# 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
Aug 2016

TABLE OF CONTENTS

LIST OF TABLES

# 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 independent and identically distributed (IID) Monte Carlo (MC) such as control variates (CV).

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

We test our algorithm on various problems including option pricing and multivariate normal probability estimation for dimensions from 4 to 64. The same tests are performed on adaptive QMC algorithm without CV as a comparision. Our results show that with good CV, the cost of adaptive QMC is greatly reduced compared to vanilla QMC.

# CHAPTER 1

# INTRODUCTION

## 1.1 The idea

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 tries to combine the reliable adaptive QMC (RAQMC) methods with control variates (CV). We justify the theory behind it, construct a practical algorithm which can be implemented and tested through high dimensional integration problems.

CV is a variance reduction technique for independent and identically distributed (IID) Monte Carlo (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 CV.

## 1.2 The challenge

The challenge is that for QMC the quadrature points are deterministic instead of random, so the variance minimization can not be used. Even if one tries to use the randomized QMC, the optimal CV coefficient for QMC is generally not the same as IID MC, which is explained by Hickernell, Lemieux, and Owen [3]. This requires us to figure out a new way to get the optimal coefficients for CV with QMC.

## 1.3 Outline

In chapter 2 we first briefly talk about QMC rules and their difference from IID MC. Then we introduce CV and the RAQMC algorithm. In Chapter 3 we derive our method and provide its justification. In chapter 4 we demonstrate results from several numerical experiments. For the final chapter we conclude the results and

discuss unsolved problems as well as possible improvements to our method.

# CHAPTER 2

# BACKGROUND

## 2.1 Problem setup

Numerical integration problems are involved in fields such as physics, mathematical finance, biology and computer graphics. It usually happens when it is hard to solve some integral analytically. Therefore, one has to use numerical methods for such problems. The MC method is a 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:

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

Then we take a sample of $N$ IID points $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\} \in [0,1)^d$ following a uniform distribution, and construct the following MC estimator:

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

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

Hence, the QMC method was introduced to address these problems. The QMC estimator is almost the same as MC estimator. The difference is that the sample points are taken from a low discrepency sequence, which is determinsticlly chosen instead of random. We will briefly review one such sequence that we used for implementation of our method.

## 2.2 Sobol sequence

Like we mentioned earlier, for QMC we do not use random points, so the question becomes how to choose such points? Naturally, in order to make good approximation we want the error to go to 0 as sample size $N$ increases, which is:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x_i}) = \int_{[0,1)^d} f(\mathbf{x})\, \mathrm{d}\mathbf{x}.$$

The answer for how to choose such sample points leads to the theory of uniformly distributed modulo one. One important property of sequences called uniformly distributed modulo one is 'fair' interval [6].

**Definition 2.2.1.** Let $\mathscr{P} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be a finite point set in $[0,1)^d$. For any given subset $J = [\mathbf{a}, \mathbf{b})$ of $[0,1)^d$, we say $J$ is *fair* if:

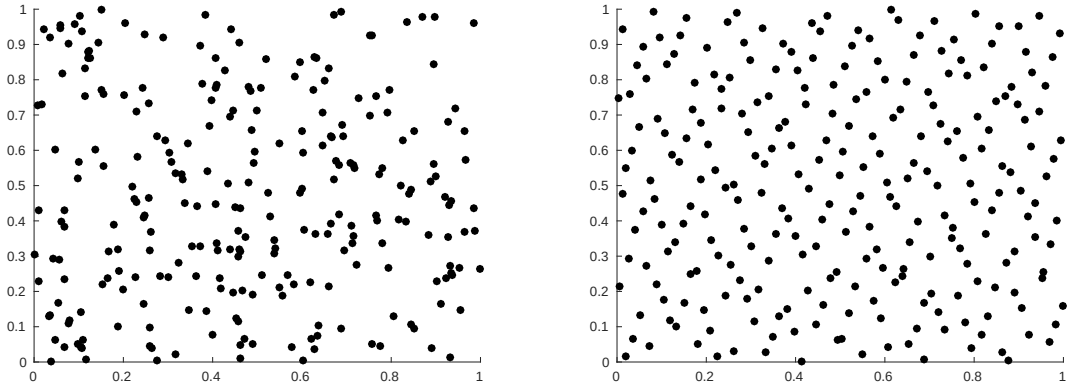
$$\frac{\sum_{i=1}^{N} 1_{[\mathbf{a},\mathbf{b})}(\mathbf{X_i})}{N} = \prod_{j=1}^{d} (b_j - a_j)$$

This simply means that a measurable subset $J$ of a unit cube is fair if the number of sample points in it (left side of equation) is equal to its volume (right side of equation). Similarly, we say a sequence$\mathscr{P}$ is fair if any given subset $J$ is fair. Such fair sequences are ideal for our purpose. Unfortunately, it is not attainable unless the demand for all intervals to be fair is weakened. This motivates the definition of $(t,d)$-sequence.

**Definition 2.2.2.** For a given dimension $d \geq 1$, an integer base $b \geq 2$, a positive integer m, and an integer t with $0 \leq t < m$, a point set $\mathscr{P} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots\}$ in $[0,1)^d$ is called a $(t,d)$-*sequence* in base $b$ if the point set $\{\mathbf{X}_{kb^m+1}, \ldots, \mathbf{X}_{kb^m+b^m}\}$ with all $m > t,\ k \geq 0$ is fair with respect to all intervals of the following form:

$$J = \prod_{i=1}^{d} \Big[\frac{a_i}{b^{s_i}}, \frac{a_i+1}{b^{s_i}}\Big),$$

where $s_i, \ldots, s_d \in \mathbb{N}_0$ with $s_1 + \cdots + s_d = k$ and $0 \leq a_i < b^{s_i}$



(a) 256 randomly generated points (b) 256 Sobol points

Figure 2.1: Comparision between MC and QMC sample points

The Sobol sequence is the first constructed $(t, d)$-sequence in base 2 [6]. Another well known $(t, d)$-sequence is Faure sequence, which is a $(0, d)$-sequences in a prime base $b \geq d$. The advantage of Sobol sequence is that it has a small base. Later in next chapter we will see this is a great help since the sample size is in form of $b^m$. Also since we use computer for simulations, we gain some advantages on bit-level operations in base 2. Figure 2.1 shows a comparision between randomly generated sample points and Sobol sequence points in dimension 2. The latter looks more evenly scattered. In fact by the definition of $(t, d)$-sequence if we divide the unit square in to 256 tiles evenly $(1 \times 256, 2 \times 128, \dots)$, there will be exactly one point in each tile.

## 2.3 Control variates

CV is a well known variance reduction technique used in MC simulation. It is ofen used when a 'simpler' version that is related to the original 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 $\int_{[0,1)^d} h(\mathbf{x})\,\mathrm{d}\mathbf{x} = \theta$. We then construct a new estimator follows:

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

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 choose $\beta_{\mathrm{MC}}$ and why. We know that the mean square error of the MC estimator is $\mathrm{Var}(\hat{I}) + \mathrm{Bias}(\hat{I}^2)$. CV method aims at efficiency improvment, so we need to reduce the mean square error. Since the estimator is unbiased, we only need to minimize its variance. Hence, the optimal $\beta_{\mathrm{MC}}$ should be the one that minmize the variance of esimator. Here we give a simple derivation of optimal $\beta_{\mathrm{MC}}$ for single CV. First, the variance of $\hat{I}_{\mathrm{CV}}$ is:

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

then the optimal $\beta_{\mathrm{MC}}$ is given by:

$$\beta_{\mathrm{MC}}^* = \frac{\mathrm{Cov}[f(\mathbf{X}_i), h(\mathbf{X}_i)]}{\mathrm{Var}[h(\mathbf{X}_i)]}. \tag{2.2}$$

Note that in practice since $\mathrm{Cov}[f(\mathbf{X}), h(\mathbf{X})]$ or $\mathrm{Var}[h(\mathbf{X})]$ is unknown we use estimations to compute $\beta_{\mathrm{MC}}^*$. In this case the variance become:

$$\mathrm{Var}(\hat{I}_{\mathrm{CV}}) = \frac{\mathrm{Var}[f(\mathbf{X}_i)]}{N}\big(1 - \mathrm{corr}^2[f(\mathbf{X}_i), h(\mathbf{X}_i)]\big),$$

and note we always have:

$$\mathrm{Var}(\hat{I}_{\mathrm{CV}}) \leq \frac{\mathrm{Var}[f(\mathbf{X}_i)]}{N} = \mathrm{Var}(\hat{I}).$$

Now we can see the merit of control variates as a variance reduction method. In the worst case, we get a completely uncorrelated $h$ 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

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 Jiménez Rugama (2014) [1] is to construct a QMC algorithm with reliable error estimatiusing 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.

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

$$\hat{I}_m(f) := \frac{1}{b^m} \sum_{i=0}^{b^m-1} f(\mathbf{X}_i) \ \mathbf{X}_i \text{ are Sobol points}$$

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

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

Here is the definition of the cone condition:

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

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

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

The first inequality ($\bigcirc \leq \Diamond$) means the sum of the larger indexed Walsh coefficients bounds a partial sum of the same coeffcients. Take $b = 2,\ l^* = 0,\ m = 12$ for example. In figure 2.2 the sum of circles should be bounded by some factor times the sum of diamonds. The second inequality ($\Diamond \leq \square$) 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 2.2 the sum of diamonds should be bounded by some relax factor times the squares.

The cone gives some meanings for the functions about how they should behave in order to use the error bound formula (2.3). This means that $|\hat{f}_\kappa|$ does not dramatically bounce back as $\kappa$ goes to infinity. Note that in Figure 2.2 we call circles the error 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.

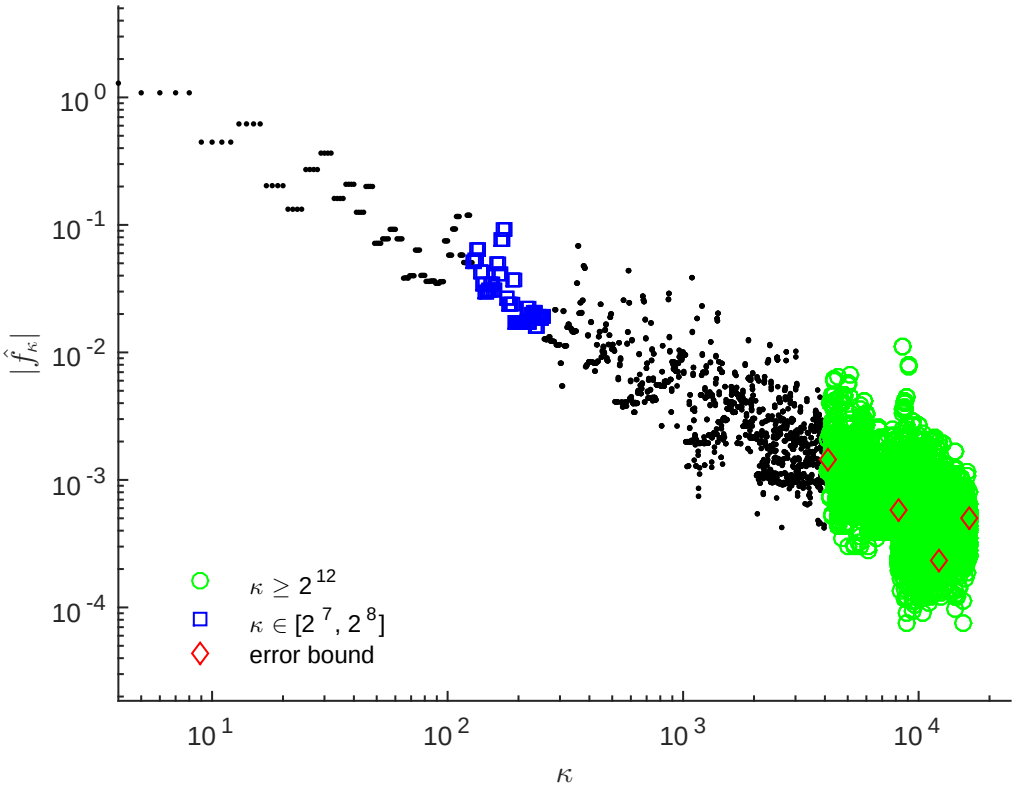



Figure 2.2: Cone condition for reliable adpative QMC algorithm

# CHAPTER 3

# RELIABLE ADAPTIVE QMC WITH CV

## 3.1 Idea to add control variates to reliable adaptive QMC

The whole method starts with an idea similar to traditional control variates technique for MC. If we know the integration of a function $\boldsymbol{h} = (h_1, \ldots, h_J)$ on the interval same as our $f$, say $\int_{[0,1)^d} h_j \, \mathrm{d}\mathbf{x} = \theta_j$, then we can define a new function $g$:

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

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

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

Now we wonder 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 CV with randomized QMC

The problem is that QMC is not a random process and we simply can't use the minimizing mean squre error trick as shown earlier anymore. However, one can use randomized QMC instead to 'restore' the randomness to QMC [7]. Randomized QMC use a different way for generating $\mathbf{X}_i$, they are still identical(i.e. from same distribution) but not independent, which will make it different from CV with MC.

Suppose $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ are generated by QMC rule, the estimator stays the same

$$\hat{I}_{cv}(f) = \frac{1}{N} \sum_{i=1}^{N} \Big[ f(\mathbf{X}_i) - \beta_{qmc}[h(\mathbf{X}_i) - \theta] \Big] \quad \mathbf{X}_i \text{ are Sobol points.}$$

We can easily prove it is still unbiased

$$\mathbb{E}(\hat{I}_{cv}) = \mathbb{E}\Big( \frac{1}{N} \sum_{i=1}^{N} \Big[ f(\mathbf{X}_i) - \beta_{qmc}[h(\mathbf{X}_i) - \theta] \Big] \Big) = I.$$

However, it's not the same case as MC like we presented before, because we do not have IID sample points this time, i.e.

$$\mathrm{Var}(\hat{I}_{cv}) \neq \frac{1}{N}\mathrm{Var}\Big(f(\mathbf{X}_i) - \beta_{qmc}[h(\mathbf{X}_i) - \theta]\Big) \text{ [3].}$$

Instead the variance becomes

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

The optimal $\beta_{qmc}$ is

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

which leave the variance to be

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

Now we are interested that if our previous formula for $\hat{\beta}_{mc}$ could still 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 formula (2.2) to see the difference.

$$\begin{aligned}
\mathrm{Cov}(\hat{I}, \hat{H}) &= \int [f(\mathbf{X}_1) + \cdots + f(\mathbf{X}_n)][h(\mathbf{X}_1) + \cdots + h(\mathbf{X}_n)]\ d\mathbf{X} \\
&= \int [\sum_{i=1}^{N} f(\mathbf{X}_i)h(\mathbf{X}_i) + \sum_{i,j=1}^{i\neq j} f(\mathbf{X}_i)h(\mathbf{X}_j)]\ d\mathbf{X} \\
&\neq \int f(\mathbf{X}_i)h(\mathbf{X}_i)d\mathbf{X}_i \\
&= \mathrm{Cov}[(f(\mathbf{X}_i), h(\mathbf{X}_i)].
\end{aligned}$$

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

### 3.3 A new way to find $\beta$

As we stated in previous section, we can not find optimal $\beta$ by minimizing variance of estimator like MC. However, if using the reliable adaptive QMC method introduced in chapter 2, we may have another way to find $\beta$.

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

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

Naturally, the new estimator becomes

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

From (3.3) it is clear that the optimal $\beta$ is the one that minimize the error term.

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

The second equivalence (3.6) is not hard to get, but the third one (3.7) may not be so obvious. Let's consider it backwards. Suppose we have a vector A and it's $\mathcal{L}_1$-norm.

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

If we replace the index, $A$ is exactly what's inside the $\mathcal{L}_1$-norm in (3.6). Hence, the third equivalence is justified. The reason we use the least square to approximate the $\mathcal{L}_1$ regression, is because it is less efficient to solve compared to existing least square methods.

### 3.4 The problem with $\theta$

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

The idea is form a observation that Walsh transform of $\theta$ in (3.5) is actually zero, since $\hat{h}_\theta = \theta\delta_{\kappa,0}$ for all nonzero $\kappa$ and the summation does not start from $\kappa = 0$.

This simplifies (3.5) to the following.

$$\begin{aligned}
\boldsymbol{\beta}^* &= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \hat{\boldsymbol{\theta}})\boldsymbol{\beta}| \\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \boldsymbol{\theta}\delta_{\kappa,0})\boldsymbol{\beta}| \\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - \hat{\boldsymbol{h}}_\kappa\boldsymbol{\beta}| \\
&= \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \widehat{\boldsymbol{H}}\boldsymbol{\beta}\|_1 \\
&\approx \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \widehat{\boldsymbol{H}}\boldsymbol{\beta}\|_2 \qquad\qquad \widehat{\boldsymbol{H}}_j = (\hat{h}_{b^{m-r-1},j}, \ldots, \hat{h}_{b^{m-r}-1,j})^T. \qquad (3.8)
\end{aligned}$$

Note that we only need the information of function $f$ and $h$ to calculate $\beta^*$, $\theta$ has been get rid of the optimization process. Hence, $\widehat{\mathbf{H}i_j}$ is redefined as in (3.8).

The same problem comes with the estimator (3.4). We have the similar solu-

tion for that.

$$\begin{aligned}\hat{I}_m(g) &= \frac{1}{b^m}\sum_{i=0}^{b^m-1} g(\mathbf{X_i}) \\ &= \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(\mathbf{X_i}) - (\boldsymbol{h}(\mathbf{X_i}) - \boldsymbol{\theta})\boldsymbol{\beta} \\ &= \frac{1}{b^m}\sum_{i=0}^{b^m-1} [f(\mathbf{X_i}) - \boldsymbol{h}(\mathbf{X_i})\boldsymbol{\beta}] + \boldsymbol{\theta}\boldsymbol{\beta}. \end{aligned} \tag{3.9}$$

After organizing it in the same format as in (3.9), $\theta$ is eliminated from the summation part. From these two revised formula (3.8) and (3.9), we managed to save $(b-1)b^{m-r-1} + b^m$ operations of subtraction.

### 3.5 The modified method

Now we make the following changes:

$$\begin{aligned} g &:= f - \beta h \\ \hat{I}_m(g) &:= \frac{1}{b^m}\sum_{i=0}^{b^m-1} g(\mathbf{X}_i). \end{aligned}$$

And we have the following equivalence:

$$\begin{aligned} \int_{[0,1)^d} f \, \mathrm{d}\mathbf{x} &= \int_{[0,1)^d} g \, \mathrm{d}\mathbf{x} + \theta\beta \\ \hat{I}_m(f) &= \hat{I}_m(g) + \theta\beta. \end{aligned}$$

So the estimation error becomes:

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

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:

$$\Big| \int_{[0,1)^d} g \, \mathrm{d}\mathbf{x} - \hat{I}_m(g) \Big| \le a(r,m) \sum_{\lfloor 2^{m-r-1} \rfloor}^{2^{m-r}-1} |\tilde{g}_{m,\kappa}|.$$

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 provide the algorithm for reliable adaptive QMC with control variates using digital sequences.

**Algorithm 1:** Reliable Adaptive QMC with control variates

**Data**: function $f$ and $\boldsymbol{H}$; value of $\int_{[0,1)^d} h_j \, \mathrm{d}\mathbf{x} = \theta_j$; tolerance $\varepsilon$

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

**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. $g = f - \boldsymbol{H}\boldsymbol{\beta}$, 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 $\hat{I}_m(g) = \sum_{i=0}^{2^m-1} f[x(i)] + \boldsymbol{\theta}\boldsymbol{\beta}$
   - 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 2 in Algorithm 1, we used an explicit way to generate it. One can find the details for that in appendix from Hickernell and Jiménez Rugama (2014) [1]. Here we reorganize it and show it in algorithm 2.

Another important point needed to be mentioned is that in our algorithm, we used an iterative way, which may require recalculating *beta* at each iteration.

**Algorithm 2:** kappa map and discrete Walsh coefficients

**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

## 4.1 Option Pricing

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 make several tests on our reliable QMC with CV algorithm with option pricing problems. Note all our experiments are implemented under geomatric 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.

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

**4.1.1 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 test our algorithm for accuracy in three cases: original reliable adaptive QMC (RAQMC), reliable adaptive QMC with CV (RAQMC_CV) for single CV and RAQMC_CV for double 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 the geomatric mean Asian option as our target function because we have the exact pricing formula under GBM model [8]. We choose to test the call options. For the 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 the exact price under the 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 several CV of which the exact solution must have clear formulas. Here we pick the European call option and the stock price at expiration as control variates. For the European call option, the payoff function is:

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

The exact price under the GBM model is given by [9]:

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

For the stock price, the expected discounted price is $S(T)e^{-rT}$, while the exact price under the GBM model is just $S(0)$.

Table 4.1 shows the set up for all options in the accuracy test. We use the in the money call option for both Asian and European options. They are monitored

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 |

weekly for a one month period with 1% interest rate and 50% volatility anunally. The test is on three scenarios: RAQMC, RAQMC_CV with one CV (RAQMC_CV$_1$) and RAQMC_CV with two CV (RAQMC_CV$_2$). The test is then performed with three different absolute error tolerances as shown in table 4.2. First we caculated the exact price for the geomatric mean Asian call option using formula (4.1). Then we run the same parameters through RAQMC_CV 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 for our algorithm as accurate as guranteed. Note for the second test, which the tolerance is $10^{-3}$, while the error seems all within a less bound $10^{-4}$. We believe this is not an 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.

**4.1.2 Efficiency.** Now that we know our algorithm provides the desired results, we continue to test the time efficiency of the algorithm. Here we do not use time complexity like big O notation. This is because the results of our algorithm depend not only on dimention but also on target function itself. Since we use an iterative way to stop we can not know the anwer a priori. Instead, we do experiments to test the 'efficiency' of our algorithm by comparing the sample size used for each caculation and the corresponding time cost. Note the results do not stay the same for each

Table 4.2: Accuracy Test of RAQMC_CV algorithm

| abstol = $10^{-2}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
|---|---|---|---|
| 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 = $10^{-3}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
| 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 = $10^{-6}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
| 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 |

time, to compensate that we take an average of 10 consecutive runs with the same set-up. This test breaks into two parts. The first one is that we want to know how our algorithm performs without using CV. Naturally the RAQMC algorithm we introduced in chapter 3 become a good reference. We know our algorithm is a bit slower compared to RAQMC without CV. The question is how small the gap is. If there is no significant difference, then it means we will not waste too much time on cases without CV or with poor CV. The second one, which is more important, is that we want to see how much time it can save by using good CV. Of course, this depends

on quality of the CV. Fortunately, several good CV are known to be use for option pricing under GBM model [10].

**4.1.2.1 RAQMC_CV without CV.** The parameter set up for efficiency test is shown in table 4.3. This time we try to price a daily monitored four months period option, which increases the dimension to 64. Note we will keep using this set up for all the following efficienty tests. We test our algorithm through four scenarios:

Table 4.3: Parameter Setup for all efficienty tests

| S0 | K | TimeVector | r | volatility | abstol | reltol |
|---|---|---|---|---|---|---|
| 120 | 130 | 1/250:1/250:64/250 | 0.01 | 0.5 | 1e-3 | 0 |

RAQMC, RAQMC_CV without CV (RAQMC_CV$_0$), RAQMC_CV with one poor CV (RAQMC_CV$_1$) and RAQMC_CV with two poor CV (RAQMC_CV$_2$). 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 4.4, all

Table 4.4: Efficiency test I of RAQMC_CV

| abstol= $10^{-3}$ | RAQMC | RAQMC_CV$_0$ | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
|---|---|---|---|---|
| Sample Size | 32768 | 32768 | 32768 | 32768 |
| Time Cost | 0.4906 | 0.4908 | 0.4915 | 0.4901 |

test end up with same sample size, which means that choice of CV does not help at all. But what we concern is that how much extra cost can it bring if our choice of CV is bad? We can see the time cost just increased around 2% for QMC with bad CV. Now we can say our method did make the cost of CV to a minimal.

**4.1.2.2 RAQMC_CV 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:

Table 4.5: Efficiency test II of RAQMC_CV algorithm with Asian Option

| | Estimate Price | Sample Size | Time Cost | $\beta^*$ |
|---|---|---|---|---|
| RAQMC | 3.5010925 | 49152 | 0.7789 | null |
| RAQMC_CV$_1$ | 3.5009629 | 16384 | 0.2658 | 0.8445 |
| RAQMC_CV$_2$ | 3.5010566 | 16384 | 0.2698 | $(0.8190,0.0028)^T$ |

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

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 [8]. Hence, for single CV we choose geomatric mean Asian call option as CV. We also want to test it for double CV so we add European call option as a second CV. Results are shown in table 4.5. We can see this choice of CV is really good. The sample size is approximately 1/4 of original size and same with time cost. The second CV does not make an obvious improvement compared to single CV but seems do offer a bit help.

Figure 4.1 is a plot of the walsh coefficients for the target function and control variates in this Asian options experiments. Recall our error of approximation is bound on this exact same coeffcients. What we'd expect is that on both methods it

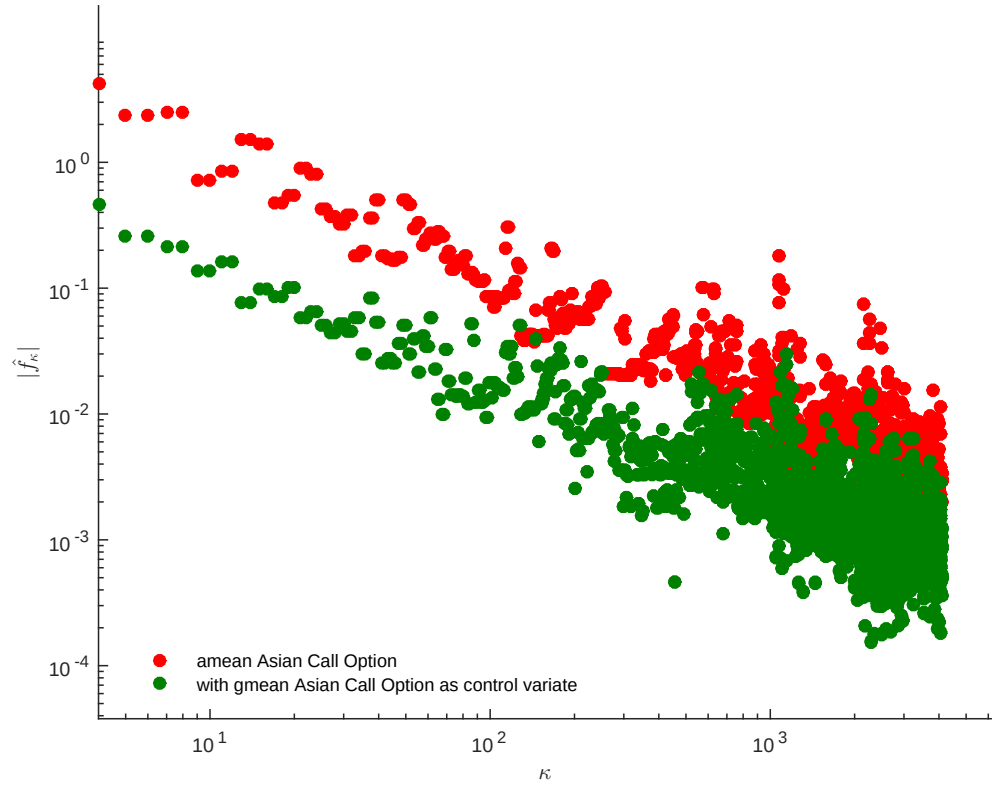



Figure 4.1: Walsh coefficients of $f$

can decrease fast but faster for CV method. The horizontal axis in the figure can be interpreted as the sample size, the vertical axis can be view as the error bound. What our algorithm does is that for a given error bound, we monitor these Walsh coefficients untill it reaches the bound. Obviously, the CV method will stop early in this case which means less compuational cost for it.

Another good example is the barrier option. The payoff function for the up and in barrier call option is:

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

The reason we choose it is for its similarity to the European option. Note if the barrier equals strike price (130), then it is just a European call option, which makes european

option a good CV. For this test we take four different barriers that gradually decrease to the strick price. We use the same parameter set-up as the previous one which is shown in table 4.3. As same as the last test, we compare oringinal RAQMC algorithm and RAQMC_CV on sample size and time cost. One merit for this example is that

Table 4.6: Efficiency test II of RAQMC_CV algorithm with Barrier Option

| Barrier | Sample Size | | Time Cost | | $\beta^*$ |
|---|---|---|---|---|---|
| | RAQMC | RAQMC_CV | RAQMC | RAQMC_CV | |
| 160 | 1048576 | 1048576 | 25.5112 | 25.0197 | 0.6613 |
| 150 | 524288 | 262144 | 11.8742 | 5.2743 | 0.8999 |
| 140 | 524288 | 32768 | 11.7936 | 0.5327 | 0.9935 |
| 130 | 524288 | 1024 | 11.3675 | 0.0609 | 1.0000 |

we can see how the cost and value of $\beta^*$ change as we change the barrier. In this example we know the optimal value should be close to 1 as shown in table 4.6 when barrier=130. Also, note we set the start sample size for iteration to be $2^{10}$, which is exactly the sample size used in this case. This is consistent with our statement earlier, it is just an European option. When we move barrier further from 130, we assume the European option will not be as good as the pervious one as a CV. Our results confirm this assumption, we can see in table 4.6 both the time cost and sample size rise as we increase the barrier. At last when barrier=160, European option completely lose its value as CV and cost of two methods are almost the same.

### 4.2 Multivariate normal probabilities

For this part tests, we are interested in computing the probability of a multivariate normal distribution in dimension 6

$$F(\mathbf{X} \in [0,1)^6) = \int_{[0,1)^6} \frac{1}{\sqrt{|\Sigma|(2\pi)^6}} e^{\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}}\, \mathrm{d}\mathbf{x},$$

where $\Sigma$ is a symmetric positive definite covariance matrix. So this time the target function is multivariate normal density function and to make it simple we set the covariance matrix in the following form

$$f(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma|(2\pi)^6}} e^{\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}},\ \Sigma = \begin{pmatrix} 1 & \sigma & \sigma & \sigma & \sigma & \sigma \\ \sigma & 1 & \sigma & \sigma & \sigma & \sigma \\ \sigma & \sigma & 1 & \sigma & \sigma & \sigma \\ \sigma & \sigma & \sigma & 1 & \sigma & \sigma \\ \sigma & \sigma & \sigma & \sigma & 1 & \sigma \\ \sigma & \sigma & \sigma & \sigma & \sigma & 1 \end{pmatrix}.$$

Then we choose the two multivariate normal distribution with diagonal covariance matrices as CV

$$h_1(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma_1|(2\pi)^6}} e^{\frac{1}{2}\mathbf{x}^T\Sigma_1^{-1}\mathbf{x}},\ \Sigma_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$h_2(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma_2|(2\pi)^6}} e^{\frac{1}{2}\mathbf{x}^T\Sigma_2^{-1}\mathbf{x}},\ \Sigma_2 = \begin{pmatrix} \sigma & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma \end{pmatrix}.$$

The reason we pick these as CV is that they are from the same multivariate gaussion family and correlate to the targe function. Computation of the integrals on these CV are rather simple since each dimension is independent in this case. We choose three different values for $\sigma$ ($0.1, 0.4$ and $0.7$) and run the test for three senarios: RAQMC,

RAQMC_CV with h1 as CV (RAQMC_CV$_1$), RAQMC_CV with h1 and h2 as CV (RAQMC_CV$_2$). The sample size and time cost in table 4.7 are average value from 20 identical trials. Because the exact value of the target integral is unknown, we use an approximation for 'exact' value to compute the error. These approximations are computed using a method developed by Genz and Bretzwith an error of $10^{-10}$ [11]. Similarly, we compute the mean of CV with the method developed by Drezner and Wesolowsky and by Genz [12] [13]. The time cost for computation of means of CV is included in the results.

We start the test with $\sigma = 0.1$. With this setting the targe function should be close to the first CV. As expected, both single and double CV perform great. The sample size is reduced to 4096 from 27852 and time cost is about half down compared to RAQMC. As we increase the value of $\sigma$, the first CV start to lose its value. When $\sigma = 0.7$ the first CV become not helpful and cost more time than RAQMC. In the mean time, the double CV method show some good effect, at $\sigma = 0.7$, the sample size is half of the other 2 and time cost is about 30% less.

Table 4.7: Test of RAQMC_CV with multivariate normal probability

| $\sigma = 0.1$, abstol=$10^{-7}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
|---|---|---|---|
| estimate value | 0.0020706728 | 0.0020706626 | 0.0020706649 |
| err=\|'exact'-estimate\| | 1.8e-9 | 4.3e-9 | 7.0e-9 |
| sample size | 27852 | 4096 | 4096 |
| time cost | 0.0414 | 0.0206 | 0.0242 |
| $\beta^*$ | null | 1.1093 | (0.9649,4.47e-5)$^T$ |
| $\sigma = 0.4$, abstol=$10^{-7}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
| estimate value | 0.0046138363 | 0.0046138346 | 0.0046138341 |
| err=\|'exact'-estimate\| | 1.9e-9 | 1.9e-9 | 2.6e-9 |
| sample size | 65536 | 32768 | 32768 |
| time cost | 0.0805 | 0.0595 | 0.0628 |
| $\beta^*$ | null | 2.2582 | (0.9673,0.0408)$^T$ |
| $\sigma = 0.7$, abstol=$10^{-7}$ | RAQMC | RAQMC_CV$_1$ | RAQMC_CV$_2$ |
| estimate value | 0.0169509335 | 0.0169509327 | 0.0169509342 |
| err=\|'exact'-estimate\| | 1.8e-9 | 6.1e-9 | 3.6e-9 |
| sample size | 524288 | 511180 | 262144 |
| time cost | 0.5235 | 0.6187 | 0.3626 |
| $\beta^*$ | null | 8.6074 | (-18.95,7.78)$^T$ |

# CHAPTER 5
# CONCLUSION

## 5.1 Discussion

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

For CV with QMC the research progress is also limited since it is hard to estimate the value of CV coefficients as we stated in chapter 3. Hickernell and Jiménez Rugama (2014) [1]'s work on building a QMC method provides a reliable and adaptive way to use QMC, as well as gives us insights into combining reliable adaptive QMC with CV. The main idea of the reliable adaptive QMC is to bound the error of estimation using summation of partial Walsh coefficients. We utilize the same idea for calculating the optimal coefficient for CV. In order to compensate the extra computation cost for CV, we used several techniques in our design to keep those cost minimal.

We test our algorithm on several option pricing problems under Black-Scholes scheme. We find the accuracy of algorithm is consistent and reliable. Comparison of QMC with CV and normal QMC is performed on different pairs of options, the results show that using the proper CV we can make improvements over vanilla QMC methods. We also provide an example of computing multivariate normal probability using CV with estimated mean. It is shown in certain cases multiple CV outperforms single CV.

## 5.2 Future work

There are several unsolved problems in our work, of which we have not found the answers yet. To start with, we use the least square regression solution to approximate the least absolute error regression for CV coefficients. Is there a way to do $\mathcal{L}_1$ regression more efficiently? Can it offer enough improvement on computational cost to compensate its own cost? This leads to the second problem. In our numerical examples we do not update CV coefficients in each iteration. The reason is that we find doing this updating requires a lot of recalculations of previous terms, which could reduce the advantage brought by CV. For further research we hope to find a way to update $\beta$ more efficiently. Another possible work for future research is that we can extend this method to rank-1 lattices [14]. The idea for getting CV coefficients is the same, but due to the different points structure compared to 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] J. Dick and F. Pillichshammer, *Digital nets and sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge University Press, 2010.

[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] 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.

[9] P. P. Boyle, "Options: A monte carlo approach," *Journal of financial economics*, vol. 4, no. 3, pp. 323–338, 1977.

[10] T. Lidebrandt, *Variance reduction: Three approaches to control variates*. PhD thesis, Citeseer, 2007.

[11] A. Genz, "Numerical computation of multivariate normal probabilities," *Journal of computational and graphical statistics*, vol. 1, no. 2, pp. 141–149, 1992.

[12] Z. Drezner, "Computation of the bivariate normal integral," *Mathematics of Computation*, pp. 277–279, 1978.

[13] A. Genz, "Numerical computation of rectangular bivariate and trivariate normal and t probabilities," *Statistics and Computing*, vol. 14, no. 3, pp. 251–260, 2004.

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