x_1, \cdots, x_n) \leq D(x_1, \cdots, x_n) \leq 2^d D^*(x_1, \cdots, x_n),$$

so for fixed dimension d the two quantities have the same order of magnitude.

2.2 Koksma-Hlawka inequality

The basis for analyzing the Quasi-Monte Carlo quadrature error is the Koksma-Hlawka inequality.

Consider the integral

$$I[f] = \int_{I^d} f(x) dx, \tag{2.5}$$

where $I = [0,1)$, and the Monte Carlo approximation

$$I_N[f] = \frac{1}{N} \sum_{n=1}^{N} f(x_n), \tag{2.6}$$

define the quadrature error by

$$\epsilon[f] = |I[f] - I_N[f]| \tag{2.7}$$

Define the variation (in the Hardy-Krause sense, see Niederreiter, 1992) of f, a continuously differential function of a single variable, as

$$V[f] = \int_0^1 \left| \frac{df}{dt} \right| dt. \tag{2.8}$$

In d dimensions, the variation is recursively defined as (Caflisch, 1998)

$$V[f] = \int_{I^d} \left| \frac{\partial^d f}{\partial t_1 \cdots \partial t_d} \right| dt_1 \cdots dt_d + \sum_{i=1}^{d} V[f_1^i] \tag{2.9}$$

in which f_1^i is the restriction of the function f to the boundary $x_i = 1$. Since these restrictions are functions of $d-1$ variables, this definition is recursive.
Theorem 1 (Koksma-Hlawka theorem)**.** *For any sequence $\{x_n\}$ and any function f with bounded variation, the integration error ϵ is bounded as*

$$\epsilon[f] \leq V[f] D_N^*. \tag{2.10}$$

Therefore, the Quasi-Monte Carlo quadrature error depends only on the discrepancy D_N^*, in other words, we can reduce the error by using low-discrepancy sequences [1].

2.3 Low-Discrepancy Sequences

In the following we give some examples of low-discrepancy sequences.

2.3.1 Van der Corput Sequences

The Van der Corput sequence is the simplest example to one-dimensional low-discrepancy sequences (Niederreiter, 1992) over the unit interval. By a *base* we mean an integer $b \geq 2$. Every positive integer k has a unique representation (called its base-b or b-ary expansion) as a linear combination of nonnegative powers of b with coefficients $a_j(k)$ in $\{0, 1, \cdots, b-1\}$. We can write this as

$$k = \sum_{j=1}^{\infty} a_j(k) b^j, \tag{2.11}$$

with all but finitely many of the coefficients $a_j(k)$ equal to zero. The radical inverse function ψ_b maps each k to a point in $[0, 1)$ by flipping the coefficients of k about the base-b "decimal" point to get the base-b fraction $.a_0 a_1 a_2 \cdots$.

More precisely,

$$\psi_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}}. \tag{2.12}$$

The base-b Van der Corput sequence is the sequence $0 = \psi_b(0), \psi_b(1), \psi_b(2), \cdots$. Its calculation is illustrated in table 2.1 for base 2.

2.3.2 Halton sequences

Halton (Halton, 1960), extending work of Hammersley (Hammersley, 1960), provides the simplest construction and first analysis of low-discrepancy sequences in arbitrary dimension d. The coordinates of a Halton sequence follow Van der Corput

[1] see section 2.3.

k	k Binary	$\psi_2(k)$ Binary	$\psi_2(k)$
0	0	0	0
1	1	0.1	$\frac{1}{2}$
2	10	0.01	$\frac{1}{4}$
3	11	0.11	$\frac{3}{4}$
4	100	0.001	$\frac{1}{8}$
5	101	0.101	$\frac{5}{8}$
6	110	0.011	$\frac{3}{8}$

TABLE 2.1: One-dimensional VDC sequence with base 2.

sequences in a distinct base. Thus, let $b_1, \cdots, b_d$ be relatively prime integers greater than 1, and set

$$x_k = (\psi_{b_1}(k), \psi_{b_2}(k), \cdots, \psi_{b_d}(k)), \quad k = 0, 1, 2, \cdots, \tag{2.13}$$

with ψ_b the radical inverse function defined in (2.12).

The requirement that the b_i be relatively prime is necessary for the sequence to fill the hypercube. What happens if not, for example, we illustrate a two-dimensional sequence defined by bases $b_1 = 2$ and $b_2 = 6$ in table 2.2 and figure 2.1. We can see there are no points in $[0, \frac{1}{2}) \times [\frac{5}{6}, 1)$, actually it has 5 big "blank" regions which means no points exist there. On the contrary, if we try bases $b_1 = 2$ and $b_2 = 3$, we can see they generate a good random sequence with low discrepancy from figure 2.2. And because we prefer smaller bases to larger bases, we usually take $b_1, \cdots, b_d$ to be the first d prime numbers.

k	$\psi_2(k)$	$\psi_6(k)$
0	0	0
1	$\frac{1}{2}$	$\frac{1}{6}$
2	$\frac{1}{4}$	$\frac{1}{3}$
3	$\frac{3}{4}$	$\frac{1}{2}$
4	$\frac{1}{8}$	$\frac{2}{3}$
5	$\frac{5}{8}$	$\frac{5}{6}$
6	$\frac{3}{8}$	$\frac{1}{36}$
7	$\frac{7}{8}$	$\frac{7}{36}$

TABLE 2.2: Two-dimensional Halton sequence with bases 2 and 6.

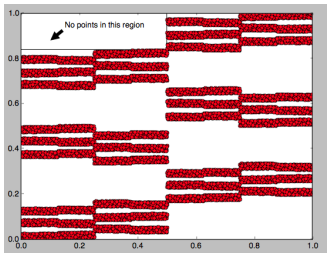

FIGURE 2.1: First 10000 points of two-dimensional Halton sequence with base 2 and 6.

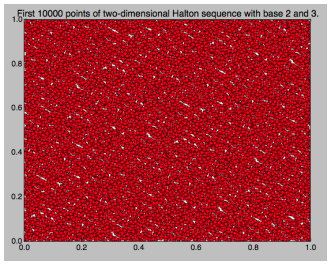

FIGURE 2.2: First 10000 points of two-dimensional Halton sequence with base 2 and 3.

2.3.3 Error analysis

We can see Halton sequences are a generalization of Van der Corput sequences. The discrepancy of a Halton sequence is bounded by

$$D_n(Halton) \leq c_d(\log n)^d n^{-1} \tag{2.14}$$

in which c_d is a constant depending on d. On the other hand, the average discrepancy of a random sequence is

$$E\left[T_n^2(\text{random})\right]^{\frac{1}{2}} = c_d n^{-\frac{1}{2}}, \tag{2.15}$$

where T_n follows the definition of equation (2.3).

For quasi-random sequences with discrepancy of size $O((\log n)^d n_{-1})$, the Koksma-Hlawka inequality implies that integration error is of the same size, that is

$$\epsilon[f] \leq c_d V[f](\log n)^d n^{-1} \tag{2.16}$$

Note that the Koksma-Hlawka inequality applies for any sequence. For quasi-random sequence, the discrepancy is small, so that the integration error is also small.

2.4 The Root Mean Square Error

A comparison of methods requires a figure of merit to evaluate there performances. For Monte Carlo methods, variance is an appropriate figure of merit, at least for

unbiased estimations with similar computing requirements (see Glasserman, 2003). The average of n responses has a variance smaller than the variance of a single response, so a comparison of variance is not tied to a particular sample size. In contrast, the integration error produced by a QMC method does depend on the number of points n, and often quite erratically. Moreover, the QMC error can be quite sensitive to parameters of the problem. This makes the comparison of QMC methods less straightforward.

As our figure of merit, we take the root mean square error over a fixed set of problem instances. Given m problems with true values $C_1, \cdots, C_m$ and n-point QMC approximations $\hat{C}_1(n), \cdots, \hat{C}_m(n)$, the root mean square is

$$\text{RMSE}(n) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{C}_i(n) - C_i)^2} \tag{2.17}$$

and the RMS relative error is

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{C}_i(n) - C_i}{C_i}\right)^2}. \tag{2.18}$$

Since there is no closed form Black-Scholes formula to price basket options, we will use approximate prices by doing high enough iterations of the QMC methods and the MC methods to replace the true prices $\{C_i\}_{i=1}^m$. We can see from figure 2.3, the difference of prices becomes less as iteration increases. And based on firgure 2.4, we regard the price calculated by the QMC method or the MC method with iteration 20000 as true option price [2], since when iterations larger than 20000, for each method, the difference of prices less than 0.1.

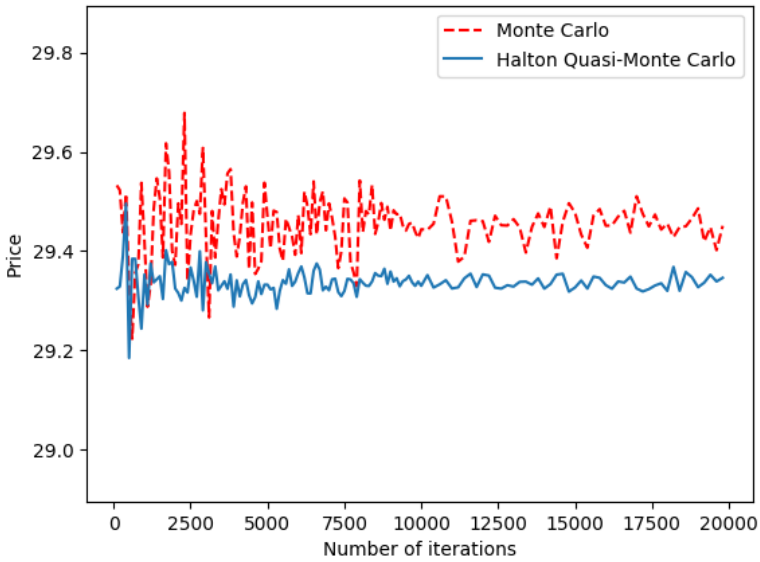

FIGURE 2.3: Prices of a basket option generated by QMC and MC with T = 0.15, dt = 0.01, and iterations from 100 to 20000.

[2] In case of $T = 0.15$, $dt = 0.01$

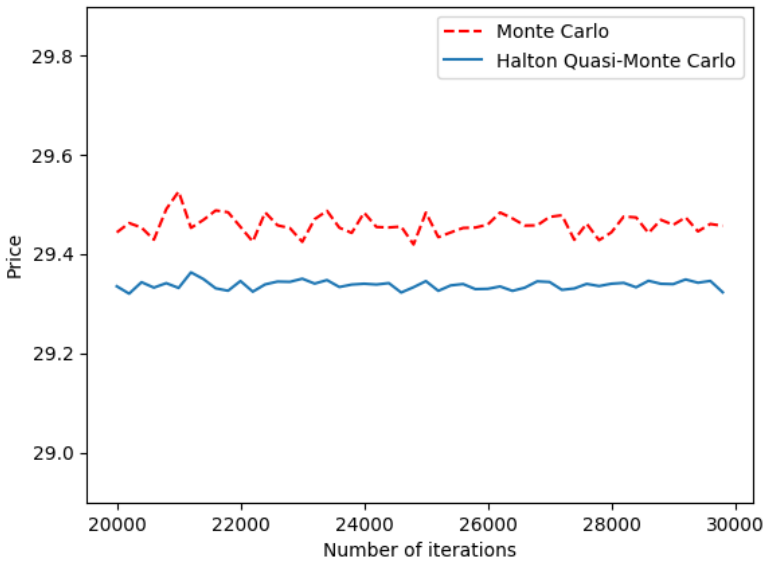

FIGURE 2.4: Prices of a basket option generated by QMC and MC with T = 0.15, dt = 0.01, and iterations from 20000 to 30000.

2.5 Randomized QMC

There are two good reasons for randomizing QMC. The first reason is that this can improve accuracy. A particularly remarkable result of this type is a theorem of Owen (Owen, 1997), showing that the root mean square error of integration using a class of randomized nets is $O(1/n^{1.5-\epsilon})$, whereas the error without randomization is $O(1/n^{1-\epsilon})$. The second reason is we need to use randomizing QMC methods to generate n different QMC sequences for pricing options [3].

Denote such a point set generically by

$$P_n = \{x_1, \cdots, x_n\}, \tag{2.19}$$

where each x_i is an element of $[0,1)^d$. We introduce 4 methods, which are Random Shift method, Random Permutation of Digits method, Scrambled Nets method and Linear Permutation of Digits method for randomizing QMC.

2.5.1 Random Shift

The Random Shift is a method for randomizing QMC. The simplest randomization of the point set P_n generates a random vector U uniformly distributed over the d-dimensional unit hypercube, and shifts each point in P_n by U, modulo 1:

$$P_n(U) = \{x_i + U \mod 1, i = 1, \cdots, n\}. \tag{2.20}$$

The reduction mod 1 applies separately to each coordinate.

[3] see section 3.2

2.5.2 Random Permutation of Digits

Another mechanism for randomizing QMC applies a random permutation of $0, 1, \cdots, b-1$ to the coefficients in the base-b expansion of the coordinates of each point. Consider the one-dimensional case and write $x_k = 0.a_1(k)a_2(k)\cdots$ for a b-ary representation of x_k. Let $\pi_j, j = 1, 2, \cdots$ be independent random permutations of $\{0, 1, \cdots, b-1\}$, uniformly distributed over all $b!$ permutations of the set. Randomize P_n by mapping each point x_k to the point $0.\pi_1(a_1(k))\pi_2(a_2(k))\cdots$, applying the same permutations π_j to all points x_k. For a d-dimensional point set, randomize each coordinate in this way, using independent permutations for different coordinates.

2.5.3 Scrambled Nets

For the one-dimensional case. Suppose x has the b-ary representation $0.a_1a_2a_3\cdots$. The first coefficient a_1 is mapped to $\pi(a_1)$, with π as random permutation of $\{0, 1, \cdots, b-1\}$. The second coefficient is mapped to $\pi_{a_1}(a_2)$, the third coefficient to $\pi_{a_1,a_2}(a_3)$, and so on; the random permutations $\pi, \pi_{a_1}, \pi_{a_1,a_2}, \cdots,\ a_j = 0, 1, \cdots, b-1,\ j = 1, 2, \cdots$, are independent with each uniformly distributed over the set of all permutations of $\{0, 1, \cdots, b-1\}$. To scramble a d-dimensional point set, apply this procedure to each coordinate, using independent sets of permutations for each coordinate.

2.5.4 Linear Permutation of Digits

This method maps a base-b expansion $0.a_1a_2\cdots$ to $0.\hat{a}_1\hat{a}_2\cdots$ using

$$\hat{a}_j = \sum_{i=1}^{j} h_{ij}a_i + g_j \ \ mod\, b,$$

with the h_{ij} and g_j chosen randomly and independently from $\{0, 1, \cdots, b-1\}$ and the h_{ij} required to be positive.

Chapter 3

Setting

3.1 The Market

We operate in the context of a complete, standard financial market $\mathcal{M}$, with constant risk-free rate r and volatility matrix $\boldsymbol{\sigma}$. The price processes of the assets in this market are governed by a set of stochastic differential equations (SDEs). We propose pricing methods for European-call basket options in a Black & Scholes framework.

Consider a basket consisting of N assets with prices $S_i(t)$, $i = 1, \cdots, n$, which are described, with risk-free rate r, along with money market account $B(t)$, by

$$dB(t) = rB(t)dt, \qquad B(0) = 1; \tag{3.1}$$

$$dS_i(t) = \mu_i S_i(t)dt + \sigma_i S_i(t)dW_i(t), \tag{3.2}$$

in which $\{W_i(t), t > 0\}$ are standard Brownian motions associated with the price of asset i and $\sigma_i (\geq 0)$ are the corresponding volatilities.

Equations (3.1) and (3.2) can be solved by the development of a risk-free measure $\mathbb{Q}$ and straightforward use of Ito's formula (Shreve, 2004) leading to the following expressions for the price processes:

$$B(t) = e^{rt}, \qquad B(0) = 1; \tag{3.3}$$

$$S_i(t) = S_i(0) \exp\left[\left(r - \frac{1}{2}\sigma_i^2\right) t + \sigma_i W_i(t)\right], \quad i = 1, \cdots, N. \tag{3.4}$$

in which $\{W_i(t), t > 0\}$ are Brownian motions under the risk-free measure $\mathbb{Q}$.

Further, we assume that the different assets returns are correlated in a constant way i.e.

$$Cov(dW_i, dW_j) = \rho_{ij}dt. \tag{3.5}$$

Here ρ_{ij} is the correlation coefficient between the ith and jth Brownian motion. The symmetric matrix with ρ_{ij} as the entry in the ith row and jth column is called the correlation matrix. For example, if we have five underlyings $N = 5$, the correlation matrix will look like this:

$$\begin{pmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} & \rho_{15} \\ \rho_{21} & 1 & \rho_{23} & \rho_{24} & \rho_{25} \\ \rho_{31} & \rho_{32} & 1 & \rho_{34} & \rho_{35} \\ \rho_{41} & \rho_{42} & \rho_{43} & 1 & \rho_{45} \\ \rho_{51} & \rho_{52} & \rho_{53} & \rho_{54} & 1 \end{pmatrix}$$

Note that $\rho_{ii} = 1$ and $\rho_{ij} = \rho_{ji}$. The correlation matrix is positive definite, so that $y^T Dy \geq 0$.

Then we can write (3.4) as

$$S_i(t) = S_i(0) \exp\left[\left(r - \frac{1}{2}\sigma_i^2\right) t + \sigma_i Z_i\right], \tag{3.6}$$

in which $\mathbf{Z} = \{Z_1, \cdots, Z_5\}$ is multivariate normal vector with mean zero and covariance matrix $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$. In this paper, we generate $\mathbf{Z}$ by a Cholesky factorization (Glasserman, 2003) of Σ.

3.2 The value of the Claim

For a European call option can be expressed as

$$V(t) = \exp^{-r(T-t)} \mathbb{E}\left[\varphi(S(T))\middle|\mathcal{F}_t\right] \tag{3.7}$$

The function $\varphi(S(T))$ is depending on the type of European Call Option (ECO). We are developing a pricing formula for an ECO on a basket of assets where the different assets are correlated. That is

$$\sum_{n=1}^{N} \alpha_n = 1, \tag{3.8}$$

$$\varphi(S(T)) = \left(\sum_{n=1}^{N} \alpha_n S_n(T) - K\right)^+, \quad n = 1, \cdots, N, \tag{3.9}$$

in which $\{\alpha_n\}_{n=1}^N$ are proportionality coefficients, and K is the strke price. This leads to the price process of the particular ECO

$$V(t) = \exp^{-r(T-t)} \mathbb{E}\left[\left(\sum_{n=1}^{N} \alpha_n S_n(T) - K\right)^+ \middle| \mathcal{F}_t\right] \tag{3.10}$$

$$= \exp^{-r(T-t)} \mathbb{E}\left[\left(\sum_{n=1}^{N} \alpha_n S_n(0) \exp\left[\left(r - \frac{1}{2}\sigma_n^2\right) T + \sigma_n W_n(T)\right] - K\right)^+ \middle| \mathcal{F}_t\right]. \tag{3.11}$$

For simplicity we prefer to consider the value at $t = 0$, which is the value at the time you are buying the ECO. The expression then becomes

$$V(0) = \exp^{-rT} \mathbb{E}\left[\left(\sum_{n=1}^{N} \alpha_n S_n(0) \exp\left[\left(r - \frac{1}{2}\sigma_n^2\right) T + \sigma_n W_n(T)\right] - K\right)^+\right]. \tag{3.12}$$

Then we can use MC integration, that is, for obtaining an approximation to $V(0)$ by calculating

$$\exp^{-rT}\left(\frac{\varphi(S^1(T)) + \varphi(S^2(T)) + \cdots + \varphi(S^n(T))}{n}\right) \tag{3.13}$$

in which $\varphi(S^i(T)), i = 1, 2, \cdots, n$ are the ith $S(T)$ is generated by the ith normal distributed sequence (see section 3.3), which is generated by the ith QMC sequence.

3.3 Generating of Normally Distributed Sequences

For using a standard Monte Carlo method for the evaluation of the expression (3.12) we could simply use the fact that $W_n(T) \sim N(0, \sqrt{T})$ in (3.12) and use some known algorithms like the **Box Muller algorithm** (Box and Muller, 1958), **the Polar Marsaglia method** and **the Marsaglia Bray method** (Ripley, 1987) to simulate values from the standard normal distribution, the last two are known as rejection methods which mean that some combinations of the uniform distributed variables used in the algorithms are rejected. However, these methods are not good to apply to Quasi-Monte carlo calculations because they may damage the low discrepancy properties of the sequence by altering the order of the sequence or scrambling the sequence uniformity.

Thus in this paper, we use an algorithm created by Beasley, Springer & Moro (Beasley and Springer, 1977) for generating normal distribution sequences.

Recall that the CDF of a standard normal distribution is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt. \tag{3.14}$$

Let Φ^{-1} be the inverse of the cumulative distribution Φ, the algorithm divides the domain for uniform sample U into two regions:

1. The region of the distribution, $0.5 \leq x < 0.92$, $x \sim U[0, 1]$, is modeled as in (Beasley and Springer, 1977) :

   $$\Phi^{-1}(x) = \frac{\sum_{n=0}^{3} a_n (x - 0.5)^{2n}}{1 + \sum_{n=0}^{4} b_n (x - 0.5)^{2n}}$$

   wherea_n, b_n are from the table below:

n	a_n	b_n
0	2.50662823884	-8.4735109309
1	-18.61500062529	23.08336743743
2	41.39119773534	-21.06224101826
3	-25.44106049637	3.13082909833

2. And for, $0.92 \leq x < 1$, Φ^{-1} is modeled by:

   $$\Phi^{-1}(x) = \sum_{n=0}^{8} c_n (\ln(-\ln(1 - x)))^n \tag{3.15}$$

   where c_n are from the table below:

n	c_n
0	0.3374754822726147
1	0.9761690190917186
2	0.1607979714918209
3	0.0276438810333863
4	0.0038405729373609
5	0.0003951896511919
6	0.0000321767881768
7	0.0000002888167364
8	0.0000003960315187

3. Finally, for $0 \leq x < 0.5$, Φ^{-1} is modeled by:

$$\Phi^{-1}(x) = -\Phi^{-1}(1-x). \tag{3.16}$$

3.4 Calibration of Implied Volatility

We try to set up the calibration of implied volatility in a way that the calculated prices are as close as possible to the actual market data. Denote the parameter set $\sigma = \{\sigma_1, \cdots, \sigma_n\}$, the model prices by C_i^σ and the true prices by C_i, we calibrate it by minimizing the sum of the squared errors

$$\arg\min_{\sigma} \sum_i |C_i^\sigma - C_i|^2$$

Such a procedure is done numerically via the scipy optimize package in python [1].

3.5 Assets Correlation

In this paper, we use Exponential Weighted moving Average (EWMA) (Hull, 2012, Chapter 10) to get the correlation matrix of all assets. The EWMA is a statistic for monitoring the process that averages the data in a way that gives less and less weight to data as they are further removed in time.

Define $\bar{\sigma}_n$ as the volatility per day of market variable on day n, as estimated at the end of day $n-1$. Suppose that the value of the market variable at the end of day i is S_i. Define μ_i as the continuously compounded return during day i (between the end of day $i-1$ and the end of day i) so that

$$\mu_i = \ln \frac{S_i}{S_{i-1}}. \tag{3.17}$$

[1] More detail can see from `https://python4mpia.github.io/fitting_data/least-squares-fitting.html`

In general, for risk management purposes, we can compute $\bar{\sigma}_n^2$ in this way (Hull, 2012, Chapter 10):

$$\bar{\sigma}_n^2 = \frac{1}{m}\sum_{i=1}^{m}\mu_{n-i}^2. \tag{3.18}$$

Equation (3.18) gives equal weights to all $\mu_{n-1}^2, \cdots, \mu_{n-m}^2$. Our objective is to estimate $\bar{\sigma}_n$, the volatility on day n. It therefore makes sense to give more weight to recent data. Then the EWMA model is

$$\sum_{i=1}^{m}\alpha_i = 1 \tag{3.19}$$

$$\bar{\sigma}_n^2 = \sum_{i=1}^{m}\alpha_i\mu_{n-i}^2, \tag{3.20}$$

where α_i decrease exponentially as we move back through time. Specifically, $\alpha_{i+1} = \lambda\alpha_i$ where λ is a constant between zero and one. The RiskMetrics database, which was originally created by JPMorgan and made publicly available in 1994, used the EWMA model with $\lambda = 0.94$ for updating daily volatility estimates in its RiskMetrics database. The company found that, across a range of different market variables, this value of λ gives forecast of the variance rate that come closest to the realized variance rate (Morgan, 1995). Thus in this paper, we choose $\lambda = 0.94$.

It turns out that this weighting scheme leads to a particularly simple formula for updating volatility estimates. The formula is

$$\bar{\sigma}_n^2 = \lambda\bar{\sigma}_{n-1}^2 + (1-\lambda)\mu_{n-1}^2. \tag{3.21}$$

Similarily, we can compute the covariance of any two assets by replacing μ_{n-1}^2 with the cross product of two returns.

We know that the correlation of assets x and y can be calculated using the following formula

$$Corr = \frac{Cov_{xy}}{\bar{\sigma}_x\bar{\sigma}_y}$$

Thus once we can calculate the two volatility $\bar{\sigma}_x, \bar{\sigma}_y$ and Cov_{xy}, it is easy to get the correlation of assets x and y.

Chapter 4

Implementation

In this chapter, we use 3 months risk-free rate r = 1.11% with initial date at 10/26/2017, collected from U.S. Department of the Treasury [1].

4.1 Model parameters

We calculated the implied volatility of each stock for every asset as shown in table 4.1. We collected stock prices and strke prices of call options on 3-month from Yahoo Finance on 10/26/2017 at close 4:00 pm. In general, for each stock call option, we want to use 20 strikes which each strike price has a trading volume of more than 10 and we use the marke mid-price (average between bid and ask) to calibrate implied volatility from Yahoo Finance, however, there is just one stock, which is AAPL, more than 20 strike prices with trading volume of more than 15.

Stock	Stock Price	No. of strikes	Implied Volatility
AAPL	157.41	20	0.219
NVDA	195.69	19	0.383
IBM	153.06	15	0.155
F	12.27	14	0.173
MSFT	78.76	17	0.166

TABLE 4.1: Implied Volatilities.

For calculating the correlation matrix, we use the historical data from Yahoo Finance from 10/26/16 to 10/26/17 of each asset, the result as shown in figure 4.1.

Ticker	AAPL	IBM	NVDA	MSFT	F
AAPL	1	0.022424137225	0.34873147445	0.484380230661	0.115175203267
IBM	0.022424137225	1	0.0832905077635	0.146826726755	0.211643476696
NVDA	0.34873147445	0.0832905077635	1	0.333234860688	0.155428206702
MSFT	0.484380230661	0.146826726755	0.333234860688	1	0.114982535877
F	0.115175203267	0.211643476696	0.155428206702	0.114982535877	1

FIGURE 4.1: Correlation Matrix.

[1] http://www.ams.org/tex/amslatex.html

4.2 Pricing European Call Basket option

We consider a European call basket option within on a basket consisting of stocks AAPL (Apple), NVDA (NVIDIA Corporation), IBM (International Business Machines Corporation), F (Ford Motor Company) and MSFT (Microsoft Corporation) with initial time on 10/26/2017 and maturity time on 1/23/2018. And the proportionality coefficient of each asset are 35%, 30%, 10%, 10% and 15% as shown in table 4.2.

Stock	Proportionality coefficient
AAPL	35%
NVDA	30%
IBM	10%
F	10%
MSFT	15%

TABLE 4.2: Proportionality coefficient.

For different strike prices, we calculate the prices of this basket option with both Quasi-Monte Carlo (QMC) method and standard Monte-Carlo (MC) method, the results are shown as in figure 4.2. We can see the two methods yield similar results, in general, the prices calculated by QMC method are slightly smaller than those calculated by standard MC method.

Value	K = 100	K = 105	K = 110	K = 115	K = 120	K = 125	K = 130	K = 135
Quasi-Monte-Carlo	29.3360513016	24.3287440998	19.3556283175	14.3667238	9.4014188587	4.40336613131	0.358593420579	0.0
Monte-Carlo	29.4591811752	24.4579591921	19.5024458992	14.4692288497	9.52260913096	4.71126387701	1.25095163056	0.135689923793

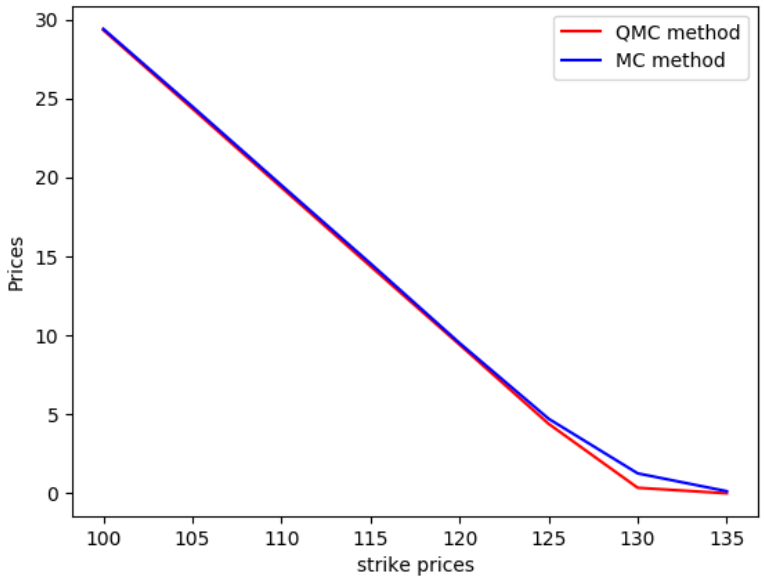

FIGURE 4.2: Prices of European call basket option with different strike prices. With initial stock prices and implied volatilities as shown in table 4.1 column Stock Price and column Implied Volatility.

We compare the root mean square errors (RMSE) for the two methods as shown in figure 4.3. As we can see, the two RMSEs are declining as the number of iteration increasing, but it is obvious that QMC method has lower RMSE than standard MC method, which means OMC method can achieve the given accuracy with less iteration number than standard MC method. And then, we compare the the convergence rate for the two methods as shown in figure 4.4, it shows the convergence rate of price calculated by QMC method is faster than that calculated by standard MC method.

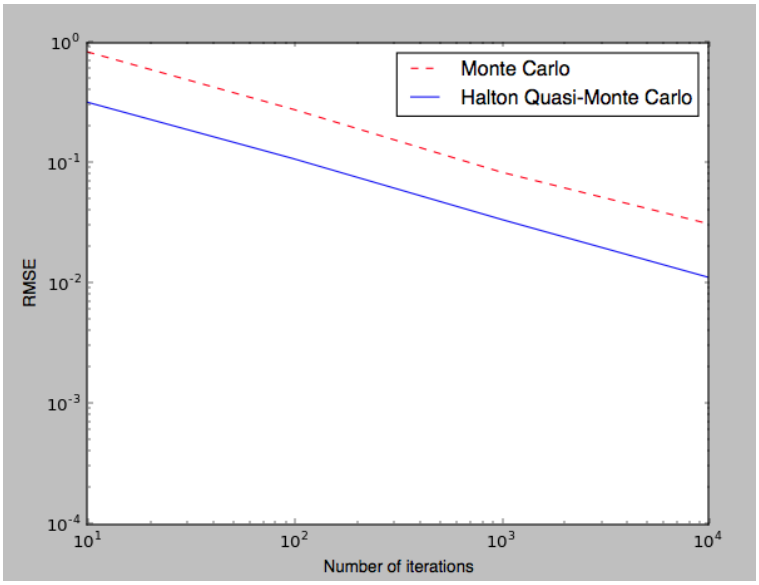

FIGURE 4.3: Root mean square errors in pricing basket option with 5 assets.

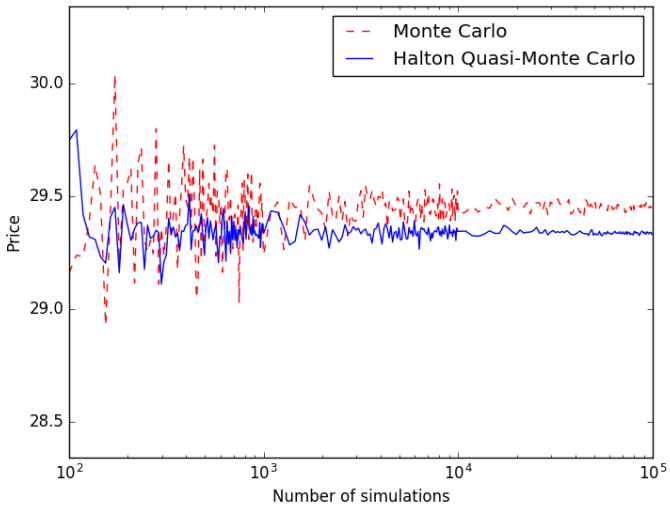

FIGURE 4.4: Convergence Rate for MC methods and QMC methods.

4.3 Conclusion

This report has presented how to use QMC methods to price basket option, the procedures are :

- Generate low-discrepancy sequences (Halton sequences)
- Generate normal distributed sequences
- Calibrate implied volatility
- Construct correlation matrix
- Generate correlated asset paths
- Price option

And we take an European call basket option as an example to compare the difference of QMC method and standard MC method. The OMC method will generate slightly lower prices than MC method. Furthermore, the use of Quasi Monte Carlo method gives faster convergence than the use of the standard Monte Carlo method for the European cal basket option we are studying.

Bibliography

Beasley, J. D. and S. G. Springer (1977). "The Percentage Points of the Normal Distribution". In: *Applied Statistics* 23.1, pp. 118–121.

Box, G. E. P. and Mervin E. Muller (1958). "A Note on the Generation of Random Normal Deviates". In: *The Annals of Mathematical Statistics*, pp. 610–611.

Caflisch, Russel E. (1998). "Monte Carlo and quasi-Monte Carlo methods". In: *Acta Numerica*, pp. 1–49.

Glasserman, Paul (2003). *Monte Carlo Methods in Financial Engineering*. Springer Science, New York.

Halton, J. H. (1960). "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals". In: *Numerische Mathematik 2:84-90*.

Hammersley, J. M. (1960). "Monte Carlo methods for solving multivariable problems". In: *Annals of the New York Academy of Sciences 86:884-874*.

Hua, L. K. and Y. Wang (1981). *Applications of Number Theory to Numerical Analysis*, Springer, Berlin/New York.

Hull, John C. (2012). *Risk Management and Financial Institutions*. John Wiley & Sons, Hoboken, New Jersey.

Kuipers, L. and H. Niederreiter (1974). *Uniform Distribution of Sequences*, Wiley, New York.

Morgan, J. P. (1995). *RiskMetrics Monitor*.

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*, Society for Industrial and Applied Mathematics, Philadelphia.

Owen, A. B. (1997). "Scrambled net variance for integrals of smooth functions". In: *Annals of Statistics* 25, pp. 1541–1562.

Ripley, B. D. (1987). *Stochastic Simulation*. John Wiley & Sons.

Shreve, Steven E. (2004). *Stochastic Calculus for Finance II: Continuous - Time Models*, Springer Science, New York.