Distribution Agreement

Hanqiu Xia April 8, 2015

Pricing Multi-asset Path-dependent Options Through Monte Carlo Simulations

by

Hanqiu Xia

James Nagy
Advisor

Department of Mathematics and Computer Science

James Nagy

Advisor

David Zureick-Brown

Committee Member

Kaiji Chen

Committee Member

2015

Pricing Multi-asset Path-dependent Options Through Monte Carlo Simulations

By

Hanqiu Xia

James Nagy
Advisor

An abstract of a thesis submitted to the Faculty of Emory College
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2015

Abstract

Pricing Multi-asset Path-dependent Options Through Monte Carlo Simulations
By Hanqiu Xia

Monte Carlo simulation (MC) is an approach that is widely used in high-dimensional numerical integration and one of its main financial applications is option pricing. The aim of this thesis is to evaluate the price of an Asian option using standard Monte Carlo method and Quasi-Monte Carlo method (QMC) respectively. Since QMC's convergence rate is determined by nominal problem dimension, the convergence rate of QMC increases as the problem dimension increases, which limits the performance of QMC in high dimensions. Hence, in this thesis, we also consider several techniques which are proposed to capture the effective dimensions and improve the efficiency of QMC in high-dimensional situations. The techniques include principal component analysis (PCA) and Kronecker product approximation (KPA) and they are applied for both constant and time-dependent volatilities. Finally, we conduct numerical experiments and compare the precision and computational time between Quasi-Monte Carlo and Monte Carlo methods.

Pricing Multi-asset Path-dependent Options Through Monte Carlo Simulations

By

Hanqiu Xia

James Nagy
Advisor

A thesis submitted to the Faculty of Emory College
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Options are derivative contracts that provide buyers with a right, but not an obligation, to sell or buy assets at a predetermined price during a specified time period (American style options) or at a certain time (European style options). The essence of an option is to estimate the value of the right that option can provide, and is referred to as the option price. Determining an arbitrage-free price is at the center of valuing an option's price. Evaluation of an option's price requires significant computational effort, involving mathematical functions in high dimensions. Besides, there are several dynamic factors associated with an option's price, therefore, accurately estimating options price is a necessary and challenging task for financial analysts. The aim of this thesis is to investigate an approach to evaluate the price of high-dimensional European style Asian options through standard Monte Carlo methods and Quasi-Monte Carlo methods respectively. In this chapter, we present the background of options pricing and the mathematical model we will use to evaluate Asian options; more detailed background can be found in [4].

1.1 Background

In current financial markets, most options are either European or American options. European style options can only be exercised on the date of expiration;

therefore, its payoff is only determined by the underlying assets price on that day, while American options can be exercised at any time before expiration. The options whose payoffs depend on the path of the underlying assets price just like American options are called path-dependent options.

In this thesis, we will introduce another important example of path-dependent options, called Asian options. An Asian option's payoff is based on the difference between the average price of the underlying asset during the contract term and the strike price. Nowadays, Asian options have become increasingly popular in options markets, because of their lower volatilities and cheaper premium [7].

The premium, also know as the option price, is the most essential part in option trading. Setting a good price for an option can protect the interest of both buyers and sellers in any market situations. For buyers, this price sets the limit of the maximum loss the buyers could suffer. For the sellers, it represents the amount of money they can receive from buyers without delivering the underlying assets immediately, although they have the obligation to exercise the contract when the buyers require, regardless of the movement in the market.

However, the procedure of option pricing is not easy, since the option price depends on many elements, such as strike price, length of maturity, time-dependent volatilities and the spot price of the underlying assets; some of these elements are even dynamic and hard to predict. The theoretical definition of an Asian options price is determined by several forecasting-needed factors, such as asset price. Hence, a theoretical analysis is usually inapplicable in practical cases, since it is under an unrealistic assumption that we know, for example, the exact asset price at time t in the future. Therefore, we need to develop computational methods for option price evaluation.

With the booming of financial industry, vast studies are conducted in this field and several new methods have been proposed to price path-dependent options. Specifically, when it comes to Asian options, Monte Carlo simulations are commonly used.

1.2 Mathematical Model

Suppose we focus our analysis on European Style Asian options in a complete and standard financial market under the multi-dimensional Black Scholes framework. The market contains M risky assets and 1 risk free asset. The following formulas are based on the works of Sabino [7, 6].

We denote the price of a risk free asset at time t as $S_0(t)$, then:

$$S_0(t) = S_0(0)e^{rt}, \tag{1.1}$$

where $S_0(0)$ is the initial price of the risk free asset, and r is the constant interest rate under the assumption of continuous compounding interest.

We assume the risky assets' price behavior follows Geometric Brownian motion and satisfies the stochastic differential equation (SDE:)

$$dS_i(t) = rS_i(t)dt + \sigma_i(t)S_i(t)dW_i(t), \qquad i = 1, 2, \ldots, M \tag{1.2}$$

where $S_i(t)$ is the i-th risky asset price at time t, $\sigma_i(t)$ denotes the time-dependent volatility of the i-th asset return, which determines how far $S_i(t+\Delta)$ can wander during a time interval of length Δ. W represents the M-dimensional standard Brownian motion. The former term is used to simulate deterministic trends, while the latter term represents the unpredictable events occurring during this motion.

According to Ito's Lemma, the solution for equation (1.2) is [4]:

$$S_i(t) = S_i(0)\exp\left[\int_0^t \left(r - \frac{\sigma_i^2(s)}{2} ds \right) + \int_0^t \sigma_i(s) dW_i(s) \right]. \tag{1.3}$$

This solution is a multi-dimensional geometric Brownian motion, GMB($r, \int_0^t \frac{\sigma_i^2(s)}{2} ds$).

We work with a finite time set $\{t_1, \ldots, t_N\}$ in evaluating Asian option prices, so that we can rewrite the above equation as:

$$S_i(t_j) = S_i(0)\exp\left[\int_0^{t_j} \left(r - \frac{\sigma_i^2(s)}{2} \right) ds + Z_i(t_j) \right] \tag{1.4}$$

for the time-dependent case, or

$$S_i(t_j) = S_i(0)\exp\left[\left(r - \frac{\sigma_i^2}{2} \right) t_j + Z_i(t_j) \right] \tag{1.5}$$

for the constant volatilities case. We define $Z_i(t_j) = \int_0^{t_j} \sigma_i(s) dW_i(s)$, and Z is an $M \times N$ matrix whose entries are normal random variables with zero mean and covariance matrix Σ_{MN}.

For time-dependent volatilities, Σ_{MN} has the block matrix structure:

$$\Sigma_{MN} = \begin{pmatrix} \Sigma(t_1) & \Sigma(t_1) & \ldots & \Sigma(t_1) \\ \Sigma(t_1) & \Sigma(t_2) & \ldots & \Sigma(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(t_1) & \Sigma(t_2) & \ldots & \Sigma(t_N) \end{pmatrix}, \quad \Sigma_{i,k}(t_j) = \int_0^{t_j} \rho_{ik}\sigma_i(t_j)\sigma_k(t_j) \tag{1.6}$$

where $i, k = 1, \ldots, M$. ρ_{ik} is the constant instantaneous correlation between W_i and W_k. Each block $\Sigma(t_j)$ is a matrix of size $M \times M$, and $\Sigma_{i,k}(t_j)$ is its (i, k) entry. For each element in Σ_{MN}:

$$((\Sigma_{MN})_{i,k})_{lm} = \int_0^{min(t_l,t_m)} \rho_{ik}\sigma_i(t)\sigma_k(t)dt, \tag{1.7}$$

For constant volatilities, we have a similar block structure, but with constant $\Sigma(t_j)$; that is,

$$\Sigma_{MN} = \begin{pmatrix} t_1\Sigma & t_1\Sigma & \dots & t_1\Sigma \\ t_1\Sigma & t_2\Sigma & \dots & t_2\Sigma \\ \vdots & \vdots & \ddots & \vdots \\ t_1\Sigma & t_2\Sigma & \dots & t_N\Sigma \end{pmatrix}, \quad \Sigma_{i,k} = \rho_{ik}\sigma_i\sigma_k. \tag{1.8}$$

The price at time t of a simple European Style option with maturity T is:

$$V(t) = \exp(-r(T-t))E\left[\ \phi(T)\middle|\mathscr{I}_t\ \right], \tag{1.9}$$

where $\phi(T)$ denotes the payoff of the option at maturity time T, and $\mathscr{I}_t$ is the information filtration, which is basically all the information we have at time t. Since the option may be traded several times, traders may want to have a daily value of this option in order to evaluate its risks. Therefore, the formula of a simple European Style option is a function of t. By evaluating the function in equation (1.9) at the payoff of Asian options, which is determined by the average underlying assets' price over a pre-set period of time, we get the theoretical definition of Asian options:

$$a(t) = \exp(-r(T-t))E\left[\ \max\left(\frac{\displaystyle\int_0^T \sum_{i=1}^{M} w_i S_i(t_j)dt}{T} - K, 0\right)\middle|\mathscr{I}_t\ \right], \tag{1.10}$$

where we assume the start time of the option is $t = 0$. K denotes the strike price of the option, and w_i represents the weight of i-th risky asset with $\sum_{i=1}^{M} w_i = 1$.

The formula above works with continuous time; it is usually unrealistic in practice, so we instead work with a finite discrete time grid $\mathscr{T} = \{t_1, t_2, \dots, t_N\}$, $t_1 < t_2 <$

$\ldots < t_N \leq T$. Based on this discrete time grid, the formula that approximates the theoretical Asian options price by sums is as follows:

$$a(t) = \exp(-r(T-t))E\left[\ \max\left(\ \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}S_i(t_j) - K, 0\ \right)\middle|\mathscr{I}_t\ \right]. \tag{1.11}$$

We should note that $\exp(-r(T-t))$ here works as the discount factor to convert the payoff of the option to the current option price at time t.

Equation (1.11) can be rewritten as an integral on a hypercube $[0,1]^{MN}$, so that we can use MC and QMC to estimate the integral

$$a(t) = \exp(-(T-t))\int_{[0,1]^{NM}} \max\left(\ g(F^{-1}(\mathbf{u})) - K, 0\ \right)\mathbf{du}. \tag{1.12}$$

The integrated part is the payoff of Asian options at maturity T, where $F^{-1}(\mathbf{u})$ is the inverse cumulative distribution function of normal variables Z, $\mathbf{u}$ represents the random vectors that lie in $[0,1]^{NM}$

and

$$g(\mathbf{Z}) = \sum_{k=1}^{M\times N} \exp(\mu_k + Z_k). \tag{1.13}$$

For time-dependent volatilities,

$$\mu_k = \ln(w_{k_1k_2}S_{k_1}(0)) + rt_{k_2} - \frac{\int_0^{t_{k_2}} \sigma_{k_1}^2(t)dt}{2}, \tag{1.14}$$

and for constant volatilities,

$$\mu_k = \ln(w_{k_1k_2}S_{k_1}(0)) + \left(\ r - \frac{\sigma_{k_1}^2}{2}\ \right)t_{k_2}. \tag{1.15}$$

The index $k_1 = 1 + (k-1) \mod M$, and $k_2 = \lfloor (k-1)/M \rfloor + 1$, here $\lfloor\cdot\rfloor$ denotes the floor function; that is, $\lfloor z \rfloor$ computes the greatest integer less than or equal to z.

Equation (1.12) is the focus of this chapter, and we will use this formula to estimate the Asian options price by using a standard Monte Carlo method and a quasi-Monte Carlo method in the following chapters. The thesis is organized as follows: In Chapter 2, we present the basic ideas of MC and QMC. In Chapter 3, we introduce some techniques that can fulfill the purpose of selecting the effective dimensions. Chapter 4 conducts numerical tests for the simulation procedures we adopt. Chapter 5 draws some conclusions.

Chapter 2

Monte Carlo Methods

There are two Monte Carlo simulations that are usually applied in evaluating high-dimensional path-dependent options price – standard Monte Carlo (MC) and Quasi-Monte Carlo (QMC). The main idea of MC and QMC are quite similar, they both estimate high-dimensional integrals by computing the average of a great number of simulations, the only difference is MC conducts the simulation on pseudorandom sequences, while QMC is based on low-discrepancy sequences, which is more deterministic than the former one. Because of this difference, QMC generally has a lower convergence rate than MC, which means QMC needs a fewer number of samples to compute the results. However,the lower convergence rate of QMC is related to the problem dimension; specifically, QMC has a better convergence rate for lower dimensional integrals. It is therefore necessary to select the effective dimensions for QMC by analysis of variance (ANOVA) techniques when we deal with high-dimensional path-dependent options. In this chapter, we illustrate the basic idea of Monte Carlo simulations, and present strengths and weakness of MC and QMC, respectively.

2.1 Standard Monte Carlo Methods

The standard Monte Carlo method can be used to numerically evaluate an integral in a hypercube by drawing a number of uniform random points from $[0, 1]^d$ and then computing the arithmetic average.

We can illustrate this idea through a simple integration example in $d = 1$ dimensions. Suppose we need to calculate the integral of a function $f(x)$ from a to b in Figure 2.1.

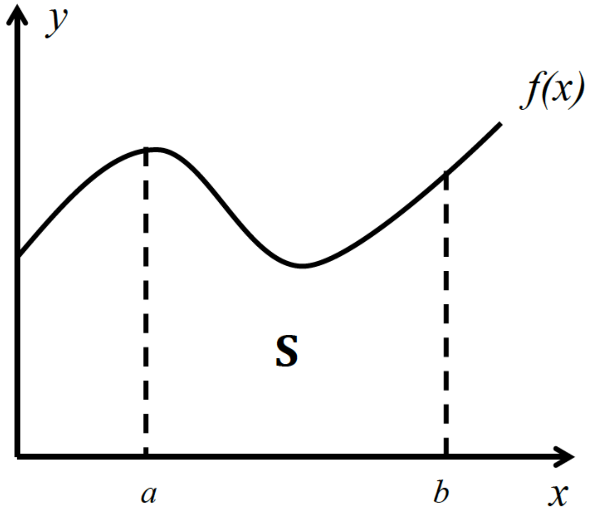

Figure 2.1: Integral of $f(x)$ on the interval $[a, b]$, where $x \in \Re$.

In a geometric sense, the integral is the area S under the graph of $f(x)$, above the x-axis and also bounded by the vertical lines $x = a$ and $x = b$. According to the first mean value theorem for integration, there exists a number x in $[a, b]$ such that

$$\int_a^b f(t)dt = f(x)(b - a), \tag{2.1}$$

where $f(x)$ is the mean value of $f(t)$ defined in interval $[a, b]$. Therefore, we can approximate the integral by taking a number of random samples $x_i, x_i \in \Re$, and replace the mean value of $f(t)$ with a numerical average of the $f(x_i)$ values; that is,

$$\int_a^b f(t)dt \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)(b-a). \tag{2.2}$$

If $b = 1$ and $a = 0$, then the integral approximately equals the average value of $f(x_i)$, that is:

$$\int_0^1 f(t)dt \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i). \tag{2.3}$$

Without loss of generality, as for estimating integrals in the hypercube $[0,1]^d$, *e.g.*, $I = \int_{[0,1]^d} f(\mathbf{t})\mathbf{dt}$ we can draw n independent random samples $\mathbf{x}_i$ from a uniform distribution on $[0,1]^d$, each $\mathbf{x}_i$ is a column vector of d elements, and compute the arithmetic average of $f(\mathbf{x}_i)$ [5]:

$$\int_{[0,1]^d} f(\mathbf{t})\mathbf{dt} \approx \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\int_{[0,1]^d}\mathbf{dt} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i) = \hat{I}. \tag{2.4}$$

According to the Law of Large Numbers (LLN), which states that the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed, $\hat{I}$ converges to I as n increases.

Based on the work of Sabino [7], $I - \hat{I}$ has a normal sampling distribution $N(0, \sigma/\sqrt{n})$, where σ is the true standard deviation of $f(\mathbf{x})$, which is generally unknown. So, we usually estimate it by the sample standard deviation s, which is the standard deviation of $f(\mathbf{x}_i)$ in the sample:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\Big(\ f(\mathbf{x}_i) - \hat{I}\ \Big)^2}. \tag{2.5}$$

Hence, the estimated standard deviation or standard error of the sampling distribution of $I - \hat{I}$ is $\dfrac{s}{\sqrt{n}}$. Recall from probability theory, we can compute the smallest number of simulations n needed to guarantee the standard error not greater than ε_α through the following expression [2] :

$$P\Big(\Big| \ I - \hat{I} \ \Big| < \varepsilon_\alpha \ \Big) \geq (1 - \alpha). \tag{2.6}$$

This probability inequality is based on t-distribution with $\alpha\%$ level of significance and $\varepsilon_\alpha = \dfrac{2s}{\sqrt{n}}$.

MC chooses a set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ by using a pseudorandom sequence. As its name suggests, a pseudorandom sequence is a sequence of numbers that appears to be random, and it can be easily generated by pseudorandom number algorithms (for example, MATLAB's `rand` function). The points from a pseudorandom number source are randomly and irregularly distributed in the space. An illustration in 2-dimension is shown in Figure 2.2 :

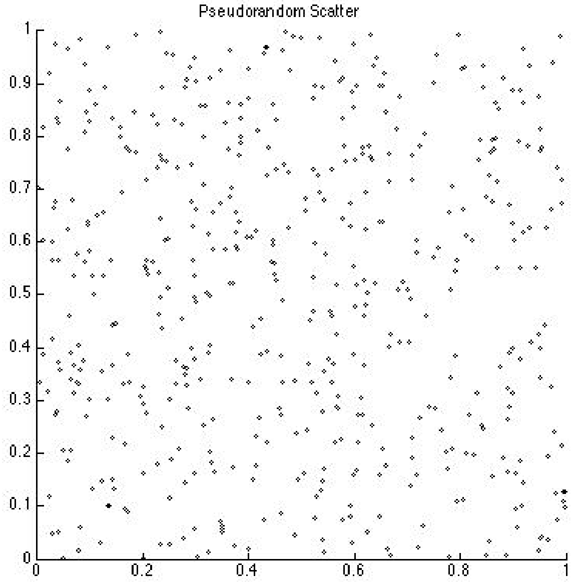

Figure 2.2: 500 points from a pseudorandom sequence in 2 dimensions.

Because it is relatively simple to generate a pseudorandom sequence, MC can be easily applied in many cases, even some complicated integrations. Moreover, since MC's convergence rate is $O\left(\frac{1}{\sqrt{n}}\right)$ that does not depend on the dimensionality of integral, MC has some advantages in pricing derivative contracts in high dimensions. However, this convergence rate also puts limits on MC's usefulness. The square root form in the denominator of the convergence rate means that we need to quadruple the number of points if we want to halve the root mean square error (RMSE). In other words, MC's low order of convergence requires a large number of simulations. Another weakness of MC is its results are statistical in nature. The function (2.6) just sets a probabilistic error bound, which means the estimated integral we compute can be wrong with some probability [2]. To sum up, the standard Monte Carlo method is not a perfect technique for numerical integrations, and refinements are needed.

2.2 Quasi-Monte Carlo Methods

The quasi-Monte Carlo method works in almost the same way as the standard Monte Carlo method does, except it is based on low-discrepancy sequences. A low-discrepancy sequence is a set of d-dimensional points with a low star discrepancy.

The definition of the star discrepancy of a point set $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in $[0,1)^d$ is:

$$D_N(p) = \sup \left| \ \frac{A(J;n)}{n} - \lambda(J) \ \right|, \tag{2.7}$$

where $J = \displaystyle\prod_{j=1}^{d} [0, u_j), 0 \leq u_j \leq 1$; $\lambda(J)$ is the Lebesgue measure of J; and $A(J;n)$ represents the number of $\mathbf{x}_i$ that fall into the set J . Hence, the discrepancy of sequence P is low if the percentage of points $\mathbf{x}_i$ in J is close to the Lebesgue measure of J [2].

Compared to a pseudorandom sequence, a low-discrepancy sequence works with more deterministic points not simply random ones. Two classical types of low-discrepancy sequences in 2-dimensions are shown in Figure 2.3:

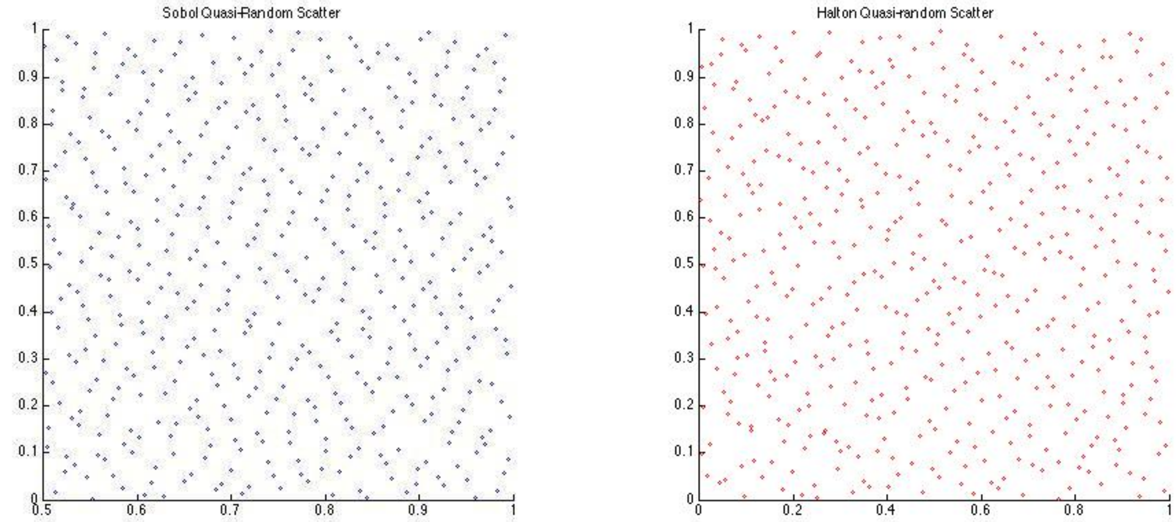

Figure 2.3: 500 points from a Sobol sequence (left); 500 points from a Halton sequence (right)

From the above figures, we can see that the points from a low-discrepancy sequence fills the space more evenly than random points (compare Figure 2.2 with Figure 2.3). The fact that QMC works with well-chosen points brings it several advantages. First, this property makes QMC overcome the probabilistic problems that MC could face, and result in better point estimates. Moreover, using a low-discrepancy sequence makes QMC have a generally lower convergence rate – $O\left(\frac{(\ln n)^d}{n}\right)$, which means QMC can converge faster than MC and require fewer number of simulations to obtain a similar integration level as MC does [2].

Although QMC has all of the above-mentioned advantages, it is not a panacea for numerical integration, since it also has some noticeable drawbacks. For instance, based on the Koksma-Hlawka inequality, the error of QMC is bounded by:

$$|I - \hat{I}_{QMC}| \leq D_n^* V_{HK}(f), \tag{2.8}$$

where D_n^* denotes discrepancy of the point set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ that QMC focuses on, and $V_{HK}(f)$ is the Hardy-Krause variation of the function f. Inequality (2.8) only provides the upper bound of the error of QMC, which fails to give us much information. Besides, it is hard to estimate this bound, because we have difficulties in computing $V_{HK}(f)$ even in very simple cases [6].

In order to estimate the error of the QMC method, we can use a randomized quasi-Monte Carlo method (RQMC) [7]. Compared to QMC, the feature of RQMC is randomizing the points obtained from a low-discrepancy sequence. The simplest randomization procedure is through random shifting. The idea is we first sample N random vectors $\mathbf{u}_i$ and then mix them with points, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a low-discrepancy sequence. For each new point we have

$$\mathbf{y}_i = (\mathbf{x}_i + \mathbf{u}_i) \bmod 1. \tag{2.9}$$

Here mod 1 does not work as the regular modulus. If x is a real number in $[1, 2]$, then x mod 1 implies the decimal part of x; if x is in $[0, 1]$, then x mod 1 equals to x. The reason we use mod 1 here is to make sure that the new points $\mathbf{y}_i$ are in the interval $[0, 1]^d$. After this transformation, we use the point set $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ to implement RQMC instead of $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Since RQMC is conducted on random points, we can use formula (2.5) to compute the approximation error of RQMC.

In addition, we should notice that the dimensionality of the problem has an effect on QMC's order of convergence, which implies that $O\left(\dfrac{(\ln n)^d}{n}\right)$ is smaller than $O\left(\dfrac{1}{\sqrt{n}}\right)$ only if n is large enough and d is small. Therefore, this property limits QMC's superiority in high-dimensional integrals. Recently, a significant amount of research has been done on this issue and aimed to extend QMC's superiorities in high dimensions. Sabino has stated in his work that a substantial amount of financial experiments have proved that the number of dimensions that really matter is usually lower than the nominal one and also suggested that we can capture the most essential dimensions of the problem by analysis of variance (ANOVA) [6] .

In summary, although the standard MC and QMC use the same idea in the equation (2.4) to estimate the numerical integral, they obtain point sets from different sources. MC just simply selects points from a pseudorandom sequence. As for QMC, besides constructing point sets by using low-discrepancy sequences, we also need to reduce the nominal dimensions of the problem before we compute the average of simulations. We will introduce the techniques for selecting the effective dimensions in the next chapter.

Chapter 3

Effective Dimensions Selection

The majority of multi-assets path-dependent options usually involve functions in high dimensions, which can result in expensive computational cost and inapplicability of QMC. Moreover, variables in many financial markets are highly correlated with each other, which implies many variables in the data set share only a few essential sources of information [3]. Therefore, we can select the most important uncorrelated sources in the multivariate system to significantly reduce the nominal dimensions. Sabino suggests that the effective dimensions can be extracted by the analysis of variance (ANOVA) method [7]. The basic technique underlying ANOVA is Principal Component Analysis (PCA), whose goal is to search for the first few principal components that can account for most of the overall variance of multivariate data $\mathbf{x} = [x_1, \ldots, x_n]^T$ [3]. In the constant volatilities case, it can be shown that Σ_{MN} can be written as a Kronecker product of two smaller matrices. That is,

$$\Sigma_{MN} = R \otimes \Sigma = \begin{pmatrix} t_1\Sigma & t_1\Sigma & \ldots & t_1\Sigma \\ t_1\Sigma & t_2\Sigma & \ldots & t_2\Sigma \\ \ldots & \ldots & \ddots & \ldots \\ t_1\Sigma & t_2\Sigma & \ldots & t_N\Sigma \end{pmatrix},$$

where R is the $N \times N$ matrix. We can we can improve the computational speed of PCA by taking advantage of some properties of Kronecker products. While for the time-dependent volatilities case, since Σ_{MN} cannot be written as the Kronecker product of R and Σ, we introduce a new technique called Kronecker product approximation (KPA) to improve the efficiency of PCA.

3.1 Principal Component Analysis

Principal component analysis (PCA) is a data transformation tool that can transform a group of possibly correlated variables into a set of linearly uncorrelated variables. In general, after transformation, the number of linearly uncorrelated variables is less than or equal to the number of original variables. These linearly uncorrelated variables are called principal components. PCA is commonly used in financial markets, since many financial systems have high collinearity between returns [3].

3.1.1 Mathematical Background of PCA

PCA is a statistical procedure that focus on the analysis of the eigenvalues and eigenvectors of a covariance matrix. According to the definition of PCA, the first principal component express as much of the variability in the data as it can, that is, it has the largest possible variance, the second principal component has the second largest possible variance given that it is uncorrelated to the preceding principal component, and so forth and so on. The following formulas are from the work of Xing and Lai [3].

Let $\mathbf{x} = [x_1, \ldots, x_n]^T$ be a normal random vector with known mean μ and covariance matrix $\mathbf{V}$. $\mathbf{V}$ is a $n \times n$ symmetric nonnegative covariance matrix with n eigenvalues λ_i, sorted in decreasing order $\lambda_1 > \lambda_2 > \ldots \lambda_n$, and corresponding eigenvector $\mathbf{e}_i$. Note also that if $\mathbf{e_i}$ is an eigenvector, then $c\mathbf{e_i}$ is also an eigenvector, for any scalar $c \neq 0$. PCA defines the ith principal component of $\mathbf{x}$ as $\mathbf{e}_i^T\mathbf{x}$. To maximize the variance of $\mathbf{e}^T\mathbf{x}$ over all unit eigenvectors $\mathbf{e}$, we can use calculus to show:

$$\frac{\partial}{\partial e_j}(Var(\mathbf{e}^T\mathbf{x})) = \frac{\partial}{\partial e_j}(\mathbf{e}^T\mathbf{V}\mathbf{e}) = 0. \tag{3.1}$$

Since $\mathbf{e}^T\mathbf{e} = ||\mathbf{e}||^2 = 1$, we can introduce a Lagrange multiplier λ:

$$\frac{\partial}{\partial e_j}(\mathbf{e}^T\mathbf{V}\mathbf{e} + \lambda(1 - \mathbf{e}^T\mathbf{e})) = 0 \tag{3.2}$$

$$\Rightarrow \mathbf{V}\mathbf{e} = \lambda\mathbf{e}.$$

This implies that λ is an eigenvalue of $\mathbf{V}$, and $\mathbf{e}$ is the corresponding eigenvector. So, we have $\max(\mathbf{e}^T\mathbf{V}\mathbf{e}) = \max(\mathbf{e}^T\lambda\mathbf{e}) = \max(\mathbf{e}^T\mathbf{e}\lambda) = \max(\lambda) = \lambda_1$. The first principal component of $\mathbf{x}$ is $\mathbf{e}_1^T\mathbf{x}$, and with variance λ_1.

For second principal component, we consider to maximize $\mathrm{Var}(\mathbf{e}^T\mathbf{x})$ under the constraint that $\mathbf{e}$ is orthogonal to $\mathbf{e}_1$, i.e. $\mathbf{e}_1^T\mathbf{e} = 0$, and $||\mathbf{e}|| = 1$. We introduce Lagrange multipliers λ and η:

$$\frac{\partial}{\partial e_j}(\mathbf{e}^T\mathbf{V}\mathbf{e} + \lambda(1 - \mathbf{e}^T\mathbf{e}) + \eta\mathbf{e}_1^T\mathbf{e}) = 0 \tag{3.3}$$

$$\Rightarrow \mathbf{V}\mathbf{e} = \lambda\mathbf{e}, \quad \mathbf{e}_1^T\mathbf{e} = 0.$$

This implies that λ is an eigenvalue of $\mathbf{V}$ other than λ_1, because the corresponding eigenvector $\mathbf{e}$ is orthogonal to $\mathbf{e}_1$. So, $\max(\mathbf{e}^T\mathbf{V}\mathbf{e}) = \max(\lambda) = \lambda_2$. The second principal component of $\mathbf{x}$ is $\mathbf{e}_2^T\mathbf{x}$, with variance λ_2.

Proceeding inductively in this way, we get

$$\lambda_{k+1} = \max_{||\mathbf{e}||=1, \mathbf{e}_j^T\mathbf{e}=0, \text{ for } 1\leq j\leq k} Var(\mathbf{e}^T\mathbf{V}\mathbf{e}). \tag{3.4}$$

The maximizer $\mathbf{e}$ in equation (3.4) is the eigenvector $\mathbf{e}_{k+1}$ corresponding to λ_{k+1}, and $\mathrm{Var}(\mathbf{e}_{k+1}^T\mathbf{x}) = \lambda_{k+1}$.

The aim of PCA is to find the first few principal components whose overall variance that can account for most of the overall variance, that is:

$$\frac{\displaystyle\sum_{i=1}^{k} Var(\mathbf{e}_i^T\mathbf{x})}{\displaystyle\sum_{i=1}^{n} Var(x_i)} = p, \text{ for } p \approx 1. \tag{3.5}$$

Since $\mathbf{V} = cov(\mathbf{x})$, the diagonal entries of $\mathbf{V}$ are the variances of x_i. Hence, $\displaystyle\sum_{i=1}^{n} Var(\mathbf{x}_i) = trace(\mathbf{V}) = \lambda_1 + \cdots + \lambda_n$. We know $\displaystyle\sum_{i=1}^{k} Var(\mathbf{e}_i^T\mathbf{x}) = \sum_{i=1}^{k} \lambda_i$ from the inductive procedure above, then we can rewrite the ratio in equation (3.5) as:

$$\frac{\displaystyle\sum_{i=1}^{k} \lambda_i}{\displaystyle\sum_{i=1}^{n} \lambda_i} = p. \tag{3.6}$$

Here k is the effective dimension we can identify by fixing p with a value very close to 1, for example, we can choose $p = 99\%$.

3.1.2 New Covariance Matrix after PCA

After applying PCA to reduce the nominal dimensions, we obtain a new multivariate random vector $\mathbf{z}$ which is made up by the first k principal components of $\mathbf{x}$:

$$\mathbf{z} = (\mathbf{e}_1^T\mathbf{x}, \mathbf{e}_2^T\mathbf{x}, \ldots, \mathbf{e}_k^T\mathbf{x})^T = E\mathbf{x} \tag{3.7}$$

where $E = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)^T$. Since PCA is a linear transformation, a linear transformation of a multivariate normal random vector $\mathbf{x}$ also has a multivariate normal distribution, but with a new mean and covariance matrix. According to work of Xing and Lai, the new random vector $\mathbf{z}$ has a multivariate normal distribution with mean

$$E(\mathbf{z}) = E\mu \tag{3.8}$$

and $k \times k$ covariance matrix

$$Var(\mathbf{z}) = E\mathbf{V}E^T \tag{3.9}$$

3.1.3 Improvement of PCA

PCA is a fundamental method in determining the effective dimension of high-dimensional integrals. Reducing dimensionality cannot only shorten the computation time, but also help QMC be feasible in high-dimensional financial cases.

Since PCA is based on the eigenvalues of a covariance matrix, if the dimension is high, it would take much time to compute the eigenvalues of the covariance matrix directly. For this drawback, we can remedy it by exploiting properties of Kronecker products. The idea is that we rewrite the covariance as the Kronecker product of two smaller matrices and analyze the eigenvalues of the two smaller matrices instead.

In the following we illustrate this improvement of PCA with the Asian options example that we have mentioned in chapter 2. In the constant volatilities case, because the covariance matrix Σ_{MN} is characterized by a special structure, that is, $\Sigma_{MN} = R \otimes \Sigma$, we can reduce calculation by computing the eigenvalues of the two smaller matrices R and Σ. An important property of Kronecker products is that, if $\mathbf{e}$ and $\mathbf{v}$ are vectors containing the eigenvalues of A and B, respectively, then $\mathbf{e} \otimes \mathbf{v}$ is a vector containing the eigenvalues of $A \otimes B$ [7]. Therefore, we can get the eigenvalues of Σ_{MN} by computing the eigenvalues of R and Σ:

$$eig(\Sigma_{MN}) = eig(R) \otimes eig(\Sigma), \tag{3.10}$$

where $eig(\cdot)$ denotes a column vector of all eigenvalues of the matrix. By applying this analysis, we convert the analysis of an $MN \times MN$ matrix into the analysis of two smaller $M \times M$ and $N \times N$ matrices, and then significantly reduce the computational effort from $O((MN)^3)$ to $O(M^3 + N^3)$ [7].

In the time-dependent volatilities case, the covariance matrix Σ_{MN} has time-dependent elements, so it no long can be formed as a Kronecker product of two smaller matrices. Hence, we introduce another new approach called Kronecker product approximation (KPA) in the next section to improve the efficiency of PCA for the time-dependent case.

3.2 Kronecker Product Approximation

In the time-dependent volatilities market, the covariance matrix of the asset returns is not constant anymore; it changes with the time-dependent volatilities functions and instantaneous correlation [7]. Therefore, the covariance matrix Σ_{MN} cannot bewritten exactly as a Kronecker product, which means that we cannot use the same way as we applied in constant volatilities case to reduce the computational cost, even though PCA is still feasible in determining the effective dimension. Hence, in order to improve the efficiency of PCA in the time-dependent volatilities case, we propose a new technique which is based on approximation with a Kronecker product. According to the works of Van Loan and Pitsianis, the basic purpose of Kronecker product approximation is to find two matrices $B \in \Re^{m_1 \times n_1}$ and $C \in \Re^{m_2 \times n_2}$that can minimize the Frobenius norm [9, 8]:

$$\Phi_A(B, C) = ||A - B \otimes C||_F^2, \tag{3.11}$$

where $A \in \Re^{m \times n}$ is a known matrix with $m = m_1 m_2$ and $n = n_1 n_2$. As for the time-dependent volatilities case, A is the covariance matrix Σ_{MN}.

KPA's applicability is based on the hypothesis that the principal components of the problem do not change a lot after the transformation, which implies that the effective dimensions of $B \otimes C$ is approximately the same as those of Σ_{MN}. Thus, we can covert the problem of reducing the nominal dimensions of Σ_{MN} into an easier one – selecting the effective dimensions of $B \otimes C$, which is similar to the situation we deal with in constant volatilities case so that we can decrease the computational burden significantly by using the same method.

In the following subsections, we analyze two cases to solve this nearest Kronecker product problem. The first case is under the assumption that one of matrix B and C is fixed. In the second case, we assume that neither B nor C are fixed.

3.2.1 One of Two Matrices is Fixed

In the constant volatilities case, the former matrix R in the Kronecker product $R \otimes \Sigma$ is the auto-covariance matrix of a single Brownian motion and the latter matrix Σ is the covariance matrix of asset returns. Hence, as for the time-dependent case, we consider to find two matrices that are approximations of R and Σ, respectively.

We assumed in chapter 1 that the risky assets' price behavior is characterized by the multi-dimensional Brownian motion, and we generate Brownian motion through the covariance matrix of each Brownian motion R. Based on the definition of standard Brownian motion, the elements in the auto-covariance matrix R are defined by $R_{l,m} =$

$cov(W_i(t_l), W_i(t_m)) = min(var(W_i(t_l)), var(W_i(t_m))) = min(t_l, t_m), l, m = 1, \ldots, N$, therefore, R has the following boomerang shape [7]:

$$R = \begin{pmatrix} t_1 & t_1 & \cdots & t_1 \\ t_1 & t_2 & \cdots & t_2 \\ \vdots & \vdots & \ddots & \vdots \\ t_1 & t_2 & \cdots & t_N \end{pmatrix}. \tag{3.12}$$

This matrix is constant and independent of the market features, no matter it is in the constant or time-dependent volatilities case.

Based on the above analysis, we can assume matrix B in equation (3.11) to be the auto-covariance matrix of a single brownian motion R. After we fix B, the problem to minimize equation (3.11) becomes a linear least square problem that aims to find the unknown matrix C:

$$\Phi_{\Sigma_{MN}}(R, C) = \min ||\Sigma_{MN} - R \otimes C||_F. \tag{3.13}$$

The paper of Van Loan and Pitsianis presents the theorem that specifies the solution of this least square problems [9]. That is, if $R \in \Re^{m_1 \times n_1}$ is fixed, then $C \in \Re^{m_2 \times n_2}$ that minimizes $||\Sigma_{MN} - R \otimes C||_F$ is defined as:

$$c_{ij} = \frac{tr(\hat{\Sigma}_{MN}^T R)}{tr(R^T R)}, \quad i \in [1, m_2],\ j \in [1, n_2], \tag{3.14}$$

where $tr(\cdot)$ represents the trace of the matrix, which equals to the sum of the diagonal entries of the matrix. And $\hat{\Sigma}_{MN} = \Sigma_{MN}(i : m_2 : m, j : n_2 : n)$, which is a boomerang shape block matrix. The notation $i : m_2 : m$ is used, as is done in MATLAB, to specify indices $i, i + m_2, i + 2m_2, \ldots, m$. Because of the special structure of a boomerang matrix, we can evaluate C in an efficient way by computing the denominator and numerator in equation (3.14) in following ways [7]:

$$tr(R^T R) = tr(R^2) = \sum_{j=1}^{N} (2(N-j)+1)t_j^2 \tag{3.15}$$

and

$$tr(\hat{\Sigma}_{MN}^T R) = tr(\hat{\Sigma}_{MN} R)) = \sum_{j=1}^{N} (2(N-j)+1)\sigma_{jj} b_{jj}, \tag{3.16}$$

where σ_{jj} and b_{jj} are the diagonal entries of matrix Σ_{MN} and B.

In summary, the above analysis is based on the premise that B is known and fixed, and we derive the result through the idea of least squares optimization.

3.2.2 Neither Matrices are Fixed

Our original aim is to search for the matrices B and C that minimize the Frobenius norm $||\Sigma_{MN} - B \otimes C||_F$. In the previous case, we have one matrix fixed, so the solution might not be optimal. In this section, we assume that neither matrix B nor C in equation (3.11) is fixed. In particular, as for the example of Asian options, we consider the problem of finding matrices B and C whose Kronecker product constructs an optimal approximation to Σ_{MN}:

$$\Phi_{\Sigma_{MN}}(B, C) = \min ||\Sigma_{MN} - B \otimes C||_F. \tag{3.17}$$

Recall from the previous chapter, Σ_{MN} is a block matrix with $N \times N$ blocks, and the size of each block is $M \times M$. Van Loan showed that the general method to solve this KPA can be obtained from the singular value decomposition of a permuted version of Σ_{MN} [9]. Specifically, the above equation can be rewritten as :

$$\Phi(B, C) = \min ||\tilde{\Sigma}_{MN} - vec(B) \otimes vec(C)||_F, \tag{3.18}$$

where $\tilde{\Sigma}_{MN}$ is a $N^2 \times M^2$ that is obtained by rearranging the entries of the matrix Σ_{MN} :

$$\tilde{\Sigma}_{MN} = \begin{pmatrix} vec((\Sigma_{MN})_{11})^T \\ \vdots \\ vec((\Sigma_{MN})_{n1})^T \\ \vdots \\ vec((\Sigma_{MN})_{1n})^T \\ \vdots \\ vec((\Sigma_{MN})_{nn})^T \end{pmatrix}, \tag{3.19}$$

where $vec(\cdot)$ means reshaping a matrix $X \in \Re^{m\times n}$ into a column vector $vec(X) \in \Re^{mn}$ by stacking the columns of X as follows [1]:

$$X = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{pmatrix} \Rightarrow vec(X) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}. \tag{3.20}$$

According to Van Loan's approach, we solve equation (3.17) by analyzing the singular value decomposition of $\tilde{\Sigma}_{MN}$ [8]. Suppose we know the SVD of $\tilde{\Sigma}_{MN}$, that is, $\tilde{\Sigma}_{MN} = U\Sigma V^T$. Firstly, we check the signs of the first column of U and V to make sure they are nonnegative. If they are negative, then replace U with $-U$ and replace V with $-V$. Then, the optimum solution B and C are defined by the first column of U and V, respectively. Using MATLAB colon notation this can be written as:

$$vec(B) = \sqrt{\sigma_1}U(:,1) \;\; and \;\; vec(C) = \sqrt{\sigma_1}V(:,1), \tag{3.21}$$

where σ_1 is the largest singular value of $\tilde{\Sigma}_{MN}$. Next, we reshape the $vec(B)$ into a N-by-N matrix B, and reshape $vec(C)$ into a M-by-M matrix C.

In this situation, the solutions B and C construct the optimal approximation to

Σ_{MN}, which implies that the corresponding $\Phi(B, C)$ should be less than or equal to the $\Phi(R, C)$ obtained from equation (3.13).

In summary, PCA is the general approach to reduce the nominal dimensions, and it is based on the analysis of eigenvalues of the covariance matrix (see equation (3.6)). In order to improve the efficiency of PCA, we can rewrite the covariance matrix exactly or approximately into a Kronecker produce of two smaller matrices, which is shown in equation (3.13) and (3.17). And then we reduce the computational efforts by applying the properties of Kronecker products (equation (3.10)).

Chapter 4

Numerical Experiments

In this chapter, we perform all numerical experiments mentioned in the previous parts. Our experiments involve standard Monte Carlo and Randomized Quasi Monte Carlo with Sobol sequence, which are utilized in both the constant and time-dependent volatilities cases. All computations were done using MATLAB.

Suppose we are interested in estimating the price of Asian option on a basket of $M = 10$ risky assets with strike price $K = 100$. The option will expire in $T = 1$ year and has $N = 100$ equally spaced sample points during its life time. So, the nominal dimension is $d = 1000$, which is so high that RQMC would be inapplicable. The parameters chosen for the simulation in the constant volatilities case are listed in Table 4.1.

$S_i(0)$	$=$	100
K	$=$	100
r	$=$	4%
T	$=$	1
σ_i	$=$	$10\% + \frac{i-1}{9} 40\% \quad \text{for} \ \ i = 1, \ldots, 10$
ρ_{ij}	$=$	$0 \quad \text{for} \ \ i, \ j = 1, \ldots, 10$

Table 4.1: Input parameters used in the constant volatilities case

The parameters chosen for the simulation in the time-dependent volatilities case are listed in Table 4.2.

$S_i(0)$	$=$	100
K	$=$	100
r	$=$	4%
T	$=$	1
τ_i	$=$	1.5 year for $i = 1, \ldots, 10$
$\sigma_i(0)$	$=$	$10\% + \frac{i-1}{9} 40\%$ for $i = 1, \ldots, 10$
$\sigma_i(+\infty)$	$=$	9% for $i = 1, \ldots, 10$
ρ_{ij}	$=$	0 for $i,\ j = 1, \ldots, 10$

Table 4.2: Input parameters used in the time-dependent volatilities case

We assume the time-dependent volatilities have an exponential decaying expression:

$$\sigma_i(t) = \hat{\sigma}_i(0) \exp(-\frac{t}{\tau_i}) + \sigma_i(+\infty), \tag{4.1}$$

where $\sigma_i(0)$ is the initial volatility for the i-th asset, $\sigma_i(+\infty)$ represents its asymptotic volatility and τ_i its decay constant, and $\hat{\sigma}_i(0) = \sigma_i(0) - \sigma_i(+\infty)$. Based on this function, we can derive the solution for each element in covariance matrix Σ_{MN}:

$$\begin{aligned} ((\Sigma_{MN})_{ik})_{lm} &= \int_0^{min(t_l, t_m)} \rho_{ik} \sigma_i(t) \sigma_k(t) dt \\ &= \hat{\sigma}_i(0) \hat{\sigma}_k(0) \tau_{ik} \left(1 - \exp\left(\frac{t}{\tau_{ik}}\right)\right) + \hat{\sigma}_i(0) \sigma_k(+\infty) \tau_i \left(1 - \exp\left(\frac{t}{\tau_i}\right)\right) + \\ &\quad \hat{\sigma}_k(0) \sigma_i(+\infty) \tau_k \left(1 - \exp\left(\frac{t}{\tau_k}\right)\right) + \sigma_i(+\infty) \sigma_k(+\infty) t, \end{aligned}$$

where $\tau_{ik} = \tau_i \tau_k / (\tau_i) + \tau_k$.

This chapter presents two numerical tests, including a computation time comparison among PCA, KP and KPA, and estimations of Asian options price by RQMC and standard MC. For each experiment, we state its goal first, and describe the steps of simulation procedure, then provide illustrations of the results, and finally analyze the results.

4.1 Computational Time Comparison

This numerical test is aimed to compare the computational time among different dimension reduction techniques, including PCA, KP and KPA. We conduct the tests in the constant and time-dependent volatilities cases, respectively.

In the constant volatilities case, we compare the computational time used by PCA and KP, while in the time-dependent volatilities situation, we make the computational time comparison between PCA and KPA. We consider two cases for KPA, the first case is under the assumption that one of two smaller matrices is fixed. In the second case, we assume that neither of them are known.

We implement the same steps for two tests. We set the ratio p in expression (3.5) from 0.9 to 0.99 with a 0.01 increment. And for each ratio p, we compute the average computational time for 10 separate runs. The illustration of the computational time comparison between PCA and KP in the constant volatilities case is shown in Figure 4.1. Figure 4.2 illustrates the computational time comparison among PCA and two cases of KPA in the time-dependent volatilities case.

In Figure 4.1, PCA needs a time almost 10 times higher than KP for all selected p. And in Figure 4.2, both methods of KPA use less time than PCA to select the effective dimensions. This is because both KP and KPA convert the computation of eigenvalues of an $MN \times MN$ matrix into the analysis of two smaller $M \times M$ and $N \times N$ matrices, and then significantly reduce the computational effort from $O((MN)^3)$ to $O(M^3 + N^3)$.

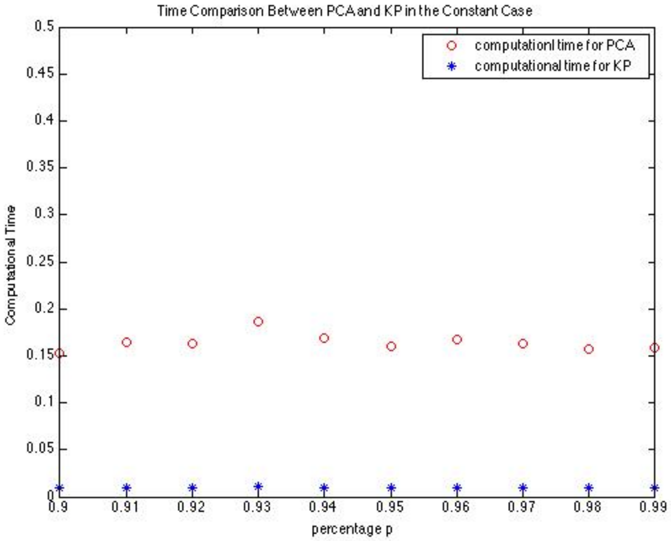

Figure 4.1: Computational time comparison between PCA and KP in the constant volatilities case.

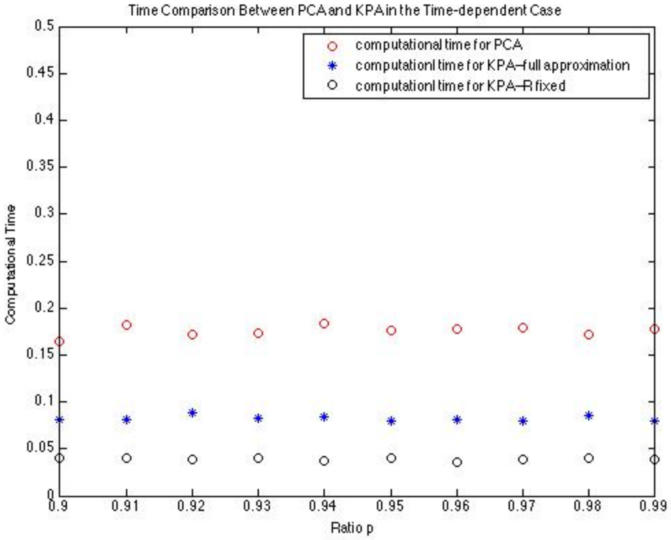

Figure 4.2: Computational time comparison between PCA and KPA in the time-dependent volatilities case.

4.2 Estimation of Option Price

In this test, we estimate the price of an Asian option by MC and RQMC at time $t_0 = 0$, which is the initial price of the Asian option. In addition, we also present a comparison between the precision of standard MC and RQMC.

The test consists of the following steps.

- First, we generate point sets for MC simulations. We use MATLAB's `rand` and `sobolset` function to generate the point set for standard MC and RQMC, respectively. In order to extend the applicability of RQMC, we randomize the points based on the idea in equation (2.9).

- Next, we define the $MN \times MN$ covariance matrix Σ_{MN} in equation (1.6) and (1.8) according to our chosen parameters. For high dimensional problems, we need to apply the dimension reduction techniques (such as PCA, KP and KPA) to select the effective dimensions for RQMC first, and then regenerate the new covariance matrix based on expression (3.9).

- Then, we apply MATLAB's `norminv` function to compute the inverse of the cumulative distribution of the normal random vector Z, $F_Z^{-1}(\mathbf{u})$, with corresponding mean zero and covariance matrix Σ_{MN}.

- Finally, we plug $F_Z^{-1}(\mathbf{u})$ into the function (1.12) and use the Monte Carlo method to estimate the Asian option's price and the RMSEs.

In this test, we set the number of simulations $n = 1000$ and 10 replications. The final result is obtained from the average of these 10 replications. Table 4.3 and 4.4 present the estimated prices and RMSEs coming from standard MC and RQMC in the constant volatilities and the time-dependent volatilities case, respectively.

Constant Volatilities Case	**Standard MC**	**RQMC**
Estimated Price	2.0877	2.0704
RMSE	0.0264	0.0104

Table 4.3: Option prices and RMSEs for the constant volatilities case

Time-dependent Volatilities Case	**Standard MC**	**RQMC**
Estimated Price	2.1193	2.1109
RMSE	0.0234	0.0133

Table 4.4: Option prices and RMSEs for the time-dependent volatilities case

All of the computed prices in the above tables are statistically consistent, but present a different accuracy. In both cases, the RQMC simulation provides smaller RMSEs and shows a better accuracy than the standard MC simulation. This is due to the fact that RQMC works with well-chosen points, which helps RQMC overcome the probabilisitic problems that MC could face and result in a lower convergence rate. Hence, RQMC requires fewer simulations to obtain a similar integration level as MC does, and therefore, when they use the same number of simulations, RQMC will provide a more accurate integration level.

Chapter 5

Conclusions

In this thesis, we have employed two integration methods to estimate an Asian option's price – standard MC and RQMC. Both methods estimate the integration $I = \int_{[0,1]^d} f(\mathbf{x})d\mathbf{x}$ by drawing n independent samples $\mathbf{x}_i$ from $[0,1]^d$ and then computing the arithmetic average of $f(\mathbf{x}_i)$, the only difference is they generate the set of points in different ways. MC generates points from a pseudorandom sequence with a convergence rate $O\left(\frac{1}{\sqrt{n}}\right)$,while RQMC uses a low-discrepancy sequence with a lower convergence rate $O\left(\frac{(\ln n)^d}{n}\right)$. However, RQMC's convergence rate is dependent of the nominal dimension d, which would limit the performance of RQMC in estimating high-dimensional integrals. Therefore, in order to extend RQMC's superiority in high-dimensional problems, we have applied several dimension reduction techniques, including PCA, KP (for the constant volatilities case) and KPA (for the time-dependent volatilities case), to select the effective dimensions and then reduce the nominal dimensions. We have conducted several numerical experiments to illustrate and compare the efficiency and accuracy among the dimension reduction techniques and the two integration methods.

We first made a comparison of the computational time among different dimension reduction techniques. The results are shown in Figure 4.1 and Figure 4.2. KPA and KP require less time than the straightforward PCA since both of them utilize the properties of the Kronecker product, which significantly reduce their computational burden. Hence, from the view of computational cost, KPA with fixed R and KP are the most efficient dimension reduction techniques for the time-dependent volatilities case and constant volatilities case, respectively.

In addition, we conducted a numerical test in pricing high-dimensional Asian options with multi-assets in a Black-Scholes model by standard MC and RQMC. The results are shown in Table 4.3 and Table 4.4. RQMC provides smaller RMSEs in both constant and time-dependent volatilities cases, which confirms the fact that RQMC can make more precise approximation than the standard MC in estimating numerical integrals.

Future work on the topics discussed in this thesis could include investigation of alternative sampling methods, such as Latin supercube sampling (LSS), to randomize the points for RQMC. Previous work [6] has shown that it can improve the accuracy of the numerical integration for high dimensions.

Bibliography

[1] J. Kamm and J. G. Nagy. Optimal Kronecker product approximation of block toeplitz matrices. *SIAM Journal on Matrix Analysis and Applications*, 22:155–172, 2000.

[2] I. Krykova. Evaluation of path-dependent securities with low discrepancy methods. Master's thesis, Worcester Polytechnic Institute, Worcester, MA, 2003.

[3] T. Lai and H. Xing. *Statistical Models and Methods for Financial Markets*. Springer New York, New York, NY, 2008.

[4] S. N. Neftci. *An Introduction to the Mathematics of Financial Derivatives*. Academic Press, San Diego, CA, 2000.

[5] D. P. O'Leary. *Scientific Computing with Case Studies*. SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[6] P. Sabino. Monte Carlo methods and path-generation techniques for pricing multi-asset path-dependent options. Technical Report 36/07, Dipartimento di Matematica, Universitá degli studi di Bari, 2007. Code arXiv:0710.0850v1.

[7] P. Sabino. Efficient quasi-Monte simulations for pricing high-dimensional path-dependent options. *Decisions in Economics and Finance*, 32:49–63, 2009.

[8] C. F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123:85–100, 2000.

[9] C. F. Van Loan and N. Pitsianis. Approximation with Kronecker products. In M. S. Moonen, G. H. Golub, and B. L. de Moor, editors, *Linear Algebra for Large Scale and Real-Time Applications*, pages 293–314. Kluwer Academic Publisher, Dordrecht, 1993.