RELIABLE QUASI-MONTE CARLO

WITH CONTROL VARIATES

BY

DA LI

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Applied Mathematics
in the Graduate College of the
Illinois Institute of Technology

Approved ______________________
Advisor

Chicago, Illinois
May 2016

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
$\mathbb{N}$	Positive Integers
$\mathbb{N}_0$	Nonnegative Integers
$\mathbb{Z}$	Integers
$\mathbb{R}$	Real numbers
$\oplus$	Digital addition

ABSTRACT

Recently Quasi Monte Carlo (QMC) methods have been implemented in a guaranteed adaptive algorithm. This raises the possibility of combining adaptive QMC with efficiency improvement techniques for IID Monte Carlo (MC) such as control variates.

The challenge for adding control variates to QMC is that optimal control variate coefficient for QMC is generally not the same as that for MC. Here we propose a method for computing the optimal control variate coefficients with a guaranteed adaptive QMC algorithm. One merit of control variates is that it is theoretically no worse than using no control variates. Our method is implemented in an efficient way so that the extra cost for control variates is not significant.

Our new adaptive QMC algorithm with control variates is illustrated by two financial problems. One is pricing an arithmetic mean Asian option with geometric mean Asian option as control variates and the other is barrier option with European option as control variates. Our results show that with good control variates, the cost of adaptive QMC is greatly reduced compared to vanilla QMC.

CHAPTER 1

INTRODUCTION

1.1 What are we going to do?

Recently there are some great results from construction of Quasi Monte Carlo (QMC) methods that can adaptively choose a sample size for given error tolerances [1]. Our work is trying to combine reliable QMC methods with control variates. We will justify the theory behind it, construct a practical algorithm which can be implemented and tested through high dimensional integration examples.

1.2 Why this is a good idea?

Control Variates (CV) is a variance reduction technique for IID MC methods. QMC can be viewed as a deterministic version of IID MC, which outperforms MC for many integrals [2]. Naturally we wonder if QMC can also benefit from the CV technique. If that is possible, it can be especially useful for problems where we can easily find good control variates.

1.3 What's the challenge?

The challenge is that the optimal control variate coefficient for QMC is generally not the same as for simple Monte Carlo, as explained by Hickernell, Lemieux, and Owen [3]. This requires us to figure out a right way to get the optimal coefficients for control variates with Quasi-Monte Carlo.

1.4 Outline

In chapter 2 we first will briefly introduce Quasi-Monte Carlo rule and it's difference between Monte-Carlo. Then we will briefly talk about digital sequence and layout several concepts which will be used later. Chapter 3 will show the derivations and theories of our methods along with the corresponding algorithm. In chapter 4 we will demonstrate results from several numerical experiments. We choose several option pricing problems for our target. For the final chapter we will discuss the results and future extension of the method.

CHAPTER 2

BACKGROUND

2.1 Problem Setup

Numerical integration problems are involved in fields such as physics, mathematical finance, biology, computer graphics, and many others fields. It usually happens when it is hard to solve some integral analytically. Therefore, one has to use numerical methods for such problems. MC method is the general way to solve problems in such case [4]. The method can be simply explained in the following way.

Suppose we have the following standard integration approximation problem whose format is:

$$I = \int_{[0,1)^d} f(x)dx. \tag{2.1}$$

Then we take sample of n points $\{\mathbf{x_0}, \ldots, \mathbf{x_n}\} \in [0,1)^d$ follow the uniform distribuation randomly, and construct the following MC estimator:

$$\hat{I}(f) = \frac{1}{n}\sum_{i=1}^{n} f(X_i).$$

However, there are several problems with IID MC method [5]. First, it is difficult to generate truly random samples. Second, error bound for IID MC works only probabilistic sense. Second, in many applications the convergence rate of MC is considered not fast enough.

Hence, QMC method were introduced to address these problems. For QMC method the estimator is almost the same with MC. The difference is that the sample points are taken from low discrepency sequence, which is determinsticly chosen instead of random. We will briefly rewiew one method for constructing such sequence that we used for our application in the next section.

2.2 Sobol Sequence

2.3 Control Variates

CV is a well known variance reduction technique used in MC simulation. It is ofen used when a 'simpler' version of the origin problem can be solved explicitly. In this section we briefly review the ideas and main results of the method.

Suppose we want to solve the integration problem (2.1) showed earlier, now we have a known function h and its value on the interval $\int_{[0,1)^d} h(x)dx = \theta$. We then construct a new estimator as the following

$$\hat{I}_{\mathrm{CV}}(f) = \frac{1}{n}\sum_{i=1}^{n}\Big[f(X_i) - \beta_{\mathrm{MC}}[h(X_i)-\theta]\Big] \quad s.t.\ X_i \sim \mathcal{U}[0,1),\ i.i.d.$$

We can easily see it's an unbiased estimator, i.e. $\mathbb{E}(\hat{I}_{\mathrm{CV}}) = I$. Now the question is how should we give β_{MC} and why is that. The idea is rather straitforward. We know the mean square error of MC estimator is $\mathrm{Var}(\hat{I}) + \mathrm{Bias}(\hat{I}^2$ []. CV method aims at efficiency improvment, so we need to reduce mean square error. Since the estimator is unbiased, we only need to minimize its variance. Hence, the optimal β_{MC} should be the one that minmize the variance of esimator. Here we give a simple derivation of optimal β_{MC} for single CV. First, the variance of $\hat{I}_{\mathrm{CV}}$ is

$$\begin{aligned}
\mathrm{Var}(\hat{I}_{\mathrm{CV}}) =& \mathrm{Var}\Big(\frac{1}{n}\sum_{i=1}^{n}\big[f(X_i) - \beta_{\mathrm{MC}}[h(X_i)-\theta]\big]\Big)\\
=&\frac{1}{n}\mathrm{Var}\Big(f(X_i) - \beta_{\mathrm{MC}}[h(X_i)-\theta]\Big) \quad \text{by } X_i \text{ i.i.d}\\
=&\frac{1}{n}\mathbb{E}\Big(\big[f(X_i) - \beta_{\mathrm{MC}}[h(X_i)-\theta] - I\big]^2\Big)\\
=&\frac{1}{n}\mathbb{E}\Big(\big[[f(X_i) - I] - \beta_{\mathrm{MC}}[h(X_i)-\theta]\big]^2\Big)\\
=&\frac{1}{n}\mathbb{E}\big([f(X_i)-I]^2 - 2\beta_{\mathrm{MC}}[f(X_i)-I][h(X_i)-\theta] + \beta_{\mathrm{MC}}^2[h(X_i)-\theta]^2\big)\\
=&\frac{1}{n}\Big(\mathrm{Var}[f(X_i)] - 2\beta_{\mathrm{MC}}\mathrm{Cov}[f(X_i),h(X_i)] + \beta_{\mathrm{MC}}^2\mathrm{Var}[h(X_i)]\Big)\\
=&\frac{1}{n}\Big(\mathrm{Var}[h(X_i)](\beta_{\mathrm{MC}} - \frac{\mathrm{Cov}[f(X_i),h(X_i)]}{\mathrm{Var}[h(X_i)]})^2 + \\
&\qquad \mathrm{Var}[f(X_i] - \frac{\mathrm{Cov}^2[f(X_i),h(X_i)]}{\mathrm{Var}[h(X_i)]}\Big)
\end{aligned}$$

, then the optimal β_{MC} is given by

$$\beta^*_{\text{MC}} = \frac{\text{Cov}[f(X_i), h(X_i)]}{\text{Var}[h(X_i)]} \tag{2.2}$$

. In this case the variance become

$$\text{Var}(\hat{I}_{\text{CV}}) = \frac{\text{Var}[f(X_i)]}{n}\big(1 - \text{corr}^2[f(X_i), h(X_i)]\big)$$

, and note we always have

$$\text{Var}(\hat{I}_{\text{CV}}) \leq \frac{\text{Var}[f(X_i)]}{n} = \text{Var}(\hat{I})$$

Now we can see the merit of control variates as a variance reduction method. In the worst case, we get a completely uncorrelated g that leads correlation to zero, and we have variance exactly the same as not using control variates. On the other hand, the more correlated our control variates is to the target function, the more variance we can get rid of by using the method.

2.4 Reliable Adaptive QMC with digital sequence

2.4.1 Idea of adaptive cubature algorithm.

One practical problem for QMC method is that how to get the sample size big enough for a required error tolerance. The idea in work of Hickernell and Jimnez Rugama(2014) [1] is to construct a QMC algorithm with reliable error estimation on digital sequence. Here we briefly summarize their results.

The error of QMC method on digital sequence can be expressed in terms of

Walsh coefficients of the integrand on certain cone conditions.

$$\text{if } f \in \mathscr{C} \text{ then } \left| \int_{[0,1)^d} f(x)dx - \hat{I}_m(f) \right| \leq a(r,m) \sum_{\lfloor 2^{m-r-1} \rfloor}^{2^{m-r}-1} |\tilde{f}_{m,k}| \tag{2.3}$$

$$\hat{I}_m(f) := \frac{1}{b^m} \sum_{i=0}^{b^m-1} f(z_i \oplus \Delta)$$

$$\tilde{f}_{m,k} = \text{ discrete Walsh coefficients of } f$$

$$a(r,m) = \text{ inflation factor that depends on } \mathcal{C}$$

.

Here is the defination of the cone condition.

$$\mathscr{C} := \left\{ f \in L^2[0,1)^d : \bigcirc \leq \hat{\omega}(m-l)\Diamond,\ l \leq m; \quad \Diamond \leq \mathring{\omega}(m-l)\Box,\ l^* \leq l \leq m \right\}$$

$$\bigcirc := \sum_{\kappa=\lfloor b^{l-1} \rfloor}^{b^l-1} \sum_{\lambda=1}^{\infty} |\hat{f}_{\kappa+\lambda b^m}|, \quad \Box := \sum_{\kappa=b^{l-1}}^{b^l-1} |\hat{f}_\kappa|, \quad \Diamond := \sum_{\kappa=b^m}^{\infty} |\hat{f}_\kappa| \tag{2.4}$$

$$l^* \in \mathbb{N} \text{ be fixed }; \forall m \in \mathbb{N}, \hat{\omega}(m), \mathring{\omega}(m) \geq 0, \text{ and } \lim_{m\to\infty} \hat{\omega}(m) = 0, \ \lim_{m\to\infty} \mathring{\omega}(m) = 0$$

The first inequality($\bigcirc \leq \Diamond$) means the sum of the larger indexed Walsh coefficients bounds a partial sum of the same coeffcients. Take $l = 0, m = 12$ for example, in Figure 2.1 the the sum of circles should be bounded by some factor times the sum of diamonds. The second inequality($\Diamond \leq \Box$) requires the sum of the larger Walsh coefficients be bounded by the sum of smaller indexed Walsh coefficients. Take $l = 8$ at this time, which means in Figure 2.1 the sum of diamonds should be bounded by some relax factor times the squares.

The cone give some meanings for the functions about how they should behave to get the err bound formula (2.3). This means that $|\hat{f}_\kappa|$ does not dramatically bounce back as κ goes to infinity. Note that in Figure 2.1 we call circles the err bound, this is proven to be true and under the cone conditions we can estimate it using discrete Walsh coefficients instead of true Walsh coefficients.

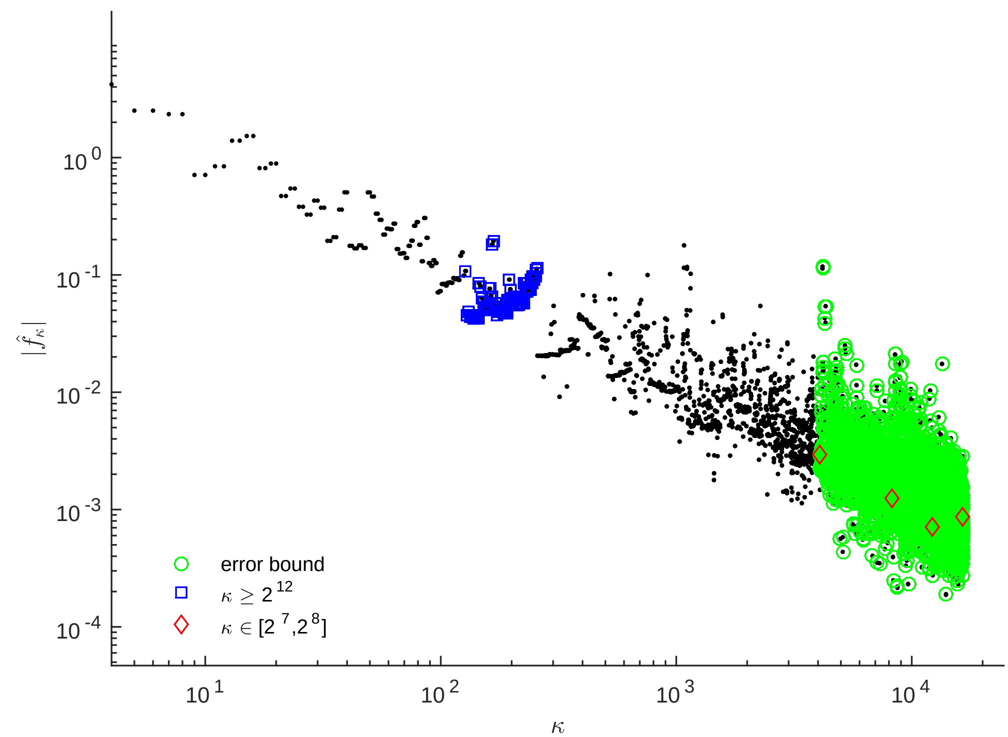

Figure 2.1. Cone condition for reliable adpative QMC algorithm

CHAPTER 3

RELIABLE ADAPTIVE QMC SOBOL WITH CV

3.1 Idea to add control variates to QMC with Digital Sequence

The idea is similar to traditional control variates technique for Monte-Carlo. If we know the integration of a function $\boldsymbol{h} = (h_1, \ldots, h_J)$ on the interval same as our f, say $\int_{[0,1)^d} h_j dx = \theta_j$, then we can define a new function g

$$\begin{aligned} g &:= f - (\boldsymbol{h} - \boldsymbol{\theta})\boldsymbol{\beta} \qquad (3.1) \\ \text{s.t. } \boldsymbol{\theta} &= (\theta_1, \ldots, \theta_J), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^T \end{aligned}$$

Then easily we can find that if we replace f with g, the integration stays the same.

$$\int_{[0,1)^d} g dx = \int_{[0,1)^d} f - (\boldsymbol{h} - \boldsymbol{\theta})\boldsymbol{\beta} dx = \int_{[0,1)^d} f dx$$

Now we are wondering if we can still use the same method as MC, the answer is no and the reason is in the next section.

3.2 The problem of C.V. with QMC

As we pointed out earlier, QMC use a different way for generating X_i, they are still identical(from same distribution) but not independent anymore, which caused the problem for control variates.

Suppose $X_1, \ldots, X_n$ are generated by QMC rule, the estimator stays the same

$$\hat{I}_{cv}(f) = \frac{1}{n} \sum_{i=1}^{n} \Big[f(X_i) - \beta_{qmc}[h(X_i) - \theta] \Big] \quad X_i \in \mathcal{U}(0,1)$$

We can easily prove it is still unbiased

$$\mathbb{E}(\hat{I}_{cv}) = \mathbb{E}\Big(\frac{1}{n} \sum_{i=1}^{n} \Big[f(X_i) - \beta_{mc}[h(X_i) - \theta] \Big] \Big) = I$$

However, it's not the same case as MC like we presented before, because we do not have i.i.d for X_i this time

$$\text{Var}_{qmc}(\hat{I}_{cv}) \neq \frac{1}{n}\text{Var}\Big(f(X_i) - \beta_{mc}[h(X_i) - \theta]\Big)$$

Instead the variance become

$$\begin{aligned}
\text{Var}\hat{I}_{cv}) &= \text{Var}\Big(\hat{I} - \beta_{qmc}\hat{H}\Big) \quad s.t.\ \hat{I} = \sum_{i=1}^{n} f(X_i),\ \hat{H} = \sum_{i=1}^{n}[h(X_i) - \theta] \\
&= \text{Var}(\hat{I}) - 2\beta_{qmc}\text{Cov}(\hat{I}, \hat{H}) + \beta_{qmc}^2\text{Var}(\hat{H}) \\
&= \text{Var}(\hat{H})\Big(\beta_{qmc} - \frac{\text{Cov}(\hat{I}, \hat{H})}{\text{Var}(\hat{H})}\Big)^2 + \text{Var}(\hat{I}) - \frac{\text{Cov}(\hat{I}, \hat{H})}{\text{Var}(\hat{H})}
\end{aligned}$$

The optimal β_{qmc} is

$$\beta_{qmc}^* = \text{Var}(\hat{H})^{-1}\text{Cov}(\hat{I}, \hat{H}) \tag{3.2}$$

which leave the variance to be

$$\text{Var}_{qmc}(\hat{I}_{cv}) = \text{Var}(\hat{I})\big(1 - \text{corr}^2[\hat{I}, \hat{H}]\big)$$

Now we are interested that if our previous formula for $\hat{\beta}_{mc}$ could be an estimation for $\hat{\beta}_{qmc}$. The fact is that they are generally not the same. Let's take the covariance part of formula (3.2) and (2.2) to see the difference.

$$\begin{aligned}
\text{Cov}(\hat{I}, \hat{H}) &= \int [f(X_1) + \cdots + f(X_n)][h(X_1) + \cdots + h(X_n)]\ d\mathbf{X} \\
&= \int [\sum_{i=1}^{n} f(X_i)h(X_i) + \sum_{i,j=1}^{i \neq j} f(X_i)h(X_j)]\ d\mathbf{X} \\
&\neq \int f(X_i)h(X_i)dX_i \\
&=\text{Cov}x[(f(X_i), h(X_i)]
\end{aligned}$$

There is also a very good example from Hicknell and Lemieux(2005) [3]'s paper, showing that β_{mc} and β_{qmc} can make a quite different results in some cases.

3.3 A new way to find β

As we stated in previous section, we can not find optimal β by minimizing variance of estimator like Monte-Carlo. However, if using the guaranteed adaptive QMC method introduced in chapter 2, we may have another way to find β.

Recall equation (2.3) the error bound for new estimator of g still holds

$$\Big|\int_{[0,1)^d} g dx - \hat{I}_m(g)\Big| \le a(r,m) \sum_{\lfloor 2^{m-r-1}\rfloor}^{2^{m-r}-1} |\tilde{g}_{m,k}| \tag{3.3}$$

Naturally, the new estimator become

$$\hat{I}_m(g) := \frac{1}{b^m} \sum_{i=0}^{b^m-1} g(z_i + \Delta) \tag{3.4}$$

From (3.3) it is clear that the optimal β is the one that minimize the error term.

$$\begin{aligned}
\boldsymbol{\beta}^* &= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{g}_\kappa| && && (3.5)\\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \hat{\boldsymbol{\theta}})\boldsymbol{\beta}| && \hat{\boldsymbol{h}}_\kappa = (\hat{h}_{\kappa,1}, \dots, \hat{h}_{\kappa,J}), \hat{\boldsymbol{\theta}} = (\hat{\theta}_{\kappa,1}, \dots, \hat{\theta}_{\kappa,J}) && (3.6)\\
&= \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_1 && \hat{\boldsymbol{f}} = (\hat{f}_{b^{m-r-1}}, \dots, \hat{f}_{b^{m-r}-1)})^T && (3.7)\\
&\approx \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_2 && \hat{\boldsymbol{H}} = (\hat{\boldsymbol{H}}_1, \dots, \hat{\boldsymbol{H}}_J) && (3.8)\\
& && \hat{\boldsymbol{H}}_j = (\hat{h}_{b^{m-r-1},j} - \hat{\theta}_j, \dots, \hat{h}_{b^{m-r}-1,j} - \hat{\theta}_j)^T
\end{aligned}$$

The second equivalence is not hard to get, but the third one may not be so obvious. Let's consider it backwards. Suppose we have a vector A and it's $\mathcal{L}_1$-norm.

$$A = \begin{pmatrix} f_1 - z_1 \\ f_2 - z_2 \\ \dots \\ f_n - z_n \end{pmatrix}, \quad \|A\|_1 = \sum_{i=1}^n |f_i - z_i|, \quad z_i := (\boldsymbol{h}_i - \boldsymbol{\theta})$$

If we replace the index,A is exactly what's inside the $\mathcal{L}_1$-norm in (3.7). Hence we justified the third equivalence. The reason we use an approximation instead, i.e. the $\mathcal{L}_1$-norm, is because there is no efficient way to solve it compared to existing least square methods.

3.4 The problem with θ

We noticed a problem in solution for optimal β, which is we do a lot of subtractions with θ. This could be a large cost when we have difficult functions which means b^{m-r} could be very large number. Therefore we present a way to avoid that part.

The idea is form a observation that Walsh transform of θ in (**??**) is actually zero, since $\hat{h}_\theta = \theta\delta_{\kappa,0}$ and the summation is not start from $\kappa = 0$.

This simplifies (3.6) to the following. Note that we only need the information of function f and h to calculate β^*, θ has been get rid of the optimization process.

$$\begin{aligned}
\boldsymbol{\beta}^* &= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \hat{\boldsymbol{\theta}})\boldsymbol{\beta}| && (3.9)\\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \boldsymbol{\theta}\delta_{\kappa,0})\boldsymbol{\beta}| \\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - \hat{\boldsymbol{h}}_\kappa\boldsymbol{\beta}| && \hat{\boldsymbol{f}} = (\hat{f}_{b^{m-r-1}}, \dots, \hat{f}_{b^{m-r}-1)})^T \\
&= \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_1 && \hat{\boldsymbol{H}} = (\hat{\boldsymbol{H}}_1, \dots, \hat{\boldsymbol{H}}_J) \\
&\approx \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_2 && \hat{\boldsymbol{H}}_j = (\hat{h}_{b^{m-r-1},j}, \dots, \hat{h}_{b^{m-r}-1,j})^T
\end{aligned}$$

The same problem happened with the estimator (3.4). We have the similar solution for that.

$$\begin{aligned}
\hat{I}_m(g) &= \frac{1}{b^m}\sum_{i=0}^{b^m-1} g(z_i+\Delta) \\
&= \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(z_i+\Delta) - (\boldsymbol{h}(z_i+\Delta) - \boldsymbol{\theta})\boldsymbol{\beta} \\
&= \frac{1}{b^m}\sum_{i=0}^{b^m-1} [f(z_i+\Delta) - \boldsymbol{h}(z_i+\Delta)\boldsymbol{\beta}] + \boldsymbol{\theta}\boldsymbol{\beta} \qquad (3.10)
\end{aligned}$$

After organize it the in format of (3.10), θ is eliminated from the summation part. From these two parts of work on θ we managed to save $\frac{b-1}{b}b^{m-r}+b^m$ operations of subtraction.

3.5 The modified method

Now we make the following changes

$$g = f - \beta h$$

$$\hat{I}_m(g) = \frac{1}{b^m}\sum_{i=0}^{b^m-1} g(z_i+\Delta)$$

And we have the following equivalence

$$\int_{[0,1)^d} f dx = \int_{[0,1)^d} g dx + \theta\beta$$

$$\hat{I}_m(f) = \hat{I}_m(g) + \theta\beta$$

So the estimation error becomes

$$\Big|\int_{[0,1)^d} f dx - \hat{I}_m(f)\Big| = \Big|\int_{[0,1)^d} g dx - \hat{I}_m(g)\Big|$$

Here if our g is in the cone we introduced earlier (2.4), then we can use the results from Hickernell and Jimnez Rugama(2014) [1], the error is bounded by

$$\Big|\int_{[0,1)^d} g dx - \hat{I}_m(g)\Big| \leq a(r,m)\sum_{\lfloor 2^{m-r-1}\rfloor}^{2^{m-r}-1} |\tilde{g}_{m,k}|$$

This leads to the same algorithm suggested by Hickernell and Jimnez Rugama(2014) [1], but since we are using control variates, several modifications have to be made.

3.6 The Algorithm

We now give the algorithm for reliable adaptive QMC with control variates using digital sequence.

Algorithm 1: Reliable Adaptive QMC with control variates

Data: function f and $\boldsymbol{H}$; value of $\int_{[0,1)^d} h_j dx = \theta_j$; tolerance ε

Result: $\hat{I}(f)$; samples size; optimal β

begin

1. $m, r =$ start numbers, $x = 2^m$ sobolset points
2. get kappa map($\tilde{\kappa}$) and Walsh coefficients($\tilde{f}, \tilde{\boldsymbol{H}}$) using algorithm 2
3. $\boldsymbol{\beta} = \tilde{H}\{\tilde{\kappa}[x(a:b)]\}\backslash\tilde{f}\{\tilde{\kappa}[x(a:b)]\}, (a:b) = (2^{m-r-1} : 2^{m-r} - 1)$
4. $g = f - \boldsymbol{H\beta}$, repeat step 2 on g
5. $\tilde{S}_{m-r,m}(g) = \sum_a^b \left|\tilde{g}\{\tilde{\kappa}[x(a:b)]\}\right|$
6. check whether g is in the cone
7. **if** $a(m,r)\tilde{S}_{m-r,m}(g) \leq \varepsilon$ **then**
    - return $\hat{I}_m(g) = \sum_{i=0}^{2^m-1} f[x(i)] + \boldsymbol{\theta\beta}$
    - return $\boldsymbol{\beta}, n = 2^m$
8. **for** $m = m+1 : mmax$ **do**
    - $xnext =$ next 2^{m-1} sobolset points
    - repeat step 2 on $[x, xnext]$
    - repeat step 5, 6, 7

Note that for generating kappa map, i.e. step **??**in Algorithm 1, we used an explicit way to generate it.

Another importent point need to be mentioned is that in our algorithm, we used an iterative way, which may require recalculating *beta* each time m increment.

Algorithm 2: kappa map and discrete Walsh coeffcients

Data: function f; $Y_v^{(m)}$; $m \in \mathbb{N}_0$

Result: $\tilde{\kappa}$; $\tilde{S}_{m-r,m}(f)$

begin

- **if** $m = 0$ **then**
  - $\mathring{\mathbf{v}}(0) = 0$
- **if** $m \geq 1$ **then**
  - **for** $m : 1 : -1$ **do**
    - $\mathring{\mathbf{v}}_m(\mathbf{k}) = \mathring{\mathbf{v}}_m - 1(\mathbf{k})$
    - $\mathring{\mathbf{v}}_m(\mathbf{k}) = \mathbf{k},\ \mathbf{k} = b^{m-1}, \ldots, b^m - 1$
- **for** $l = m - 1 : max(1, m - r) : -1$ **do**
  - **for** $k = 1 : b^l - 1$ **do**
    - $\forall a \in \mathbb{F}_b$, find a s.t. $|\, Y^{(m)}_{\mathring{\mathbf{v}}(k+a^* b^l)}| \geq |\, Y^{(m)}_{\mathring{\mathbf{v}}(k+ab^l)}|$

CHAPTER 4

NUMERICAL EXPERIMENT

Option Pricing has always been a challenging topic in financial mathematics. Although there are several other methods for pricing options, Monte Carlo performs better when solving high dimension problems. In this chapter we makes several tests our reliable QMC with CV algorithm with option pricing problems. Note all our experiments are implemented under brownian motion (GBM) pricing model on non dividend paying stock. Since the GBM model is a well known model for option pricing, we only lay out the parameters for option formulas we use later and not dig into the model itself.

$$
\begin{aligned}
S(jT/d) &= \text{current asset price at time } jT/d,\ j = 1, \dots, d \\
K &= \text{strike price} \\
T &= \text{expiration time} \\
\sigma &= \text{volatility} \\
r &= \text{interest rate} \\
d &= \text{number of time steps.}
\end{aligned}
$$

4.1 Accuracy

The first thing we want to test is whether our algorithm provides the 'accurate' solution. This means if the function satisfies the cone condition, the difference between our estimation and true value should be bounded by the pre-defined error tolerance. We will test our algorithm for accuracy in three cases: without CV, single CV and multiply CV.

To do this we have to know the exact value of our integral to calculate the exact error of our results. Therefore, we choose geomatric mean Asian option as our target function because they have exact solution for it under GBM model. We choose to test call options, so for geomatric mean Asian call option the payoff function is:

$$C_T^{\text{gmean}} = \max\Big(\Big[\prod_{j=1}^{d} S(jT/d)\Big]^{\frac{1}{d}} - K, 0\Big)e^{-rT}.$$

The closed formula for exact price of it under GBM model is:

$$\begin{aligned} C_T^{\text{gmeanExact}} &= S(0)e^{-(r+\sigma^2/2)T/2}\Phi(\tilde{d}_1) + Ke^{-rT}\Phi(\tilde{d}_2) \\ \tilde{d}_1 &= \frac{\ln(S(0)/K) + (r + \frac{n-1}{6(n+1)}\sigma^2/2)T/2}{\sqrt{\frac{2n+1}{6(n+1)}}\sigma\sqrt{T}} \\ \tilde{d}_2 &= \frac{\ln(S(0)/K) + (r - \sigma^2/2)T/2}{\sqrt{\frac{2n+1}{6(n+1)}}\sigma\sqrt{T}}. \end{aligned} \tag{4.1}$$

Then we need a couple control variates of which the exact solution must have clear formulas. Here we pick European call option and stock price at expiration as control variates. For European call option, the payoff function is:

$$C_T = \max(S(T) - K, 0)e^{-rT}.$$

The exact price of it under GBM model is given by:

$$\begin{aligned} C_T^{\text{euroExact}} &= S(0)\Phi(d_1) - Ke^{-rT}\Phi(d_2) \\ d_1 &= \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 &= \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}. \end{aligned}$$

For stock price, the payoff function is:

$$C_T = S(T)e^{-rT},$$

while exact price of it under GBM model is just $S(0)$.

The following is the set up for all options in the accuracy test: . We use in the money call option for both Asian and European options. They are monitored weekly for one month period with 1% interest and 50% volatility anunally.

Table 4.1. Parameter Setup for accuracy test

S0	K	TimeVector	r	volatility
120	100	1/52:1/52:4/52	0.01	0.5

The test is then perform with three different absolute error tolerances as shown in table 4.1. First we caculated the exact price for the geomatric mean Asian call option using formula (4.1). Then we run the same parameters through our reliable QMC algorithm to get the estimates. Finally we caluate the absolute value of difference between estimates and true prices as errors. We used three different absolute error tolerance set ups for this: $10^{-2}, 10^{-3}, 10^{-6}$. From the results we can see all errors lied within the preset tolerance. This give us confidence that our algorithm is at least accurate as guranteed.Note for second test, which the tolerance is 10^{-3}, while the error seems all within a lesser bound 10^{-4}. We believe this is not a error for the algorithm or the code. It happened since each iteration our sample size is doubled and this could be more than satisfied for the predefined tolerance level.

4.2 Efficiency

Now that we know our algorithm provides the desired results, we will go on test the time efficiency of our algorithm. Here we can not talk about time complexity using big O notation for the results depend not only on dimention but also on target function itself. Instead, we do experiments to test the 'efficiency' of our algorithm by compare the sample size used for caculation and the corresponding time cost. This test breaks into two parts. The first one is that we want to know how our algorithm performs without using control variates. Naturally the cubature sobol algorithm [1] become a good reference. We know our algorithm is a bit slower compared to it when not using control variates. The question is how small the gap is. If there is

Table 4.2. Accuracy Test of reliable QMC algorithm

abstol = 10^{-2}	no CV	single CV	double CV
exact price	1.2926641930	1.2926641930	1.2926641930
estimate price	1.2938177355	1.2937890386	1.2930809161
err=|exact-estimate|	1.1535e-3	1.124e-3	4.167e-4
abstol = 10^{-3}	no CV	single CV	double CV
exact price	1.2926641930	1.2926641930	1.2926641930
estimate price	1.2926529108	1.2925687148	1.2925793049
err=|exact-estimate|	1.12821e-5	9.54782e-5	8.48881e-5
abstol = 10^{-6}	no CV	single CV	double CV
exact price	1.2926641930	1.2926641930	1.2926641930
estimate price	1.2926643000	1.2926639635	1.296646297
err=|exact-estimate|	1.07e-7	2.295e-7	4.367e-7

no significant difference, then it means we will not waste too much time on cases without control variates or with poor control variates. The second one, which is more important, is that we want to see how much time it can save for using good control variates. Of course this depends on quality of the control variates. Fortunately, several good CV is known to be use for option pricing under GBM model [**?**].

4.2.1 reliable QMC without CV. Parameter set up for efficiency test is shown in table4.2.1. This time we try to price a weekly monitored 1-year period option, which increases the dimension to 52. Note we will keep using this set up for all the folloing efficienty tests. We test our algorithm through four scenarors: the original adpaptive

Table 4.3. Parameter Setup for efficienty test

S0	K	TimeVector	r	volatility	abstol	reltol
120	130	1/52:1/52:52/52	0.01	0.5	1e-3	0

QMC, our QMC_CV without CV, QMC_CV with single poor CV and QMC_CV with two poor CV. For target option we still use geomtric mean Asian option European option. For single CV we use European option and add stock price for double CV. As shown in table **??**

Table 4.4. Efficiency test I of reliable QMC algorithm

Sample Size		Time Cost	
cubSobol	cubSobolcv	cubSobol	cubSobolcv
65535	9011	0.2783	0.0673

4.2.2 reliable QMC with CV. There are two types of asian options, depends on which types of mean one use. In last section we introduced geomatric mean Asian option, this time we choose arithematic mean asian call option as our target function, whose payoff function is:

$$C_T^{Amean} = \max\left(\frac{1}{d}\sum_{j=1}^{d} S(jT/d) - K, 0\right)e^{-rT}.$$

For this option there is no close formula for exact price under GBM model. However, recall we introduced another Asian option earlier of which the price has a clear formula. It is known that geomatric mean Asian option was first used as a CV for pricing arithmatic mean Asian option [6]. We start with this pair for efficiency test part II.

Table 4.5. Efficiency test II of reliable QMC algorithm with Asian Option

Sample Size		Time Cost	
cubSobol	cubSobolcv	cubSobol	cubSobolcv
65535	9011	0.2783	0.0673

4.2.3 Barrier Option. Another good example is barrier option. This is the payoff function for up and in barrier call option:

$$C_T^{\mathrm{U\&I}} = (S(T) - K)^+ 1_{\{\max S(jT/d) \geq \mathrm{Barrier}\}}.$$

The reason we choose it is because of its similaritis to the European option. Note if the barrier equals strike price, then it is just an European call option, which makes european option a good control variates. For this test we take three different barriers which gradually grow further from the strick price. As same from last test, we compared oringinal reliable QMC algorithm and our CV version on sample size and time cost. We can see from the results in table 4.2.3 that CV method takes nearly % time

Table 4.6. Results of cubSobol, cv_old and cv_new with Barrier Option

Barrier	Sample Size		Time Cost	
	no CV	with CV	no CV	with CV
140	524288	65535	1.874	0.2743
135	524288	6963	1.959	0.0519
130	524288	1024	1.876	0.0199

lesser than.

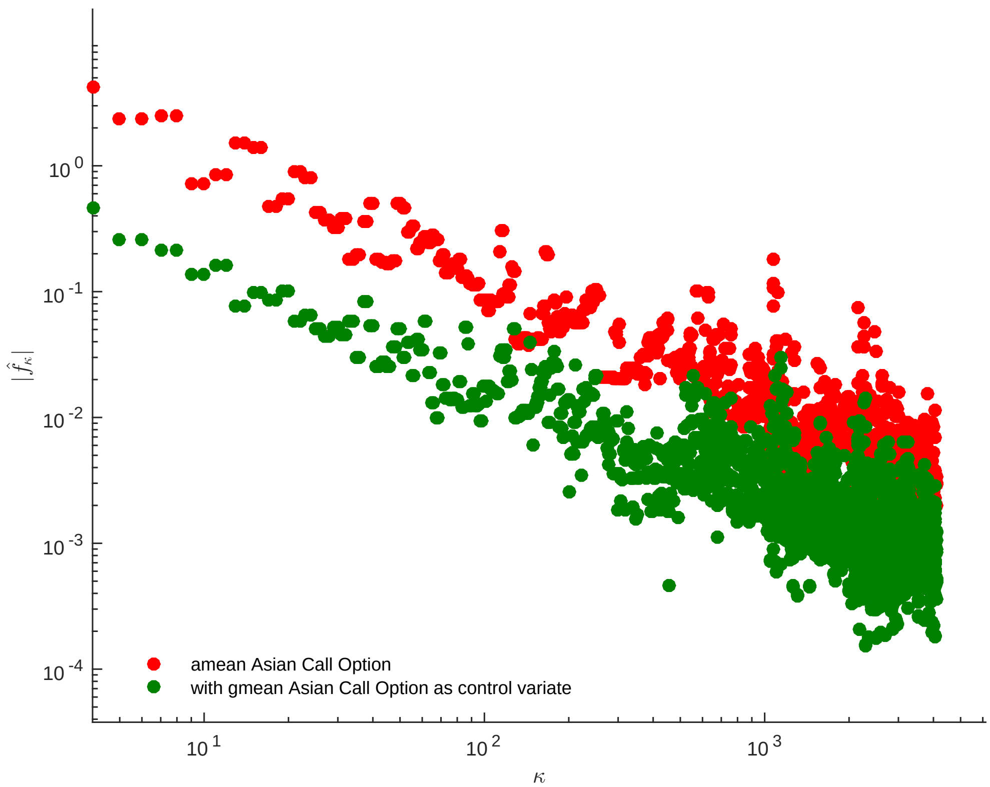

Figure 4.1. Walsh coefficients of f

CHAPTER 5
CONCLUSION

5.1 Discussion

So far there are only few Quasi_Monte Carlo algorithms that can adaptively determine the sample size needed based on integrand values. This is because the estimation of error for QMC is hard. Several studies show that if using Quasi standard error there will be some serious drawbacks [7]. There is also a way using internal replications of i.i.d. Randomized QMC rules, but the number of replications are not known [3].

For control variates the research progress is also limited since its hard to estimate the value of *beta* as we stated in chapter 3.

In our two numerical examples, using the proper control variates gave great results. Also our modified algorithm for the θ problem works as expected.

5.2 Future work

Currently we only have method implemented in Sobol sequences. One further possible work is that we can extend this methods on rank-1 lattices [8]. The idea for getting CV coefficients is the same, but due to the different structure for digital sequences, some effort has to be done for adaption of the method.

BIBLIOGRAPHY

[1] F. J. Hickernell and L. A. J. Rugama, "Reliable adaptive cubature using digital sequences," *arXiv preprint arXiv:1410.8615*, 2014.

[2] A. N. Avramidis and J. R. Wilson, "Integrated variance reduction strategies for simulation," *Operations Research*, vol. 44, no. 2, pp. 327–346, 1996.

[3] F. J. Hickernell, C. Lemieux, A. B. Owen, *et al.*, "Control variates for quasi-monte carlo," *Statistical Science*, vol. 20, no. 1, pp. 1–31, 2005.

[4] G. Fishman, *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, 2013.

[5] H. Niederreiter, *Quasi-Monte Carlo Methods*. Wiley Online Library, 2010.

[6] A. G. Kemna and A. Vorst, "A pricing method for options based on average asset values," *Journal of Banking & Finance*, vol. 14, no. 1, pp. 113–129, 1990.

[7] A. B. Owen, "On the warnock-halton quasi-standard error," *Monte Carlo Methods and Applications mcma*, vol. 12, no. 1, pp. 47–54, 2006.

[8] L. A. J. Rugama and F. J. Hickernell, "Adaptive multidimensional integration based on rank-1 lattices," *arXiv preprint arXiv:1411.1966*, 2014.