Quasi-Monte Carlo Methods:

Applications in Finance and Actuarial Science

by

Ken Seng Tan

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Statistics

Waterloo, Ontario, Canada, 1998

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

In recent years, there has been a growing interest in the application of quasi-Monte Carlo methods in finance and actuarial science. A common application of the Monte Carlo method is in the evaluation of multi-dimensional integrals. Quasi-Monte Carlo uses specially selected deterministic sequences rather than random sequences as in Monte Carlo. These special sequences are known as low discrepancy sequences and have the property that they tend to be evenly dispersed throughout the unit cube. For many applications in finance, the use of low discrepancy sequences seems to provide more accurate answer than random sequences. Nevertheless there are several drawbacks of this method. First, there is no simple criterion to assess the accuracy of the estimates in applications of this technique. Second, the integrand should have certain smoothness properties. This can be a restrictive condition in some situations. Third, any additional smoothness of the integrands is not reflected in the the error bound of the quasi-Monte Carlo method. In this thesis, we address these issues and examine ways of overcoming these problems so that the efficiency of the quasi-Monte Carlo method can be enhanced.

Acknowledgements

Uniformity plays an important role in Quasi-Monte Carlo methods. Ironically (fortunately and unfortunately), my life at Waterloo has never been able to achieve such uniformity. These experiences have made my life in a seemingly tameness town more challenging. Miraculously, I survive!

First, I would like to express my deepest gratitude to my supervisor, Professor Phelim P. Boyle. He has been encouraging, patient, understanding, and never grudging the time required however pressing his own work. His insights, guidance, and the numerous stimulating discussions in "Second Cup" were instrumental in accomplishing this thesis. I will always remember the interesting time we shared in Salt Lake City, New York City, Cairns, Sydney, ... as well as the endless days and nights in our involvement in another book project.

My appreciation also goes to Professors Adam Kolkiewicz, Don L. McLeish, Harry H. Panjer, Kenneth R. Vetzal for their constructive comments as well as serving on my examining committee. In particular, Professor Harry H. Panjer gave me the opportunity concurrently (and painfully) of working on another book; otherwise this thesis would have been finished much earlier. I am also grateful to Professor Paul Glasserman of Columbia University, who acted as my external examiner and provided me with a great deal of useful input.

Special thanks go to the departmental chair Professor Mary E. Thompson who offered me the opportunity to be part of her team; Professor Gordon E. Willmot for the coffee break that I desperately needed; Professor Elias S. Shiu of University of Iowa for the recommendation letters he wrote; Professor Art B. Owen of Stanford

University for the discussion with respect to Chapter 4; Dr. David X. Li for the discussion with respect to Chapter 6; Professor Mary R. Hardy for her being a teacher/friend/colleague. I treasure many inspiring conversations that we shared beyond the scope of this thesis. I recall when I first met Phelim, he appeared to be trading in an "incomplete market". But thanks to Mary who evidently completes his market! Thanks also to other faculty members (Professors Robert L. Brown, Jerry F. Lawless, Keith P. Sharp, William J. Welch, retired Mike A. Bennett and Jim B. Whitney), staff (Nandanee Basdeo, Lucy Simpson, Linda Lingard, Amy Aldous, retired Mary Chen) and the gracious students at the University of Waterloo, who made my years at Waterloo meaningful and enjoyable. Also to Marta Poterski for kindly giving me a "noon call", ensuring I would not miss or skip my lunch.

Further thanks go to my parents and my siblings, especially my sister Yoke Peng who just started her university life at Waterloo. Friends in other parts of the world — Malaysia, Singapore, Hong Kong, Taiwan, US (Dr. Shaun Wang, Esther Yang, Chon Keat Wong, Professor Xiaodong Lin), and those who are still in Canada (Angela and Kurt, Hanh and Chong, Lan and Eduard, Lepain and Jake). They can finally stop asking me the question "Are you done with your PhD yet?".

The acknowledgement would not be complete if I do not mention a special lady — Winnie Cheung, to whom this thesis is dedicated. Undoubtedly, she has allowed me to savour extremums of my life.

Finally, I would like to acknowledge the research support from University of Waterloo, Natural Sciences and Engineering Research Council of Canada, Ontario Graduate Scholarship and the Society of Actuaries.

Contents

List of Tables

List of Figures

Chapter 1

Overview

The Monte Carlo simulation method is a powerful and flexible approach for providing numerical solutions to a large class of complex problems. In recent years, the Monte Carlo approach has been extensively used in the finance and investment area. Initially, the applications were mainly concerned with calculations related to the pricing of complex financial instruments and the computation of related hedging parameters. Examples of such instruments are mortgage-backed securities and various complex exotic options. More recently, Monte Carlo methods have been used to estimate the distribution of returns of entire portfolios. Applications include the calculation of credit risk and market risk and value at risk computations. Boyle, Broadie and Glasserman [10] provide a recent survey of the applications of Monte Carlo methods to financial calculations.

We now discuss briefly why the Monte Carlo method is useful in the investment and finance area. In modern financial economics, security prices are modelled as stochastic processes to reflect future uncertainty. The current price of a security

can be represented as the expected value of the future payouts on the security. This follows from the assumption of no-arbitrage. The expectation is taken with respect to a probability measure that is induced by the current price system. Under a suitable normalization, the revised prices become martingales under this probability measure. Hence this probability measure is often called the equivalent martingale measure. If a complex financial instrument has a payout that depends on the prices of several underlying securities or a payout that depends on the price path of an existing security, then its price can be written as a multi-dimensional integral. There are many different types of financial instruments of this nature. In some cases, the number of dimensions is quite large; for example, under mortgage-backed securities, the number of dimensions is as high as 360.

For high dimensional problems, the Monte Carlo method has strong advantages over alternative numerical integration schemes. The rate of convergence for the Monte Carlo method is $\mathcal{O}(N^{-1/2})$ which is independent of the dimension. This is in contrast to most quadrature methods which suffer from the "curse of dimensionality" with a convergence rate of $\mathcal{O}(N^{-2/s})$. One disadvantage of the standard Monte Carlo method is that in some cases, notably for large scale problems, the rate of convergence is very slow. Different methods of speeding up the convergence have been proposed. These techniques are known as variance reduction techniques. Recently, so-called quasi-Monte Carlo methods or the low discrepancy (LD) methods have been used in finance applications. These methods rely on the use of specially selected deterministic sequences instead of random sequences. These deterministic sequences have the property that they are well dispersed throughout the unit cube

and are known as (LD) low discrepancy sequences. The monograph by Niederreiter [68] provides an excellent discussion of these sequences. Applications of low discrepancy sequences to finance problems are discussed by Acworth, Broadie and Glasserman [1], Boyle, Broadie and Glasserman [10], Caflisch, Morokoff and Owen [16], Galanti and Jung [27], Joy, Boyle and Tan [46], Ninomiya and Tezuka [72] and Paskov and Traub [80]. There are differences in the relative efficiency of LD methods versus standard Monte Carlo for the typical finance problems and for other more general applications. In the case of finance problems, the LD methods appear to outperform standard Monte Carlo for some high dimensional problems. For example, in mortgage-backed security applications with dimensions as high as 360, Paskov and Traub [80] report good results using Sobol' sequences while Ninomiya and Tezuka [72] conclude that generalized Niederreiter sequences are superior. For more general non-finance integrals — which can be non smooth or highly periodic — the superiority of low discrepancy algorithms vanishes for dimensions around 30 or even lower. For a comparison of the two approaches in the case of more general integrands see Bratley, Fox and Niederreiter [14] and Morokoff and Caflisch [59, 60]. The advantages of LD methods for finance applications appears to stem from the smoothness of the integrand in many applications and the fact that the effective dimension in finance applications is sometimes lower than the actual dimension. However, it is not always the case that the classical LD methods dominate standard Monte Carlo for finance applications. The relative effectiveness depends on several factors including the nature of the integrand and the properties of the sequence used to evaluate it.

It is worth noting that many of the techniques introduced in this thesis can also be applied to problems in actuarial science. Indeed, one of the most popular insurance contracts in the United States is the so-called equity-indexed annuity (EIA) which can contain different types of exotic options. For example, the contract could include a minimum guaranteed amount plus a variable payoff based on the performance of a stock market index during the life time of the contract. These EIAs feature a whole range of exotic options such as lookback options (high-water mark), cliquet options and Asian options. The methods introduced in this thesis provide an attractive alternative to the standard Monte Carlo approaches that are often employed to value these contracts.

We now provide a brief summary of the chapters in this thesis. Chapter 2 provides a brief overview of the Monte Carlo and LD methods. Since the notion of discrepancy plays a crucial role in characterizing the uniformity of the sequences as well as the error bound of the LD methods, we provide a more detailed discussion.

In Chapter 3, we introduce different types of low discrepancy point sets and low discrepancy sequence. The notion of elementary intervals, (t, m, s)-nets and (t, s)-sequences play an important role in low discrepancy point sets and low discrepancy sequence. These concepts are abstract and sometimes difficult to understand. In Section 3.3, we use a graphical representation to describe various important properties of elementary intervals, (t, m, s)-nets and (t, s)-sequences. We also generalize the conventional elementary intervals, (t, m, s)-nets and (t, s)-sequences which allows variable bases in defining the point sets and sequences. This is in contrast to conventional (t, m, s)-nets and (t, s)-sequences which confine to a common base.

In Chapter 4, we examine the two-dimensional orthogonal projections of the low discrepancy sequences. We find that undesirable pattern occurs even in the first two dimensions of a non-telescopic sequence while a similar pattern is observed in higher dimensions for telescopic sequence.[1]

Until recently a major drawback of the classical LD approach has been the absence of a reliable practical error bound. Even though there exists a deterministic upper bound based on Koksma-Hlawka inequality, this theoretical bound significantly overestimates the actual error. This is in contrast to the crude Monte Carlo method for which the accuracy of the estimate can be assessed by constructing the confidence limits.

In Section 4.3, we describe a technique developed by Owen [74] which provides a probabilistic error bound for the quasi-Monte Carlo method by randomizing the points while preserving the low discrepancy property. By replicating this procedure, we obtain several independent estimates for the integral of interest and thus provide an estimate of the standard error. From this we can derive probabilistic error bounds and overcome one of the main disadvantages of the classical quasi-Monte Carlo approach. Owen's original method, however, is computationally infeasible when the dimension is greater than 10. Consequently, we propose a partial randomization technique based on Owen's method. The proposed partial randomization technique has the advantage of computational feasibility in high dimensions. We investigate the effectiveness of this approach and provide numerical illustrations which compare it to Owen's approach in the case of low-dimensional problems. We

[1]For definition on telescopic and non-telescopic sequences, see Section 3.6 of Chapter 3.

also apply the proposed sequences to high-dimensional option problems and conclude that substantial variance reduction can be achieved even when compared to Monte Carlo methods using antithetic variable technique.

In Chapter 5, we examine the impact of the discontinuity in LD methods. Examples which contain discontinuity include integrating a characteristic function or the decision to accept or reject in an acceptance-rejection algorithm. Unless the boundary of the set which defines the discontinuity is parallel to the coordinate axes, the variation of the function is infinite. While the unbounded variation does not affect the conventional convergence rate of Monte Carlo methods, it has an adverse effect on the efficiency of the low discrepancy sequences. This also follows from the Koksma-Hlawka inequality which leads to an infinite error bound in these situations. In this chapter, we discuss methods of dealing with discontinuities when using LD methods. We provide a general framework for avoiding the discontinuity by smoothing the integrands. This method is inspired by the smoothing technique proposed by Moskowitz and Caflisch [61]. Our empirical studies indicate the superiority of our generalization over Moskowitz and Caflisch's smoothing technique. The generalized technique also has the advantage that the resulting estimate is unbiased, in contrast to Moskowitz and Caflisch's method which is biased.

In this chapter, we consider functions which have additional smoothness than that required by the Koksma-Hlawka error bound. It follows from the Koksma-Hlawka inequality that any additional regularity of the integrand is not reflected in the magnitude of the error bound. This is in contrast to classical one-dimensional integration rules, such as the Gaussian formulas, are designed to exploit the par-

ticular regularity class of the integrand to achieve a higher rate of convergence. In Chapter 6, we discuss another family of numerical quadrature formulas known as method of good lattice points. This method is designed to handle periodic functions and achieves a rate of convergence of order $\mathcal{O}(N^{-\alpha}\log^{\alpha\beta} N)$, where the constant $\alpha > 0$ depends on the smoothness class of the function while $\beta > 0$ depends on the dimension. For certain values of α and β, this rate compares favourably to the rate $\mathcal{O}(N^{-1}\log^{s-1} N)$ attained by LD methods with fixed point set. We apply these lattice rules to low-dimensional financial instruments such as the spread options, generalized rainbow options, lookback options and Asian options. We find that this method yields an extremely high level of precision even for a relatively small number of integration nodes. This method also compares favourably to other methods.

In Chapter 7, we consider a specific application of the low discrepancy sequences. In particular, we propose an efficient technique for simulating through trees. The need to simulate through trees arises in situation when the values of a derivative security is path-dependent or when the tree is not recombining. In this chapter, we describe how to combine the simulation technique with low discrepancy sequences. We also establish condition under which simulating through tree using low discrepancy sequences leads to the same exact discrete pricing as the backward induction method provided we use the correct number of points. This requires an additional uniformity in the conventional low discrepancy sequences. The numerical examples conducted also indicate that the resulting refined low discrepancy sequences perform better than conventional low discrepancy sequences.

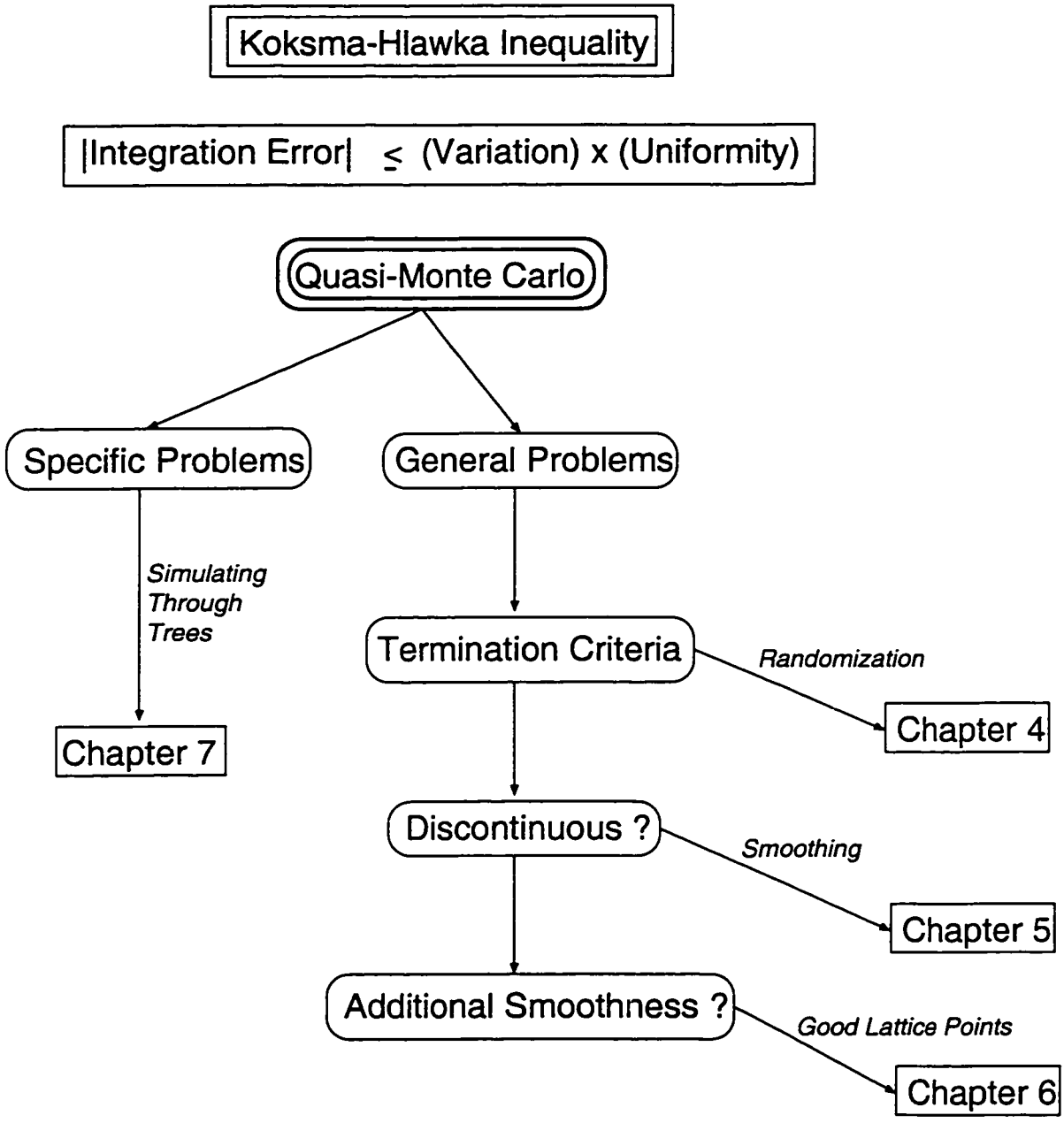


Figure 1.1: Flow Chart of the Chapters

To summarize, Figure 1.1 provides a "flow chart" for the last four chapters. Essentially, Chapters 4, 5 and 6 identify some of the disadvantages using LD methods and describe methods of overcoming these problems. These problems can be analyzed in term of the error bound given by the classical Koksma-Hlawka inequality which states that the absolute integration error is bounded by a product of two components. (See also Chapter 2.) These two components correspond to the *smoothness* of the function and the *uniformity* of the deterministic points used in approximating the function. Chapter 7, on the other hand, considers a specific application using low discrepancy sequences for simulating through trees.

In the last chapter of the thesis; i.e. Chapter 8, we provide a general guideline for using the LD methods and list potential research topics that could be conducted in near future.

Chapter 2

Monte Carlo and Quasi-Monte Carlo Methods

2.1 Introduction

This chapter lays out the background which is essential for the remainder of the thesis. A key distinction between quasi-Monte Carlo and Monte Carlo methods is that the former uses deterministic points while the latter uses random points. Typically we have in mind the problem of evaluating a multidimensional integrals. The sequence of points can be used as the integration nodes for evaluating such high-dimensional integrals. It is well known that using a random sequence gives rise to a rate of convergence $\mathcal{O}(N^{-1/2})$. On the other hand, using a systematically constructed sequence, which is more evenly distributed than a randomly generated sequence, can result in a better rate of convergence. Section 2.2 provides a brief introduction to the classical Monte Carlo methods with particular application to

integration. The advantages and disadvantages of this method are discussed. The motivation for using deterministic points is discussed in Section 2.3. Since the notion of discrepancy plays a central role in analyzing these deterministic sequences and in evaluating the performance of the quasi-Monte Carlo methods, an extensive discussion of this concept is given in Section 2.4.

2.2 Monte Carlo Integration

Consider the problem of evaluating a multiple integral of the form

$$\int_{[0,1)^s} f(\boldsymbol{x})d\boldsymbol{x} = \int_0^1 dx^{(1)} \int_0^1 dx^{(2)} \cdots \int_0^1 dx^{(s)} f(x^{(1)}, x^{(2)}, \ldots, x^{(s)}) = \theta \quad (2.2.1)$$

where $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(s)}) \in [0,1)^s$. We also assume the function f is Riemann integrable in $[0,1)^s$ and that $|\theta| < \infty$.

In many practical problems of interest, the function f is either too complicated to evaluate or that the dimensionality of the problem, s, is too large to be handled by either traditional methods such as the quadrature formula. One viable technique is the Monte Carlo method.

The basic principle of Monte Carlo method is to use independent, uniformly distributed random numbers in $[0,1)^s$ as the source of integration nodes to approximate θ. Suppose $\{\boldsymbol{x}_n\}$ is a canonical random sequence such that $\{\boldsymbol{x}_n\}$ are independently and identically distributed in $[0,1)^s$, the Monte Carlo estimate of θ

is given by the average of f evaluated at the point $\{\boldsymbol{x}_n\}$. In other words,

$$\text{Monte Carlo estimate } (\hat{\theta}) = \frac{\sum_{n=1}^{N} f(\boldsymbol{x}_n)}{N} \tag{2.2.2}$$

where N is the total number of random points. The strong law of large number asserts that the Monte Carlo estimate based on (2.2.2) converges almost surely.[1] It also follows that the Monte Carlo estimate is unbiased; i.e.

$$\mathrm{E}(\hat{\theta}) = \mathrm{E}(f) = \theta \tag{2.2.3}$$

and with variance

$$\begin{aligned} \sigma_N^2 = \mathrm{Var}(\hat{\theta}) &= \mathrm{E}[(\hat{\theta} - \mathrm{E}(\hat{\theta}))^2] \\ &= \frac{\sigma^2(f)}{N} \end{aligned} \tag{2.2.4}$$

where $\sigma^2(f)$ is the variance of f defined by

$$\sigma^2(f) = \int_{[0,1)^s} f^2(\boldsymbol{x})d\boldsymbol{x} - \left(\int_{[0,1)^s} f(\boldsymbol{x})d\boldsymbol{x}\right)^2. \tag{2.2.5}$$

Note that the Monte Carlo method still converges even if $\sigma^2(f)$ is infinite. We assume $\sigma^2(f) < \infty$ so that

$$\lim_{N\to\infty} \sigma_N^2 = \lim_{N\to\infty} \frac{\sigma^2(f)}{N} = 0.$$

[1] $g_n \to g$ almost surely if $\{\omega \in \Omega : g_n(\omega) \to g(\omega)\}$ as $n \to \infty$ is an event with probability 1. In our context, if $\{\boldsymbol{x}_n\}$ is a sequence of random vectors, the event that $\hat{\theta}$ does not tend to θ as $N \to \infty$ has probability zero.

An important feature Monte Carlo method is the probabilistic error bound. The Central Limit Theorem guarantees that the distribution of $\hat{\theta}$ converges asymptotically to a normal distribution with mean θ and variance σ_N^2 as $N \to \infty$. Therefore,

$$\lim_{N\to\infty} \text{Prob}\left(\frac{c_1\sigma(f)}{\sqrt{N}} \le \hat{\theta} - \theta \le \frac{c_2\sigma(f)}{\sqrt{N}}\right) = \frac{1}{\sqrt{2\pi}}\int_{c_1}^{c_2} e^{-t^2/2}\,dt$$

for any constants $c_1 < c_2$ and $0 < \sigma(f) < \infty$. For this reason, it is more meaningful to provide the confidence interval of the Monte Carlo estimate, rather than just the point estimate $\hat{\theta}$ given by (2.2.2). For instance, the 95% confidence interval is usually reported and is constructed as follows:

$$\left[\hat{\theta} - 1.96\frac{\sigma(f)}{\sqrt{N}},\ \hat{\theta} + 1.96\frac{\sigma(f)}{\sqrt{N}}\right], \tag{2.2.6}$$

where

$$\frac{1}{\sqrt{2\pi}}\int_{-1.96}^{1.96} e^{-t^2/2}\,dt = 0.95$$

and $\sigma(f)$ is the standard deviation of the function f (as given by (2.2.5)). In practice, $\sigma(f)$ is often estimated using the following consistent estimator

$$\hat{\sigma}(f) \approx \sqrt{\frac{\sum_{n=1}^{N}(f(\boldsymbol{x}_n) - \hat{\theta})^2}{N-1}}.$$

The interval constructed according to (2.2.6) can be interpreted that with 95% confident the true value θ lies in this range. The precision of the Monte Carlo estimate depends on how narrow this interval is. The tighter the interval, the

better the Monte Carlo estimate. There are two ways of improving the accuracy of the standard Monte Carlo estimate:

1. reducing the sampling variance, $\hat{\sigma}^2(f)$, or

2. increasing the number of trials, N.

The techniques used to improve the efficiency of the Monte Carlo estimate by reducing the variability of the underlying function are referred to as *variance reduction techniques*. Examples of variance reduction techniques include the control variate technique, the antithetic variable method, stratified sampling and importance sampling. An alternative to variance reduction techniques is to resort to the brute force approach by increasing the number of simulations until a desired level of accuracy is reached. This method, however, can be very expensive and time-consuming. To improve the Monte Carlo estimate by 1/10 requires an increase in the number of simulations by 100-fold. This follows from the rate of convergence of the Monte Carlo method which is of order $\mathcal{O}(N^{-1/2})$. Nevertheless, one advantage of the Monte Carlo method is that the rate of convergence is independent of the dimensions of the problems. This is in contrast to most other techniques which usually suffer from a phenomenon called the "curse of dimensionality". The classical quadrature method with an error bound of $\mathcal{O}(N^{-2/s})$ is an example.

Despite these advantages, the Monte Carlo method does have several deficiencies which hinder its usefulness. First is the probabilistic nature of the error estimate. As discussed earlier, it is customary to report the confidence interval as opposed to just the point estimate $\hat{\theta}$. Due to the random fluctuation of the estimate, the point estimate $\hat{\theta}$ by itself can be a bad estimate of the true value θ even though

the confidence interval constructed satisfies the statistical notion of the confidence level. Furthermore, the constructed confidence interval does not completely rule out the possibility that the true value may not lie in the interval, although such event is less probable. In situations where high precision of the estimate is critical, such as the sensitivity analysis, the probabilistic nature of the estimate becomes undesirable.

Another fundamental difficulty with Monte Carlo method lies on the requirement that $\{x_n\}$ is a random sample that is independently and identically distributed. In fact, a truly random sequence does not exist. What is actually done in practice is to use so-called *pseudo-random sequences.* Such a sequence has the same relevant statistical properties as a random sequence and presumably will behave like a random sequence even though it is generated through a deterministic algorithm. The success of any Monte Carlo calculation therefore crucially depends on the "ability" of these sequences to "imitate" the true random sequence. When a bad (and undetected) pseudo-random number generator is used, the resulting Monte Carlo estimate becomes meaningless.

Although the theory concerning the construction and analysis of (pseudo-)random number generators is well developed (see for example, Knuth [49] and L'Ecuyer [57]), unreliable and dangerous generators still exist in the scientific literature and on computer systems. The following quote from L'Ecuyer [57] indicates the potential danger in blindly relying on "black-box" (pseudo-)random number generators.

> *Many of the "default" [(pseudo-)random number] generators currently offered in popular computer softwares, or suggested in some simulation*

textbooks, are old ones, and are not competitive with those based on the more recent theory. Much worse, many bad generators are still proposed every year in (supposedly serious) journal articles.

Despite the fact that the rate of convergence does not depend on the dimension, the Monte Carlo method is generally regarded as a time-consuming tool, with rate of convergence $\mathcal{O}(N^{-1/2})$. As pointed out earlier, variance reduction techniques can be used in conjunction with the Monte Carlo method to enhance the efficiency of the underlying method. The effectiveness of these variance reduction techniques depends on how well they reduce the sampling variance component $\hat{\sigma}^2(f)$. Quasi-Monte Carlo or the low discrepancy method is another approach that has been proposed in the literature. This method potentially has a much better rate of convergence than the Monte Carlo method. This thesis explores the application of quasi-Monte Carlo methods to numerical problems in finance.

2.3 Motivation for Using Deterministic Points

Let us assume for simplicity that the problem of interest is in the evaluation of one-dimensional integrals. Recall that Monte Carlo integration relying on points that are randomly, independently and uniformly distributed in $[0, 1)$ can achieve on average an absolute error of magnitude $N^{-1/2}$. This implies that there must exist a set of nodes for which the integration error is no larger than the average. For instance, in the case of 1-dimensional problem, it is well known that we could achieve a much better convergence rate using quadrature methods such as trapezoidal or midpoint rules. In contrast to pseudo-random points which try to imitate

the randomness as closely as possible, quadrature methods seek *derandomization.* Under the quadrature methods, the integration nodes are spread more evenly over the integration domain $[0,1)$ than a typical random sample from a uniform distribution. For example, if we were to use an evenly spaced grid points on $[0,1)$ such as $\{\frac{1}{N}, \frac{3}{N}, \ldots, \frac{N-1}{N}\}$, the integration error is $\mathcal{O}(N^{-2})$, which is a drastic improvement over the Monte Carlo method. The method just described is typically known as the N-panel method (or midpoint rule).

The experience from the 1-dimensional problem suggests that among the basic ingredients (randomness, independencies, and uniformity) of a random sequence, the randomness and independencies seem to play a secondary role in Monte Carlo integrations. A sequence with a better uniformity than a random sequence leads to better rate of convergence. Generalizing this idea to high dimensions suggests that using a sequence that is more uniformly distributed in $[0,1)^s$ than a random sequence could lead to a better convergence rate. This indicates the essential feature of the quasi-Monte Carlo or the low discrepancy method. The basic idea of the quasi-Monte Carlo or the low discrepancy method is to replace the random sample in a Monte Carlo method by carefully chosen deterministic points. The criteria for the choice of deterministic points is such that the sequence in $[0,1)^s$ has a better uniformity than a random sequence. To measure the uniformity we use the concept of *discrepancy* which is the topic of the next section. The ultimate goal of quasi-Monte Carlo or the low discrepancy method is to achieve an integration error that is significantly smaller than that given by Monte Carlo method.

In view of the advantages of quasi-Monte Carlo method, it is interesting to

compare it with the Monte Carlo method. The disadvantages of Monte Carlo method was discussed in previous section. Since the underlying principle of quasi-Monte Carlo method is to abandon the randomness and independencies and to adopt completely deterministic nodes in a Monte Carlo calculation, the resulting integration error is therefore deterministic, rather than probabilistic. Furthermore, when the deterministic nodes are well chosen, the quasi-Monte Carlo method yields an error bound of $\mathcal{O}(N^{-1}\log^{s} N)$ for an infinite sequence and $\mathcal{O}(N^{-1}\log^{s-1} N)$ for a finite point set, assuming function f is sufficiently regular. These rates are superior to the Monte Carlo rate of convergence for large N.

It was argued that another problem with the Monte Carlo method lies in the random sample generation. It is difficult to generate a sequence that is a truly representative of a random sequence since "randomness" is a statistical concept. This difficulty, on the other hand, evaporates when a quasi-Monte Carlo method is used. In the quasi-Monte Carlo method, there exists a precise construction algorithm for generating the required nodes.

2.4 Discrepancy

This section describes the concept of discrepancy. For a more complete treatment on this topic, we refer to Kuipers and Niederreiter [53] or Niederreiter [68].

Definition 2.1 (Discrepancy) *Consider a s-dimensional sequence of N points* $\{\boldsymbol{x}_n\} \in [0,1)^s$. *For any hyper-rectangular box of the form* $\mathcal{J} = \prod_{i=1}^{s}[u^{(i)}, v^{(i)})$, *with*

$0 \leq u^{(i)} < v^{(i)} \leq 1$, *the* ***(local)discrepancy*** *over* $\mathcal{J}$, $D_N(\mathcal{J})$ *is defined by*

$$D_N(\mathcal{J}) = \frac{\mathcal{X}(\mathcal{J})}{N} - v(\mathcal{J}), \tag{2.4.1}$$

where $v(\mathcal{J}) = \prod_{i=1}^{s}(v^{(i)} - u^{(i)})$ *is the volume of* $\mathcal{J}$ *and* $\mathcal{X}(\mathcal{J})$ *represents the number of points of* $\{\boldsymbol{x}_n\}$ *whose coordinates satisfy* $u^{(i)} < x_n^{(i)} \leq v^{(i)}$ *for* $1 \leq i \leq s$; *i.e.* $\mathcal{X}(\mathcal{J})$ *counts the number of points that belong to the hyper-rectangular box* $\mathcal{J}$.

Figure 2.1 illustrates the concept of local discrepancy in two dimensions. Consider a sequence of 10 elements $\in [0,1)^2$. Let $(u^{(1)}, u^{(2)}) = (0.3, 0.4)$ and $(v^{(1)}, v^{(2)}) = (0.8, 0.6)$ so that the rectangle $\mathcal{J}$ is of the form

$$\mathcal{J} = [0.4, 0.8) \times [0.3, 0.6)$$

with $v(\mathcal{J}) = 0.4 \times 0.3 = 0.12$. From the graph, we see that two points lie in

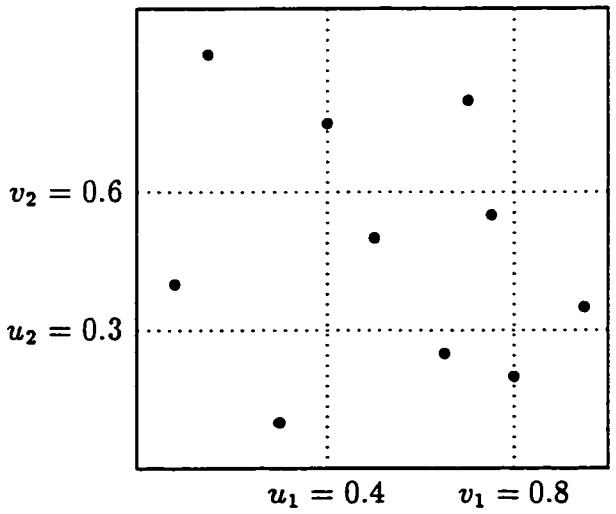


Figure 2.1: ILLUSTRATION OF LOCAL DISCREPANCY

rectangle $\mathcal{J}$. Hence, the local discrepancy over $\mathcal{J}$ can be computed as

$$D_{10}(\mathcal{J}) = \frac{2}{10} - 0.12 = 0.08 \ .$$

By taking different norm of $D_N(\mathcal{J})$ over $[0,1)^s$, we obtain different discrepancy measures such as the extreme, the star and the mean square discrepancies. In general, these measures give a global measure of uniformity of the sequence. More importantly, these discrepancies play an important role in the error bounds for quasi-Monte Carlo integration. We will discuss this in greater detail in Subsection 2.4.1. We now describe some of the more important types of discrepancies.

Definition 2.2 (Extreme Discrepancy) *The L_∞ norm (**extreme discrepancy**) of a finite sequence $\{\boldsymbol{x}_n\}$ is defined as*

$$D_N = \sup_{J \in \mathcal{J}} |D_N(J)| \tag{2.4.2}$$

where the supremum is extended over all subintervals in $[0,1)^s$ of the form $\mathcal{J} = \prod_{i=1}^{s}[u^{(i)}, v^{(i)}]$, with $0 \le u^{(i)} < v^{(i)} \le 1$.

Definition 2.3 *The L_2 norm is defined as*

$$T_N = \left\{ \int_{(\boldsymbol{u},\boldsymbol{v}) \in [0,1)^{2s}, u^{(i)} < v^{(i)}} [D_N(J(\boldsymbol{u},\boldsymbol{v}))]^2 d\boldsymbol{u} d\boldsymbol{v} \right\}^{\frac{1}{2}} \tag{2.4.3}$$

where $J(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{J}$.

If we impose an additional condition that $u^{(i)} = 0$ for all j, then the counterpart measures of discrepancy exist and are defined respectively as follows:

Definition 2.4 (Star Discrepancy) *The L_∞ norm (**star discrepancy**) is defined as*

$$D_N^* = \sup_{J \in \mathcal{J}^*} |D_N(J)| \tag{2.4.4}$$

where $\mathcal{J}^*$ is the family of the subintervals $\mathcal{J}$ such that the hyper-rectangle box contains a corner at $\mathbf{0}$.

Definition 2.5 (Mean Square) *The L_2 norm (mean square) is defined as*

$$T_N^* = \left\{ \int_{[0,1)^s} [D_N(J(\mathbf{0}, \boldsymbol{v}))]^2 d\boldsymbol{v} \right\}^{\frac{1}{2}} \tag{2.4.5}$$

where $J(\mathbf{0}, \boldsymbol{v}) \in \mathcal{J}^*$.

A more general concept of discrepancy is suggested by Fang and Wang [23]. This can be described as follow:

Definition 2.6 *Let $F(\boldsymbol{x})$ be a cumulative density function in $\Re^s$ and $\{\boldsymbol{x}_n, 1 \leq n \leq N\}$ be a set of points on $\Re^s$. Then the **F-discrepancy** of the point set with respect to $F(\boldsymbol{x})$ is defined as*

$$\hat{D}_N^* = \sup_{\boldsymbol{x} \in \Re^s} |F_N(\boldsymbol{x}) - F(\boldsymbol{x})|,$$

where $F_N(\boldsymbol{x})$ is the empirical distribution of $\{\boldsymbol{x}_n, 1 \leq n \leq N\}$.

Note that $\hat{D}_N^*$ is also the *Kolmogorov-Smirnov statistic* for the goodness of fit test of $F(\boldsymbol{x})$. When $F(\boldsymbol{x})$ is the uniform distribution on $[0,1)^s$, the F-discrepancy becomes the star discrepancy described above.

The classical result in the theory of uniform distribution of sequences states that an infinite sequence $\{\boldsymbol{x}_n\}$ is uniformly distributed in $[0,1)^s$ if and only if $\lim_{N\to\infty} D_N^* = 0$ (or equivalently $\lim_{N\to\infty} D_N = 0$). In other words, the extreme and the star discrepancies measure the uniformity of a sequence. They can be viewed as a quantitative measure for the deviation from the uniform distribution. Sequences that have small discrepancy (see Equation (2.4.7)) are known as low discrepancy sequence. The significance of the discrepancy will become clear when we discuss the error bound for quasi-Monte Carlo integration in Subsection 2.4.1. Before we introduce this result, let us consider various properties and relationships among these discrepancy measures. The simplest such relationship is given by the following proposition (see Niederreiter [68, Proposition 2.4]).

Proposition 2.1 *For any finite sequence in* $[0,1)^s$*, we have the following relationship between* D_N^* *and* D_N

$$D_N^* \leq D_N \leq 2^s D_N^*.$$

The first inequality is trivial since $\mathcal{J}^*$ is a subset of $\mathcal{J}$. The second inequality arises from the fact that any sets in $\mathcal{J}$ can be expressed as a 2^s sets in $\mathcal{J}^*$. Another trivial relationships is

$$T_N^* \leq D_N^*,$$

which is obvious since L_∞ norms are larger than L_2 norms. Additional relationships between D_N and T_N^* was established in Niederreiter [68] while the relationship

between T_N and T_N^* was discussed in Morokoff and Caflisch [59].

Since discrepancy measures the uniformity of a sequence and since the efficiency of quasi-Monte Carlo method relies on the uniformity of the sequence, two questions arise at this point.

1. Is it possible to construct sequences that have greater uniformity than the random sequence ?

2. If such sequences exist, is it possible to construct an optimal point set in the sense that it has the lowest possible discrepancy ?

It turns out that it is possible to construct a sequence that has a smaller discrepancy than the random sequence. Using the law of iterated logarithms, the expectation of a random sequence can be shown to be bounded by $N^{-1/2}(\log\log N)^{1/2}$ (Chung [17] or Kiefer [48]). An example of a sequence which has a better uniformity that the random sequence is one-dimensional sequence with nodes given by the midpoint rule; i.e. $x_n = \frac{2n-1}{N}$. In fact in one-dimensional sequence of N elements, the points constructed according to the midpoint rule yields the lowest discrepancy, as indicated by the following theorems:

Theorem 2.1 *If* $0 \le x_1 \le \cdots \le x_N \le 1$, *then*

$$D_N^* = \frac{1}{2N} + \max_{1\le n\le N}\left|x_n - \frac{2n-1}{2N}\right|.$$

Theorem 2.2 *If* $0 \le x_1 \le \cdots \le x_N \le 1$, *then*

$$D_N = \frac{1}{N} + \max_{1\le n\le N}\left(\frac{n}{N} - x_n\right) - \min_{1\le n\le N}\left(\frac{n}{N} - x_n\right).$$

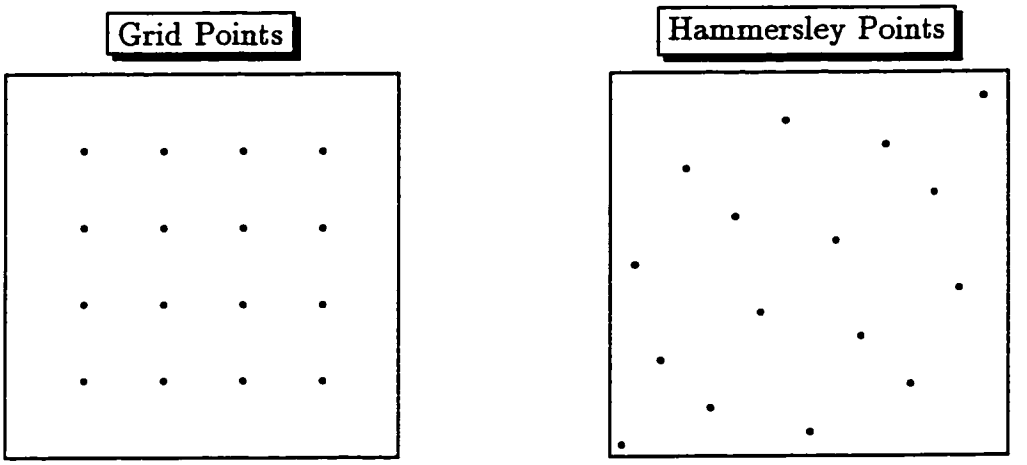


Figure 2.2: COMPARISON OF TWO DIFFERENT POINT SETS

For proofs of the above theorems, see Niederreiter [68, p. 15]. Theorem 2.1 suggests that for any 1-dimensional finite sequence of N points, the lowest possible star discrepancy is attained by setting $x_n = \frac{2n-1}{N}$, which coincides with the midpoint rule.

It is tempting to generalize the equally spaced grid point to higher dimensions based on the Cartesian product rule; i.e. $N = M^s$. Unfortunately, such a regular grid points (or the rectangular lattice) only work well in one dimension. To understand why they fail to produce satisfactory results in high dimensions, it is instructive to consider the following example where $s = 2$.[2] The left panel of Figure 2.2 depicts a sample of $N = 4^2$ grid points, i.e. $M = 4$, while the right panel gives 16 Hammersley points.[3] Suppose the function $f(\boldsymbol{x})$, $\boldsymbol{x} \in [0,1)^2$ under consideration depends strongly on one of the variables. The rectangular lattice yields

[2]We follow the argument made in Sobol' [95], see also Fox [26].

[3]A Hammersley sequence is an example of a low discrepancy point set. See Chapter 3.

only 4 different values with each repeated 4 times. The Hammersley points, on the other hand, produce 16 different values of either $f(x_1)$ or $f(x_2)$. If these points are used to create scenarios in a solvency or cash-flow testing, the implication is that we really obtain 4 different scenarios for the first set while 16 different scenarios for the second set.

In general, when a function $f(\boldsymbol{x})$, $\boldsymbol{x} \in [0,1)^s$, depends strongly only on $s' < s$ leading variables, the rectangular lattice results in only $M^{s'} = N^{s'/s}$ different values. Hence $N - N^{s'/s}$ functional evaluations become redundant. The problem becomes more significant when $s' \ll s$ and as N increases.

The discrepancy of the rectangular lattice can be shown to be

$$D_N^* = (M+1)^{-1} \sim N^{-1/s}.$$

This order is better than the random sequence only when $s = 1$. The discrepancy of the rectangular lattice becomes progressively large as s increases. This is also the underlying reason that the quadrature methods do not perform well in high dimensions.

Extensive research has been carried out to construct an optimum s-dimensional sequence that has the lowest possible discrepancy. Unfortunately, even with $s = 2$, such point set exists only for $N \leq 6$. For instance, with one point, the optimum position of the point is $((\sqrt{5}-1)/2, (\sqrt{5}-1)/2)$ with $D_1^* = (\sqrt{5}-1)/2$. See White [105] for the solution for $N \leq 6$. Other attempts have also been made to search for a uniformly best sequence, see for example Aird and Rice [2] and Lambert [54].

In discussing the discrepancy bound, it is important to make a distinction between a point set and a sequence. Informally, a point set consisting of N elements $\in [0,1)^s$ is called a low discrepancy point set when D_N^* is small. For low discrepancy sequence, additional points can be introduced to the existing set of points while still preserving low discrepancy. Informally, a low discrepancy sequence can be defined as a sequence of elements $\in [0,1)^s$ for which D_N^* is small for all $N > 1$. Because of the flexibility in the low discrepancy sequence, the discrepancy of a low discrepancy sequence is higher than a low discrepancy point set. This is also reflected in the discrepancy bound discussed below.

For a s-dimensional point set or sequence, it has been established that its discrepancy must be no smaller than the famous Roth [87] lower bound which states that

$$\begin{aligned} D_N^* &\geq B_s \frac{\log^{(s-1)/2} N}{N} && \text{for a finite point set,} \\ D_N^* &\geq B_s \frac{\log^{s/2} N}{N} && \text{for an infinite sequence,} \end{aligned}$$

where $B_s > 0$ depends only on s.

Unfortunately, except in one dimension, it is not known if such lower bound is attainable. Improvements to Roth lower bound has been suggested by Schmidt [90] and Beck [5], but only restricted to cases with $s = 2$ and 3. It is widely believed

that the lower bound of Roth can further be improved to

$$D_N^* \geq B'_s \frac{\log^{s-1} N}{N} \qquad \text{for a finite point set,} \qquad (2.4.6)$$

$$D_N^* \geq B'_s \frac{\log^{s} N}{N} \qquad \text{for an infinite sequence,} \qquad (2.4.7)$$

where $B'_s > 0$ depends only on s. Conjecture (2.4.6) is true for $s = 1, 2$ but remains open for $s \geq 3$. For $s = 1$, (2.4.6) follows immediately from Theorem 2.1 while for $s = 2$, (2.4.6) was established by Schmidt [90]. Unfortunately, $s = 1$ is the only case for which (2.4.7) is known. This result is again due to Schmidt [90]. Expository accounts of this topic are given in Kuipers and Niederreiter [53].

Although (2.4.6) and (2.4.7) have not been proven in higher dimension, many sequences have been constructed that attain the conjectured lower bound. For this reason, it is customary to speak of low discrepancy sequences when such bounds are achieved. We will discuss these sequences in later sections.

2.4.1 Error Bounds

The importance of discrepancy can be seen from the Koksma-Hlawka inequality. The inequality states that the absolute error in approximating an integral by a series of point evaluations is bounded by a product of two measures, the discrepancy of the sequence and the variation of the integrand. In the 1-dimensional case, the Koksma inequality is stated as follows:

Theorem 2.3 (Koksma Inequality) *If f has bounded variation $V(f)$ on $[0,1)$, then, for any N points $\{x_n\} \in [0,1)$, we have*

$$\left| \frac{\sum_{n=1}^{N} f(x_n)}{N} - \int_0^1 f(u)du \right| \leq V(f) D_N^*.$$

where D_N^* is the star discrepancy of the sequence $\{x_n, 1 \leq n \leq N\}$.

The generalization of Koksma inequality to multidimensional case requires an appropriate concept of bounded variation for functions of multiple variables. In particular, the one which is of interest to us is the Hardy-Krause variation, which is defined as follows:

Definition 2.7 (Hardy-Krause Variation) *For a function $f(\boldsymbol{x}) = f(x^{(1)}, \ldots, x^{(s)})$ on $[0,1)^s$ with $s \geq 2$, let us consider a partition $\mathcal{P}$ of $[0,1)^s$ which is composed of a set of s finite sequences $u_0^{(j)}, u_1^{(j)}, \ldots, u_{m_j}^{(j)}$ such that $0 = u_0^{(j)} \leq u_1^{(j)} \leq \cdots \leq u_{m_j}^{(j)} = 1$ for $j = 1, \ldots, s$. For such partition, define an operator Δ_j, $j = 1, \ldots, s$, by*

$$\Delta_j f(x^{(1)}, \ldots, x^{(j-1)}, u_i^{(j)}, x^{(j+1)}, \ldots, x^{(s)}) =$$

$$f(x^{(1)}, \ldots, x^{(j-1)}, u_{i+1}^{(j)}, x^{(j+1)}, \ldots, x^{(s)}) - f(x^{(1)}, \ldots, x^{(j-1)}, u_i^{(j)}, x^{(j+1)}, \ldots, x^{(s)})$$

*for $i = 0, \ldots, m_j - 1$. Then the **variation** of f on $[0,1)^s$ in the sense of **Vitali** is defined by*

$$V^s(f) = \sup_{\mathcal{P}} \sum_{i_1=0}^{m_1-1} \cdots \sum_{i_s=0}^{m_s-1} |\Delta_{1,\ldots,s} f(u_{i_1}^{(1)}, \ldots, u_{i_s}^{(s)})|, \tag{2.4.8}$$

where $\Delta_{1,\ldots,s}$ *denotes* $\Delta_1 \cdots \Delta_s$ *and the supremum is extended over all partitions* $\mathcal{P}$ *of* $[0,1)^s$.

Now, for $1 \leq k \leq s$ *and* $1 \leq i_1 < \cdots < i_k \leq s$, *let* $V^{(k)}(f; i_1, \ldots, i_k)$ *denote the Vitali variation of the function* f *restricted to the* k*-dimensional subspace such that*[4] $(x_1, \ldots, x_s)$ $x_j = 1$ *for* $j \neq i_1, \ldots, i_k$. *Then*

$$V(f) = \sum_{k=1}^{s} \sum_{1 \leq i_1 < \cdots < i_k \leq s} V^{(k)}(f; i_1, \ldots, i_k)$$

is called the ***variation*** *of* f *on* $[0,1)^s$ *in the sense of* ***Hardy-Krause*** *and* f *is of* ***bounded variation*** *in the Hardy-Krause sense if* $V(f)$ *is finite.*

With this notion of variation, the Koksma inequality is generalized to the multidimensional case as follows:

Theorem 2.4 (Koksma-Hlawka Inequality) *For any sequence of* N *points* $\{\boldsymbol{x}_n\}$ *in* $[0,1)^s$ *and any function* f *of bounded variation in the sense of Hardy-Krause, then we have*

$$\left| \frac{\sum_{n=1}^{N} f(\boldsymbol{x}_n)}{N} - \int_{[0,1)^s} f(\boldsymbol{u}) d\boldsymbol{u} \right| \leq \sum_{k=1}^{s} \sum_{1 \leq i_1 < \cdots < i_k \leq s} V^{(k)}(f; i_1, \ldots, i_k) D_N^*(i_1, \ldots, i_k) \tag{2.4.9}$$

where $D_N^*(i_1, \ldots, i_k)$ *denotes the star discrepancy of the orthogonal projection of the sequence* $\{\boldsymbol{x}_n\}$ *on to the appropriate* k*-dimensional subspace of* $[0,1)^s$. *Since* $D_N^* \geq D_N^*(i_1, \ldots, i_k)$ *for* $k \leq s$ *and together with the definition of variation of* f *in*

[4] *For example, if* f *is a 5-dimensional function, then* $V^{(3)}(f; 1, 3, 4)$ *would imply* $V^{(3)} f(x_1, 1, x_3, x_4, 1)$.

the sense of Hardy-Krause, inequality (2.4.9) can also be written as

$$\left| \frac{\sum_{n=1}^{N} f(\boldsymbol{x}_n)}{N} - \int_{[0,1)^s} f(\boldsymbol{u}) d\boldsymbol{u} \right| \leq V(f) D_N^*. \tag{2.4.10}$$

Note that when f is sufficiently smooth[5] on $[0,1)^s$, the Vitali variation (2.4.8) of f on $[0,1)^s$ can be expressed more compactly as

$$V^{(s)}(f) = \int_{[0,1)^s} \left| \frac{\partial^s f}{\partial u_1 \cdots \partial u_s} \right| d\boldsymbol{u} \tag{2.4.11}$$

and $V^{(k)}(f; i_1, \ldots, i_k)$ as

$$V^{(k)}(f; i_1, \ldots, i_k) = \int_{[0,1)^s} \left| \frac{\partial^s f}{\partial u_{i_1} \cdots \partial u_{i_k}} \right|_{u_j = 1, j \neq i_1, \ldots i_k} du_{i_1} \cdots u_{i_k}.$$

Theorem 2.4 was proved by Koksma [50] while the generalization to multidimensional case was proved by Hlawka [38]. The proof of these two theorems can also be found in Kuipers and Niederreiter [53], Hua and Wang [41] and Niederreiter [68].

One important distinction between the error bound given by Koksma-Hlawka inequality and error bound given by Monte Carlo method is that the former is deterministic while the latter is probabilistic. Furthermore, the Koksma-Hlawka bound is only asymptotic in nature. The empirical observation indicates that the number of points required for the asymptotic behavior to begin to operate grows exponentially with the dimensions (see Morokoff and Caflisch [59]). In practice, only a relatively small sample of points will ever be used. The bounds implied by

[5]The function f is sufficiently smooth on $[0,1)^s$ in the sense that the partial derivatives in (2.4.11) exist.

Koksma-Hlawka inequality, as a result, may not be representative of the true errors. In addition, the bounds given by (2.4.10) is not sharp.

Despite the difficulties pertaining to the Koksma-Hlawka bound, this inequality does provide a theoretical justification for the application of low discrepancy sequences to Monte Carlo methods. Several important insights can be gleaned.

- The Koksma-Hlawka inequality effectively separates the integration error into two components: these correspond to the smoothness of the function and the discrepancy of the deterministic nodes used in approximating the function.

- The inequality also implies that for a fixed function, sequences with lower discrepancy should lead to lower integration error. Since the asymptotic discrepancy for a finite sequence of N random points, N grid points and N low discrepancy points are $\mathcal{O}((\log\log N)^{1/2}N^{-1/2})$, $\mathcal{O}(N^{-1/s})$ and $\mathcal{O}(N^{-1}\log^s N)$ respectively, the Koksma-Hlawka inequality implies that the deterministic error bound of these 3 methods are also of the same order as their asymptotic discrepancy. This suggests that for large N and high dimension, the low discrepancy method is the most effective while the grid point method is the least.

- The factor $\log^s N$ in the discrepancy bound of the low discrepancy sequence becomes dominant for high dimensions, suggesting that the efficiency of the low discrepancy sequence deteriorates with dimensions. This is consistent with the computational results shown in later chapters.

- So far much attention has been devoted to the nature of the sequence used in approximating the function, not much has been discussed regarding the function itself. For the inequality to hold, the function must be reasonably smooth and continuous. Discontinuous functions generally violate the hypothesis of bounded variation and hence adversely affect the efficiency of the low discrepancy methods. We will explore this issue in Chapter 5.

Chapter 3

Low Discrepancy Sequences

3.1 Introduction

The purpose of this chapter is to introduce different types of low discrepancy point sets or low discrepancy sequences. Most of these point sets or sequences will be used in the computational studies in subsequent chapters. To have a better understanding of these point sets or sequences, we provide a simple and intuitive approach using diagrams to illustrate various properties of an important class of low discrepancy point sets and low discrepancy sequences known as (t, m, s)-nets and (t, s)-sequences. Many of the popular low discrepancy point sets or low discrepancy sequences can be considered as special cases of the (t, m, s)-nets and (t, s)-sequences. We also generalize the conventional (t, m, s)-nets and (t, s)-sequences so that the generalized class encompasses a broader family of low discrepancy point sets or low discrepancy sequences. For instance, the low discrepancy sequence constructed by Halton [32] belongs to the generalized class while it does not belong to

the conventional (t, s)-sequence.

The rest of the chapter is organized as follows: Section 3.2 introduces various types of low discrepancy point sets and low discrepancy sequences. Section 3.3 describes (t, m, s)-nets and (t, s)-sequences and their generalizations. Good lattice point sets and good point sequences are discussed in Sections 3.4 and 3.5. Section 3.6 concludes the chapter by pointing out the difference between a telescopic and a non-telescopic sequence (point set).

3.2 Low Discrepancy Sequences

In this section we briefly describe a few popular low discrepancy sequences. It is widely believed that the discrepancy of any sequence is bounded by the following conjecture lower bound:

$$D_N^* \geq C_s \frac{\log^s N}{N} + \mathcal{O}\left(\frac{\log^{s-1} N}{N}\right) \tag{3.2.1}$$

where C_s is constant in N but is sequence-dependent and dimension-dependent. Sequences that attain such lower bound is customary known as low discrepancy sequences. Explicit algorithms for constructing these sequences are given by van der Corput, Halton [32], Sobol' [93], Faure [24], Niederreiter [65], Tezuka [101], and Lapeyre-Pagès [78]. Efficient algorithms for generating van der Corput or Halton sequences can be found in Halton and Smith [33], Schatte [89] and Lécot [56]. Implementations of the Faure, Halton, Sobol', and Niederreiter sequences in Fortran programming language are available in Fox [26], Bratley and Fox [13],

Bratley and Fox and Niederreiter [14, 15] and Sobol', Turchaninov and Shukhman [96].

For our purpose, we find it convenient to express the generation of the low discrepancy sequences using the following matrix notation. For arbitrary large R, we define

$$\boldsymbol{C}_i = \begin{bmatrix} c_{i10} & c_{i11} & \cdots & c_{i1\,R-1} \\ c_{i20} & c_{i21} & \cdots & c_{i2\,R-1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{iR0} & c_{iR1} & \vdots & c_{iR,R-1} \end{bmatrix}, \quad \boldsymbol{A}_n = \begin{bmatrix} a_0(n) \\ a_1(n) \\ \vdots \\ a_R(n) \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_R \end{bmatrix}, \quad \boldsymbol{Q}_{ni} = \begin{bmatrix} q_{ni1} \\ q_{ni2} \\ \vdots \\ q_{niR} \end{bmatrix}$$

where the matrix $\boldsymbol{C}_i$ is known as the generation matrix for the i-th component of the low discrepancy sequences and the vector $\boldsymbol{A}_n$ is the base b representation corresponding to the integer n; i.e.

$$n = a_0 + a_1 b + \cdots + a_{R-1} b^{R-1} \equiv (a_{R-1} a_{R-2} \cdots a_0)_b \tag{3.2.2}$$

and $\boldsymbol{Q}_{ni}$ is the vector obtained from matrix multiplication $\boldsymbol{C}_i \boldsymbol{A}_n$. Hence we have

$$\boldsymbol{Q}_{ni} = \boldsymbol{C}_i \boldsymbol{A}_n. \tag{3.2.3}$$

The matrix multiplication and addition is assumed to be carried out over the field $\mathbf{F}_b$. Each component of $\boldsymbol{x}_n = (x_{n1}, \ldots, x_{ns})$ is generated from the vector $\boldsymbol{Q}_{ni}$ as

follows:

$$\begin{aligned} x_{ni} &= \frac{(q_{ni1}q_{ni2}\cdots q_{niR})_b}{b^R} \\ &= \frac{q_{ni1}b^{R-1}+\cdots+q_{niR}}{b^R} \\ &= \sum_{r=1}^{R}\frac{q_{nir}}{b^r}. \end{aligned}$$

Many of the low discrepancy sequences have the above representation. For instance, if the generation matrix is chosen to be

$$c_{ijr} = \begin{cases} 1 & \text{for } j-r=1, \\ 0 & \text{otherwise,} \end{cases} \tag{3.2.4}$$

so that $\boldsymbol{C}_i$ is an identity matrix for $1 \le i \le s$, we obtain the famous van der Corput in base b when $s = 1$ and any prime base b. If

$$c_{ijr} = \begin{cases} 0 & \text{for } 0 \le r < j-1, \\ \binom{r}{j-1}(i-1)^{r-j+1} & \text{for } r \ge j-1, \end{cases} \tag{3.2.5}$$

so that $\boldsymbol{C}_i = \boldsymbol{P}^{i-1}$, $1 \le i \le s$, where $\boldsymbol{P}$ is the Pascal matrix, we obtain the Faure sequence when b is chosen to be the smallest prime base $\ge s$. Other choices of the generation matrices are possible. Niederreiter [65] provides a general construction principle for obtaining the generation matrix that yields sequences with small discrepancy.

The advantage of expressing the generation method as described above is that

efficient algorithm exists for such representation. When $b = 2$, we have a "super-fast" algorithm suggested by Antonov and Saleev [3] which exploits the combinatoric structure of the Gray code [30]. When $b \neq 2$, we also have an efficient algorithm based on a generalization of the Gray code. This is described in Tezuka [101].

3.3 (t, m, s)-Nets and (t, s)-Sequences in Base b

This section discusses the theory related to (t, m, s)-nets and (t, s)-sequences which is first introduced by Sobol' [93] and later formalized by Niederreiter [65]. The (t, m, s)-net and (t, s)-sequence form a general class of low discrepancy sequences in which many of the popular low discrepancy sequences can be considered as special cases. Because of the strict combinatorial property imposed by the net, sequence satisfies the notion of net is guaranteed to have low discrepancy.

Definition 3.1 (Elementary Interval) *An **elementary interval** in base b is an interval E in $[0,1)^s$ of the form*

$$E = \prod_{i=1}^{s} \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right)$$

with $d_i \geq 0, 0 \leq a_i < b^{d_i}$ and a_i, d_i are integers.

An elementary interval E is thus a subinterval of the unit-cube $[0,1)^s$ whose i-th axis has length $1/b^{d_i}$. When we divide the i-th axis into b^{d_i} equal slices and repeat the division for other axes, the subinterval obtained is the elementary interval having volume $b^{-\sum_{i=1}^{s} d_i}$.

Definition 3.2 *Let $0 \leq t \leq m$ be integers. A (t, m, s)-net in base b is a finite point set with b^m points from $[0,1)^s$ such that every elementary interval in base b of volume b^{t-m} contains exactly b^t points of the sequence.*

Definition 3.3 *An infinite sequence of points $\{\boldsymbol{x}_n\} \in [0,1)^s$ is a (t, s)-sequence in base b if for all $k \geq 0$ and $m > t$, the finite sequence $\boldsymbol{x}_{kb^m}$, ..., $\boldsymbol{x}_{(k+1)b^m-1}$ forms a (t, m, s)-net in base b.*

Another definition of the nets introduced by Owen [75] is as follows:

Definition 3.4 *Let s, m, t, b, λ be integers with $s \geq 1, m \geq 0, 0 \leq t \leq m, b \geq 2$, and $1 \leq \lambda < b$. A sequence $\{\boldsymbol{x}_n\}$ of λb^m points is called a (λ, t, m, s)-net in base b if every elementary interval in base b of volume b^{t-m} contains λb^t points of the sequence and no elementary interval in base b of volume b^{t-m-1} contains more than b^t points of the sequence.*

From the above definitions, it is easy to see that a (t, m, s)-net in base b is a $(1, t, m, s)$-net in base b. Also if $\{\boldsymbol{x}_n\}$ is a (t, s)-sequence in base b, then $\boldsymbol{x}_{kb^{m+1}}$, ..., $\boldsymbol{x}_{kb^{m+1}+\lambda b^m-1}$ is a (λ, t, m, s)-net in base b for integers $k \geq 0$ and $1 \leq \lambda < b$.

The importance of the net can also be seen from the following theorem which asserts that if a sequence satisfies a (t, s)-sequence in base b, low discrepancy is ensured.

Theorem 3.1 *For any $N \geq 2$, the discrepancy of the first N points of a (t, s)-sequence in base b satisfies*

$$D_N^* \leq C(t, s, b)\frac{(\log N)^s}{N} + \mathcal{O}\left(\frac{b^t(\log N)^{s-1}}{N}\right) \tag{3.3.1}$$

where

$$C(t,s,b) = \frac{b^t}{s}\left(\frac{b-1}{2\log b}\right)^s$$

if either $s = 2$ *or* $b = 2, s = 3, 4$*; otherwise*

$$C(t,s,b) = \frac{b^t}{s!}\cdot\frac{b-1}{2\lfloor b/2\rfloor}\cdot\left(\frac{\lfloor b/2\rfloor}{\log b}\right)^s.$$

Sobol' [93] describes how to construct (t,m,s)-nets and (t,s)-sequences in base 2. Faure [24] provides a construction of the $(0,m,s)$-net and $(0,s)$-sequence in a prime base greater than or equal to s. Niederreiter [65] generalizes the construction of the Sobol' sequence to arbitrary bases and the Faure sequence to bases that are of prime power greater than or equal to s.

For the remainder of this subsection, we illustrate graphically how the uniformity of a sequence is maintained when the sequence satisfies the net property. We use the construction due to Faure [24] to generate the $(0,s)$-sequences in base b. For simplicity, we only consider the $(0,2)$-sequence so that the points can be plotted on a graph and hence the distribution of the points in $[0,1)^2$ can be seen. By definition, the finite subsequence

$$\boldsymbol{x}_{kb^m}, \ldots, \boldsymbol{x}_{(k+1)b^m-1} \tag{3.3.2}$$

of a $(0,s)$-sequence is a $(0,m,2)$-net in base 2 for all $k \geq 0$ and $m > 0$. More specifically, let consider $k = 8$ and $m = 3$ in sequence (3.3.2). The resulting sequence is therefore a $(0,3,2)$-net in base 2 with $2^3 = 8$ elements. It follows from

Definition 3.2 that every elementary interval in base 2 (or rectangle in this case) with area 2^{-3} contains only one point of this subsequence. The rectangles of interest are of the form

$$\left[\frac{a_1}{2^{d_1}}, \frac{a_1+1}{2^{d_1}}\right) \times \left[\frac{a_2}{2^{d_2}}, \frac{a_2+1}{2^{d_2}}\right) \tag{3.3.3}$$

in $[0,1)^2$ with integers a_i, d_i such that $d_i \geq 0$ and $0 \leq a_i < 2^{d_i}$ for $i = 1, 2$. Another constraint on d_1 and d_2 can be derived by recognizing that for this subsequence, the elementary interval has area 2^{-3} and since the rectangle produced from (3.3.3) has area $2^{-(d_1+d_2)}$, hence we have

$$2^{-(d_1+d_2)} = 2^{-3},$$

for integers $d_1, d_2 \geq 0$. This implies

$$d_1 + d_2 = 3.$$

Since d_1 and d_2 are integers, the above equation yields 4 sets of solutions, namely $(0,3), (3,0), (1,2)$ and $(2,1)$, where the first coordinate refers to d_1 and the second coordinate refers to d_2. It is clear from the above analysis that, in general, there is no unique way of characterizing the elementary intervals in base b. Figure 3.1 demonstrates the 4 possible representations of the elementary intervals in base 2 corresponding to the 4 sets of solution of (d_1, d_2). In this figure and the subsequent figures, the horizontal and vertical axes represent, respectively, the first and second dimension of the point. A remarkable feature is that irrespective of how the

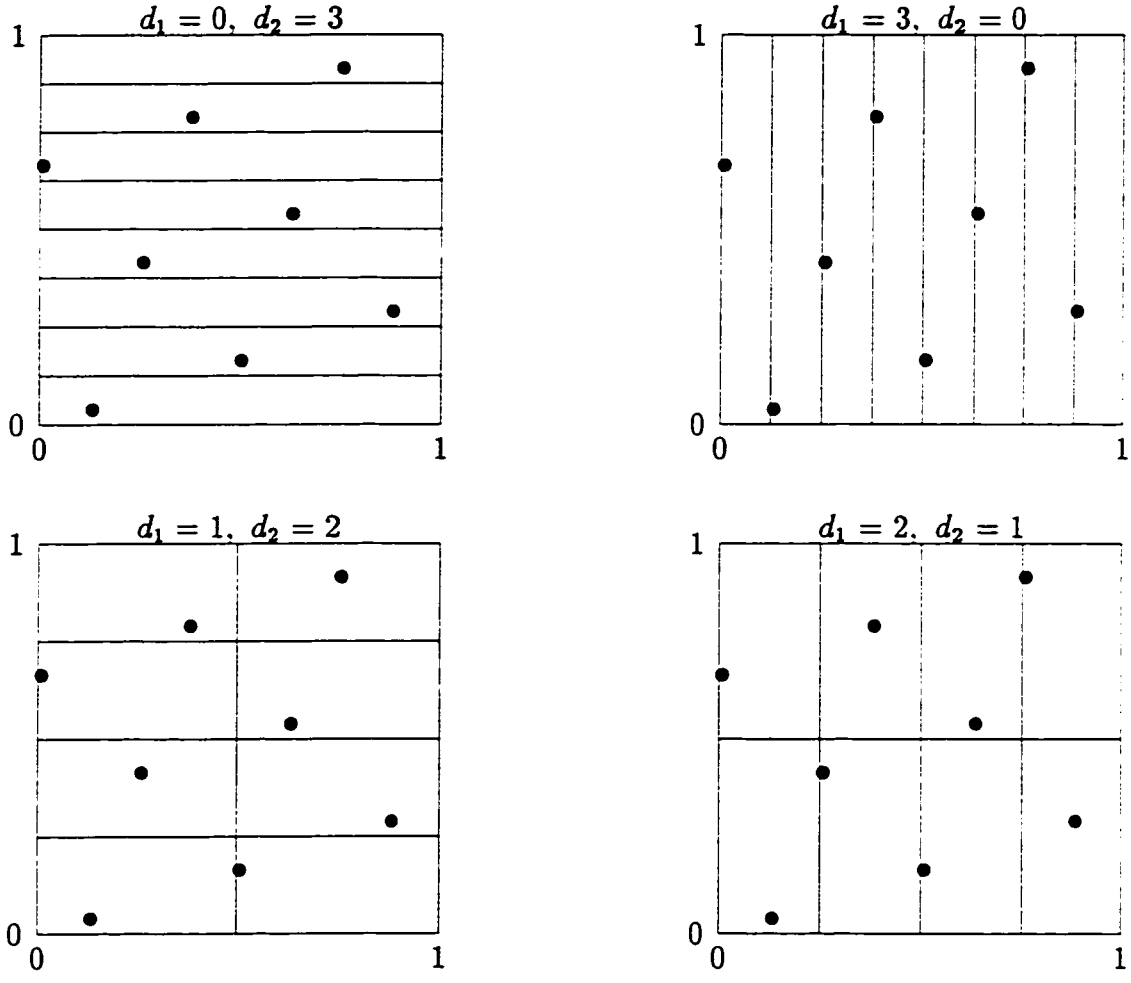


Figure 3.1: 4 Characterizations of Elementary Intervals for the First 8 Points

elementary interval is constructed, as long as the elementary intervals satisfy the necessary condition, each elementary interval will contain exactly one point, as it should.

One compelling advantage of a (t, s)-sequence is that points can be subsequently added in group of b^m points without distorting the uniformity of the sequence. This is in contrast to other techniques such as lattice rules or stratified sampling methods where the number of points has to be preset. To see how the uniformity of the sequence is maintained when more points are added, let us consider introducing an additional 8 points to our existing sequence in the above example. The result is shown in Figure 3.2. The first set of the 8 points are denoted by the "$\bullet$" while the subsequent set of the 8 points are denoted by "$\times$". With 16 points in the sequence, the area of the elementary interval reduces to 2^{-4}. Each revised rectangle still contains only a single point. This should not be surprising since the sequence of 16 points in fact is a $(0, 4, 2)$-net in base 2 ($k = 4$ and $m = 4$ in (3.3.2)). Following the same argument as before, it is easy to derive that there are five possible ways to characterize the elementary intervals as shown in Figure 3.2.

The above phenomenon can be explained as follows: consider the lower right panel of Figure 3.1. Suppose each of the rectangles is cut into 2 identical squares with area 2^{-4}. Sixteen identical squares are produced but only 8 of them contain a point. If points are to be added subsequently while maintaining the overall uniformity, the natural positions for these newcomers are those squares without any points. Consequently, each empty square is successively filled up by the newly added points. When exactly 8 points are added so that each of the 16 squares has

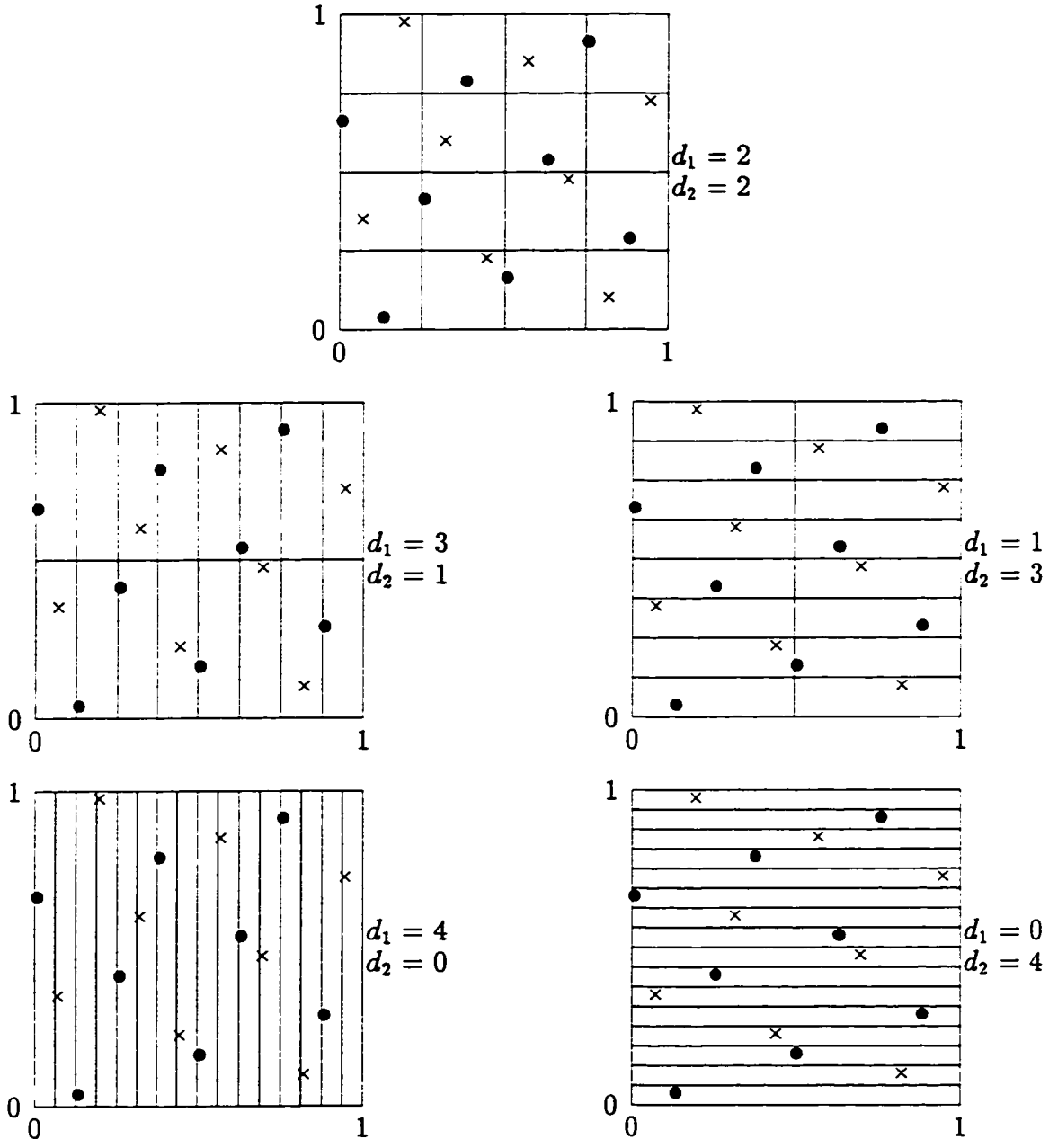


Figure 3.2: Different Characterization of Elementary Intervals with 8 More Points ("•" denotes the first 8 points while "×" denotes the second 8 points)

exactly one point, a $(0, 4, 2)$-net in base 2 is formed as shown in Figure 3.2.

The basic argument applies when even more points are needed. For instance, if points are continually incremented to the existing sequence of 16 points, each of the squares in the upper panel of Figure 3.2 is further subdivided into 2 smaller identical rectangles so that 16 of the rectangles contain no points. To construct the next level of net; i.e. $(0, 6, 2)$-net in base 2, exactly 16 points as opposed to 8 points are needed to fill up each of the empty rectangles. The two panels in Figure 3.3 indicates how the points are distributed throughout the unit square as additional 16 and 32 points are subsequently added to the sequence. The two panels in Figure 3.3 correspond to $(0, 5, 2)$-net and $(0, 6, 2)$-net in base 2 and demonstrate only one particular partition of the elementary intervals.

Note that as points are introduced to the existing sequence, the optimal uniformity is achieved only when the next higher level of net is formed with each elementary interval contains only a single point. The number of points required to accomplish such task unfortunately grows exponentially in base b. In general, to construct a $(t, m+1, s)$-net in base b from a (t, m, s)-net in base b, exactly b^m points are needed. Hence, this is only feasible for low values of b and m. Any finite sequence of N points for which $b^m < N < b^{m+1}$ represents the transition zone. This transient state becomes exponentially long as m increases. The cycle of moving from optimal uniformity to transition zone and then back to optimal uniformity has an important implication on the efficiency of of the low discrepancy sequence for quasi-Monte Carlo integration, particularly when b is large as in the high-dimensional Faure sequence considered in Chapter 4.

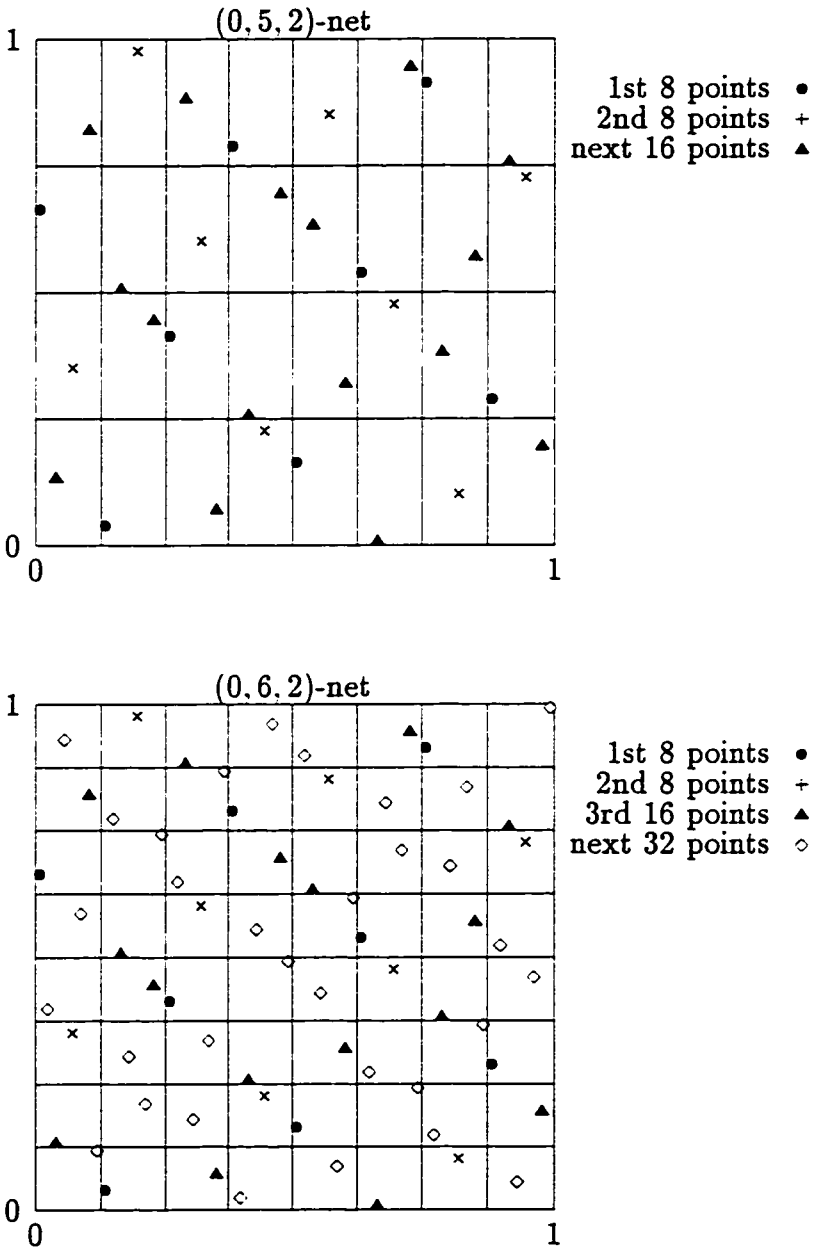


Figure 3.3: Distribution of Points as More Points are Added

The strategy of successively occupying the elementary intervals as points are introduced ensures that the resulting sequence maintains good uniformity for any N even though the optimal number of points is not reached. This property is important in computational practice since in most cases, we do not know in advance how many simulations will be needed. We would prefer to have the flexibility of continuing a simulation until we achieve a given level of precision.

So far we have only considered a special case where each elementary interval contains only a single point ($t = 0$). There exists a special relation between a $(0, m, s)$-net in base b and a (t, m, s)-net in base b for $1 \leq t \leq m$. We now explore such relationship.

The points in each panel of Figure 3.4 are identical to Figure 3.2 except that rectangles are drawn differently. As noted carlier, this sequence corresponds to a $(0, 4, 2)$-net in base 2. However, if we examine the top left panel of Figure 3.4, interestingly a $(1, 4, 2)$-net in base 2 is implied. To see this, first notice that each rectangle of top left panel of Figure 3.4 is composed of 2 disjoint elementary intervals of Figure 3.2 (say top or middle panel). This rectangle therefore has area 2^{-3} and is also an elementary interval in base 2. Second, there are exactly 2 elements in each of these rectangles. Consequently, we have 8 rectangles each with area 2^{-3} and containing 2 points, satisfying all the properties of a $(1, 4, 2)$-net in base 2. Applying the same argument to the top right, lower left and lower right reveals that the sequence is respectively a $(2, 4, 2)$-net, $(3, 4, 2)$-net and $(4, 4, 2)$-net in base 2. This illustrate a remarkable feature of the net theory that different type of nets can be produced by restructuring the elementary intervals differently. More

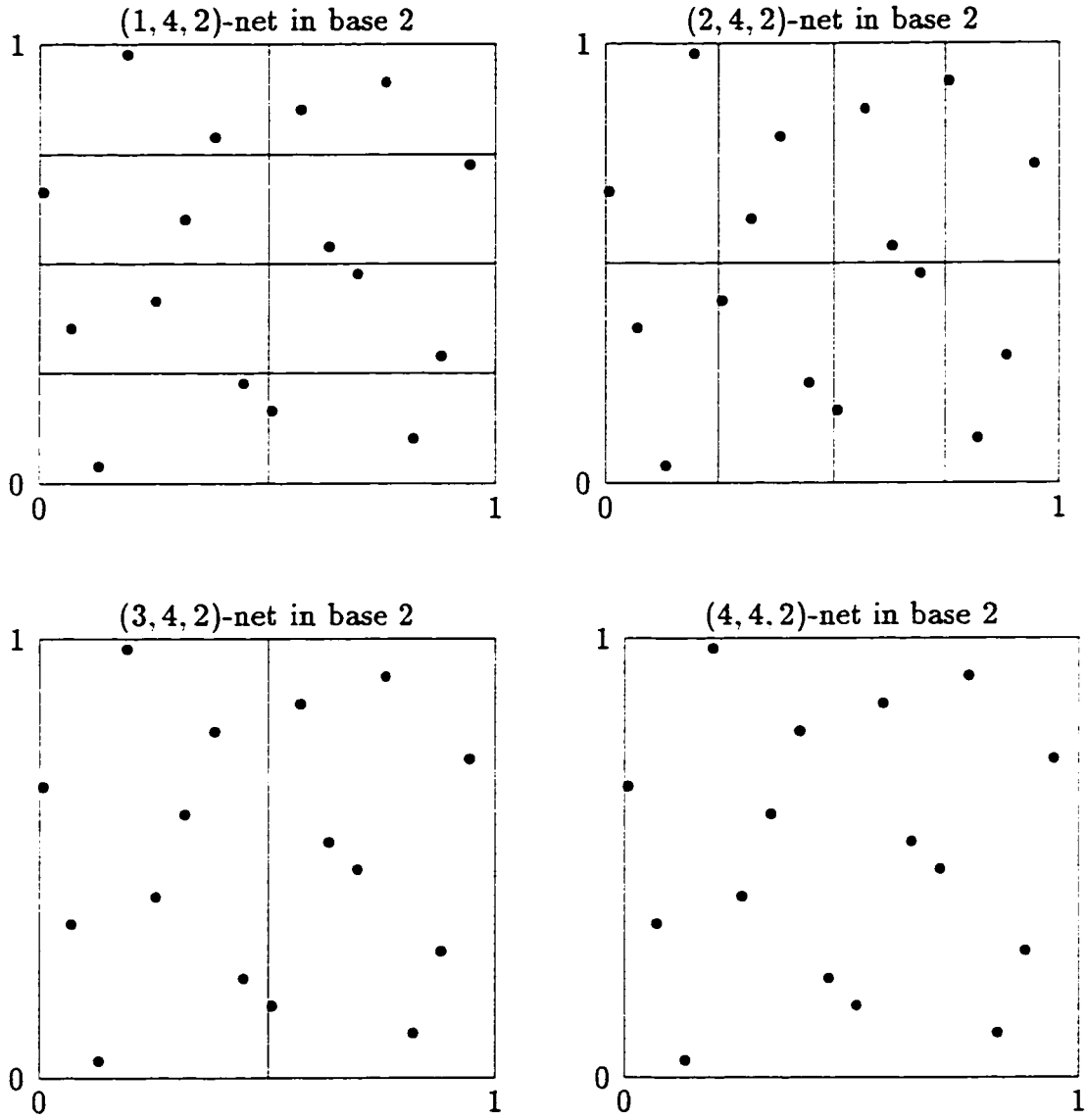


Figure 3.4: Different Ways of Characterizing (t, m, s)-nets

formally, we have the following proposition

Proposition 3.1 *Any (t, m, s)-net in base b is also a (u, m, s)-net in base b for integers $t < u \leq m$ and any (t, s)-sequence in base b is also a (u, s)-sequence in base b for integers $u \geq t$.*

A consequent of the above proposition is that every elementary interval of a (u, m, s)-net in base b can be expressed as a disjoint union of the elementary intervals of a (t, m, s)-net in base b for $t < u$.

The parameter t plays an important role in the theory of net. It determines the number of points in each elementary interval and smaller values of t imply stronger regularity properties. Hence, t is sometimes known as "quality parameter" or "uniformity parameter". When $t = m$, (m, m, s)-net reduces to a random sequence.

By now, it should also be obvious that a (t, m, s)-net in base b and a s-dimensional stratification share many similarities. For instant, a (t, m, s)-net in base b can be interpreted as a s-dimensional stratification in which $[0, 1)^s$ is partitioned into b^{m-t} hyper-rectangles (elementary intervals) of volume b^{t-m} each containing exactly b^t points. The disadvantage of stratification is that partitioning of points in higher dimension become a difficult task. Furthermore, the number of points must be preset before carrying the stratification. If more points are to be added subsequently, the earlier points have to be discarded. The (t, m, s)-net, especially the (t, s)-sequence, on the other hand, provide a systematic way of sampling the points from $[0, 1)^s$ and still ensuring uniformity of the sequence.

3.3.1 Generalized (t, m, s)-net and (t, s)-sequence in Base b

From the conventional definition of the elementary interval, each axis is divided into equal slices based on a common base b. Nevertheless, such restriction can be relaxed by assigning a different base to different axis. The subinterval obtained from this kind of partition becomes a more generalized version of the elementary interval. More formally, Definition 3.1 can be generalized as follow:

Definition 3.5 (Generalized Elementary Interval) *A **generalized elementary interval** in base $\boldsymbol{b}$ is an interval E' in $[0,1)^s$ of the form*

$$E' = \prod_{i=1}^{s} \left[\frac{a_i}{b_i^{d_i}}, \frac{a_i + 1}{b_i^{d_i}} \right)$$

with $d_i \geq 0, 0 \leq a_i < b_i^{d_i}$, a_i, d_i are integers and $\boldsymbol{b} = (b_1, \ldots, b_s)$.

Hence, the generalized elementary interval has volume equal to $\prod_{i=1}^{s} b_i^{-d_i}$. Similarly, we can also obtain a generalized version of (t, m, s)-net and (t, s)-sequence based on the generalized elementary interval as follows:

Definition 3.6 (Generalized $(t, \boldsymbol{m}, s)$-net) *Define vectors $\boldsymbol{b} = (b_1, \ldots, b_s)$, $\boldsymbol{t} = (t_1, \ldots, t_s)$ and $\boldsymbol{m} = (m_1, \ldots, m_s)$ such that each component is integer satisfying $0 \leq t_i \leq m_i$ for $i = 1, \ldots, s$. A $(\boldsymbol{t}, \boldsymbol{m}, s)$-net in base $\boldsymbol{b}$ is a finite sequence with $\prod_{i=1}^{s} b_i^{m_i}$ points from $[0,1)^s$ such that every generalized elementary interval in base $\boldsymbol{b}$ of volume $\prod_{i=1}^{s} b_i^{t_i - m_i}$ contains exactly $\prod_{i=1}^{s} b_i^{t_i}$ points of the sequence.*

Definition 3.7 (Generalized (t, s)-sequence) *An infinite sequence of points $\{\boldsymbol{x}_i\} \in [0,1)^s$ is a $(\boldsymbol{t}, s)$-sequence in base $\boldsymbol{b}$ if for all $k \geq 0$ and $m_i > t_i$, the finite sequence $\boldsymbol{x}_{kn}, \ldots, \boldsymbol{x}_{(k+1)n-1}$ with $n = \prod_{i=1}^{s} b_i^{m_i}$ forms a $(\boldsymbol{t}, \boldsymbol{m}, s)$-net in base $\boldsymbol{b}$.*

The conventional elementary interval, (t, m, s)-net and (t, s)-sequence are trivially obtained from these generalized concepts. For instance, when $b_1 = \cdots = b_s = b$, the generalized elementary interval in base $\boldsymbol{b}$ reduces to conventional elementary interval in base b. Furthermore, if $\sum_{i=1}^{s} m_i = m$ and $\sum_{i=1}^{s} t_i = t$, Definitions 3.6 and 3.7 become the classical (t, m, s)-net and (t, s)-sequence in base b.

To the best of our knowledge, Definitions 3.5, 3.6 and 3.7 are new. The construction of such sequences for arbitrary bases $\boldsymbol{b}$ is not known. The only known situation is when all the bases b_i, $1 \leq i \leq s$ are relatively coprime. Using the Chinese Remainder Theorem, it can be shown when b_i, $1 \leq i \leq s$ are relatively coprime, the resulting generalized sequences with $t_i = 0$, $1 \leq i \leq s$ become the famous Halton sequences. (See Halton [32].)

Now, Proposition 3.1 can be restated as follows:

Proposition 3.2 *Any generalized $(\boldsymbol{t}, \boldsymbol{m}, s)$-net in base $\boldsymbol{b}$ is also a $(\boldsymbol{u}, \boldsymbol{m}, s)$-net in base $\boldsymbol{b}$ where $\boldsymbol{u}$ is an integer vector satisfying $t_i \leq u_i \leq m_i$ for $1 \leq i \leq s$. Any generalized $(\boldsymbol{t}, s)$-sequence in base $\boldsymbol{b}$ is also a $(\boldsymbol{u}, s)$-sequence in base $\boldsymbol{b}$ where $u_i \geq t_i$ for $1 \leq i \leq s$.*

3.4 Good Lattice Point Sets

The *good lattice point (g.l.p.)* set is defined as follows:

Definition 3.8 *Let $(N; z_1, \ldots, z_s)$ be a vector with integral components satisfying $1 \leq z_i < N$, $z_i \neq z_j (i \neq j)$, $s < n$ and the greatest common divisors $(N, z_i) = 1$,*

$1 \leq i \leq s$. *Let*

$$x_{ni} = \left\{ \frac{n}{N} z_i \right\}$$

where $\{y\}$ *denotes the fractional part of* y. *Then the point set* $\{\boldsymbol{x}_n = (x_{n1}, x_{n2}, \ldots, x_{ns}), 1 \leq n \leq N\}$ *is called a* ***lattice point*** *set of the generator* $(N; z_1, \ldots, z_s)$. *If the point set* $\boldsymbol{x}_n$ *has the smallest discrepancy among all possible generating vectors, then the set is called* ***good lattice point*** *set.*

The g.l.p. is introduced by Korobov [51] and Hlawka [38]. They show that with an appropriate choice of the generators $(N; z_1, \ldots, z_s)$, the discrepancy of the point set is $\mathcal{O}(N^{-1+\epsilon})$ for $0 < \epsilon \ll 1$. In Chapter 6, we will discover that this type of point sets is very effective for sufficiently smooth functions.

3.5 Good Point Sequences

The good point sequence is defined as follow:

Definition 3.9 *Let* $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_s) \in \Re^s$. *If the first* N *terms of the sequence*

$$\boldsymbol{x}_n = (\{n\gamma_1\}, \ldots, \{n\gamma_s\}), \qquad \text{for } n = 1, 2, \ldots,$$

has discrepancy $D_N^* = \mathcal{O}(N^{-1+\epsilon})$ *as* $n \to \infty$, *then the point set is called a* ***good point set*** *and* z *and* $\boldsymbol{\gamma}$ *a* ***good point.***

Several choices of good point $\boldsymbol{\gamma}$ have been proposed in the literature. The following are a few samples:

(a) The *square root sequence* with

$$\gamma = (\sqrt{p_1}, \ldots, \sqrt{p_s})$$

where all p_i, $1 \leq i \leq s$, are different primes.

(b) The *cyclotomic field method* with

$$\gamma = \left(\left\{ 2\cos\frac{2\pi}{p} \right\}, \left\{ 2\cos\frac{4\pi}{p} \right\}, \ldots, \left\{ 2\cos\frac{2\pi s}{p} \right\} \right)$$

where p is a prime $\geq 2s + 3$. This is proposed by Hua and Wang [41].

(c) Let p be a prime and $q = p^{1/(s+1)}$, then the good point is obtained as

$$\gamma = (q, q^2, \ldots, q^s).$$

3.6 Conclusion

In this chapter, we introduced different types of low discrepancy point sets and low discrepancy sequences. Before we conclude this chapter, we make the following three remarks which have an important implications in quasi-Monte Carlo methods.

1. Recall that in Chapter 2, we discuss the difference between a low discrepancy point set and a low discrepancy sequence. We noted that a low discrepancy sequence has the flexibility of introducing more points to the existing sequence while still maintain low discrepancy. This is a desirable property in practice

since we can keep increasing the number of point evaluations until a required precision is reached while utilizing all the previously calculated values. On the other hand, using a low discrepancy point set would require redo the entire computation every time the number of point evaluations is changed. The flexibility of the sequence is achieved at the expense of increasing the discrepancy. This implies that in situation where such flexibility is not needed, it is preferred to use a point set since we can achieve a better discrepancy bound when N is fixed. Examples of low discrepancy sequences are the Halton, Faure, Sobol', Niederreiter and the good points while examples of low discrepancy point sets are the Hammersley points and the good lattice points.

2. Another feature of the low discrepancy sequences (or point sets) is the distinction between telescopic and non-telescopic sequences (point sets). A sequence is telescopic when it satisfies the following property: Suppose we have already generated a s-dimensional low discrepancy sequence, $\boldsymbol{x}_n = (x_{n1}, x_{n2}, \ldots, x_{ns})$, $n \geq 1$. If a $(s+1)$-dimensional low discrepancy sequence can be produced by stacking an additional component $x_{n,s+1}$, $n \geq 1$, to the existing s-dimensional low discrepancy sequence, we describe the sequence as telescopic. Examples of telescopic sequences (point sets) are the Sobol', the Halton, and the good points. Faure sequence, on the other hand, is non-telescopic.

3. The distinction between nominal dimension and effective dimension. Loosely speaking, for a function with nominal dimension s, the effective dimension is approximately the same as the nominal dimension when all the compo-

nents are of equal importance. When the function is dominated by a few components, the effective dimension is less than the nominal dimension. A more formal discussion of this issue is given in Caflisch, Morokoff and Owen [16] where the effective dimension is defined to be the smallest subset of the components that captures 99% of the total variance. Note that in most finance applications, the effective dimensions are in general considerably less than the nominal dimensions. Furthermore, the dominating components are usually the first few dimensions. For most other problems, the dominating components can be any subset of the dimensions.

Because of the nature of telescoping, the telescopic sequences would have greater uniformity in the initial components of the sequences. Non-telescopic sequences require constructing the entire sequences from scratch and hence have a better uniformity when all the dimensions are considered together. The implication of this property is that non-telescopic sequences would be more efficient applying to problems where the effective dimension is approximately the same as the nominal dimension. When the effective dimension is a lot smaller than the nominal dimension, the telescopic sequences would be preferred. In particular, the initial dimensions of the telescopic sequences should be used to estimate those dominating components.

Chapter 4

Randomization Techniques

4.1 Introduction

In the first part of this chapter, we evaluate the uniformity of the sequences by examing the two-dimensional orthogonal projections of the low discrepancy sequences. The evidences from these investigations suggest that undesirable patterns behaviour of the points exist when we examine subsets of the dimensions.

Until recently a major drawback of the classical LD approach has been the absence of a reliable practical error bound. Even though there exists a deterministic upper bound, this theoretical bound significantly overestimates the actual error in practice. This is in contrast to the crude Monte Carlo method for which a statistical estimate of the error is readily available. The purpose of the second part of this chapter is to investigate a modification of the technique proposed by Owen [74] for overcoming this problem. We investigate the effectiveness of this approach and provide numerical illustrations which compare it to Owen's approach in the case of

low-dimensional problems.

The rest of the chapter is organized as follows: Section 4.2 evaluates the uniformity of the sequences by examing their two-dimensional orthogonal projections . Section 4.3 describes a technique developed by Owen which provides a probabilistic error bound for quasi-Monte Carlo by randomizing the points in a such a way to preserve the low discrepancy property. In Subsection 4.3.1, we propose a simplification to Owen's randomization technique that is feasible for high dimensional problems. Section 4.4 discusses two different ways of obtaining the variance of the estimated values from the randomized nets (or sequences). Section 4.5 describes the class of complex derivative securities that we will use for our numerical calculations. The numerical comparisons, which consist of two parts, are conducted in Section 4.6. The first part (see Subsection 4.6.1) deals with low dimensional examples so that the efficiency of our proposed randomization can be compared and assessed to Owen's randomization technique. Statistical tests are also provided to validate our studies. The second part (see Subsection 4.6.2) applies the proposed method to higher dimensional examples. Section 4.7 concludes the chapter.

4.2 Uniformity of Low Discrepancy Sequences

In the last chapter, we discussed the uniformity of sequence in high dimension and noted that the star discrepancy, D_N^*, is a measure of uniformity. Unfortunately, computing D_N^* is only feasible for dimensions as low as two or three and for small number of point sets. Hence we need other methods to assess the uniformity. A common approach is to examine the orthogonal projections of the sequences.

This method provides a mean of investigating the pairing of the points and is restricted to two dimensions. This provides useful information on the uniformity of the sequences. We discuss this method in the following subsection.

4.2.1 Orthogonal Projections

In this subsection, we provide evidence that undesirable patterns occur in low discrepancy sequences. Since it is not possible to plot the points in $[0,1)^s$, for $s > 2$ or 3, we use a simpler approach by examining the two-dimensional orthogonal projection of the low discrepancy points. We are assuming that if a sequence is uniformly dispersed in $[0,1)^s$, then any two-dimensional orthogonal projections should also be uniformly dispersed. For instance, Figure 4.1 plots the first and second coordinates of a $(0,4,7)$-net in base 7. (We use Faure's generation algorithm for obtaining these nets.) These points appear to be uniformly dispersed throughout the unit-square. Figure 4.2 provides a similar comparison except that random points are generated. From the graph, one can see that the random points tend to cluster and tend to have gaps. These are typical features of random points.

Even though $(0,s)$-sequences in high dimensions still maintain low discrepancy, undesirable features exist when we focus on their orthogonal projections. In Figure 4.3 we plot nine pairs of the orthogonal projections of a $(0,3,19)$-net in base 19. These nine pairs were selected at random. By merely increasing the dimensions from seven to nineteen, the orthogonal projection reveals an interesting characteristic of the nets. The graph clearly suggests that undesirable correlation exists between these points. Such patterns have also been pointed out by Morokoff and Caflisch

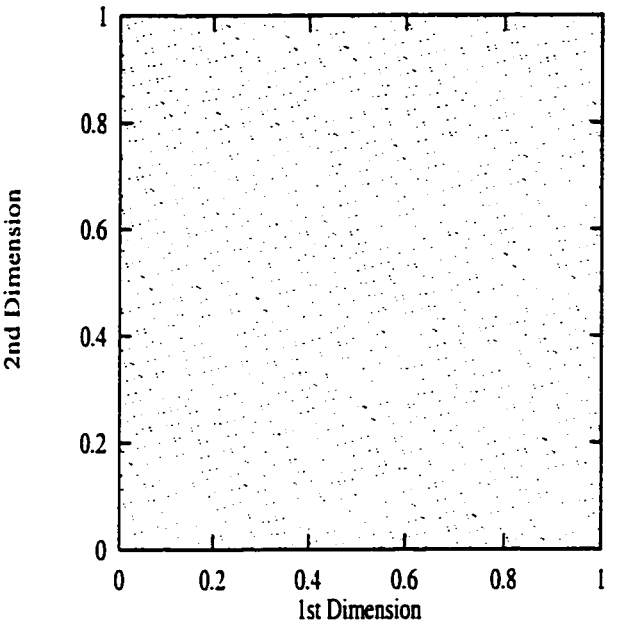


Figure 4.1: Orthogonal Projection of a $(0, 4, 7)$-Net

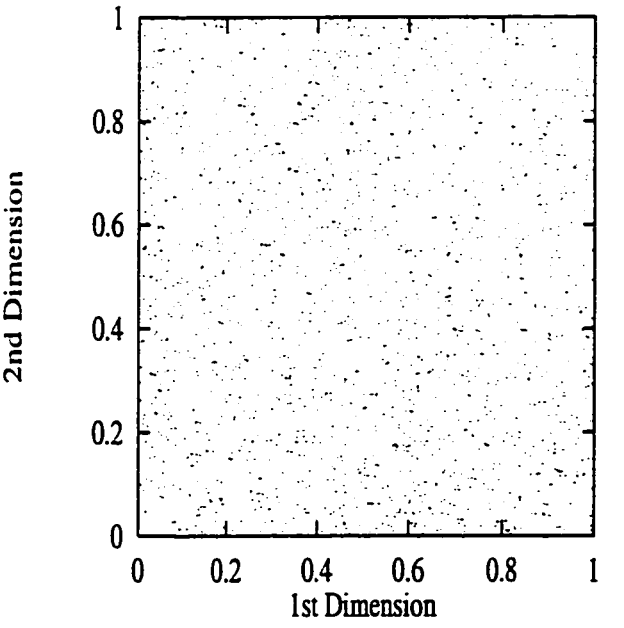


Figure 4.2: Orthogonal Projection of a 7-dimension Random Points

[59] and Boyle, Broadie and Glasserman [10]. As argued by Morokoff and Caflisch [59], this phenomenon is due to the large prime base used to generate $(0, m, s)$-nets and can be explained by Figure 4.4. The grid shows subsets of the elementary intervals with volume 19^{-3} for the subsets of the points in the first panel of Figure 4.3. From net property, the $(0, 3, 19)$-net in base 19 must have exactly one point in each elementary interval of volume 19^{-3}. This is confirmed by Figure 4.4. The points between successive elementary intervals, on the other hand, form a regular pattern and are not uniformly distributed within these intervals. This behaviour leads to the highly correlated structure as shown in Figure 4.3.

It should be pointed out that such patterns are not unique to Faure sequences. They exist in other type of sequences. For telescopic sequences, this kind of pattern only happens in higher dimensions due to the way that these sequences are telescoped. For instance, Figure 4.5 provides an orthogonal projection of the 7-th and 8-th dimensions for the first 5491 Halton points. The corresponding pair of prime numbers for generating these two dimensions are 17 and 19. It was discussed in the previous chapter that the first $N = b_1^{m_1} b_2^{m_2} \cdots b_s^{m_s}$ points of the Halton sequence is a generalized $(\mathbf{0}, \boldsymbol{m}, s)$-net where b_i is the i-th prime number used to generate the i-th dimension of the Halton sequence. In Figure 4.5, there are $5491 = 17^2 \cdot 19$ Halton points. This implies that the two-dimensional elementary interval of the form

$$\left[\frac{i}{17^2}, \frac{i+1}{17^2}\right) \times \left[\frac{j}{19}, \frac{j+1}{19}\right)$$

where $0 \leq i < 17^2$ and $0 \leq j < 19$ should contain exactly one point. This is

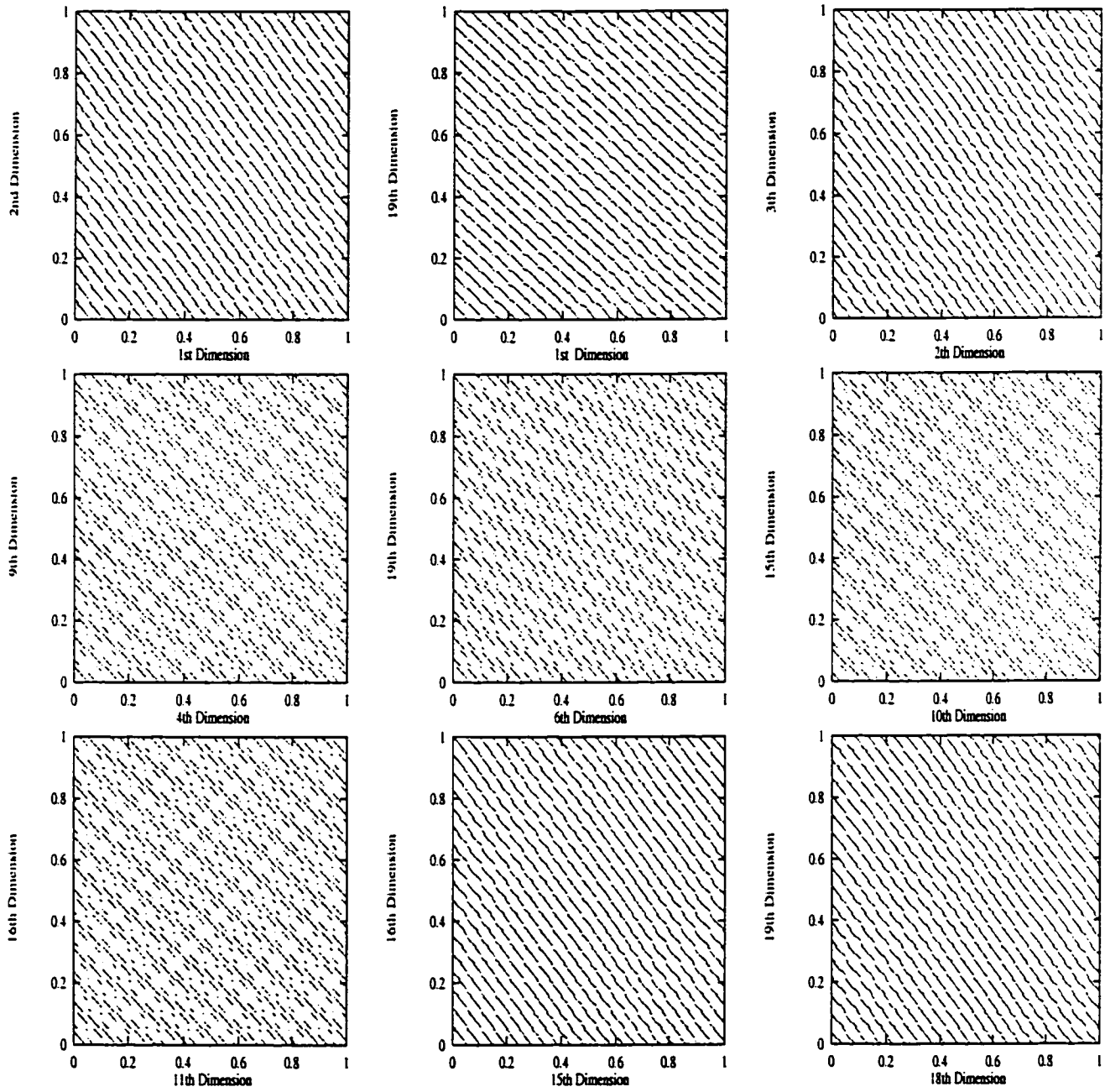


Figure 4.3: Orthogonal Projection of a Classical $(0, 3, 19)$-Net

confirmed in Figure 4.6 where subset of the points in intervals $[0, 5/17^2) \times [0, 1)$ is extracted. The horizontal axis is divided into five equal slices of length $1/17^2$ and the vertical axis is divided into 19 equal slices of length 1/19. Similar to the Faure sequence, the points between the adjacent elementary intervals are arranged in a specific pattern that give rise to the correlated structure shown in Figure 4.5. The use of orthogonal projections to study Halton sequences was suggested by Braaten and Weller [12]. The orthogonal projection for other types of sequences can be found in Morokoff and Caflisch [59].

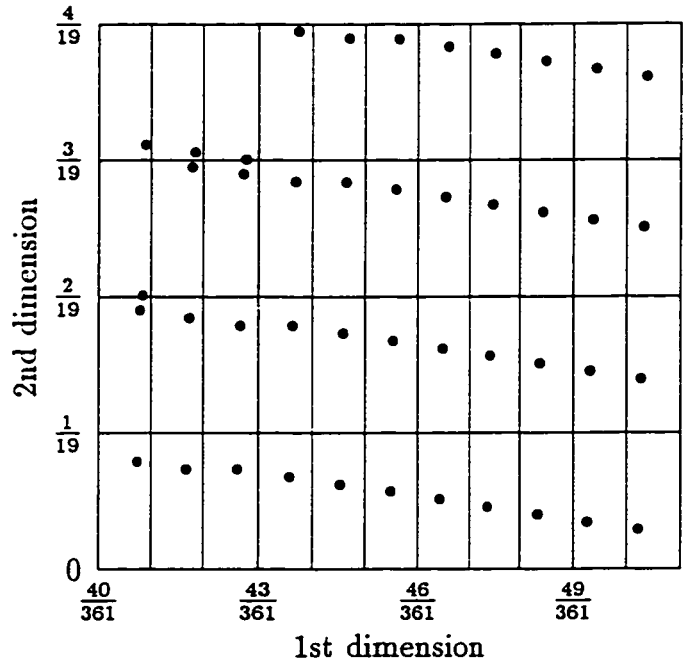


Figure 4.4: Blowup of Subsets of $(0, 3, 19)$-Net from First Panel of Figure 4.3

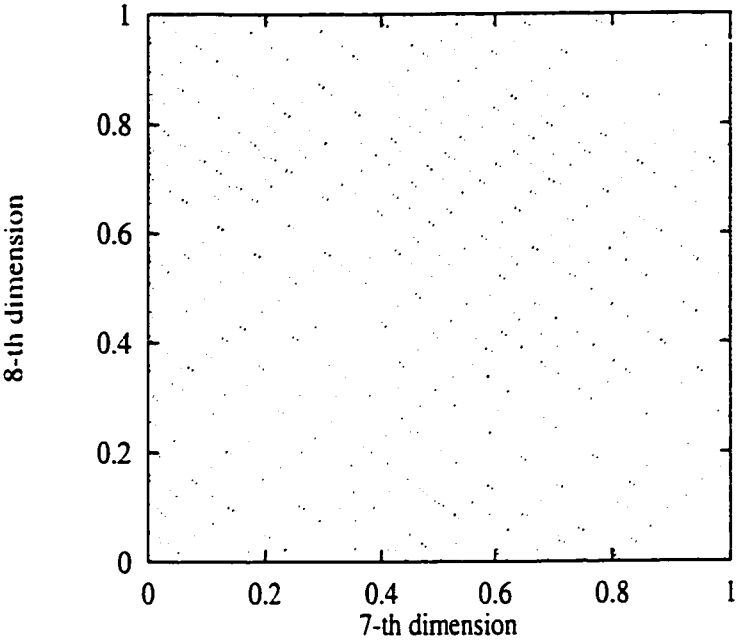


Figure 4.5: Orthogonal Projection Halton Sequence (7-th and 8-th Dimensions)

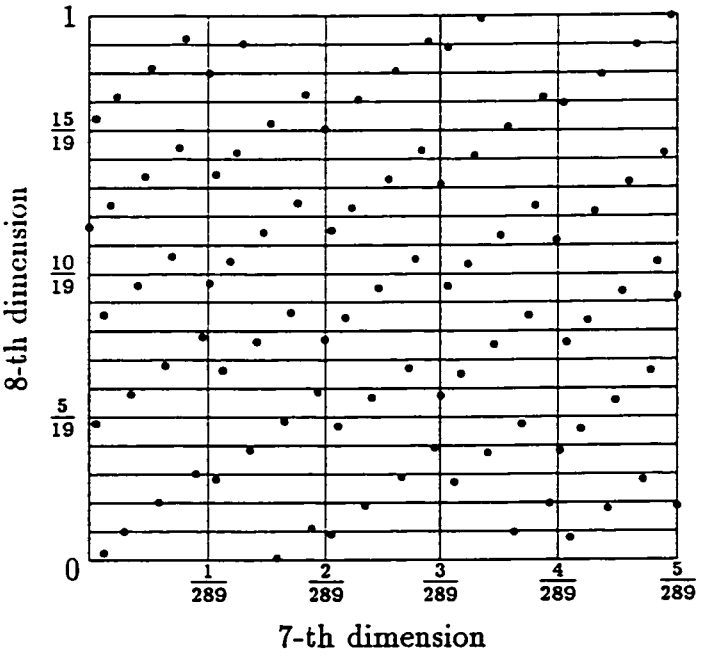


Figure 4.6: Subsets of Halton Points in Figure 4.5

4.3 Randomization

In this section we describe the scrambled (or randomized) (t, m, s)-nets or (t, s)-sequences proposed by Owen [74]. In the following subsection, we describe our modifications of Owen's method. The idea of combining Monte Carlo and LD methods has been proposed by several authors. Cranley and Patterson [19] first introduced the technique in the context of number-theoretic methods. Braaten and Weller [12] randomly permute the Halton sequence. Joe [45] randomizes lattice rules. Faure [25] provides an optimal permutation for a one-dimensional low discrepancy sequence.

Our interpretation of Owen's randomization technique may be described as follows: Suppose $\{\boldsymbol{A}\}$ is a (t, m, s)-net or a (t, s)-sequence in base b. Let $\boldsymbol{A}_n = (A_{n1}, A_{n2}, \ldots, A_{ns})$ denote the n-th term in the net (or sequence). Each component of $\boldsymbol{A}_n$ can be expressed in its base b representation as

$$A_{ni} = \sum_{j=1}^{\infty} a_{nij}\, b^{-j}$$

where $0 \leq a_{nij} < b$ for all n, i, j.

A scrambled (or randomized) version of $\{\boldsymbol{A}\}$ is a net (or sequence) $\{\boldsymbol{X}\}$ with components $\boldsymbol{X}_n = (X_{n1}, \ldots, X_{ns})$ defined in terms of a_{nij} as follow:

$$\begin{aligned} X_{ni} &= \sum_{j=1}^{\infty} x_{nij} b^{-j} \\ &= \frac{\pi_i(a_{ni1})}{b} + \frac{\pi_{ia_{ni1}}(a_{ni2})}{b^2} + \cdots + \frac{\pi_{ia_{ni1}\cdots a_{nik-1}}(a_{nik})}{b^k} + \cdots . \end{aligned} \tag{4.3.1}$$

The functions π are random permutation of the digits $\{0, 1, ..., b-1\}$ and are mutually independent with each permutation function uniformly distributed over $b!$ permutations. The first digit x_{ni1} in the base b expansion of the scrambled sequence X_{ni} is obtained by permuting a_{ni1} using the randomly chosen permutation function π_i. The second digit x_{ni2} is obtained by permuting a_{ni2} using the appropriate randomly generated permutation function $\pi_{ia_{ni1}}$ which depends on the value of first digit; i.e., a_{ni1}. In general, the permutation function applied to the k-th digit a_{nik} depends on the first $k-1$ values a_{nij}, $j = 1, \ldots, k-1$.

The scrambled sequences $\{\boldsymbol{X}_n\}$ defined above not only inherit the equidistribution properties of the unscrambled sequences $\{\boldsymbol{A}_n\}$, each individual point of the sequence is also uniformly distributed on $[0,1)^s$. This implies that the sample estimate $\frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{X}_n)$ is an unbiased estimator for $\int f(\boldsymbol{X})d\boldsymbol{X}$. These two properties follow from the following two propositions of Owen [74].

Proposition 4.1 *If $\{\boldsymbol{A}_n\}$ is a (t, m, s)-net in base b, then $\{\boldsymbol{X}_n\}$ is a (t, m, s)-net in base b with probability 1. If $\{\boldsymbol{A}_n\}$ is a (t, s)-sequence in base b, then $\{\boldsymbol{X}_n\}$ is a (t, s)-sequence in base b with probability 1.*

Proposition 4.2 *Let $\{\boldsymbol{X}_n\}$ be a randomized (t, m, s)-net or (t, s)-sequence in base b as described in equation (4.3.1). Then each element $\{\boldsymbol{X}_n\}$ has the uniform distribution on $[0,1)^s$. That is, for any Lebesgue measurable $\mathcal{G} \subseteq [0,1)^s$, $P(\boldsymbol{X}_n \in \mathcal{G}) = \lambda_s(\mathcal{G})$, the s-dimensional Lebesgue measure of $\mathcal{G}$.*

As a corollary to Proposition 4.1, if $\{\boldsymbol{A}_n\}$ is a (λ, t, m, s)-net in base b, then $\{\boldsymbol{X}_n\}$ is also a (λ, t, m, s)-net in base b with probability 1.

Several important asymptotic results pertaining to the randomized nets have also been obtained in a series of papers by Owen [75, 76] and Hickernell [37]. For instance, Owen [76] shows that for a sufficiently smooth integrand f; i.e., the mixed partial derivative $g(\boldsymbol{x}) = \partial^s f(\boldsymbol{x})/\partial \boldsymbol{x}^s$ is Lipschitz continuous which means that $|g(\boldsymbol{x}) - g(\boldsymbol{x}^*)| \leq B||\boldsymbol{x} - \boldsymbol{x}^*||^\beta$ for some $B \geq 0$, $\beta \in (0, 1]$ and $||z||$ is the Euclidean norm of z, the variance of the estimate of the randomized net is of order $N^{-3}\log^{s-1} N$ as $N = \lambda b^m \to \infty$. Thus the integration errors are of order $N^{-3/2}\log^{(s-1)/2} N$ in probability, which compares favourably to the rate $N^{-1}\log^{s-1} N$ attained by unrandomized nets. However, extreme care must be taken when interpreting these two asymptotic rates. The former rate describes the average case over random permutations for a fixed function f. The latter rate, on the other hand, describes the worst case over functions for a fixed set of integration points. Note that the worst case result remains valid for randomized nets. This implies that it never "hurts" to scramble the nets. In the more favourable situation, we can achieve a superior rate of $N^{-3/2}\log^{(s-1)/2} N$ for smooth functions while in the worst case, we still have the upper error bound of the Koksma-Hlawka inequality. A further advantage of scrambling the nets lies in the ease of estimating the attained accuracy. We will explore this issue in greater detail in Section 4.4.

Theorem 3 of Owen [75] (or Theorem 1 of Owen [76]) also establishes that for any square-integrable integrand f, the variance of the estimate based on a scrambled $(0, m, s)$-net is never more than $e \approx 2.718$ times the Monte Carlo variance while the variance of the scrambled $(\lambda, 0, m, s)$-net is never more than $1 + e \approx 3.718$ times the Monte Carlo variance. Thus the scrambled net variance cannot be much worse

than the Monte Carlo variance.

Another asymptotic rate studied by Hickernell [37] is that the worst case integrand of bounded variation, averaged over permutations of a net, attains a rate of $N^{-1}\log N^{(s-1)/2}N$. Comparing to the asymptotic rate achieved by scrambled nets on smooth integrands, this rate is a factor of $N^{-1/2}$ less efficient. Hence, if the integrand is chosen pessimistically after observing how the net was randomized, no real improvement can be expected from scrambling the nets.

For both scrambled and unscrambled nets, the asymptotic rate cannot be expected to set in until $N = b^{t+s}$. For $(0, m, s)$-net in base b, this implies that $N \approx s^s$. The number of sample points required therefore becomes unrealistically large even for moderate dimensions. Hence the benefit of the (scrambled) nets seems unattainable. The nets nevertheless can still be very useful in many applications since in most cases, the effective dimensions of the problems are considerable less than the nominal dimensions. This is certainly to be expected in typical finance applications. This also partially accounts for the success reported in finance articles (see e.g. Caflisch, Morokoff and Owen [16], Joy, Boyle and Tan [46], Ninomiya and Tezuka [72] and Paskov and Traub [80]) even for very high-dimensional problems, which seems to contradict the findings in other fields (see e.g. Bratley, Fox and Niederreiter [68] and Fox [26]).

4.3.1 Partial Randomization

The implementation of Owen's randomization procedure can be represented graphically in Figure 4.7. It is convenient to describe Owen's technique in terms

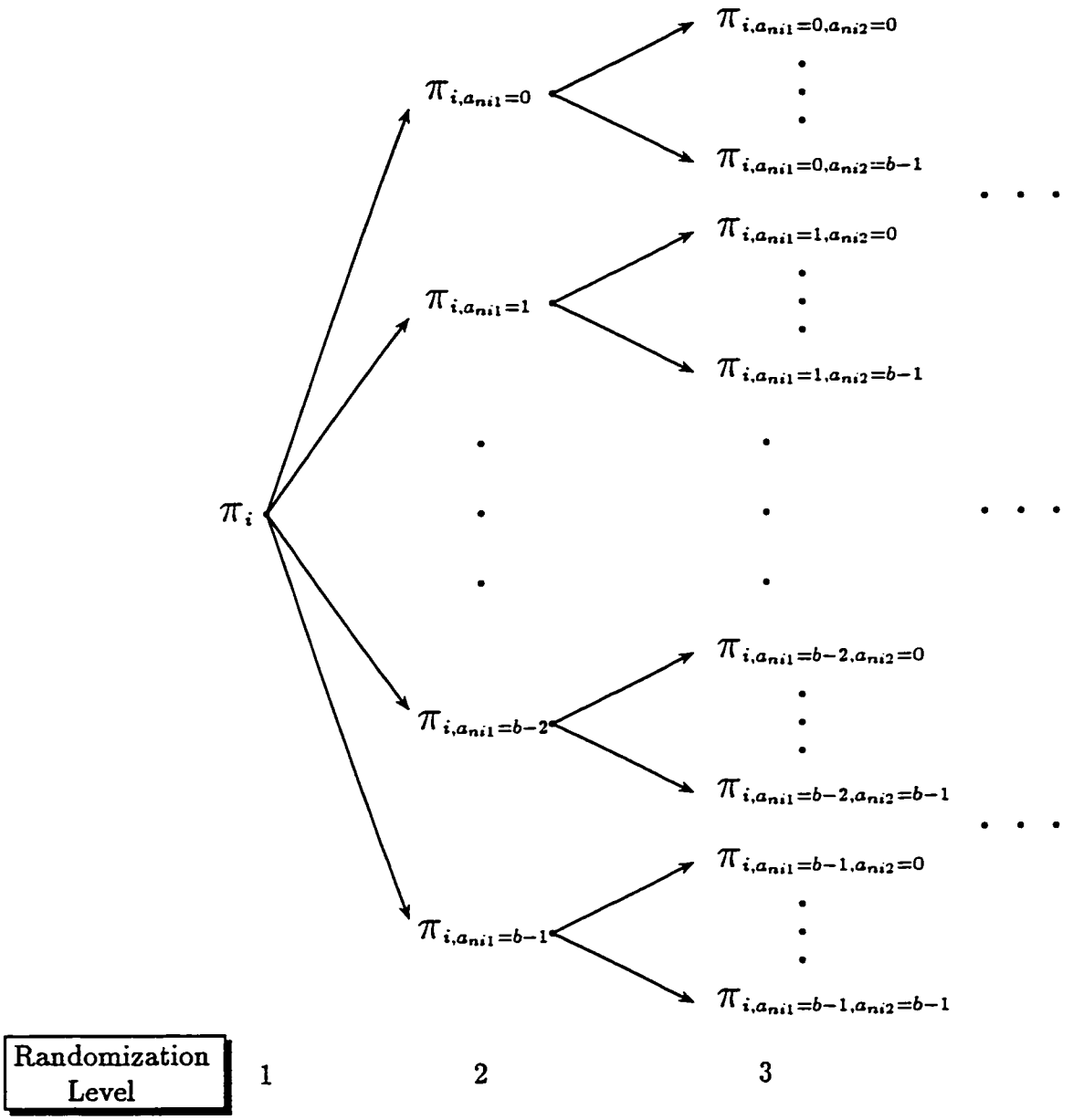


Figure 4.7: Implementation of Owen's Randomization Algorithm

of levels of randomization. In the first level of randomization, we require only one particular permutation function π_i to permute the first digit in the base b expansion of A_{ni}. In the second level of randomization, there are b possible choices of permutation functions $\{\pi_{i,a_{ni1}=0}, \pi_{i,a_{ni1}=1}, \cdots, \pi_{i,a_{ni1}=b-1}\}$, depending on the value of a_{ni1}. In general, there are b^{k-1} possible choices of permutation functions in the k-th level of randomization. The appropriate permutation function used to permute the k-th digit a_{nik} is determined by the $k-1$ values $\{a_{nij}, j = 1, \ldots, k-1\}$. This illustrates the "path-dependency" of the underlying randomization algorithm.

This particular property of the randomization functions implies two potential difficulties in implementing the algorithm. The first problem is that the expansion of X_n must be truncated at some finite k^*. One choice, suggested by Owen, is to take k^* large enough so that b^{-k^*} is small compared to the error committed in truncating the expansion. An alternate choice is to take $k^* = M$ if at most b^M points will ever be used.

The second practical issue is the memory storage problem. Suppose the expansion of A_n is truncated at k^*, Figure 4.7 indicates that scrambling a s-dimensional net (or sequence) requires $s(1 + b + b^2 + \cdots + b^{k^*-1}) = s\frac{b^{k^*}-1}{b-1}$ independent permutations. For large s, large b or large k^*, the underlying algorithm is very memory intensive and hence computationally infeasible. This also explains why the numerical examples in Owen [74] and Owen and Tavella [77] are based on low dimensional problems (such as 6 and 10).

In this chapter, we consider a modification of Owen's randomization technique so that we can apply the randomized technique to problems with higher dimensions.

We now describe our proposed modification. Suppose, for $1 \leq i \leq s$, the first k digits $\{a_{ni1}, a_{ni2}, \ldots, a_{nik}\}$ have already been randomized according to Owen's randomization technique. To proceed to the next level of randomization, instead of using $\pi_{ia_{ni1}\cdots a_{nik}}$ to scramble the digit a_{nik+1} as suggested by Owen's randomization technique, we use $\pi_{ia_{ni1}\cdots a_{nik-1}}$ to permute a_{nik+1}. In fact, the same set of permutation functions $\pi_{ia_{ni1}\cdots a_{nik-1}}$ are used to permute the digits $\{a_{nij}, k \leq j \leq k^*\}$. Therefore, the above algorithm is consistent with Owen's randomization technique only up to level k. The path-dependency of the permutation functions is destroyed after the k-th level. Hence we have $\pi_{ia_{ni1}\cdots a_{nik-1}} = \pi_{ia_{ni1}\cdots a_{nik}} = \pi_{ia_{ni1}\cdots a_{nik^*}}$. We denote this method as randomization of level k. In the trivial case where $k = 0$, no randomization is performed. As the randomization level increases, the proposed randomization converges to Owen's technique. At the other extreme where $k = k^*$, the randomization of level k converges to Owen's algorithm. In this context, when $0 < k < k^*$, the randomization of level k can be considered as a partial randomization technique.

The advantage of using "partial" randomization rather than the "full" randomization lies in its ease of implementation. For the randomization of level k, we only need $s\frac{b^k-1}{b-1}$ permutation functions to permute an s-dimensional net (or sequence). The number of permutation functions required is dramatically reduced when k is small relative to k^*. This allows us to scramble the sequence in a much higher dimensions where it would not be feasible under full randomization. Partial randomization, however, does not come at no cost. Suppose the m-th and n-th terms of the i-th component of the sequence share the same $k-1$ digits in base

b expansion of A_{mi} and A_{ni}; i.e., $a_{mij} = a_{nij}$, for $1 \leq j \leq k-1$. These two terms will be scrambled by the same set of permutation functions regardless of the values of a_{mij} and a_{nij}, for $k \leq j \leq k^*$. This destroys the independency of the digits after the k-th expansion. Hence, Proposition 4.2 need not hold for the scrambled sequences generated from the partial randomization technique. However, since the first k digits are scrambled correctly in the sense of Owen's randomization technique, the maximum deviation between the scrambled point using the partial randomization of level k and the corresponding point using the full randomization is $(b-1)/b^{k+1}$. The above argument assumes that the permutation functions used to scramble the first k digits are identical in both techniques. When b is large, the maximum difference $(b-1)/b^{k+1}$ will be insignificant even when k is small. Hence, the independency issue becomes much less of a concern for high dimensional nets (or sequences) and these are the ones we are primarily interested. This is certainly true for $(0, m, s)$-nets where the nets can exist only if $b \geq s-1$. As an illustration, consider scrambling a $(0, 4, 100)$-net in base 101. Since there are only 101^4 points in this net, we let $k^* = 4$. If full randomization were used to scramble the net, we would require 104,060,400 permutation functions. On the other hand, consider partial randomization of level 2. In this case, we only need $100 \times 102 = 10{,}200$ permutation functions. This results in a significant reduction of 10,202 times in the number of permutation functions required to scramble the net. The reduction of the permutation functions is achieved at the expense of giving up the independency in the third and fourth digits of the expansion. The effect is negligible since for each point, the maximum deviation between the full and partial randomizations of level

2 is $100/101^3 \approx 0.000097$. In other words, the corresponding scrambled point for the full and partial randomizations of level 2 can only differ after the third decimal places. This provides an intuitive justification for using partial randomization. By giving up total randomizing at the third digit and beyond, we achieve a substantial reduction in the number of permutation functions. This allows us to apply the randomization technique to much higher dimensions. Furthermore, the difference occurs at the terms in the expansion (in base b) that only have a minor role in determining the value of the points.

4.3.2 Evaluation of Uniformity

In Subsection 4.2.1, we assessed the uniformity of the low discrepancy sequences based on the orthogonal projections. We observed that undesirable behaviour of the points exists for the classical $(0, s)$-sequences.

In this subsection, we apply the same technique to the randomized sequences. First, we use the proposed partial randomization of level 2 to the same sets of points reported in Figure 4.3. These correspond to the first panel in Figure 4.3. Figure 4.8 shows subsets of the elementary intervals after randomization. The randomized procedure effectively destroys the regular structure displayed in Figure 4.4 while retains the property that each elementary interval still contains a single point. Figure 4.9 gives the same set of the orthogonal projections for the randomized $(0, 3, 19)$-net in base 19. The randomized net appears to eliminate the regularities observed in the classical $(0, 3, 19)$-net. The randomized points are more uniformly dispersed throughout the unit square and do not follow any specific structure.

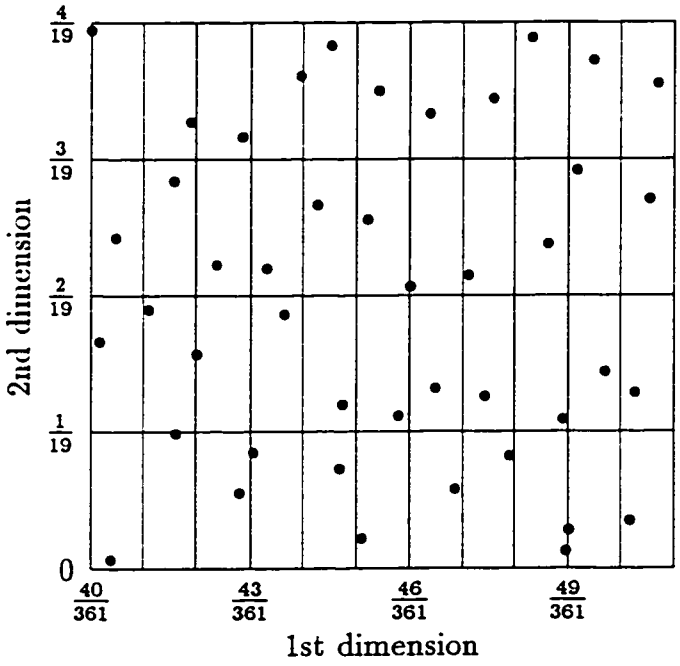


Figure 4.8: Blowup of Subsets of Randomized $(0, 3, 19)$-Net

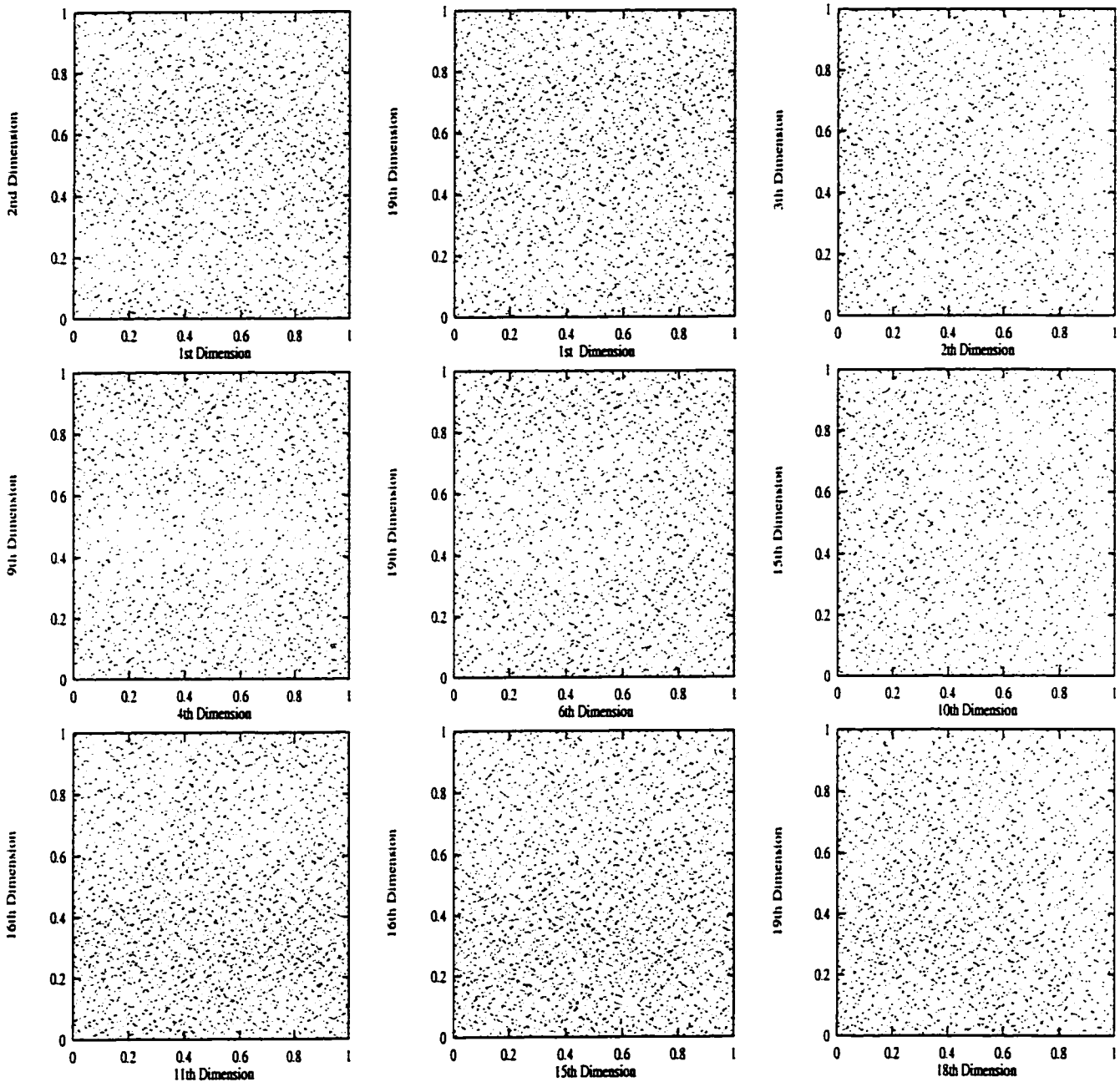


Figure 4.9: Orthogonal Projection of a Randomized $(0, 3, 19)$-Net

4.4 Variance Estimation

The underlying advantage of randomizing the classical $(0, m, s)$-net or $(0, s)$-sequence is in obtaining a statistical error bound of the estimates. Owen [75] discusses two different approaches to estimate the variance of the estimate, $V(\hat{f})$. The simplest way of estimating $V(\hat{f})$ is by replication. A classical net (or sequence) $\{\boldsymbol{A}_n, n = 1, \ldots, N\}$ is chosen and is scrambled independently r times using r independent sets of permutation functions π so that the r estimates of the underlying value can be used to estimate $V(\hat{f})$. Suppose $\hat{f}_i$ denotes the resulting estimate from the i-th set of the independent permutation functions, then the unbiased estimate of $V(\hat{f})$ is given by

$$V(\hat{f}) = \frac{1}{r-1}\sum_{j=1}^{r}(\hat{f}_j - \bar{f})^2, \tag{4.4.1}$$

where $\bar{f} = \sum_{j=1}^{r} \hat{f}_j/r$. Once $V(\hat{f})$ is estimated, an approximate confidence interval of the estimate can be constructed from the appropriate t-distribution or normal distribution to assess the accuracy of the estimate.

Rather than using independent replications, an alternate approach to estimate $V(\hat{f})$ is to rely on a single large scrambled $(0, m, s)$-net. Recall that a $(0, m, s)$-net is a point set with b^m points $\{\boldsymbol{A}_n, n = 0, \ldots, b^m - 1\}$. Suppose the $(0, m, s)$-net is divided equally into b subsets of points as follows:

$$\underbrace{\underbrace{\boldsymbol{A}_0, \ldots, \boldsymbol{A}_{b^{m-1}-1}}_{(0,m-1,s)\text{-net}}, \underbrace{\boldsymbol{A}_{b^{m-1}}, \ldots, \boldsymbol{A}_{2b^{m-1}-1}}_{(0,m-1,s)\text{-net}}, \cdots, \underbrace{\boldsymbol{A}_{(b-1)b^{m-1}}, \ldots, \boldsymbol{A}_{b^m-1}}_{(0,m-1,s)\text{-net}}}_{(0,m,s)\text{-net}}.$$

so that each subset (or netlet) becomes a $(0, m-1, s)$-net of b^{m-1} points. Hence, a single scrambled $(0, m, s)$-net can be interpreted as a $(0, m-1, s)$-net being replicated "internally" b times. Let $\hat{f}_j$ denote the estimate obtained from the j-th internal replicate of $(0, m-1, s)$-net. Then (4.4.1) can be used to estimate $V(\hat{f})$ with $r = b$. When b is large, instead of using b internal copies, a single large scrambled $(\lambda, 0, m, s)$-net, $\lambda > 1$ is generated so that $\lambda(= r)$ internal replicates scrambled $(0, m, s)$-nets are used to estimate $V(\hat{f})$. Normally, λ should be chosen to be a lot smaller than b to allow the estimation to be manageable.

We now discuss the difference between the above two approaches. We assume that both methods use Nr points in estimating $V(\hat{f})$. First we would expect that the estimate of the underlying value using a single large randomized net of Nr points is more accurate than the corresponding estimate based on r independent replications of N points, This is certainly to be expected in the limit where N is fixed and r tends to infinity. In this situation, the independent replications can only achieve the Monte Carlo rate while the single large net can achieve the superior rate of $\mathcal{O}(\log(rN)^{s-1}/(rN))$. The drawback of using r internal copies to estimate the variance is that the estimated variance is biased. However, the variation among the r netlets should be larger than the variation based on r independent replications. If a point from a particular netlet belongs to a given elementary interval (of size b^{-m}), then other points from any of the netlets cannot appear in this elementary interval. This follows from the basic net property. In general, this leads to negative correlation among the estimates of the internal replicates and results in larger variation. Consequently, the variance estimated from r netlets overstates the true sampling

variance. This implies that any confidence statement constructed from this variance estimate is conservative. The conservativeness and higher accuracy lead us to use the second approach in obtaining the standard error of the estimate. The numerical calculations performed in Section 4.6 also indicate that the constructed confidence limits are conservative.

4.5 Derivative Securities used for Numerical Estimations

In this section, we examine the valuation of an option contract. For ease of benchmarking, it is important to use a security that admits a simple analytic solutions for any finite dimension. One of the simplest such problem in finance is the European average options where the average is taken to be the geometric average. The payoff of the call option at maturity can be represented by

$$g = \max\left[\left\{\prod_{i=1}^{s} S_i\right\}^{\frac{1}{s}} - K,\ 0\right],$$

where S_i, $1 \leq i \leq s$, is the appropriate asset price. We assume S_i is the terminal asset price for asset i so that we have a geometric portfolio average option with s underlying assets. Under the Black-Scholes framework, the distribution for the geometric average is the product of lognormal distributions and hence it is a lognormal distribution. Thus, there exists a simple closed-form representation similar to the Black-Scholes formula for a standard European call option. There is also

a closed-form solution for the case where the option is based on geometric path average for a single asset. This option is known as the geometric Asian option.

In our numerical examples, we consider the portfolio average option. We use two different approaches to obtain the required parameter values. For the first type, we consider an at-the-money option where the parameter values are fixed as follows: Initial asset prices $S_i(0) = 100$, volatility $\sigma_i = 0.3$ for $i = 1, \ldots, s$, the correlation between i-th and j-th assets is $\rho_{ij} = 0.5$ for $i, j = 1, \ldots, s$, $i \neq j$, time to maturity is 1 year and the annual interest rate is 10%. The strike price is chosen so that both call and put options have equal values.

For the second type, we randomly select the parameter values according to the following criteria: The strike price is fixed at 100 and the correlation between asset returns is fixed at 0.5. The other parameter values are generated randomly such that each initial asset price is uniformly distributed between 50 and 150, the annual volatilities are uniformly distributed between 10% and 60%, the expiration date is uniformly distributed between 6 months and 2 years and the annual interest rate is uniformly distributed between 5% and 15%. If the true option price for the set of randomly generated parameter values falls below 0.5, it is discarded and is replaced by another randomly generated set until the option price is at least 0.5. This is because very low option prices may lead to less reliable estimates of RMSE as defined below. The threshold 0.5 is suggested in Acworth, Broadie and Glasserman [1], but otherwise arbitrary.

In both cases, we compute the root-mean-squared relative error (RMSE) defined

by

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\hat{C}_j - C_j}{C_j}\right)^2}.$$

where $\hat{C}_j$ and C_j denote the simulated value and the theoretical value for the j-th option contract respectively and m represents the total number of option contracts to be evaluated.

4.6 Numerical Studies

Our numerical experiments can be divided into two parts. In the first part, we concentrate on low-dimensional option problems so that Owen's randomization technique can be implemented. This allows us to analyze the difference between the partial and full randomization methods. These are discussed in details in the following subsection. In the second part, we apply our proposed partial randomization technique to higher dimensional option contracts. The results are reported in Subsection 4.6.2.

Throughout the entire comparison, we have consistently used the scrambled versions of the $(0, m, s)$-nets in base b. When applicable the unscrambled $(0, m, s)$-nets or $(0, s)$-sequences in base b are also examined. Furthermore, the results based on crude Monte Carlo and Monte Carlo with antithetic variable technique are also reported for comparison. Briefly, the Monte Carlo estimates based on antithetic variates are obtained as follows: for each simulation run, two parallel estimates of the option prices are obtained. Suppose the first estimate, $\tilde{f}_j$, is computed from the

s independent standardized normal variates $\{\epsilon_1, \ldots, \epsilon_s\}$ while the second estimate, $\tilde{f}_j^A$, is obtained from $\{-\epsilon_1, \ldots, -\epsilon_s\}$. The overall estimate of the option price for this particular simulation trial is given by the average of these two estimates. Hence,

$$f_j = \frac{\tilde{f}_j + \tilde{f}_j^A}{2}$$

becomes the option estimate for the j-th simulation trial for the Monte Carlo method with antithetic sampling. Note that each estimate of the antithetic variable technique involves two functional evaluations. This implies that for the same number of simulations, there are twice as many functional evaluations in antithetic sampling as in crude Monte Carlo methods. Therefore, the reported results for the Monte Carlo with antithetic sampling should be adjusted approximately by a factor $1/\sqrt{2} \approx 0.707$ in order to have a fair comparison to other methods. In both cases, the random sequences are generated using the generator RAN2 from Press, Teukolsky, William and Brian [84].

4.6.1 Low-Dimensional Option Contracts

In this subsection, we compare the efficiency of both partial and full randomization techniques using 10-dimensional examples. This is done in three parts which we denote by A, B and C. To scramble the nets (or sequences), we consider Owen's full randomization and our proposed partial randomizations of levels 1 and 2. Henceforth, these three types of randomizations are referred to as partial-k^*, partial-I and partial-II, respectively. Recall that the full randomization method is implemented with truncation after k^*-th digits.

In determining the appropriate level of randomization, a preliminary calculation can be carried out in practice using the first few levels of randomization, say 1 and 2, with a small number of point set. This provides information regarding the variability of the estimates using different randomization levels and hence the appropriate level of randomization can be determined.

Suppose $\{\pi_i^p, \pi_{ia_{ni1}}^p, \cdots, \pi_{ia_{ni1}\cdots a_{nik^*-1}}^p\}$ denote the randomly generated permutation functions for randomized technique p where $p \in \{\text{partial-I, partial-II, partial-}k^*\}$. By construction, we have

$$\pi_{ia_{ni1}\cdots a_{nij}}^{\text{partial-II}} = \pi_{ia_{ni1}}^{\text{partial-II}}$$

for $2 \leq j < k^*$ and

$$\pi_{ia_{ni1}\cdots a_{nij}}^{\text{partial-I}} = \pi_i^{\text{partial-I}}$$

for $1 \leq j < k^*$ for each dimension i. For the randomized nets (or sequences) used in this study, the permutation functions are randomly generated subject to the following conditions:

$$\pi_i^{\text{partial-I}} = \pi_i^{\text{partial-II}} = \pi_i^{\text{partial-}k^*}$$

and

$$\pi_{ia_{ni1}}^{\text{partial-II}} = \pi_{ia_{ni1}}^{\text{partial-}k^*}.$$

That is, the first digit is always randomized by the same set of permutation functions in all three kinds of randomization while the second digit is always randomized by the same set of permutation function for partial-II and partial-k^* methods. When the permutation functions are controlled in this manner, any difference that arises between the partial-k method and the full randomization technique can be attributed to the effect of not randomizing the digits after the k-th places, rather than due to random variability in the permutation functions.

Part A

In the first part of the comparison, we randomly generate 1000 sets of option contracts where the parameter values are selected according to the rules outlined in the previous section. For each randomization technique, we compute the RMSE at point sets $\lambda 11^3$, $\lambda = 1, \ldots, 11$. In other words, a classical $(\lambda, 0, 3, 10)$-net in base 11 is generated and is scrambled accordingly to evaluate the option contracts.

Figure 4.10 summarizes the results. We plot the log of the RMSE against $\log(N)$ where N is the number of the sample points. For the top panel, a single scrambled net is used to evaluate the 1000 randomly generated option contracts. The middle panel is similar to the top panel except that a different scrambled net is used. From these two panels, it can be concluded that the RMSE estimated from partial-k, partial-II and partial-I methods are correlated. This is to be expected since the randomly generated permutation functions have been controlled as discussed earlier. The efficiency of the nets, however, depends on the particular choice of the permutation functions.

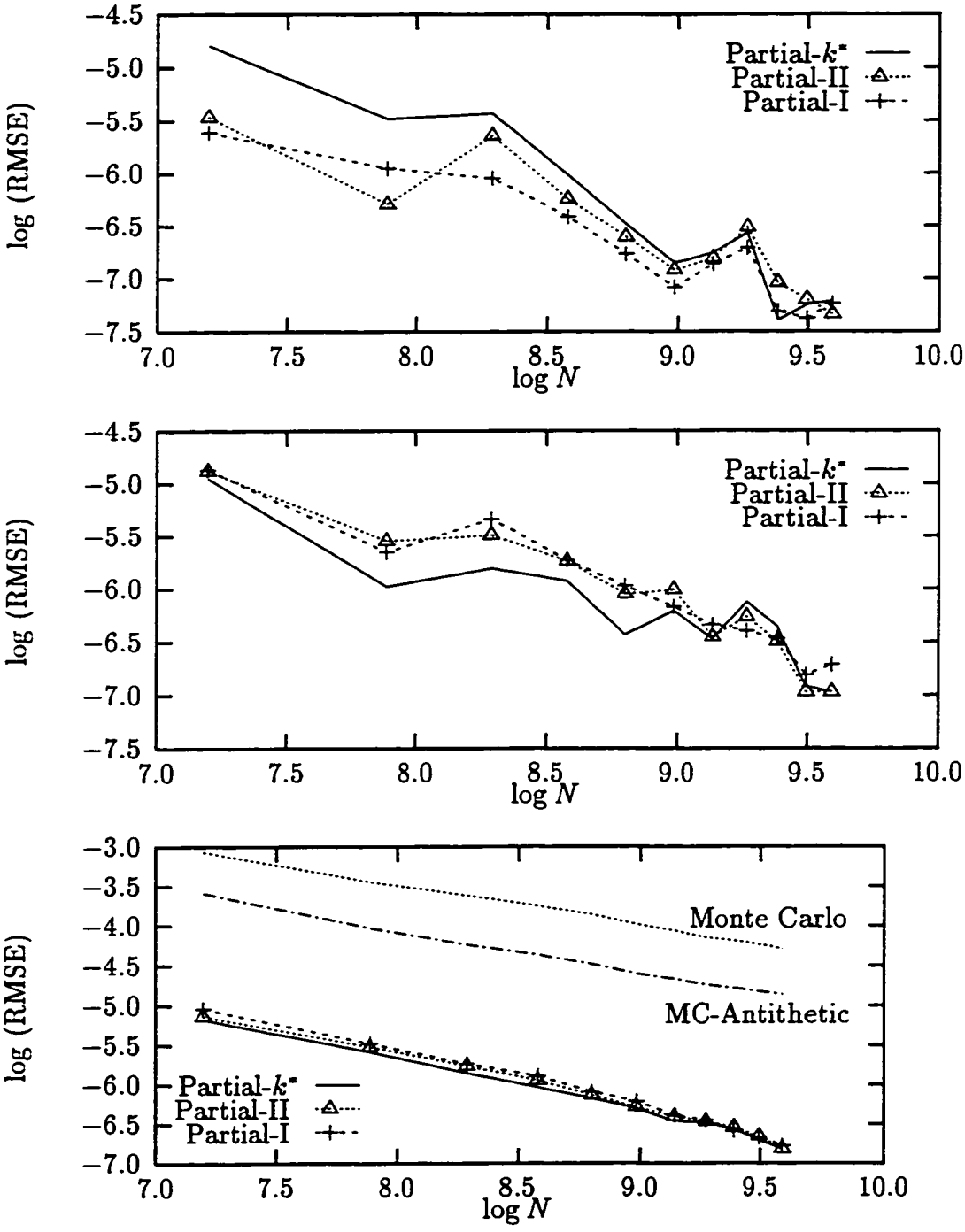


Figure 4.10: Relative Efficiency for Partial and Full Randomizations

A more appropriate assessment is to examine the average errors estimated over different independent sets of permutation functions. The result is presented at the bottom panel of Figure 4.10 where each option contract uses a different (and independent) scrambled net. Different scrambled nets are obtained by using independent set of randomly generated permutation functions. We now observe that the RMSE exhibits a smoother transition with the number of sample points. For comparison, the results obtained from the crude Monte Carlo and Monte Carlo with antithetic sampling are also included. In these cases, each option contract also uses a different set of random sequences.

On average, the results presented in the bottom panel indicate that all three of the scrambled nets essentially yield the same order of convergence although the full randomization technique seems to be consistently better than the other two scrambled nets.

Part B

We now proceed to the second part where we analyze the distribution of the estimates obtained from the scrambled nets. Statistical tests are conducted to test the hypothesis that the underlying distribution is normal. The normality assumption is important so that we can draw confidence statement about the estimated option value, hence providing a practical error bound where it would not be possible under the classical LD methods.

To facilitate our hypothesis testing, we consider a 10-dimensional at-the-money option contract using the parameter values given in Section 4.5. For each ran-

domization technique, a single large scrambled net consists of 1000×11^3 points is generated. Each consecutive 11^3 point set is used to estimate the value of the at-the-money option contract. Hence there are exactly 1,000 option estimates estimating the same underlying value. This enable us using various statistical tools to test the empirical distribution of the option estimates.

Our primary interest lies in whether the normality assumption holds for the option estimates. We now describe the statistical methods we use to conduct the hypothesis testing. These are only a small sample of the statistical tools that can be used to test the normality assumption. See D'Agostino and Stephens [20] for a comprehensive treatment on this topic.

The most common test is based on normal probability plot (also known as Q-Q or rankit plot). This can briefly be described as follows: Let $\hat{f}_j$ denote the estimate of the option using the j netlet and let $\hat{f}_{(1)} < \cdots < \hat{f}_{(r)}$ be the ordered values from all the r netlets. If $\hat{f}_{(1)} < \cdots < \hat{f}_{(r)}$ are independent and identically distributed (i.e. i.i.d.) $N(\mu, \sigma)$ random variables, then

$$\mathrm{E}[\hat{f}_{(j)}] = \mu + \sigma\gamma_j \tag{4.6.1}$$

where $\gamma_j \approx \Phi^{-1}[(j-3/8)/(r+1/4)]$ and Φ^{-1} is the inverse function of the standard normal cumulative density function. From (4.6.1), it follows that if $\hat{f}_1, \ldots, \hat{f}_r$ are i.i.d. $N(\mu, \sigma)$, a plot of $\hat{f}_{(j)}$ against γ_j would yield approximately a straight line. Such a plot is known as the normal probability plot.

Figure 4.11 provides the required plots for the three types of randomization techniques based on $r = 1000$ option estimates. These plots support the normality

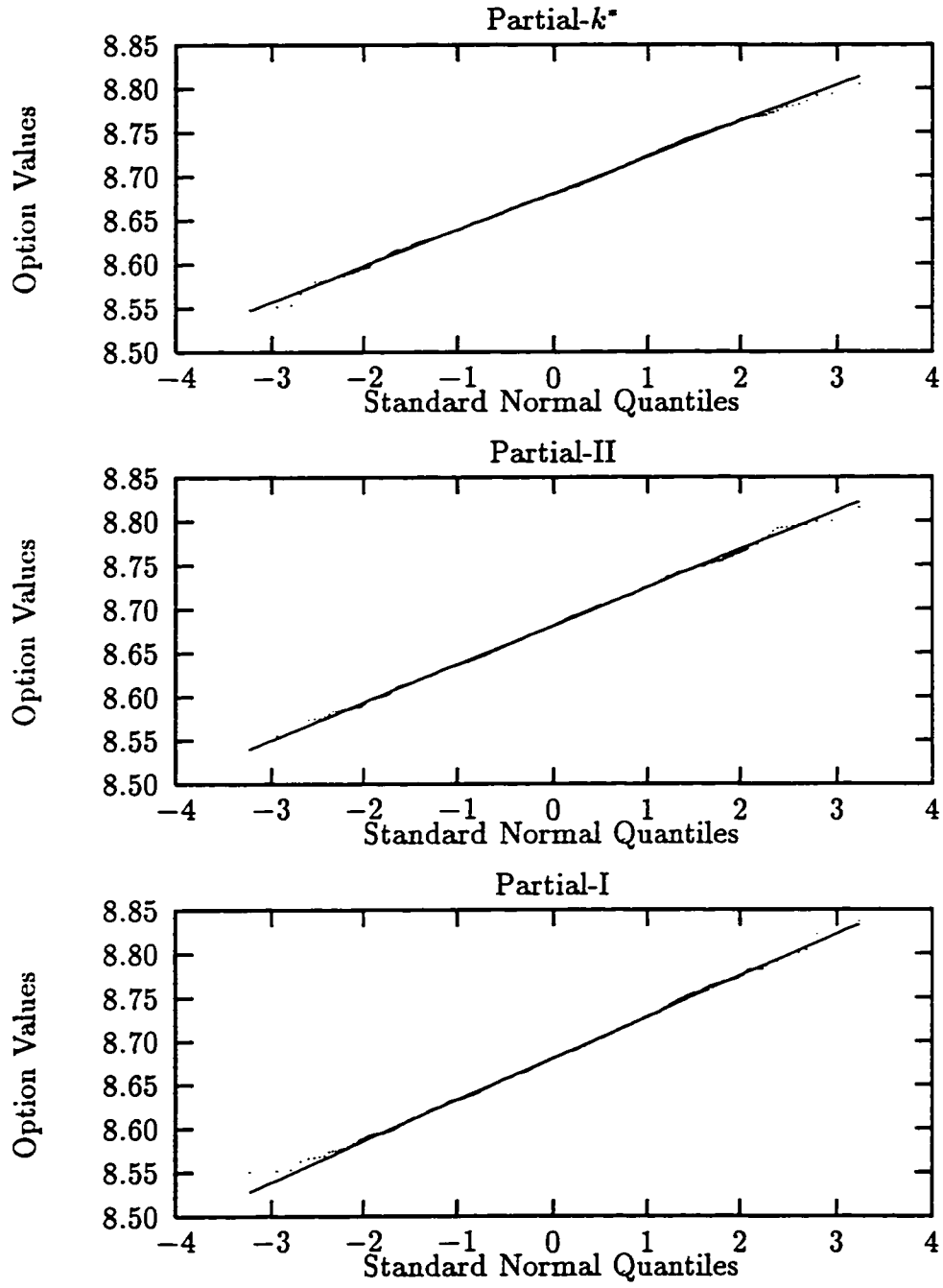


Figure 4.11: Normal Probability Plots

assumption. To confirm this observation, we conduct two other quantitative tests. The first measure is based on the correlation coefficient of points in the rankit plots. This is defined as

$$\rho_{QQ} = \frac{\sum_{j=1}^{r}(\hat{f}_{(j)} - \bar{f})(\gamma_{(j)} - \bar{\gamma})}{\sqrt{\sum_{j=1}^{r}(\hat{f}_{(j)} - \bar{f})^2}\sqrt{\sum_{j=1}^{r}(\gamma_{(j)} - \bar{\gamma})^2}} \tag{4.6.2}$$

where $\bar{f} = \sum_{j=1}^{r} \hat{f}_{(j)}/r$, $\bar{\gamma} = \sum_{j=1}^{r} \gamma_{(j)}/r$ and $r = 1000$ in this example. This statistic measures the straightness of the rankit plot and provide a powerful test of normality.

A second test is based on the closeness of the sample skewness and sample kurtosis to the theoretical values of 0 and 3 respectively under the normality assumption. This test is known as the Bowman-Shelton test for normality. The sample skewness can be estimated as

$$\text{skewness} = \frac{\sum_{j=1}^{r}(\hat{f}_{(j)} - \bar{f})^3/r}{\hat{\sigma}^3}$$

and the sample kurtosis as

$$\text{kurtosis} = \frac{\sum_{j=1}^{r}(\hat{f}_{(j)} - \bar{f})^4/r}{\hat{\sigma}^4}.$$

Here $\hat{\sigma}$ is the sample deviation defined in (4.4.1).

A statistic which tests the symmetrization and the tail of the population can

Test Statistic	Partial-k^*	Partial-II	Partial-I
ρ_{QQ}	0.9994	0.9995	0.9995
skewness	-0.0001	-0.0211	0.0747
kurtosis	2.9150	2.8300	2.8581
Bowman Shelton	0.3014	1.2781	1.7696

Table 4.1: Test Statistics based on 1000 Internal Replicates

be constructed as follow:

$$\text{Bowman-Shelton} = r\left[\frac{(\text{skewness})^2}{6} + \frac{(\text{kurtosis}-3)^2}{24}\right]. \tag{4.6.3}$$

Under the null hypothesis that the population distribution is normal, this test statistic converges to a chi-square distribution with 2 degrees of freedom as the number of observations become very large. The null hypothesis is rejected for large values of the test statistic.

Table 4.1 reports the test statistics based on the 1000 simulated option estimates. With respect to the third and fourth moments of the option estimates, there is a slight deterioration in switching from full randomization to partial randomization. However, the deterioration is well within the tolerance that hypotheses distribution is not rejected. For instance, the critical point for the Bowman-Shelton statistic at 10% significance level with $N = 800$ sample size is 4.32 and $N = \infty$ is 4.61 (chi-square distribution with 2 degree of freedom). Since the Bowman-Shelton tests obtained from the full and partial randomization methods are well below the critical points, the null hypothesis that the sample population is normal is not rejected.

Part C

We now consider the estimation of the standard errors based on the at-the-money

	Monte Carlo				Randomized $(\lambda, 0, 3, 10)$-Nets					
	Crude		Antithetic		Partial-k^*		Partial-II		Partial-I	
λ	Std. Error	%	Ratio	%	Ratio	%	Ratio	%	Ratio	%
7	0.145	6.0	1.8	5.4	9.7	2.0	9.0	1.5	8.5	2.0
8	0.136	5.9	1.8	5.8	9.6	1.7	9.0	0.9	8.5	1.6
9	0.129	4.9	1.8	5.0	9.7	1.0	9.0	0.6	8.5	1.1
10	0.123	5.5	1.8	4.2	9.7	1.1	9.1	0.4	8.6	0.3
11	0.118	4.8	1.8	4.6	9.7	0.7	9.1	0.4	8.5	0.2

Table 4.2: Comparison of the Average Standard Errors and the Percentage of the Violations based on At-the-Money Option

option example used in Part B. For each scrambling method, a $(\lambda, 0, 3, 10)$-net in base 11 is generated so that λ internal replicates (each consists of 11^3 points) are used to estimate the standard error. Five different values of $\lambda = 7, 8, \ldots, 11$ are considered. For the Monte Carlo methods with and without antithetic sampling, the same number of points $(= \lambda 11^3)$ is also used to estimate the corresponding standard error. This procedure is repeated 1000 times using different scrambled $(\lambda, 0, 3, 10)$-nets generated from independent sets of permutation functions and different random sequences.

The second column of Table 4.2 reports the average standard errors over 1000 independent standard errors obtained from Monte Carlo methods. For ease of comparison, we report the ratio of the average standard errors from the crude Monte Carlo to the corresponding average standard errors obtained from the Monte Carlo

with antithetic sampling, partial-k^*, partial-II and partial-I methods. Hence, the larger the ratio, the more favourable the underlying method. These ratios are tabulated under the heading "ratio" in Table 4.2. When $\lambda = 9$, the simulation results indicate that the average standard errors from the Monte Carlo with antithetic sampling is approximately 1.8 times smaller than the corresponding average standard errors from the crude Monte Carlo method. Since the antithetic sampling has used twice as many functional evaluations as the crude Monte Carlo method, this ratio should be reduced approximately by a factor $1/\sqrt{2}$ so that the actual improvement of the antithetic sampling is only about 1.3 times.

The scrambled nets, on the other hand, have achieved a significant improvement. For the same number of replications, the magnitude of improvement for partial-k^*, partial-II, partial-I techniques are respectively, 9.7, 9.0 and 8.5 times smaller. This implies that the average standard errors for the partial-k^*, partial-II, partial-I methods are only 0.0133, 0.0144 and 0.0159, which compare favourably to the Monte Carlo average standard error 0.129. This reinforces the findings given in Part A where we compare the efficiency by computing the RMSE.

Note that the magnitude of improvements is relatively stable across λ. We also observe that the full randomization yields the smallest average standard errors. By only randomizing the first one and first two digits as in the partial-I and partial-II approaches, we observe a slight decrease in the efficiency of the underlying method. For instance, Table 4.2 indicates that the average standard errors for partial-I and partial-II randomizations are approximately 1.14 and 1.07 times larger than the full randomization.

The main reason for estimating the standard errors is that for Monte Carlo methods, we can construct the confidence interval using the t-distribution or normal distribution, depending on the number of replications, so that the accuracy of the estimate can be gauged. This follows from the central limit theorem. This is an important advantage of the Monte Carlo method. For the full and partial randomization methods, the analysis in Part B also justifies that the use of a similar approach to estimate confidence intervals.

In addition to comparing the average standard errors, we also compute the 95% confidence interval using the t-distribution with an appropriate degree of freedom in our example. Hence we have 1000 independent sets of confidence limits for each λ and each method. From the constructed confidence limits, we determine if the theoretical option value lies within the limits. Since we have used 95% as the confidence level for obtaining the confidence limits, we would expect approximately 50 out of the 1000 constructed confidence intervals do not contain the theoretical value. This is consistent with our simulated results for the Monte Carlo with and without antithetic sampling. The percentage violations of the constructed confidence limits for these two methods are tabulated in the third and fifth columns of Table 4.2. For example, when 9 replications are used to estimate the standard errors the crude Monte Carlo results indicate that among the 1000 independent confidence limits constructed, 49 of them do not contain the theoretical option value. Similarly, the seventh, ninth and eleventh columns give the corresponding measure for the full and partial randomization of levels 2 and 1, respectively. Unlike the Monte Carlo methods, the percentage of violations for the scrambled net are

a lot smaller than the anticipated proportion. Furthermore, the percentage of violations for the partial randomization methods in most cases are less than the full randomization method. All these observations lead to the conclusion that the confidence limits constructed from the scrambled nets are conservative. This is consistent with argument given in Section 4.4.

So far the comparison of the standard errors relies only on one particular contract — the at-the-money option. Our conclusion could therefore be influenced by a particular choice of the parameter values. To avoid this possibility, we repeat the calculations using 1000 randomly generated sets of parameter values. Each option contract uses a different random point set or scrambled $(\lambda, 0, 3, 10)$-net. The results are summarized in Table 4.3. The conclusions are similar to the previous case where only one option contract is evaluated. First, the constructed confidence limits for the scrambled nets are conservative. The percentage of violations is no more than 2% for these methods. Second, the magnitude of improvement is relatively stable across the number of replications. Third, the randomized nets have achieved a substantial reduction in the estimated standard errors while the Monte Carlo with antithetic sampling is only marginally more efficient. Fourth, sacrificing the total randomization only results in a slight deterioration of the scrambled nets. Compared to the full randomization, the average standard errors obtained from randomizing only the first digit is approximately 22% ($\approx 11.7/9.6 - 1$) larger while randomizing the first two digits is roughly 10% ($\approx 11.7/10.6 - 1$) larger. The loss of accuracy is well compensated for by the tremendous reduction in the number of permutation functions required in randomizing the nets, which in turn translates

into a great saving in the memory storage requirements as well as the execution time for generating the extra permutation functions. In either case, we still observe an order of improvement around 9.6 and 10.7 times relative to the standard Monte Carlo methods.

4.6.2 High-Dimensional Option Contracts

The impetus for proposing the partial randomization lies in high-dimensional applications. In the last subsection, we have intentionally used low-dimensional option contracts ($s = 10$) so that the proposed partial randomization technique can be compared to the total randomization technique. In this subsection, we examine the efficiency of the partial randomization technique when the dimension is large. We only consider the partial-I and partial-II randomization methods since the full randomization becomes computationally infeasible for large s. We divide the studies into two parts as described below.

Part A

We first compare the efficiency of the partial randomization methods by increasing the dimensionality of the option problems. We consider $s = 100, 250$ and 365. For each case, we compute the RMSE based on 50 randomly generated option contracts as outlined in Section 4.5. Three types of LD sequences — classical $(0, s)$-sequences in base b using the generation algorithm by Faure [24] and scrambled $(0, s)$-sequences in base b using partial-I and partial-II randomization are implemented. For each sequence, we compute the RMSE at point sets $N = \lambda b^m$ for

	Monte Carlo				Randomized $(\lambda, 0, 3, 10)$-Nets					
	Crude		Antithetic		Partial-k^*		Partial-II		Partial-I	
λ	Std. Error	%	Ratio	%	Ratio	%	Ratio	%	Ratio	%
7	0.153	5.1	2.1	4.9	11.7	1.1	10.6	2.0	9.6	1.6
8	0.144	5.3	2.1	4.5	11.7	1.1	10.7	1.0	9.6	1.1
9	0.136	5.6	2.1	4.1	11.7	1.0	10.7	0.7	9.6	0.5
10	0.130	5.5	2.1	4.8	11.7	0.3	10.7	0.4	9.6	0.6
11	0.125	4.5	2.1	4.5	11.8	0.3	10.8	0.2	9.7	0.2

Table 4.3: Comparison of the Average Standard Errors and the Percentage of the Violations using 1000 Randomly Generated Option Contracts

integers λ and m satisfying $10{,}000 \leq N \leq 200{,}000$. The results (in log-log scale) are plotted in Figure 4.12. In all these cases, a single generated sequence is used to evaluate the entire portfolio of the option contracts. This explains the fluctuation in the RMSE. We also observe that the randomized sequences yield smaller RMSE than the corresponding unscrambled sequences. Furthermore, the simulation results from both partial-I and partial-II tends to be correlated. This is to be expected since we have controlled our experiment so that the first digit is always scrambled by the same set of permutation function.

Similar to Part A of last subsection, we repeat the above calculations so that each option contract uses an independent scrambled sequence generated from an independent set of permutation function. For the classical sequences, a different (and non-overlapping) part of the sequences is used to evaluate each option contract. The results are presented in Figure 4.13. For comparison, the results from Monte Carlo methods with and without antithetic sampling are also illustrated.

Compared to Figure 4.12, using different independent set of sequences has sub-

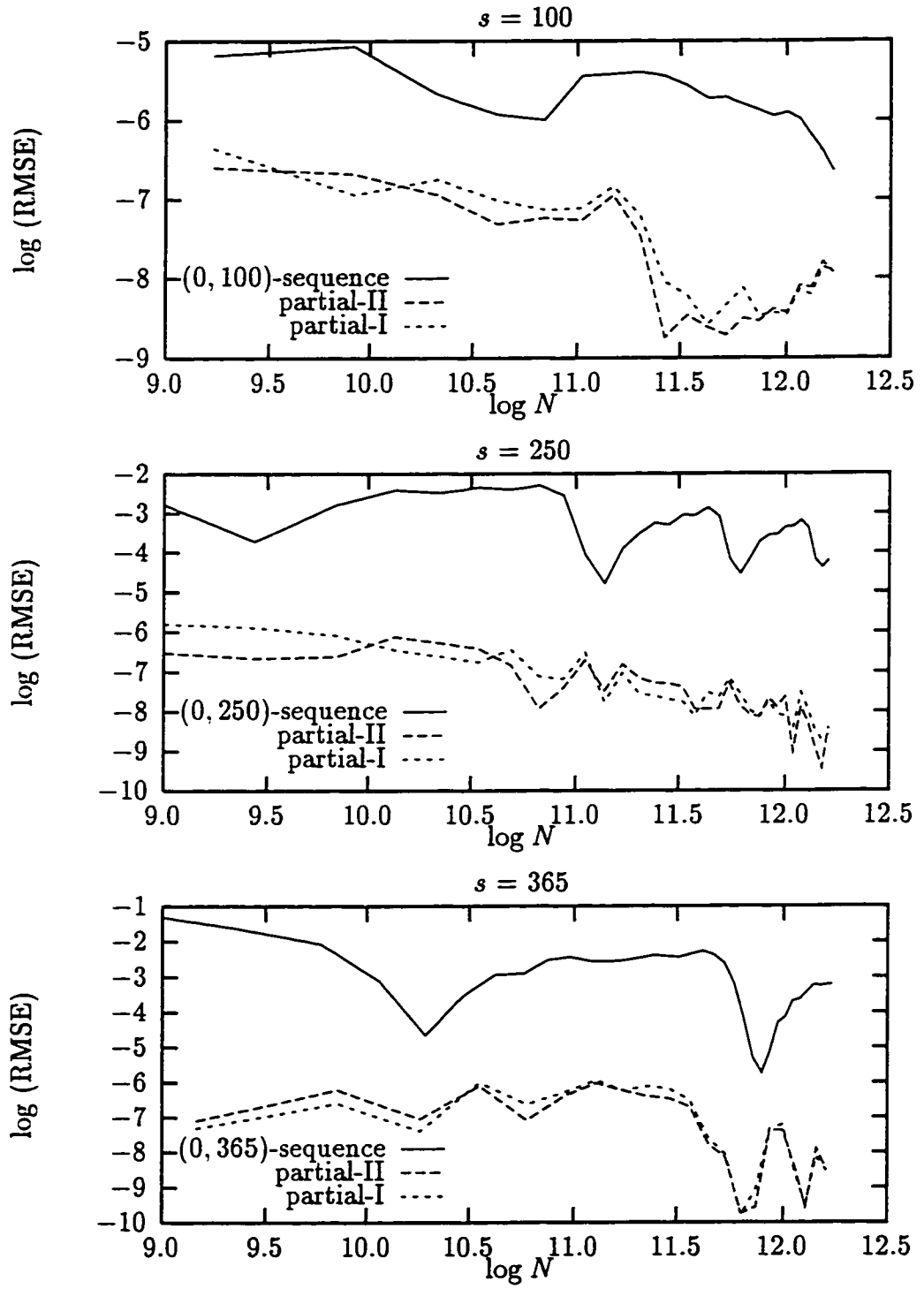


Figure 4.12: Relative Efficiency with Increasing Dimensions (Same Sequence Applies to 50 Option Contracts)

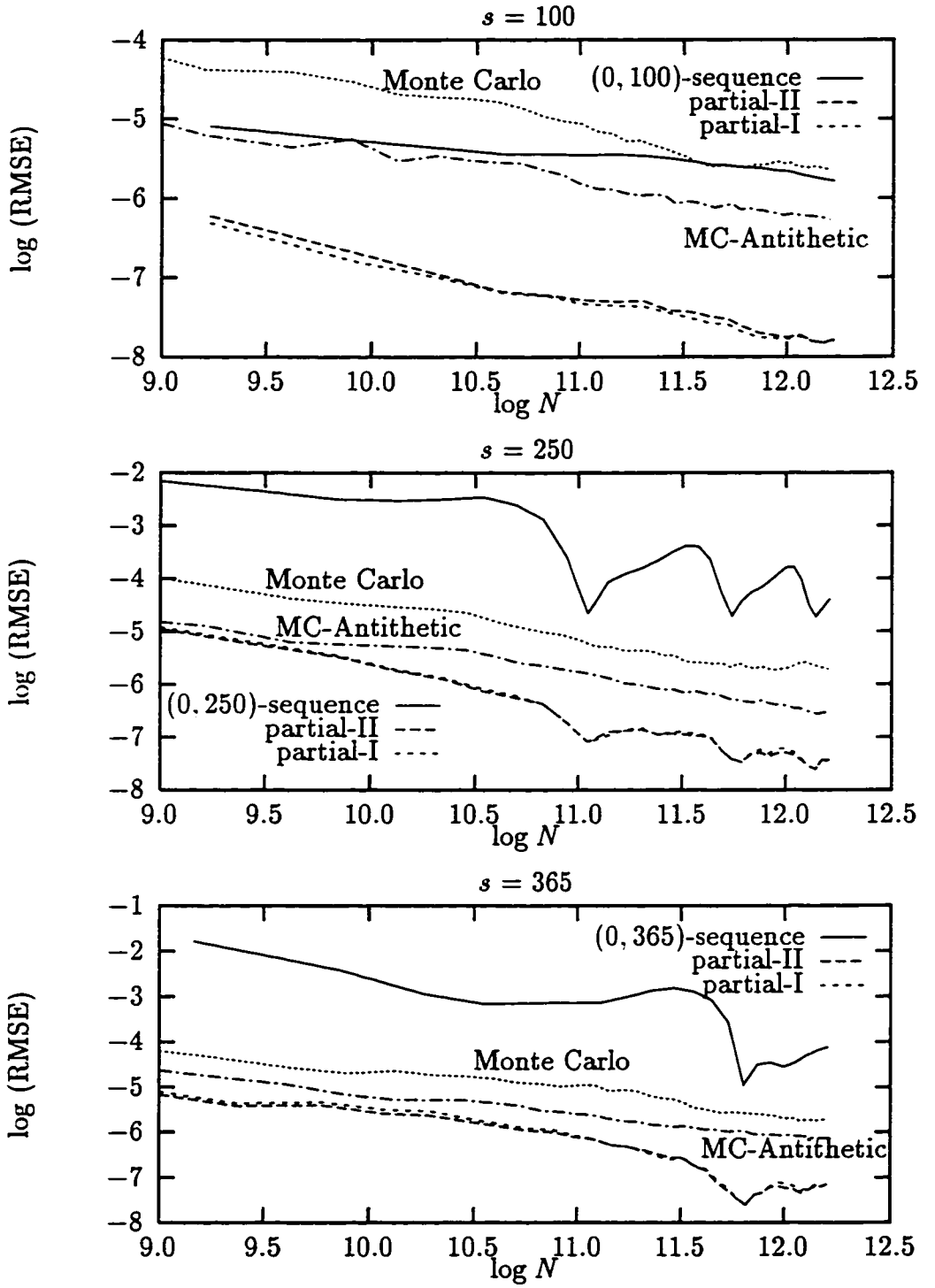


Figure 4.13: Relative Efficiency with Increasing Dimensions (Each Option Contracts Uses Different Sequences)

stantially reduced the variability in RMSE. We note the different rates of convergence exhibited by the classical and scrambled LD sequences. Theoretically, the asymptotic rate of the LD methods does not kick in until $N \approx s^s$, which is a lot larger than the number of points used in our studies. However, as discussed in Section 4.3, the benefits of the LD methods can appear even for $N \ll s^s$. This is due to the low-dimensional structures inherent in most finance problems even if the nominal dimension is large. This argument appears to hold for the scrambled sequences in Figure 4.13. The classical $(0, s)$-sequences, on the other hand, seem to require a much larger number of points to achieve the same efficiency as the scrambled sequences.

Another interesting observation is the "cyclical" convergence rate exhibited in the LD sequences, particularly more pronounced for the classical $(0, s)$-sequences. For instance when $s = 365$, the RMSE decreases as N increases until $\log N \approx 11.8$. At such point any further increments in N actually increases the RMSE. These patterns can partly be attributed to the large prime base used in generating the sequence. By construction, the $(0, s)$-sequences achieve the optimal uniformity when the number of points is equal to the power of the base so that each elementary interval contains exactly a single point. This implies that the optimal uniformity is achieved at $N = b^m$, $m = 1, 2, \ldots$ and we would expect low integration errors at these optimal point sets. For $N = \lambda b^m$, $1 < \lambda < b$, the point set is in a "transient" state in the sense that the elementary intervals are gradually filling up. Consequently using these point sets lead to slightly higher integration errors. This explains the small RMSE observed at $\log 367^2 \approx 11.8$ at the bottom panel

of Figure 4.13 since $b = 367$ is the required prime base used to generate $(0, 365)$-sequences.

Part B

To assess the effectiveness of the partial randomization methods in providing the standard errors of the estimates and the constructed confidence intervals in high dimensions, we follow the same methodology described in Part C of last subsection. In this case, we consider $s = 100$ and also use 1000 randomly generated option contracts. Each option contract uses an independent scrambled $(\lambda, 0, 3, 100)$-net in base 101. Standard errors are estimated at $\lambda = 7, 9, \ldots, 11$. The same number of points is also used for the Monte Carlo with and without antithetic sampling to estimate the standard error. From the estimated standard error, the 95% confidence limit is constructed so as to determine if the theoretical option value for that particular option contract lies in the interval.

Table 4.4 summarizes the results. As anticipated, the percentages of violation for the Monte Carlo methods fluctuate around the expected 5% level. For the scrambled nets, the percentages of violation are still less than 5% although with a much larger proportion compared to the low dimensional situation. The relative magnitude of the average standard errors for Monte Carlo with antithetic sampling, partial-II and partial-I are approximately 1.4, 8.6 and 8.4 times smaller than the corresponding measure from the crude Monte Carlo method. This indicates that a significant variance reduction can be achieved even for dimensions as high as 100 while still providing conservative confidence limits.

	Monte Carlo				Randomized $(\lambda, 0, 3, 100)$-Nets			
	Crude		Antithetic		Partial-II		Partial-I	
λ	Std. Error	%	Ratio	%	Ratio	%	Ratio	%
7	0.052	5.9	2.0	5.9	8.5	4.3	8.3	4.2
8	0.049	6.0	2.0	5.0	8.5	3.8	8.4	4.3
9	0.047	4.9	2.0	6.5	8.6	4.0	8.4	3.8
10	0.045	4.5	2.0	4.9	8.6	4.5	8.5	4.0
11	0.043	5.3	2.0	5.5	8.7	4.2	8.5	3.7

Table 4.4: Comparison of the Average Standard Errors using 1000 Randomly Generated Sets of Option Contracts ($s = 100$)

4.7 Conclusion

The classical approach to the implementation of low discrepancy sequences for the solution of problems in the finance area suffers from some drawbacks. This chapter has examined ways to rectify some of the disadvantages of LD methods. These new methods try to combine the best features of both standard (crude) Monte Carlo and classical LD methods. We found that our modification of the scrambling procedure suggested by Owen [74] improves the convergence rate as compared to the classical LD approach. More importantly, from a series of empirical studies, we observed that giving up the full randomization only results in a slight decrease in the efficiency of the sequences. This loss in efficiency is however justified by the saving of the memory storage requirements and the computation time. Statistical tests were conducted to validate the generation of the confidence intervals. The availability of confidence intervals is useful because it provides a scientific method for determining the accuracy of the estimation procedure and thus providing practical termination criteria. Hence the main conclusion of this chapter is that it is possible to modify

the randomization technique due to Owen so that it can be applied effectively to high dimensional problems.

Chapter 5

Smoothing Discontinuities

While the routine application of quasi-Monte Carlo works well for some integrands, there are situations when the efficiency of quasi-Monte Carlo methods is severely deteriorated. Two common situations are where the integrand in question is discontinuous and when the number of dimensions is very large. In this chapter, we discuss methods of dealing with discontinuities when using quasi-Monte Carlo methods. We provide a general framework for avoiding the discontinuity by smoothing the integrands. This method is inspired by the smoothing technique proposed by Moskowitz and Caflisch [61].

Section 5.1 provides some background material for the issues involved. Section 5.2 discusses an application of Monte Carlo integration which involves discontinuity. In Section 5.3, we describe the generalized smoothing technique and relate the proposed method to Moskowitz and Caflisch's smoothing technique. We also discuss the bias that arises from using Moskowitz and Caflisch's smoothing technique. In Section 5.4 , examples are presented to compare the efficiency of the

proposed generalization to the existing methods. Section 5.5 concludes the chapter.

5.1 Background

Monte Carlo integration using low discrepancy sequences has theoretical error bounds of $\mathcal{O}(N^{-1}\log^s N)$ in dimension s. This error bound is established from the Koksma-Hlawka inequality which states that for Monte Carlo integration of a function f over the domain $[0,1)^s$ of N points, the integration error is bounded by the product of two quantities (see (2.4.10)). The first quantity, D_N, depends on the uniformity of the point set and is measured by its discrepancy while the second quantity, $V(f)$, relates to the smoothness of the function for which the function has bounded variation in Hardy-Krause sense. In practice, this bound turns out to be not very useful since it is not sharp and is only correct asymptotically. Nevertheless, this bound provides a theoretical justification for using sequences that have low discrepancy. Furthermore, the Koksma-Hlawka inequality provides several "clues" for which the effectiveness of the Monte Carlo integration using low discrepancy sequences can be detrimental.

First, the factor $N^{-1}\log^s N$ is not smaller than $N^{-1/2}$ until N is very large. Furthermore, as the dimension increases, the value of N where the low discrepancy sequences start to exhibit their asymptotic behaviour also becomes exponentially larger. This implies that for fixed N, the efficiency of the low discrepancy sequence reduces in dimension s. This is in contrast to the Monte Carlo method where the rate of convergence is $\mathcal{O}(N^{-1/2})$ and is independent of the dimension.

Second, for f to be of bounded variation, the function f must be smooth.

This can be a restrictive criteria since discontinuous functions often arise in Monte Carlo applications. These functions, in general, have infinite variation unless the discontinuities occur along the direction that is parallel to the coordinate axes. The direction in which the discontinuity occurs plays an important role since low discrepancy sequences are constructed so as to minimize the discrepancy along each coordinate axis.

An example of discontinuous integrands is the characteristic function of a set, which yields a value of 1 over a certain range of values and 0 otherwise. More formally, if we denote the characteristic function of the set E as χ_E, we have

$$\chi_E(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in E \subseteq \Re^s, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the discontinuity occurs at the boundary of the set E where the value changes from 0 to 1. The variation is infinite unless the boundary of the set is parallel to the coordinate axes.

Another example which contains a discontinuity is the acceptance-rejection method used to generate independent sample points that have a certain density distribution. The acceptance-rejection method achieves this objective by transforming sample points from one distribution, say $h(\boldsymbol{x})$, to the desired distribution $f(\boldsymbol{x})$. Suppose $f(\boldsymbol{x}) \leq \gamma h(\boldsymbol{x})$, $\gamma > 1$ for all $\boldsymbol{x}$, the general acceptance-rejection algorithm can be described as follow:

Algorithm A-R-1

- REPEAT
  1. Generate $\boldsymbol{x}_n \in h(\boldsymbol{x})$.
  2. Generate $y_n \in U[0,1)$.
- UNTIL $y_n < \frac{1}{\gamma}\frac{f(\boldsymbol{x}_n)}{h(\boldsymbol{x}_n)}$ and return $\boldsymbol{x}_n$.

The above algorithm, which we have labeled as A-R-1, produces a sequence of accepted points $\boldsymbol{x}_n$ that are distributed according to f. For this algorithm to be effective, the density h should be relatively simple so that it is easy to evaluate. Furthermore, the constant γ should be close to 1 since $1/\gamma$ denotes the probability of accepting the generated trial points. When γ is large, there will be more iterations in the loop before a successful point with the desired distribution is obtained. The discontinuous nature of the acceptance-rejection method follows from the binary decision to accept or to reject the trial point.

For both examples, the discontinuity leads to infinite variation unless the boundary of the set is parallel to the coordinate axes. The Koksma-Hlawka inequality would yield an infinite error bound and the star discrepancy is no longer meaningful. The integration error bound of characteristic functions, nevertheless, can be bounded by a generalized version of a discrepancy measure known as isotropic discrepancy. The isotropic discrepancy, J_N, of a sequence $\boldsymbol{x}_n \in [0,1)^s$, $1 \le n \le N$ is defined as (see also Kuipers and Niederreiter [53])

$$J_N = \sup_{C \in \mathcal{C}} \left| \frac{A(C;N)}{N} - m(C) \right|,$$

where $\mathcal{C}$ is the class of all convex subsets of $[0,1)^s$, $A(C;N)$ is the number of $\boldsymbol{x}_n$

which are inside C, and $m(C)$ is the measure of C. Morokoff and Caflisch [60] show that the theoretic error bound for the integration of characteristic functions is $\mathcal{O}(N^{-s/(2s-1)})$, assuming an optimistic point set with $D_N = N^{-1}$ is used. This result suggests that integrating a discontinuous function in high dimension, we cannot expect to obtain a convergence that is much better than the Monte Carlo method. A heuristic argument provided by Press and Teukolsky [83] in explaining the loss of accuracy is that near the boundary of the discontinuity, the positioning of a point just inside the set and thus contributes a value of 1 or just outside the set and thus contributes a value of 0 is essentially random when the boundary of the characteristic function is not parallel to the axes. The convergence for points near the boundary can only achieve a random error. When this is combined with the rest of the points which attain a superior low discrepancy rate, we obtain a new rate of convergence that lies between these two extremes.

The computational experiments conducted by Morokoff and Caflisch [60], Moskowitz and Caflisch [61], and Press and Teukolsky [83] support the above argument. They observe a decline in the efficiency of the low discrepancy sequence when the method is used to evaluate an integrand which is discontinuous. However, the magnitude of deterioration is not as extreme as $\mathcal{O}(N^{-s/(2s-1)})$. Their studies indicate that the convergence rate is typically ranging from $N^{-1/2}$ and $N^{-2/3}$. In Moskowitz and Caflisch [61], an attempt is made to improve the convergence of low discrepancy sequence when the function of interest contains a discontinuity. They propose a "smoothed acceptance-rejection method" and show that when this method is combined with importance sampling, the superior rate of low discrepancy sequence can

be recovered. The idea underlying their method is to "smooth out" the boundary where the characteristic function is defined so that the function is no longer sharply discontinuous. In this chapter, we consider a generalization of their method and provide a general framework for the smoothing technique. We also show that Moskowitz and Caflisch's smoothing technique yields an estimate that is biased while the proposed generalized smoothing technique is unbiased.

Moskowitz and Caflisch [61] also rediscover the weighted uniform sampling method proposed by Powell and Swann [82] as an alternative to avoiding discontinuity. Moskowitz and Caflisch conclude that although the smoothed acceptance-rejection method is more efficient than the ordinary acceptance-rejection method, the weighted uniform sampling method is even more superior. In our computational examples, the generalized smoothing technique is not only more efficient than Moskowitz and Caflisch's smoothing technique, it is also competitive with the weighted uniform sampling method.

5.2 Importance Sampling Using Acceptance-Rejection Method

In standard Monte Carlo integration, the estimate of $\int_{[0,1)^s} f(\boldsymbol{x})d\boldsymbol{x}$, is obtained from the sample average using

$$\hat{A}_N = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n) \tag{5.2.1}$$

where $\boldsymbol{x}_n$ is uniformly distributed in the s-dimensional unit cube; i.e., $\boldsymbol{x}_n \sim U[0.1)^s$. It is well known that this method has a convergence rate of order $\mathcal{O}(N^{-1/2})$. In practice, the standard Monte Carlo method is often coupled with other methods known as variance reduction techniques to speed up the convergence rate by reducing the variance of the Monte Carlo estimate. Examples of these methods include the conditioning method, the use of control variates, the antithetic variable technique and importance sampling. Of particular interest to us is the importance sampling which we now describe. Suppose there exists a simpler function $g > 0$ which closely resembles f. In our context, a function is simple when it can easily be integrated analytically or numerically. The basic idea is to generate points that are distributed according to g. The points are then used to evaluate $f(\boldsymbol{x})/g(\boldsymbol{x})$ so that the importance sampled Monte Carlo estimate is given by

$$\tilde{A}_N = \sum_{n=1}^{N} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} \tag{5.2.2}$$

where $\boldsymbol{x}_n$ is distributed according to $g(\boldsymbol{x})$; i.e. $\boldsymbol{x}_n \sim g(\boldsymbol{x})$, and g is a normalized integrand such that $\int_0^1 g(\boldsymbol{x})d\boldsymbol{x} = 1$. The estimate $\tilde{A}_N$ is an unbiased estimator of $\int_0^1 f(\boldsymbol{x})d\boldsymbol{x}$ since

$$\mathrm{E}_g\left[\frac{f}{g}\right] = \int_{[0,1)^s} \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \cdot g(\boldsymbol{x})d\boldsymbol{x} = \int_{[0,1)^s} f(\boldsymbol{x})d\boldsymbol{x}.$$

The variance of the importance sampled Monte Carlo estimate is smaller than the corresponding standard Monte Carlo estimate as long as the variance of f/g is smaller than f. The efficiency of this method therefore relies on how well the func-

tion g mimics f and how easy it is to generate sample points that have probability density g.

The acceptance-rejection method discussed in the last section is one of the most commonly used tools to generate $\boldsymbol{x}_n$ with a specific probability density. Suppose the densities h and f in A-R-1 are respectively, replaced by $U[0,1)^s$ and g, we obtain the following modified algorithm which we label as A-R-2:

Algorithm A-R-2

- REPEAT
  1. Generate $(\boldsymbol{x}_n, y_n) \in U[0,1)^{s+1}$.
- UNTIL $y_n < \frac{g(\boldsymbol{x}_n)}{\gamma}$ and return $\boldsymbol{x}_n$.

This algorithm again produces a sequence of accepted points $\boldsymbol{x}_n$ having probability density g. The accepted points are then used to evaluate (5.2.2) to obtain the importance sampled Monte Carlo estimate.

The discontinuous nature of the importance sampling with acceptance-rejection method is apparent when we express the importance sampled Monte Carlo estimate (5.2.2) as

$$\tilde{A}_N = \frac{1}{\sum_{n=1}^{N^*} w(\boldsymbol{x}_n, y_n)} \sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} w(\boldsymbol{x}_n, y_n) \tag{5.2.3}$$

where N^* is the total number of trial points generated in algorithm A-R-2, and $w(\boldsymbol{x}_n, y_n)$ is the weight associated with each functional estimate $f(\boldsymbol{x}_n)/g(\boldsymbol{x}_n)$. In this case, the weight $w(\boldsymbol{x}_n, y_n)$ admits only the values of 1 or 0 depending on the decision making step in A-R-2. In other words, the weight $w(\boldsymbol{x}_n, y_n)$ is a

characteristic function which can be expressed as

$$w(\boldsymbol{x}_n, y_n) = \chi\left\{y_n < \frac{g(\boldsymbol{x}_n)}{\gamma}\right\} = \begin{cases} 1 & \text{if } y_n < \frac{g(\boldsymbol{x}_n)}{\gamma}, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the sum $\sum_{n=1}^{N^*} w(\boldsymbol{x}_n, y_n)$ denotes the total number of successful points out of the N^* trial points and the ratio $\sum_{n=1}^{N^*} w(\boldsymbol{x}_n, y_n)/N^*$ can be interpreted as the probability of acceptance. From the acceptance-rejection algorithm, this probability is exactly given by $1/\gamma$. Consequently, an alternate estimator for importance sampling with acceptance-rejection is given by

$$\tilde{A}_N = \frac{\gamma}{N^*} \sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} w(\boldsymbol{x}_n, y_n). \tag{5.2.4}$$

Using integral representation, the above estimate can be regarded as the Monte Carlo evaluation of the following integral:

$$\gamma \int_{[0,1)^s} \int_0^1 \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \chi\left\{y < \frac{g(\boldsymbol{x})}{\gamma}\right\} dy d\boldsymbol{x}. \tag{5.2.5}$$

It should now be obvious that the importance sampling with acceptance-rejection method boils down to integrating a function that is discontinuous. This implies that when a low discrepancy sequence is applied to importance sampling with acceptance-rejection, its effectiveness is severely altered.

5.3 Generalized Smoothed Acceptance-Rejection Method

Equation (5.2.5) indicates that importance sampling using the acceptance-rejection method is discontinuous. In this section, we consider a generalized smoothing technique which smooths the boundary where the binary decision of accepting or rejecting is made. Rather than having an "all-or-nothing" situation; i.e. a weight of 1 or 0, we smooth the discontinuity by introducing a "bridge" near the boundary of discontinuity so that for points in this region, the associated weight $w(\boldsymbol{x}_n, y_n)$ changes gradually from 1 to 0, instead of only restricting to 1 or 0.

More formally, suppose there exists functions g_L and g_U such that

$$0 \leq g_L(\boldsymbol{x}) \leq g^*(\boldsymbol{x}) = g(\boldsymbol{x})\gamma^{-1} \leq g_U(\boldsymbol{x}) \leq 1, \qquad \text{for all } \boldsymbol{x}. \tag{5.3.1}$$

The lower and upper bounds g_L and g_U can be fixed constants or functions that depend on $\boldsymbol{x}$. We consider the following representation of the weight in our smoothing procedure. For ease of reference, we label them as bridge A and bridge B.

Bridge A:
$$\tilde{w}^A(\boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y \in [0, g_L(\boldsymbol{x})), \\ w_1^A(\boldsymbol{x}, y) & \text{if } y \in [g_L(\boldsymbol{x}), g^*(\boldsymbol{x})), \\ w_2^A(\boldsymbol{x}, y) & \text{if } y \in [g^*(\boldsymbol{x}), g_U(\boldsymbol{x})), \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Bridge B:} \qquad \tilde{w}^B(\boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y \in [0, g_L(\boldsymbol{x})), \\ w^B(\boldsymbol{x}, y) & \text{if } y \in [g_L(\boldsymbol{x}), g_U(\boldsymbol{x})), \\ 0 & \text{otherwise.} \end{cases}$$

In either representation, the weights are chosen so that they satisfy the following two conditions:

1. Boundary conditions: For bridge A, we enforce the following constraints

$$\left.\begin{aligned} w_1^A(\boldsymbol{x}, g_L(\boldsymbol{x})) &= 1, \\ w_2^A(\boldsymbol{x}, g_U(\boldsymbol{x})) &= 0, \\ w_1^A(\boldsymbol{x}, g^*(\boldsymbol{x})) &= w_2^A(\boldsymbol{x}, g^*(\boldsymbol{x})), \end{aligned}\right\} \tag{5.3.2}$$

while for bridge B, we have

$$\left.\begin{aligned} w_1^B(\boldsymbol{x}, g_L(\boldsymbol{x})) &= 1, \\ w_1^B(\boldsymbol{x}, g_U(\boldsymbol{x})) &= 0. \end{aligned}\right\} \tag{5.3.3}$$

2. Invariant property:

$$\int_0^1 \tilde{w}^A(\boldsymbol{x}, y)dy = g^*(\boldsymbol{x}) \qquad \text{for bridge A,} \tag{5.3.4}$$

$$\int_0^1 \tilde{w}^B(\boldsymbol{x}, y)dy = g^*(\boldsymbol{x}) \qquad \text{for bridge B.} \tag{5.3.5}$$

Condition 1 ensures that when $y = g_L$ or g_U, the weights have the required values of 1 or 0. Condition 2 ensures that the distribution is correctly sampled so that the

discontinuous integrand (5.2.5) can equivalently be replaced by

$$\gamma \int_{[0,1)^s} \int_0^1 \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \tilde{w}(\boldsymbol{x}, y) dy d\boldsymbol{x}$$

where $\tilde{w}(\boldsymbol{x}_n, y_n) \in \{\tilde{w}^A(\boldsymbol{x}_n, y_n), \tilde{w}^B(\boldsymbol{x}_n, y_n)\}$. This also ensures that the resulting estimate is unbiased. Hence, an alternative Monte Carlo estimator based on the transformed weight can be formulated as

$$\tilde{A}_N = \frac{\gamma}{N^*} \sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} \tilde{w}(\boldsymbol{x}_n, y_n). \tag{5.3.6}$$

The difference between (5.2.4) and (5.3.6) lies in the weight associated with each functional evaluation $f(\boldsymbol{x})/g(\boldsymbol{x})$. In (5.2.4), $\hat{w}(\boldsymbol{x}_n, y_n)$ is restricted to be 0 or 1 while in (5.3.6), this constraint is relaxed. The weight $\tilde{w}(\boldsymbol{x}_n, y_n)$ is defined so that the resulting estimate is smooth and continuous in the required interval. This goal can be attained by an appropriate functional form of $\tilde{w}(\boldsymbol{x}_n, y_n)$ that is smooth and continuous in the intervals $[g_L(\boldsymbol{x}), g^*(\boldsymbol{x})]$ and $[g^*(\boldsymbol{x}), g_U(\boldsymbol{x})]$ for bridge A representation, and $[g_L(\boldsymbol{x}), g_U(\boldsymbol{x})]$ for bridge B representation. There are many candidates for the choice of $\tilde{w}(\boldsymbol{x}_n, y_n)$. Here we only consider piecewise polynomial function for simplicity. We have a choice of fitting two polynomials to bridge A or just one polynomial to bridge B. The decision depends on the number of parameters of the polynomial so that these parameters are uniquely determined to satisfy the boundary condition and invariant property. For the random sequence, the choice of the bridging and the functional form of $\tilde{w}(\boldsymbol{x}_n, y_n)$ have no impact on rate of convergence of both estimators (5.2.4) and (5.3.6), although the variance will be affected.

On the other hand, when the random sequence is substituted by low discrepancy sequence, estimator (5.3.6) yields a much higher convergence rate than (5.2.4) since the discontinuity is effectively eliminated.

We now illustrate the construction of the weights by assuming the weight is either linear or quadratic.

Piecewise Linear

In the first case, we assume the bridge is linear and has bridge A representation. Under the above assumption, the weights $w_1^A(\boldsymbol{x}, y) \equiv w_1^{Linear}(\boldsymbol{x}, y)$ and $w_2^A(\boldsymbol{x}, y) \equiv w_2^{Linear}(\boldsymbol{x}, y)$ can be represented as

$$w_1^{Linear}(\boldsymbol{x}, y) = a_1 + b_1[y - g_L(\boldsymbol{x})]$$

and

$$w_2^{Linear}(\boldsymbol{x}, y) = a_2 + b_2[y - g^*(\boldsymbol{x})]$$

where the parameters $a_j, b_j,\ j = 1, 2$ are determined so as the boundary conditions (5.3.2) and and the invariant property (5.3.4) are satisfied. The boundary conditions lead to

$$a_1 = 1, \tag{5.3.7}$$

$$1 + b_1[g^*(\boldsymbol{x}) - g_L(\boldsymbol{x})] = a_2 = -b_2[g_U(\boldsymbol{x}) - g^*(\boldsymbol{x})]. \tag{5.3.8}$$

From the invariant property, we have

$$\int_0^{g_L(\boldsymbol{x})} 1\,dy + \int_{g_L(\boldsymbol{x})}^{g^*(\boldsymbol{x})} 1 + b_1[y - g_L(\boldsymbol{x})]\,dy + \int_{g^*(\boldsymbol{x})}^{g^*(\boldsymbol{x})} a_2 + b_2(y - g^*(\boldsymbol{x}))\,dy = g^*(\boldsymbol{x}).$$

Evaluating the above expression yields

$$b_1(g^*(\boldsymbol{x}) - g_L(\boldsymbol{x}))^2 = b_2(g_U(\boldsymbol{x}) - g^*(\boldsymbol{x}))^2. \tag{5.3.9}$$

Thus (5.3.8) and (5.3.9) involve two equations with two unknowns. Solving these two simultaneous equations yield the following set of solution:

$$b_1 = -\frac{g_U(\boldsymbol{x}) - g^*(\boldsymbol{x})}{[g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})][g^*(\boldsymbol{x}) - g_L(\boldsymbol{x})]}$$

and

$$b_2 = -\frac{g^*(\boldsymbol{x}) - g_L(\boldsymbol{x})}{[g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})][g_U(\boldsymbol{x}) - g^*(\boldsymbol{x})]}.$$

To summarize, the generalized smoothed acceptance-rejection technique assuming the weight is piecewise linear can be formulated as follow:

$$\tilde{w}^{Linear}(\boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y \in [0, g_L(\boldsymbol{x})), \\ 1 - \dfrac{g_U(\boldsymbol{x}) - g^*(\boldsymbol{x})}{[g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})][g^*(\boldsymbol{x}) - g_L(\boldsymbol{x})]}\,[y - g_L(\boldsymbol{x})] & \text{if } y \in [g_L(\boldsymbol{x}), g^*(\boldsymbol{x})), \\ \dfrac{g^*(\boldsymbol{x}) - g_L(\boldsymbol{x})}{g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})}\left(1 - \dfrac{1}{g_U(\boldsymbol{x}) - g^*(\boldsymbol{x})}[y - g^*(\boldsymbol{x})]\right) & \text{if } y \in [g^*(\boldsymbol{x}), g_U(\boldsymbol{x})), \\ 0 & \text{otherwise.} \end{cases}$$

Because of the linearity, $\tilde{w}^{Linear}(\boldsymbol{x}, y)$ always lies between 0 and 1.

Piecewise Quadratic

Similarly, when we fit a piecewise quadratic function to the weight in the interval $[g_L(\boldsymbol{x}), g_U(\boldsymbol{x})]$ that has bridge B representation, we obtain

$$\tilde{w}^{Quadratic}(\boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y \in [0, g_L(\boldsymbol{x})), \\ 1 + a[y - g_L(\boldsymbol{x})] + b[y - g_L(\boldsymbol{x})]^2 & \text{if } y \in [g_L(\boldsymbol{x}), g_U(\boldsymbol{x})), \\ 0 & \text{otherwise,} \end{cases}$$

where

$$a = \frac{2[3g^*(\boldsymbol{x}) - g_L(\boldsymbol{x}) - 2g_U(\boldsymbol{x})]}{[g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})]^2}$$

and

$$b = -\frac{3[2g^*(\boldsymbol{x}) - g_L(\boldsymbol{x}) - g_U(\boldsymbol{x})]}{[g_U(\boldsymbol{x}) - g_L(\boldsymbol{x})]^3}.$$

In this case, the weight $\tilde{w}^{Quadratic}(\boldsymbol{x}, y)$ is no longer restricted to $[0, 1]$. In either functional form, the weight is continuous in the required intervals but at the boundary it is not differentiable. Differentiability can be accomplished by a cubic spline but we do not do this here since the computational results in Morokoff and Caflisch [60] indicate that differentiability is not as critical as continuity in Monte Carlo integration using low discrepancy sequence.

We now consider another special case of the bridging that is piecewise linear. Suppose

$$g_L(\boldsymbol{x}) = g^*(\boldsymbol{x}) - \frac{\delta}{2} \qquad \text{and} \qquad g_U(\boldsymbol{x}) = g^*(\boldsymbol{x}) + \frac{\delta}{2}, \tag{5.3.10}$$

the modified weight $\tilde{w}^{Linear}(\boldsymbol{x}, y)$, which we denote by $\tilde{w}^{M\text{-}C}(\boldsymbol{x}, y)$, becomes

$$\tilde{w}^{M\text{-}C}(\boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y \in [0, g^*(\boldsymbol{x}) - \frac{\delta}{2}), \\ \frac{1}{\delta}\left(g^*(\boldsymbol{x}) + \frac{\delta}{2} - y\right) & \text{if } y \in [g^*(\boldsymbol{x}) - \frac{\delta}{2}, g^*(\boldsymbol{x}) + \frac{\delta}{2}), \\ 0 & \text{otherwise.} \end{cases}$$

This coincides to the smoothed acceptance-rejection technique proposed by Moskowitz and Caflisch [61].

Comparing $\tilde{w}^{Linear}(\boldsymbol{x}, y), \tilde{w}^{Quadratic}(\boldsymbol{x}, y)$ and $\tilde{w}^{M\text{-}C}(\boldsymbol{x}, y)$, we note that the generalized smoothed acceptance-rejection method offers a greater flexibility in defining the weights. Both $g_L(\boldsymbol{x})$ and $g_U(\boldsymbol{x})$ can be constants or any parametric form as long as condition (5.3.1) holds. In practice, the implementation of the generalized smoothing technique is facilitated by convenient choices of $g_L(\boldsymbol{x})$ and $g_U(\boldsymbol{x})$. In particular, suppose $g^*(\boldsymbol{x})$ is a complicated function and g_L and g_U are constants or simple functions, the generalized smoothing technique reduces the computation by avoiding evaluating $g^*(\boldsymbol{x})$ whenever $y \in [0, g_L(\boldsymbol{x}))$ or $y \in [g_U(\boldsymbol{x}), 1)$. This is in contrast to Moskowitz and Caflisch's smoothed acceptance-rejection method which requires the calculation of $g^*(\boldsymbol{x})$ for all trial points.

Another inherent difficulty for the Moskowitz and Caflisch's smoothing tech-

nique is in choosing the appropriate smoothing constant δ. They argue that the constant δ should be sufficiently small. However, if δ is made too small, the smoothed acceptance-rejection method reduces to the ordinary acceptance-rejection method, hence losing the advantages of continuity. More crucially, Moskowitz and Caflisch's smoothing technique leads to an estimate that is biased for arbitrary δ. This follows from the fact the invariant property is violated since both $g_L(\boldsymbol{x})$ and $g_U(\boldsymbol{x})$ are simply defined as a flat adjustment to $g^*(\boldsymbol{x})$ (see Equation (5.3.10)). Such choices do not guarantee $g_L(\boldsymbol{x}) \geq 0$ and $g_U(\boldsymbol{x}) \leq 1$ for all $\boldsymbol{x}$. In the extreme case where $g_L(\boldsymbol{x}) < 0$ and $g_U(\boldsymbol{x}) > 1$ for all $\boldsymbol{x}$, i.e. the points are always accepted, we have

$$\begin{aligned} \int_0^1 \tilde{w}^{M\text{-}C}(\boldsymbol{x}, y)dy &= \int_0^1 \frac{1}{\delta}\left(g^*(\boldsymbol{x}) + \frac{\delta}{2} - y\right) dy \\ &= \frac{1}{\delta}\left(g^*(\boldsymbol{x}) + \frac{\delta}{2} - \frac{1}{2}\right) \end{aligned}$$

which does not reduce to $g^*(\boldsymbol{x})$ (unless $\delta = 1$) as required by the invariant property (5.3.5). This implies that if the acceptance-rejection method is replaced by the above smoothed acceptance-rejection method, the accepted point $\boldsymbol{x}_n$ with acceptance weight $\tilde{w}^{M\text{-}C}(\boldsymbol{x}_n, y_n)$ does not have the necessary density g. Consequently, the resulting estimate is biased. To ensure that the original functional value is not distorted after Moskowitz and Caflisch's transformation, we need to introduce an additional constraint. For arbitrary δ, the smoothing constant satisfies $\delta \leq \xi$ where ξ is the maximum constant for which the following two conditions hold:

$$\sup_{\boldsymbol{x}\in[0,1)^s} g^*(\boldsymbol{x}) + \frac{\xi}{2} \leq 1 \qquad \text{and} \qquad \inf_{\boldsymbol{x}\in[0,1)^s} g^*(\boldsymbol{x}) - \frac{\xi}{2} \geq 0.$$

The proof for the convergence of the smoothed acceptance-rejection method given in Moskowitz and Caflisch [61] implicitly assumes that δ is chosen such that $\delta \leq \xi$. On the other hand, even if ξ does exist, it will always be more efficient using generalized smoothed acceptance-rejection method with $g_L = \inf_{\boldsymbol{x} \in [0,1)^s} g^*(\boldsymbol{x})$ and $g_U = \sup_{\boldsymbol{x} \in [0,1)^s} g^*(\boldsymbol{x})$. The numerical examples discussed in the following section also support this argument.

5.4 Numerical Experiments

We now compare the efficiency of the proposed generalization to Moskowitz and Caflisch's smoothing method. We also consider the weighted uniform sampling method. The weighted uniform sampling method involves taking the ratio of two direct Monte Carlo integration estimates — an estimate of the original function f in the numerator, and an estimate of the importance function g in the denominator. The weighted uniform sampling method is continuous (as long as f and g are continuous) and does not require the determination of the acceptance weight for each point. This method also has the advantage that sample points need not be generated for the density g. Furthermore, a substantial variance reduction can be achieved due to the strong positive correlation between f and g.

On the other hand, for problems with large regions of low importance, the weighted uniform sampling method requires more function evaluations than if importance sampling is used. In this situation, importance sampling using acceptance-rejection method would be more efficient than the weighted uniform sampling method. The weighted uniform sampling method also has the disadvantage that

the estimate is biased although Powell and Swann [82] and Spanier and Maize [98] shown that as $N \to \infty$, the bias is negligible in comparison with the RMSE.

For the examples studied below, we consider the following estimates:

1. Direct Monte Carlo method: $\hat{A}_N^{(1)} = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n), \quad \boldsymbol{x}_n \sim U[0,1)^s.$

2. Weighted uniform sampling method:

$$\hat{A}_N^{(2)} = \frac{\sum_{n=1}^{N} f(\boldsymbol{x}_n)}{\sum_{n=1}^{N} g(\boldsymbol{x}_n)}, \quad \boldsymbol{x}_n \sim U[0,1)^s.$$

3. Acceptance-rejection method: $\hat{A}_N^{(3)} = \sum_{n=1}^{N} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)}, \quad \boldsymbol{x}_n \sim g(\boldsymbol{x}).$

4. Moskowitz and Caflisch's smoothed acceptance-rejection method:

$$\hat{A}_N^{(4)} = \frac{1}{N}\sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} \tilde{w}^{M\text{-}C}(\boldsymbol{x}_n, y_n), \quad (\boldsymbol{x}_n, \tilde{w}^{M\text{-}C}(\boldsymbol{x}_n, y_n)) \sim g(\boldsymbol{x}).$$

5. Generalized smoothed acceptance-rejection method with linear weights:

$$\hat{A}_N^{(5)} = \frac{1}{N}\sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} \tilde{w}^{Linear}(\boldsymbol{x}_n, y_n), \quad (\boldsymbol{x}_n, \tilde{w}^{Linear}(\boldsymbol{x}_n, y_n)) \sim g(\boldsymbol{x}).$$

6. Generalized smoothed acceptance-rejection method with quadratic weights:

$$\hat{A}_N^{(6)} = \frac{1}{N}\sum_{n=1}^{N^*} \frac{f(\boldsymbol{x}_n)}{g(\boldsymbol{x}_n)} \tilde{w}^{Quadratic}(\boldsymbol{x}_n, y_n), \quad (\boldsymbol{x}_n, \tilde{w}^{Quadratic}(\boldsymbol{x}_n, y_n)) \sim g(\boldsymbol{x}).$$

For the last three estimators, N^* is chosen so that the sum of the acceptance weights is within one unit of N. Note that the dimension of the sequence for the

methods involving acceptance-rejection always has one dimension more than the direct Monte Carlo or the weighted uniform sampling method. The extra dimension serves as a decision maker for determining the appropriate weight associates with each point.

The efficiency of the underlying method is assessed by comparing the resulting root-mean-squared error (RMSE). For a given value of N, the RMSE over M empirical estimates is defined as follow:

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(\hat{A}_{N,k}^{(j)} - A)^2}, \qquad j = 1, \ldots, 6$$

where $\hat{A}_{N,k}^{(j)}$ denotes the k-th empirical estimate of $\hat{A}_N^{(j)}$ and A is the true value of the underlying problem; i.e. $A = \int_{[0,1)^s} f(\boldsymbol{x})d\boldsymbol{x}$. In our examples, the true value of the integral is approximated to high degree of accuracy either using quadrature method or direct Monte Carlo method with enormous number of sample points. We set $M = 50$ and compute RMSE at $N = 2^i$, $10 \leq i \leq 19$. It is important to note that when calculating RMSE, the point set is not only distinct for each repetition, it is also different across different values of N. For instance, to estimate $\hat{A}_{N_2,k}^{(j)}$ where we have just computed $\hat{A}_{N_1,k}^{(j)}$ with $N_2 > N_1$, in practice, we often carry out the simulation by only adding $N_2 - N_1$ points to the existing N_1 points to obtain the error estimate using N_2 points. In our simulation, we deliberately use a fresh partition of the sequence consists of N_2 elements so that the points used to compute $\hat{A}_{N_1,k}^{(j)}$ are not reused in $\hat{A}_{N_2,k}^{(j)}$. This ensures the estimates $\hat{A}_{N_1,k}^{(j)}$ and $\hat{A}_{N_2,k}^{(j)}$ are not correlated so that the RMSE calculated for each N is "independent" of the

error calculated for other values of N.[1]

Once RMSEs have been computed for several different values of N, we determine the empirical convergence rate from the following relationship:

$$\text{RMSE}(\hat{A}_N^{(j)}) = cN^{-\alpha} \tag{5.4.1}$$

and

$$\log \text{RMSE}(\hat{A}_N^{(j)}) = \log c - \alpha \log N$$

The values α and c can be estimated from linear regression on the empirical data. Although the rate of convergence estimated using above relationship may not represent the asymptotic rate, it provides an adequate measure of efficiency for the finite number of sampling points considered in practice. Therefore the parameter α can be interpreted as an indicator of performance; the higher the α, the greater the rate of convergence. For random sequence, we would expect $\alpha = 1/2$ while for low discrepancy sequence, we would expect α to be greater than $1/2$, say approaching 1, and a smaller coefficient c.

Two examples are used in our comparison. The first example is a 7-dimensional problem extracted from Example 3 of Moskowitz and Caflisch [61] while second example is an extension of the first example to high dimension ($s = 20$).

---

[1]Strictly speaking, this statement is valid only for random sequences. Points in low discrepancy sequence are correlated and the resulting RMSEs are inevitably correlated even when different part of the sequence is used. In the present studies, the "independency" issue is not of critical concern.

Example 5.1 *Let* $\boldsymbol{x} \in [0,1)^7$ *and consider the function*

$$f_1(\boldsymbol{x}) = e^{1-\sum_{i=1}^{3} \sin^2(\frac{\pi}{2}x_i)} \arcsin\left(\sin(1) + \frac{\sum_{i=1}^{7} x_i}{200}\right)$$

using the positive definite importance function:

$$g_1(\boldsymbol{x}) = \frac{1}{\eta} e^{1-\sum_{i=1}^{3} \sin^2(\frac{\pi}{2}x_i)}$$

where η *is the normalizing constant such that*

$$\eta = \int_{[0,1)^7} g_1(\boldsymbol{x})\, d\boldsymbol{x} = e \cdot \left(\int_0^1 e^{-\sin^2(\frac{\pi}{2}x)} dx\right)^3$$

Example 5.2 *In this example, we assume* $\boldsymbol{x} \in [0,1)^{20}$ *and set*

$$f_2(\boldsymbol{x}) = e^{2-\sum_{i=1}^{10} \sin^2(\frac{\pi}{4}x_i)} \arcsin\left(\sin(1) + \frac{\sum_{i=1}^{20} x_i}{200}\right).$$

Similarly, we consider the importance function

$$g_2(\boldsymbol{x}) = \frac{1}{\eta} e^{2-\sum_{i=1}^{10} \sin^2(\frac{\pi}{4}x_i)}$$

where η *is the normalizing constant such that*

$$\eta = \int_{[0,1)^{20}} g_2(\boldsymbol{x})\, d\boldsymbol{x} = e^2 \cdot \left(\int_0^1 e^{-\sin^2(\frac{\pi}{4}x)} dx\right)^{10}$$

In these two examples, we set $\gamma_1 = \sup_{\boldsymbol{x}\in[0,1)^7} g_1(\boldsymbol{x})\eta^{-1}$ and $\gamma_2 = \sup_{\in[0,1)^{20}} g_2(\boldsymbol{x})\eta^{-1}$

respectively. This results in $\gamma_1 = 3.7$ for Example 5.1 and $\gamma_1 = 5.5$ for Example 5.2. Hence the acceptance-rejection methods for these two examples on average require $3.7N$ and $5.5N$ as many points before obtaining N accepted points.

For Moskowitz and Caflisch's smoothed acceptance-rejection method, we consider $\delta = 0.1$ and 0.2 for the smoothing constants. This is arbitrary but Moskowitz and Caflisch [61] have reported success with $\delta = 0.2$. For the generalized smoothing techniques, we also consider two possible candidates of $(g_L(\boldsymbol{x}), g_U(\boldsymbol{x}))$; namely

$$\text{I:} \qquad \left(\inf_{\boldsymbol{x}\in[0,1)^7} \frac{1}{\gamma_1\eta} g_1(\boldsymbol{x}), \quad \sup_{\boldsymbol{x}\in[0,1)^7} \frac{1}{\gamma_1\eta} g_1(\boldsymbol{x}) \right)$$

$$\text{II:} \qquad \left(\frac{1}{\gamma_1\eta} e^{1-\sum_{i=1}^{3} \sin(\frac{\pi}{2}x_i)}, \quad \sup_{\boldsymbol{x}\in[0,1)^7} \frac{1}{\gamma_1\eta} g_1(\boldsymbol{x}) \right)$$

for Example 5.1, and

$$\text{I:} \qquad \left(\inf_{\boldsymbol{x}\in[0,1)^{20}} \frac{1}{\gamma_2\eta} g_2(\boldsymbol{x}), \quad \sup_{\boldsymbol{x}\in[0,1)^{20}} \frac{1}{\gamma_2\eta} g_2(\boldsymbol{x}) \right)$$

$$\text{II:} \qquad \left(\frac{1}{\gamma_2\eta} e^{2-\sum_{i=1}^{10} \sin(\frac{\pi}{4}x_i)}, \quad \sup_{\boldsymbol{x}\in[0,1)^{20}} \frac{1}{\gamma_2\eta} g_2(\boldsymbol{x}) \right)$$

for Example 5.2. Note that condition (5.3.1) is satisfied in all these choices. The first set of candidates (i.e. I) is in general preferred to the second set since both g_L and g_U are constants and hence eliminate the additional computation of these two functions.

Figures 5.1, 5.2 and 5.3 provides a comparison of the simulated results using random sequence generated from RAN2 from Press et. al. [84], and low discrepancy sequences using the construction methods due to Halton [32] and Sobol' [93]. The

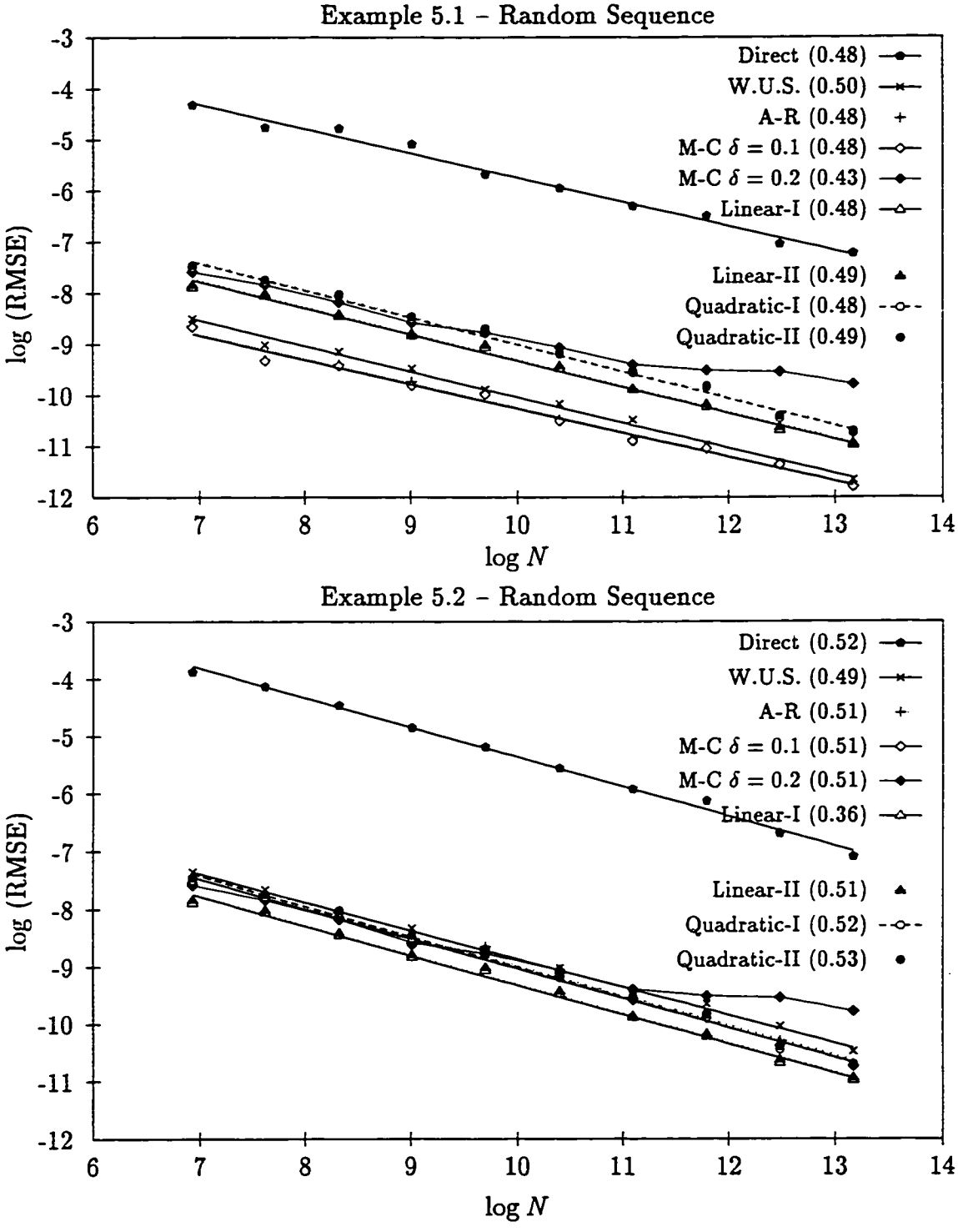


Figure 5.1: Relative Efficiency using Random Sequence

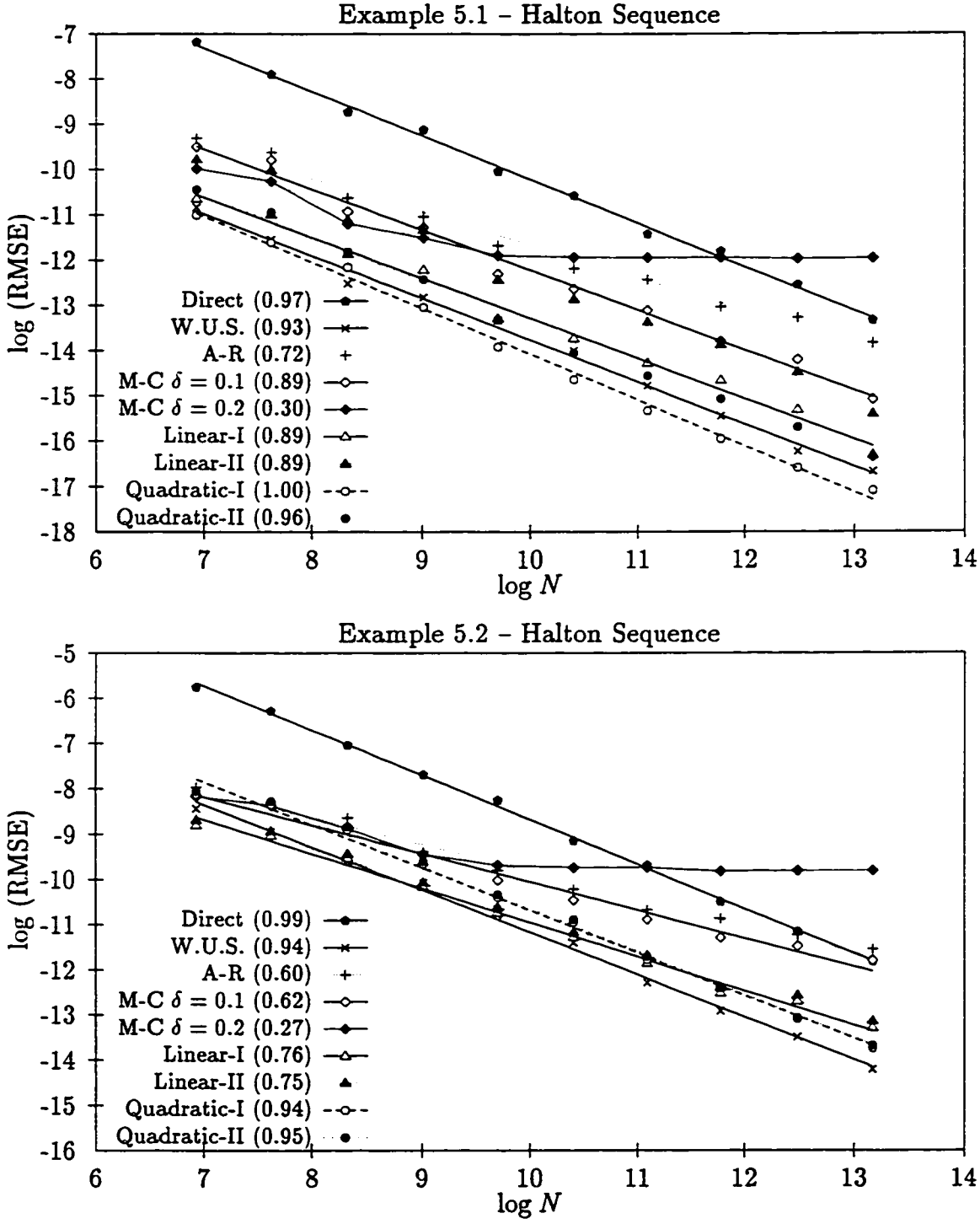


Figure 5.2: Relative Efficiency using Halton Sequence

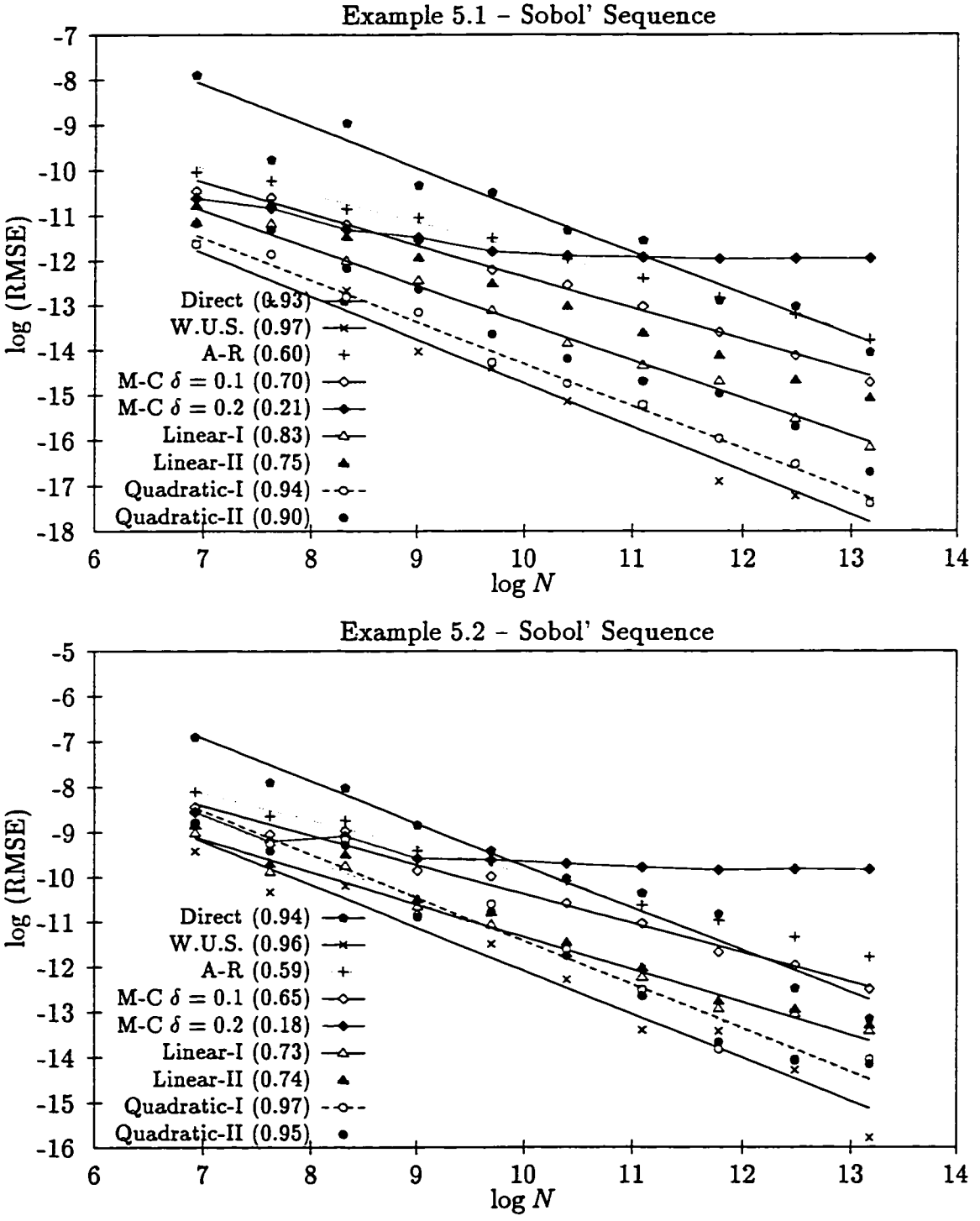


Figure 5.3: Relative Efficiency using Sobol' Sequence

results are presented on a log-log scale. The parameter α in (5.4.1) is obtained from a least squares linear fit and is reported in parenthesis for each method together with the fitted straight line (except for Moskowitz and Caflisch's smoothing technique with $\delta = 0.2$, see point 4 below). We summarize the results as follow:

1. For the range of sample points considered in these examples, the simulation methods coupled with importance function have resulted in substantial reduction of error estimates over the direct Monte Carlo method. This is evidenced by the sets of lower fitted lines for these methods; i.e. the coefficients c in (5.4.1) are smaller for Monte Carlo using importance sampling or weighted uniform sampling than the direct Monte Carlo method.

2. Although weighted uniform sampling and importance sampling using acceptance-rejection with or without smoothing are effective in reducing the error, the rate of convergence remains unaffected for the random sequence. The fitted parameter value α in Figure 5.1 is relatively stable at the anticipated level 0.5. This implies that the continuity adjustment has no impact on random sequence. In contrast, when the random sequence is replaced by the low discrepancy sequence, we observe a wide range of efficiency. In particular, there is a significant decay in the rate of convergence when switching the direct Monte Carlo method to Monte Carlo with importance sampling using the ordinary acceptance-rejection method. (Compare for example, $\alpha \approx 0.93$ for the direct Monte Carlo method to $\alpha \approx 0.6$ for importance sampling with acceptance-rejection method using Sobol' sequence.) This supports the argument that discontinuity has a detrimental impact on the efficiency of the low

discrepancy sequence.

3. The simulation result also reaches the same conclusion as Moskowitz and Caflisch [61] on the superiority of the weighted uniform sampling over importance sampling with smoothed acceptance-rejection. However, Figures 5.2 and 5.3 also indicate the proposed generalized smoothing techniques can achieve a significant improvement over Moskowitz and Caflisch's smoothing technique. In fact, the generalized smoothing technique using quadratic weights candidate I as (g_L, g_U) is as efficient as the weighted uniform sampling.

4. The results in Figures 5.1, 5.2 and 5.3 also reveal the bias in Moskowitz and Caflisch's smoothing technique with $\delta = 0.2$. This is to be anticipated since we have set $\gamma_j = \sup_{\boldsymbol{x}\in[0,1)^s} \frac{1}{\eta} g_j(\boldsymbol{x})$, and this leads to $\sup_{\boldsymbol{x}\in[0,1)^s} g^*(\boldsymbol{x}) = 1$ for both examples. This implies that with positive probability, $g^*(\boldsymbol{x}) + \frac{\delta}{2}$ can exceed 1. Hence condition (5.3.1) is violated. The severity of violation increases with δ. As noted earlier, this leads to an estimate that is biased and consequently accounts for the virtually flat RMSE for $N \geq 32{,}768$ in both examples. The estimated parameter α also becomes meaningless in this case. When $\delta = 0.1$, Moskowitz and Caflisch's smoothing technique seems to have improved over the ordinary acceptance-rejection method. For instance, consider the Sobol' sequence. The estimator α is approximately 0.70 and 0.65 for Examples 5.1 and 5.2, which are better than 0.60 and 0.59 for the ordinary acceptance-rejection method. The reason for this behaviour is that for the ranges of N considered in these examples, the bias arises from using $\alpha = 0.1$ is negligible compared to the RMSE. If N is increased to sufficiently

large, the RMSE would eventually become "flat" as in the case for $\alpha = 0.2$.

5.5 Conclusion

It is well documented that continuity of the integrand plays a crucial role in the efficiency of the Monte Carlo integration using low discrepancy sequence. Unfortunately, discontinuous functions are not uncommon in Monte Carlo methods. Examples include integrating a characteristic function or the acceptance-rejection method. In this chapter, we consider a generalized smoothing technique. This is a simple procedure for transforming a discontinuous function to a continuous function. The continuity adjustment has no effect on the rate of convergence for random sequences. However, a significant improvement can be achieved using low discrepancy sequence. Because of its simplicity and flexibility, the proposed generalization is more efficient than Moskowitz and Caflisch's smoothing technique. Furthermore, the proposed method is unbiased and is comparable to the weighted uniform sampling.

Chapter 6

Lattice Points Methods

6.1 Introduction

In earlier chapters, we have applied low discrepancy sequences to value derivative securities with dimensions as high as 365. We found that LD method, in general, is significantly more efficient than Monte Carlo method. This is also consistent with the findings reported in Acworth, Broadie and Glasserman [1], Boyle, Broadie and Glasserman [10], Caflisch, Morokoff and Owen [16], Galanti and Jung [27], Joy, Boyle and Tan [46], Ninomiya and Tezuka [72] and Paskov and Traub [80] using different types of financial instruments. All of these studies have emphasized on high-dimensional applications and overlooked those financial instruments with low dimensions. Nevertheless there are many low-dimensional financial contracts which also require efficient numerical techniques in situations where analytic solutions are not available or analytic solutions are possible but are too complicated to evaluate. One example is the spread options. Although these options only involve two

dimensions, analytic solutions are not available and hence various numerical approximation techniques have been proposed. Other examples include basket options, options on the maximum or minimum of s assets, (generalized) rainbow options or multi-strike options and exotic options such as high water mark options and Asian options which are embedded in the equity-indexed annuities. These options are typically associated with low dimensions ranging from 2 to 7. In these cases, we would expect LD methods to be very efficient relative to Monte Carlo methods, inferring from their success in much higher dimensions. In this chapter, we show that using number-theoretic methods such as the good lattice points method is superior to LD methods when an additional regularity of the integrands can be assumed.

In the last chapter, we show that discontinuity has an adverse impact on the superiority of the LD methods. On the other hand, any additional regularity of the integrand, besides having bounded variation in the sense of Hardy and Krause, is not reflected in the Koksma-Hlawka error bound. This is in contrast to classical one-dimensional integration rules, such as the Gaussian formulas, which are designed to exploit the particular regularity class of the integrand to achieve a higher rate of convergence.

In this chapter, we discuss another family of numerical quadrature formulas known as method of good lattice points. These lattice points, which are constructed based on number-theoretic methods, are specifically tailored to periodic functions. The method of good lattice points achieves a rate of convergence of order $\mathcal{O}(N^{-\alpha}\log^{\alpha\beta} N)$, for positive constants α and β which does not depend on N. For certain values of α and β, this rate compares favorably to the rate $\mathcal{O}(N^{-1}\log^{s-1} N)$

attained by LD methods with fixed point set.

The purpose of this chapter is to show that whenever an additional continuity exists for the integrands of interest, the lattice points are preferred to low discrepancy points. In general, most integrands do not have the required periodicity. Ways for circumventing such problem are also discussed in this chapter.

The rest of the chapter is organized as follows: Section 6.2 provides an overview on the method of good lattice points. Section 6.3 provides a brief discussion on pricing European-style derivative securities. Section 6.4 considers several applications using the method of good lattice points. In particular, we examine spread options, exchange options, generalized rainbow options and path-dependent options such as lookback options and Asian options. Section 6.5 concludes the chapter.

6.2 The Method of Good Lattice Points

In late 1950s and early 1960s, both N.M. Korobov and E. Hlawka introduced an interesting family of numerical quadrature formulas that are specifically tailored to periodic function of several variables. These formulas take the form

$$\int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} \approx \frac{1}{N}\sum_{k=1}^{N} f\left(\left\{\frac{k}{N}\boldsymbol{z}\right\}\right)$$

where N is the chosen number of quadrature points and $\boldsymbol{z} = \boldsymbol{z}(N) \in \mathbf{Z}^s$ is a carefully chosen integer vector that depends on N and $\{\boldsymbol{x}\} = \boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor$. Since we have assumed

f has period 1 in each variable, the above expression can be reformulated as

$$\int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} \approx \frac{1}{N}\sum_{k=1}^{N} f\left(\frac{k}{N}\boldsymbol{z}\right) \tag{6.2.1}$$

The efficiency of these rules critically depends on the number-theoretic properties of the integer vector $\boldsymbol{z}$. It has been established that, as $N \to \infty$, there exist a vector $\boldsymbol{z}(N)$ such that the quadrature error is of the following order

$$\left|\int_{[0,1)^s} f(\boldsymbol{u}) - \frac{1}{N}\sum_{k=1}^{N} f\left(\frac{k}{N}\boldsymbol{z}\right)\right| = \mathcal{O}\left(\frac{\log^{\alpha\beta} N}{N^{\alpha}}\right) \tag{6.2.2}$$

for positive constants α and β and for integrands f belonging to certain class of smooth periodic function.

The vector $\boldsymbol{z}$ that satisfies (6.2.2) is generally known as the *good lattice points* (g.l.p.) or the *optimal coefficients*. The term "g.l.p." was first coined by Hlawka [38] while the term "optimal coefficient" is due to Korobov [52]. The resulting quadrature method is known as the method of g.l.p.. More recently this quadrature rule is also known as *rank-1 rule*. See Sloan and Joe [92] for details.

6.2.1 Error Bounds

In this subsection, we discuss error bounds corresponding to the method of g.l.p.. Let $f(\boldsymbol{u}), \boldsymbol{u} \in [0,1)^s$ be a periodic function on $\Re^s$ with period 1 in each of its s

variables. Let $\hat{f}(\boldsymbol{h})$, $\boldsymbol{h} \in \mathbf{Z}^s$, denote the Fourier coefficient of f so that

$$f(\boldsymbol{u}) = \sum_{\boldsymbol{h} \in \mathbf{Z}^s} \hat{f}(\boldsymbol{h}) \exp(2\pi \, \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{u}) \tag{6.2.3}$$

where

$$\hat{f}(\boldsymbol{h}) = \int_{[0,1)^s} f(\boldsymbol{u}) \exp(-2\pi \, \mathrm{i} \boldsymbol{h} \cdot \boldsymbol{u}) d\boldsymbol{u} \tag{6.2.4}$$

and the inner product $\boldsymbol{h} \cdot \boldsymbol{u} = h_1 u_1 + h_2 u_2 + \cdots + h_s u_s$.

For fixed $\alpha > 1$, we say f belongs to the class of continuous periodic function $\mathcal{E}_\alpha^s(C)$ provided that for all nonzero $\boldsymbol{h} \in \mathbf{Z}^s$, we have

$$|\hat{f}(\boldsymbol{h})| \leq C \left[\prod_{i=1}^{s} \max(1, |h_i|) \right]^{-\alpha} = C r(\boldsymbol{h})^{-\alpha} \tag{6.2.5}$$

where $r(\boldsymbol{h}) = \prod_{i=1}^{s} \max(1, |h_i|)$ and C is a constant does not depend on $\boldsymbol{h}$. The rate of decay of the Fourier coefficient of a function is related to the smoothness of the function. Hence $\mathcal{E}_\alpha^s(C)$ can be interpreted as characterizing a class of function of certain smoothness. Zaremba [108] proved the following sufficient condition for membership of this class when $\alpha > 1$: If f is a one-periodic function on $\Re^s$ whose partial derivatives

$$\frac{\partial^{\alpha_1 + \cdots + \alpha_s} f}{\partial u_1^{\alpha_1} \cdots \partial u_s^{\alpha_s}} \qquad \text{with} \quad 0 \leq \alpha_i \leq \alpha - 1 \quad \text{for} \quad 1 \leq i \leq s$$

exist and are of bounded variation on $[0,1)^s$ in the sense of Hardy and Krause, then there exists an explicit C for which $f \in \mathcal{E}_\alpha^s(C)$.

Now consider approximating the value of the integral using the quadrature formula as given in (6.2.1). We also assume $f \in \mathcal{E}_\alpha^s(C)$ so that its Fourier series is absolutely convergent and represents f. Substituting (6.2.3) into the right-hand side of (6.2.1), we obtain

$$\begin{aligned} \frac{1}{N}\sum_{k=1}^{N} f\left(\frac{k}{N}\boldsymbol{z}\right) &= \frac{1}{N}\sum_{k=1}^{N}\sum_{\boldsymbol{h}\in\mathbf{Z}^s} \hat{f}(\boldsymbol{h})\exp\left(2\pi\,\mathrm{i}\,\frac{k}{N}\boldsymbol{h}\cdot\boldsymbol{z}\right) \\ &= \frac{1}{N}\sum_{\boldsymbol{h}\in\mathbf{Z}^s} \hat{f}(\boldsymbol{h})\sum_{k=1}^{N}\exp\left(2\pi\,\mathrm{i}\,\frac{k}{N}\boldsymbol{h}\cdot\boldsymbol{z}\right). \end{aligned} \tag{6.2.6}$$

Note that

$$\sum_{k=1}^{N}\exp\left(2\pi\,\mathrm{i}\,\frac{k}{N}\boldsymbol{h}\cdot\boldsymbol{z}\right) = \begin{cases} N, & \text{if } \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \mod N \\ 0, & \text{otherwise.} \end{cases}$$

and $\hat{f}(0) = \int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u}$. Rearranging (6.2.6) yields

$$\frac{1}{N}\sum_{k=1}^{N} f\left(\frac{k}{N}\boldsymbol{z}\right) - \int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} = \sum_{\boldsymbol{h}\neq\boldsymbol{0}} \delta_N(\boldsymbol{h}\cdot\boldsymbol{z})\hat{f}(\boldsymbol{h}) \tag{6.2.7}$$

where

$$\delta_N(\boldsymbol{h}\cdot\boldsymbol{z}) = \begin{cases} 1 & \text{if } \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \mod N \\ 0 & \text{otherwise.} \end{cases}$$

The sum in the right-hand side of (6.2.7) is over all nonzero $\boldsymbol{h}\in\mathbf{Z}^s$ with $\boldsymbol{h}\cdot\boldsymbol{z}\equiv 0$ mod N. Hence, the quadrature error is represented by the sum of certain Fourier

coefficients of f.

Since $f \in \mathcal{E}_\alpha^s(C)$, it follows from (6.2.5) that the quadrature error is bounded from above by

$$\left| \frac{1}{N} \sum_{k=1}^{N} f\left(\frac{k}{N}\boldsymbol{z}\right) - \int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} \right| \leq C P_\alpha(\boldsymbol{z}, N) \tag{6.2.8}$$

where

$$P_\alpha(\boldsymbol{z}, N) = \sum_{\boldsymbol{h} \neq \boldsymbol{0}} \frac{\delta_N(\boldsymbol{z} \cdot \boldsymbol{h})}{r(\boldsymbol{h})^\alpha}. \tag{6.2.9}$$

It follows from (6.2.8) that it is desirable to use a vector $\boldsymbol{z}$ such that $P_\alpha(\boldsymbol{z}, N)$ is small for a given N. Observe also that the quantity $P_\alpha(\boldsymbol{z}, N)$ will be small if the smallest $r(\boldsymbol{h})$ in the sum (6.2.9) is large. Hence we can define g.l.p. or the optimal coefficients independent of α as follows:

Definition 6.1 *Let $N_1, N_2, \ldots$ be an increasing sequence of positive integers. A sequence $\boldsymbol{z}(N_1), \boldsymbol{z}(N_2), \ldots$ of s-tuples of integers is a* ***good lattice points sequence*** *or an* ***optimal coefficient sequence*** *(and each $\boldsymbol{g}(N_i)$ is known as a* ***set of good lattice point*** *or a* ***set of optimal coefficient modulo*** *N_i) if there exist numbers $A > 0$ and β such that for every i, every solution $\boldsymbol{h} = (h_1, \ldots, h_s)$ of the congruence*

$$\boldsymbol{h} \cdot \boldsymbol{z}(N_i) \equiv \mod N_i$$

has

$$r(h) > A\frac{N_i}{\log^{\beta} N_i}. \tag{6.2.10}$$

The bound in (6.2.8) implies that if $z(N_i)$ is a good lattice point sequence, we can establish the following error bound

$$\left| \int_{[0,1)^s} f(u) - \frac{1}{N_i} \sum_{k=1}^{N} f\left(\frac{k}{N_i} z(N_i)\right) \right| = \mathcal{O}\left(\frac{\log^{\alpha\beta} N_i}{N_i^{\alpha}}\right) \tag{6.2.11}$$

as $i \to \infty$ and for every $f \in \mathcal{E}_{\alpha}^{s}$. This yields the same relation given in (6.2.2) with $N_i = N$.

For this reason, it is also customary to define g.l.p. z as a s-tuple of integers for which (6.2.11) is satisfied. The lowest β^* for which (6.2.10) holds is known as the *index* of the optimal coefficient sequence and it depends on the dimension s. The optimal coefficient sequences constructed by Korobov [52] has index $\beta^* \leq s$. When N is prime, Bahvalov [4] shows that β^* can be reduced to $s-1$. The work of Šargin [102] indicates that the lower bound on $\alpha\beta^*$ is at least $s-1$. It is only in the case of $s = 2$ where Šargin's lower bound is attainable. To date, the best bounds on $P_{\alpha}(z, N)$ have been those obtained by Niederreiter [69].

For a given pair of N and z, an explicit error bound for rules of the form (6.2.1) can be constructed. The idea is to find a function that belongs to the class $\mathcal{E}_{\alpha}^{s}(C)$ for which the maximum error in (6.2.8) is attained. This particular function, which

we denote by f_α, can be represented as

$$f_\alpha(\boldsymbol{u}) = \sum_{\boldsymbol{h} \in \mathbf{Z}^s} \frac{1}{r(\boldsymbol{h})^\alpha} \exp(2\pi \, \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{u}).$$

Hence, (6.2.8) becomes

$$\left| \frac{1}{N} \sum_{k=1}^{N} f_\alpha \left(\frac{k}{N} \boldsymbol{z} \right) - \int_{[0,1)^s} f_\alpha(\boldsymbol{u}) d\boldsymbol{u} \right| = \hat{P}_\alpha(\boldsymbol{z}, N). \tag{6.2.12}$$

Since f_α is the worst function in the class $\mathcal{E}_\alpha^s(C)$, it also follows from (6.2.8) and (6.2.12) that for any $f \in \mathcal{E}_\alpha^s(C)$, we have

$$\left| \frac{1}{N} \sum_{k=1}^{N} f_\alpha \left(\frac{k}{N} \boldsymbol{z} \right) - \int_{[0,1)^s} f_\alpha(\boldsymbol{u}) d\boldsymbol{u} \right| \leq C \hat{P}_\alpha(\boldsymbol{z}, N). \tag{6.2.13}$$

Therefore, the asymptotic bound (6.2.11) can be replaced by the explicit (and sharp) error bound of (6.2.13) with the assurance from (6.2.11) that if $\boldsymbol{z}$ is a g.l.p., then $\hat{P}_\alpha(\boldsymbol{z}, N)$ will go to zero rapidly as N increases.

6.2.2 Constructions of Good Lattice Points

So far we have indicated that there exists g.l.p. such that the error bound in (6.2.2) (or (6.2.11)) is attainable. We have also discussed that the asymptotic bound (6.2.11) can be replaced by the explicit (and sharp) error bound of (6.2.13). In this subsection, we discuss how to find these g.l.p.. It turns out that these g.l.p. are neither rare nor unique. There exists a simple construction for two-dimensional g.l.p.. In higher dimensions, we only have existence theorems for g.l.p.. In 1959,

Korobov demonstrated the existence of g.l.p. for N that is either prime or product of two primes. It also follows from his existence proof that for prime N, half or more of all the possible vector $\boldsymbol{z}$ can be suitable choices for g.l.p.. Several authors have subsequently relaxed the restriction on N in the existence proof of Korobov. Keast [47] shows the existence of g.l.p. of order N when N is a product of more than two primes while Niederreiter [64] shows that g.l.p. exist for all N.

We now consider the construction of the g.l.p. in two dimensions. This is the only known situation where the lower bound of Šargin is achieved. Hence we obtain an error bound of the form

$$\left| \int_{[0,1)^2} f(\boldsymbol{u}) - \frac{1}{N} \sum_{k=1}^{N} f\left(\frac{k}{N} \boldsymbol{z} \right) \right| = \mathcal{O}\left(\frac{\log N}{N^{\alpha}} \right) \tag{6.2.14}$$

for $s = 2$. The g.l.p. that yields the above rate of convergence is constructed from the Fibonacci numbers. See Bahvalov [4]. Recall that the Fibonacci numbers are defined by

$$F_1 = 1, \qquad F_2 = 1, \qquad F_m = F_{m-1} + F_{m-2}, \quad m \geq 3.$$

where F_m is the m-th Fibonacci number. For any $N = F_m$, the point set

$$\boldsymbol{u}_k = \left(\left\{ \frac{k}{N} \right\}, \left\{ \frac{k F_{m-1}}{N} \right\} \right), \qquad k = 1, \ldots, N$$

generated from the optimal coefficient $\boldsymbol{z} = (1, F_{m-1})$ is a g.l.p. that attains the error bound in (6.2.14). Hence the 2-dimensional g.l.p. exists only for specific values of N. Specifically, the first 15 Fibonacci numbers are {1, 1, 2, 3, 5, 8, 13, 21, 24, 55,

89, 144, 233, 377, 610}.

In higher dimensions, the optimal coefficients are normally found by a computer search. From Definition 6.1, the vector z is defined independent of the smoothness parameter α. This implies that if z is "optimal" or "good" in the sense of Korobov and Hlawka for a fixed value α in $\mathcal{E}_\alpha^s(C)$, it is automatically "optimal" or "good" for other classes $\mathcal{E}_\alpha^s(C)$ with other values of α.[1] Hence it is not necessary to seek g.l.p. that are tailored to the specific smoothness class of the function. The method of g.l.p. automatically exploits all the smoothness in the integrand to achieve a better rate of convergence.

Korobov shown that for fixed α, the g.l.p. sequences can be constructed by calculating $\hat{P}_\alpha(z, N)$ in (6.2.12) over all possible vectors z and choosing the z which minimizes that quantity. For convenience, we assume $\int_{[0,1)^s} f_\alpha = 1$ so that (6.2.12) becomes

$$\left| \frac{1}{N} \sum_{k=1}^{N} f_\alpha \left(\frac{k}{N} z \right) - 1 \right| = \hat{P}_\alpha(z, N).$$

Hence, searching g.l.p. reduces to finding z that minimizes $\hat{P}_\alpha(z, N)$. A more convenient representation of f_α is to express it as a product of functions of a single variable (see Haber [31]):

$$f_\alpha(u) = \prod_{i=1}^{s} F_\alpha(u_i)$$

[1] Of course, the constant C and β will vary with α.

where

$$F_\alpha(u) = \sum_{h=-\infty}^{\infty} \frac{\exp(2\pi i h u)}{\max(1, |h|)}.$$

Normally α is taken to be an even number since in these cases F_α can be expressed in terms of the Bernoulli polynomials. For $\alpha = 2, 4$ and 6, we have

$$\begin{aligned} F_2(u) &= 1 + \frac{\pi^2}{3}(3u(u-1)+1) \\ F_4(u) &= 1 + \frac{2\pi^4}{45}\left(1 - 30u^2(1-u)^2\right) \\ F_6(u) &= 1 + \frac{2\pi^6}{945}\left(1 - 21u^2 + 105u^4 - 126u^5 + 42u^6\right) \end{aligned}$$

for $u \in [0, 1]$.

This direct approach of finding a g.l.p. over all possible $\boldsymbol{z}$ becomes prohibitive for large values of N and s since there are N^s possible choices of $\boldsymbol{z}$ to search through. For this reason, Korobov [52] considers a more restricted but manageable structure of the lattice points that is of the form

$$\boldsymbol{z}(l) = (1,\ l,\ l^2 \bmod N, \ldots,\ l^{s-1} \bmod N), \qquad 1 \le l < N.$$

Each vector $\boldsymbol{z}$ thus consists of only one parameter with a total of $N - 1$ possible choices, as opposed to N^s. A further reduction in the number of vectors to be searched through is still possible by recognizing that

$$F_\alpha(u) = F_\alpha(1-u), \qquad u \in [0, 1].$$

This implies $f_\alpha(kz(l)/N) = f_\alpha((N-k)z(l)/N)$ so that the search is reduced to almost half by considering only $z(l)$ for $1 \leq l \leq \lfloor N/2 \rfloor$.

Tables of g.l.p. obtained by minimizing $\hat{P}_2(s, N)$ can be found in Haber [31]. Other tables of g.l.p. can be found in Saltykov [88], which are reproduced in Stroud [99], Hua and Wang [41], Sloan and Joe [92].

6.2.3 Periodization

We have indicated that the method of g.l.p. exploits the additional regularity of the function such as the periodicity (with respect to each component of $\boldsymbol{u}$). Most functions encountered in practice do not meet these criteria. Some functions may be periodic in some of the components but not all. In these situations we must carry out a preliminary transformation to convert a sufficiently regular nonperiodic integrand into an integrand that has the required periodicity. In doing so, we must take note of the following criteria:

1. The transformation should be relatively simple.

2. The transformation should preserve the value of the underlying integrand. In other words, we must have

$$\int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} = \int_{[0,1)^s} \phi(\boldsymbol{u})d\boldsymbol{u}$$

where f and ϕ are the original and the transformed integrands.

3. Regularity of the function should be preserved in the sense that if all the partial derivatives

$$\frac{\partial^{\alpha_1+\cdots+\alpha_s} f}{\partial u_1^{\alpha_1}\cdots\partial u_s^{\alpha_s}} \qquad \text{with } 0 \le \alpha_i \le \alpha,\ 1 \le i \le s$$

of the original function f exist and are of the bounded variation on $[0,1)^s$ in the sense of Hardy and Krause, then the transformed function ϕ should satisfy the following two properties:

- all the partial derivatives

$$\frac{\partial^{\alpha_1+\cdots+\alpha_s} \phi}{\partial u_1^{\alpha_1}\cdots\partial u_s^{\alpha_s}} \qquad \text{with } 0 \le \alpha_i \le \alpha,\ 1 \le i \le s$$

are of bounded variation on $[0,1)^s$ in the sense of Hardy and Krause:

- $$\left|\frac{\partial^{\alpha_1+\cdots+\alpha_s} \phi}{\partial u_1^{\alpha_1}\cdots\partial u_s^{\alpha_s}}\right|_{u_i=0} = \left|\frac{\partial^{\alpha_1+\cdots+\alpha_s} \phi}{\partial u_1^{\alpha_1}\cdots\partial u_s^{\alpha_s}}\right|_{u_i=1}$$

with $0 \le \alpha_i \le \alpha - 1,\ 1 \le i \le s$.

A good account of the methods for periodizing the integrands can be found in Hua and Wang [41], Zaremba [109] and Beckers and Haegemans [6]. In this subsection, we only describe a particular class of these techniques. These techniques are based on nonlinear transformation such that each variable is defined as

$$u = \psi(t)$$

where ψ is a smooth increasing function which maps $[0,1]$ onto $[0,1]$ and $\psi^{(j)}(0) = \psi^{(j)}(1) = 0$ for $1 \leq j \leq \alpha$. From the standard change-of-variable techniques, the transformed periodized ϕ has the following representation

$$\phi(\boldsymbol{u}) = f(\psi(u_1), \ldots, \psi(u_s))\psi'(u_1)\cdots\psi'(u_s).$$

We now consider two potential choices of ψ. These transformations will be used extensively in our numerical studies.

A. Polynomial Transformations:

Let ψ_m, $m \geq 2$ be a function that satisfies

$$\psi_m(t) = (2m-1)\binom{m-1}{2m-2}\int_0^t (x - x^2)^{m-1}dx.$$

Then for fixed m, the polynomial transformations is given by

$$\psi(t) = \psi_m(t)$$

with $\alpha = m - 1$. The first three polynomial transformations are

$$\begin{aligned}
\psi_2(t) &= 3t^2 - 2t^3 \\
\psi_3(t) &= 10t^3 - 15t^4 + 6t^5 \\
\psi_4(t) &= 35t^4 - 84t^5 + 70t^6 - 20t^7.
\end{aligned}$$

B. $\sin^m$-Transformations: Sidi [91]

Define

$$\psi_m(t) = \frac{\theta_m(t)}{\theta_m(1)}$$

where

$$\theta_m(t) = \int_0^t (\sin \pi x)^m dx, \qquad m = 1, 2, \ldots .$$

Then the first four $\sin^m$-transformations are

$$\begin{aligned}
\psi_1(t) &= \frac{1}{2}(1 - \cos \pi t), \\
\psi_2(t) &= \frac{1}{2\pi}(2\pi t - \sin 2\pi t), \\
\psi_3(t) &= \frac{1}{16}(8 - 9\cos \pi t + \cos 3\pi t), \\
\psi_4(t) &= \frac{1}{12\pi}(12\pi t - 8\sin 2\pi t + \sin 4\pi t).
\end{aligned}$$

These transformations have the advantages of simplicity and easily calculated derivatives. Furthermore, we have

$$\psi_m^{(j)}(0) = \psi_m^{(j)}(1) = 0$$

for $1 \leq j \leq m$.

6.2.4 Error Estimation

In Chapter 4, we have discussed how to obtain an error estimate for quasi-Monte Carlo methods using Owen's methods. Owen's idea was based on randomization which was originally suggested by Cranley and Patterson [19] for obtaining the error estimate in the context of number theoretic methods. Let $Q(\boldsymbol{z}, N)f$ denote the number theoretic rule of order N with g.l.p. $\boldsymbol{z}$. In other words, we have

$$Q(\boldsymbol{z}, N)f = \frac{1}{N}\sum_{k=1}^{N} f\left(\left\{\frac{k}{N}\boldsymbol{z}\right\}\right).$$

For each $\boldsymbol{v} \in [0,1)^s$, we define the shifted number theoretic rule of order N with g.l.p. $\boldsymbol{z}$ as

$$Q(\boldsymbol{z}, N, \boldsymbol{v})f = \frac{1}{N}\sum_{k=1}^{N} f\left(\left\{\frac{k}{N}\boldsymbol{z} + \boldsymbol{v}\right\}\right).$$

Cranley and Patterson show that if $\boldsymbol{v}$ is a vector chosen randomly from a multivariate uniform distribution on $[0,1)^s$ so that each component of $\boldsymbol{v}$ is independently and uniformly distributed on $[0,1]$, then

$$\mathrm{E}[Q(\boldsymbol{z}, N, \boldsymbol{v})f] = If.$$

Consequently, if $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q$ are q independent random vectors drawn from a multivariate uniform distribution on $[0,1)^s$, then

$$\bar{Q}(\boldsymbol{z}, N)f = \frac{1}{q}\sum_{j=1}^{q} Q(\boldsymbol{z}, N, \boldsymbol{v}_j)f$$

is an unbiased estimate of If. This suggests that it is possible to construct the confidence interval by replication with the estimate of the standard error, $\hat{\sigma}$, calculated as

$$\hat{\sigma} = \left\{ \frac{1}{q(q-1)} \sum_{j=1}^{q} [Q(z, N, v_j)f - \bar{Q}(z, N)f]^2 \right\}^{1/2}.$$

6.3 Option Pricings

In this section, we provide a brief overview on pricing European-style derivative securities. Consider an economy with s risky assets with prices $\boldsymbol{S}_t = (S_{1t}, \ldots, S_{st})$ at time t. We assume the risk neutralized asset prices processes satisfy

$$dS_{it} = (r - \delta_i)S_{it}\, dt \; + \; \sigma_i S_{it}\, dW_{it}, \qquad 1 \leq i \leq s$$

where the parameters δ_i, σ_i, $1 \leq i \leq s$, are the annualized dividend yield, and volatility for asset i and r is annualized risk-free rate. The process $\boldsymbol{W}_t = (W_{1t}, \ldots, W_{st})$ are correlated Brownian motions where each W_{it} has drift 0 and variance 1, and the i-th and j-th components have correlation ρ_{ij}. Note that each asset price is a geometric Brownian motion process. Let $\theta_{it} = \log S_{it}$, $1 \leq i \leq s$, then $\boldsymbol{\theta}_t = (\theta_{1t}, \ldots, \theta_{st})$ is normally distributed with mean

$$\begin{aligned} \mu_t &= (\mu_{1t}, \ldots, \mu_{st}) \\ &= \left(\log S_{10} + (r - \delta_1 - \frac{1}{2}\sigma_1^2)t, \ldots, \log S_{s0} + (r - \delta_s \frac{1}{2}\sigma_s^2)t \right) \end{aligned} \quad (6.3.1)$$

and covariance matrix

$$\sum_t = (\sigma_{ij}) = \rho_{ij}\sigma_i\sigma_j t. \tag{6.3.2}$$

Under the risk-neutral measure, the current price of the derivative security is equal to the expected value of its payoff; i.e. the current price of the derivative security is computed as

$$\begin{aligned} V_0 &= e^{-rT}\mathrm{E}_Q[g(\boldsymbol{S}_T)] \\ &= e^{-rT}\int_{\mathcal{A}(\boldsymbol{S}_T)} g(\boldsymbol{S}_T)f(\boldsymbol{S}_T)\, d\boldsymbol{S}_T \end{aligned} \tag{6.3.3}$$

where $g(\boldsymbol{S}_T) = g(S_{1T}, \ldots, S_{sT})$ is the payoff of the European derivative security and depends on the terminal asset prices $(S_{1T}, \ldots, S_{sT})$, $\mathcal{A}(\boldsymbol{S}_T) \in \Re^s$ is the integration domain and $f(\boldsymbol{S}_T)$ is a s-variate lognormal distribution with probability density function

$$f(\boldsymbol{S}_T) = \frac{1}{\sqrt{|\sum|(2\pi)^s}S_{1T}\cdots S_{sT}} \exp\left\{-\frac{1}{2}(\log \boldsymbol{S}_T - \boldsymbol{\mu})'\sum{}^{-1}(\log \boldsymbol{S}_T - \boldsymbol{\mu})\right\}$$

where $\log \boldsymbol{S}_T = (\log(S_{1T}), \ldots, \log(S_{sT}))$, $\boldsymbol{\mu}_T$ and $\sum_T$ are defined in (6.3.1) and (6.3.2).

The price of an European derivative security therefore reduces to evaluating the integral of the form (6.3.3). The complexity of the problem depends on the structure of the payoff function $g(\boldsymbol{S}_T)$. It is only in rare cases that analytic solution are available. In most situations, it is necessary to approximate (6.3.3) using numerical

methods such as finite difference methods, tree approaches or Monte Carlo methods.

6.4 Applications

In this section, we consider the valuation of several exotic options. We demonstrate that for reasonably low dimensions, the multiple integral (6.3.3) can be effectively approximated to high precision using method of g.l.p.. This method also compares favorably to other numerical techniques. The following subsection considers spread options, Subsection 6.4.2 examines a 5-dimensional example based on generalization of the spread options and Subsection 6.4.3 considers path-dependent options such as lookback options and Asian options.

6.4.1 Spread Options

Spread options are options whose payoff depends on the difference in the prices of two underlying assets. In the general case, the payoff of a spread option at maturity can be represented as

$$g(\boldsymbol{S}_T) = \begin{cases} \max[w_2 S_{2T} - w_1 S_{1T} - K,\, 0] & \text{for call option} \\ \max[K - (w_2 S_{2T} - w_1 S_{1T}),\, 0] & \text{for put option} \end{cases}$$

where K is the strike price of the option and the weights w_1 and w_2 are assumed to be positive. When $K = 0$, the spread options reduce to exchange options. The pricing formula for this particular case with $w_1 = w_2 = 1$ is given in Margrabe [58] and Stultz [100].

For non-zero strike prices, analytic expressions for the values of spread options are not known and numerous approximation methods have been proposed. A simple approach suggested by Wilcox [106] is to assume that the spread is normally distributed. This assumption leads to a very simple pricing formula. The drawback of this approach is its inconsistency with the lognormal framework of the asset prices. A more accurate approach is given by Pearson [81]. Other approaches to pricing the spread options include the bivariate binomial and trinomial approximations proposed by Boyle [9], Boyle, Evnine and Gibbs [11] and He [34]. Simulation is also an alternative approach although the drawbacks of this method, as according to Pearson, are "*somewhat limited accuracy and the computational effort involved.*" In this subsection, we show that the number-theoretic methods using g.l.p. can be a very effective tool for pricing these options. The numerical examples also indicate that this method yields much higher accuracy than the approximation algorithm proposed by Pearson [81].

We now discuss how to use the method of g.l.p. to price the spread options. Substituting $g(\boldsymbol{S}_T) = \max[w_2 S_{2T} - w_1 S_{1T} - K,\ 0]$ into (6.3.3), we obtain the current price of the spread call option V_0 as

$$\begin{aligned} e^{rT} V_0 &= \mathrm{E}_Q\big[\max[w_2 S_{2T} - w_1 S_{1T} - K, 0]\big] \\ &= \int_0^\infty \int_0^\infty \max[w_2 S_{2T} - w_1 S_{1T} - K, 0] f(\boldsymbol{S}_T) d\boldsymbol{S}_T \qquad (6.4.1) \\ &= \int_0^\infty \int_{(w_1 S_{1T}+K)/w_2}^\infty (w_2 S_{2T} - w_1 S_{1T} - K) f(\boldsymbol{S}_T) dS_{2T} dS_{1T} \\ &= \int_{-\infty}^\infty \int_{\log(w_1 e^{\theta_{1T}+\mu_{1T}}+K)-\log(w_2)-\mu_{2T}}^\infty \\ &\qquad (w_2 e^{\theta_{2T}+\mu_{2T}} - w_1 e^{\theta_{1T}+\mu_{1T}} - K) h(\boldsymbol{\theta}_T) d\boldsymbol{\theta} \qquad (6.4.2) \end{aligned}$$

where

$$h(\boldsymbol{\theta}_T) = \frac{1}{\sqrt{|\sum_T|(2\pi)^2}} \exp\left(-\frac{1}{2}\boldsymbol{\theta}_T' \sum_T^{-1} \boldsymbol{\theta}_T\right).$$

To obtain (6.4.2), we have used the substitution $S_{iT} = \exp(\theta_{iT} + \mu_{iT})$, $i = 1, 2$.

Note that the integration domain of the integral (6.4.2) is not in $[0,1)^2$. For either method of g.l.p. or Monte Carlo integration, we always assume that the integration domain lies in $[0,1)^s$. This can be accomplished by the following sequence of transformations: first, we let $\boldsymbol{\theta}_T = \boldsymbol{C}\boldsymbol{y}$ where $\boldsymbol{C}$ is the Cholesky decomposition of the covariance matrix $\sum_T$; i.e., $\boldsymbol{C}\boldsymbol{C}' = \sum_T$ where

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1\sqrt{T} & 0 \\ \rho_{12}\sigma_2\sqrt{T} & \sqrt{1-\rho_{12}}\sigma_2\sqrt{T} \end{bmatrix}.$$

Applying this transformation to (6.4.2), we obtain the following expression

$$\frac{1}{\sqrt{(2\pi)^s}} \int_{a_1'}^{b_1'} \int_{a_2'}^{b_2'} (e^{c_{21}y_1 + c_{22}y_2 + \mu_{2T}} - e^{c_{11}y_1 + \mu_{1T}} - K) e^{-\frac{1}{2}(y_1^2 + y_2^2)}\, dy_2 dy_1, \qquad (6.4.3)$$

where $a_1' = -\infty$, $b_1' = b_2' = \infty$ and

$$a_2' = \frac{\log(w_1 e^{c_{11}y_1 + \mu_{1T}} + K) - \log(w_2) - \mu_{2T} - c_{21}y_1}{c_{22}}.$$

Now consider the second phase of the transformation by letting $y_i = \Phi^{-1}(z_i)$.

This results in

$$\int_{d_1}^{e_1}\int_{d_2}^{e_2}(e^{c_{21}\Phi^{-1}(z_1)+c_{22}\Phi^{-1}(z_2)+\mu_{2T}} - e^{c_{11}\Phi^{-1}(z_1)+\mu_{1T}} - K)\,dz, \tag{6.4.4}$$

where $d_1 = \Phi(-\infty) = 0$, $e_1 = e_2 = \Phi(\infty) = 1$ and

$$d_2 = \Phi\left(\frac{\log(w_1 e^{c_{11}\Phi^{-1}(z_1)+\mu_{1T}} + K) - \log(w_2) - \mu_{2T} - c_{21}\Phi^{-1}(z_1)}{c_{22}}\right).$$

The final change-of-variable $z_i = d_i + u_i(e_i - d_i)$ leads to

$$\int_0^1\int_0^1 (1-d_2)(w_2 e^{c_{21}\Phi^{-1}(u_1)+c_{22}\Phi^{-1}(d_2+u_2(1-d_2))+\mu_{2T}} - w_1 e^{c_{11}\Phi^{-1}(u_1)+\mu_{1T}} - K)du, \tag{6.4.5}$$

which has the required integration domain $[0,1)^2$.

It follows from (6.4.5) that the Monte Carlo or quasi-Monte Carlo integration estimate is given by

$$\hat{V}_0 = \frac{e^{-rT}}{N}\sum_{n=1}^{N}\hat{h}(u_{n1}, u_{n2}) \tag{6.4.6}$$

where

$$\hat{h}(u_{n1}, u_{n2}) = (1-d_2)(w_2 e^{c_{21}\Phi^{-1}(u_{n1})+c_{22}\Phi^{-1}(d_2+u_{n2}(1-d_2))+\mu_{2T}} - w_1 e^{c_{11}\Phi^{-1}(u_{n1})+\mu_{1T}} - K)$$

and $\boldsymbol{u}_n = (u_{n1}, u_{n2}) \in [0,1)^2$ denotes the n-th term of a random or low discrepancy

sequence. Alternatively, a crude estimate of the spread options can be obtained as

$$\tilde{V}_0 = \frac{e^{-rT}}{N} \sum_{n=1}^{N} \max[w_2 e^{c_{21}\Phi^{-1}(u_{n1})+c_{22}\Phi^{-1}(u_{n2})+\mu_2 T} - w_1 e^{c_{11}\Phi^{-1}(u_{n1})+\mu_1 T} - K, 0]. \tag{6.4.7}$$

Comparing these two approaches of estimating the spread options, we would expect estimate (6.4.6) to be more efficient for a given N. This estimate can be interpreted as a conditioning estimate; the component $w_2 e^{c_{21}\Phi^{-1}(u_{n1})+c_{22}\Phi^{-1}(d_2+u_{n2}(1-d_2))+\mu_2 T} - w_1 e^{c_{11}\Phi^{-1}(u_{n1})+\mu_1 T} - K$ denotes the payoff of the option conditional on being in-the-money while the adjustment factor $1-d_2$ denotes the probability of the option being in-the-money. Hence this eliminates the max function in (6.4.7). The additional smoothness can further be exploited by using lattice methods such as the g.l.p. methods. This requires an additional transformation so that the function of interest is periodic. Accordingly, estimate (6.4.6) is revised as

$$\bar{V}_0 = \frac{e^{-rT}}{N} \sum_{n=1}^{N} \hat{h}(\psi(u_{n1}), \psi(u_{n2}))\psi'(u_{n1})\psi'(u_{n2}) \tag{6.4.8}$$

where (u_{n1}, u_{n2}) is the g.l.p. and ψ is an appropriate transformation for periodizing the integrands. The function ψ can be one of the transformations discussed in Subsection 6.2.3.

We now demonstrate the efficiency of using the g.l.p. in evaluating (6.4.8). This is carried out in two parts as described below:

Part A

In the first part of the analysis, we consider a special case of spread options where $K = 0$ and $w_1 = w_2 = 1$. This becomes the exchange options and we can obtain an analytical expression for these options as

$$e^{-rT}\mathrm{E}\big[\max[S_{2T} - S_{1T}, 0]\big] = S_{20}e^{-\delta_2 T}\Phi(-D_2) - S_{10}e^{-\delta_1 T}\Phi(-D_1) \qquad (6.4.9)$$

where

$$\begin{aligned} D_1 &= \frac{\log(S_{20}/S_{10}) + (\delta_1 - \delta_2 - \sigma^2/2)T}{\sigma\sqrt{T}} \\ D_2 &= D_1 + \sigma\sqrt{T} \\ \sigma^2 &= \sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2. \end{aligned}$$

Note that the risk-free rate r does not play any role in determining the value of the exchange options. The analytic values of these options serve as benchmark against the results using the method of g.l.p.. As in the previous chapters, our assessment is based on the RMSE calculated over 50 randomly generated sets of option contracts.

The parameter values for the 50 option contracts are determined as follows: For the first asset, we set $S_{10} = 100$, $\sigma_1 = 30\%$, and $\delta_1 = 5\%$. Holding these parameter values fixed allows us to have a better control over the options being in-the-money, out-of-money or at-the money by manipulating the parameter values of the second asset. Hence the parameter values of the second asset are generated randomly according to the following rules: the second asset price S_{20} is uniformly distributed between 50 and 130, the volatility σ_2 is uniformly distributed between

10% and 50%, the dividend yield δ_2 is uniformly distributed between 1% and 10%. the correlation between asset 1 and 2, ρ_{12}, is uniformly distributed between -0.8 and 0.8 and the time until maturity T is uniformly distributed between 6 months to 5 years. The option price for the randomly selected option contract is again enforced to be at least 0.5.

The two-dimensional g.l.p. used in our computation is generated from Fibonacci numbers F_m. This point set has the advantage that it is the only known situation where Šargin's lower bound is attainable. One possible drawback of this point set is that it only admits specific values of N. For instance, in our examples we consider the Fibonacci numbers F_m with $m = 7, 8, \ldots, 16$. This implies that for $10 < N < 1000$, we can only compute the RMSEs at 10 possible values of N, namely $N = 13, 89, \ldots, 610, 987$. Nevertheless, we can always find an appropriate g.l.p. for arbitrary point set using the searching technique outlined in Subsection 6.2.2.

When using (6.4.8), we consider the following types of periodization techniques:

$$\psi(u) = \begin{cases} \text{Polynomial-Transformation with } m = 3, 4 \\ \sin^m\text{-Transformation with } m = 2, 3 \end{cases}$$

Furthermore, the same set of g.l.p. is also applied to (6.4.6), i.e. without any periodization. This allows to examine the impact of periodizing the integrands. Table 6.1 reports the RMSE in percentage for each method. Three conclusions can be drawn from these results. First, the method of g.l.p. yields extremely high precision, even for a small point set. For instance, with a mere 55 points, the largest RMSE across different periodization techniques is only 0.05% to as low as

0.017%. Second, the efficiency of the periodized integrands varies across different methods of transformations. Within the same class of transformation, it appears that higher level of transformation gives a much better result. For example, the

	RMSE in Percentage				
	Periodization Transformation				No Periodization
N	Polynomial-3	Polynomial-4	$\sin^2$	$\sin^3$	
13	2.76679	3.96266	1.78029	5.42768	25.65
21	0.70174	0.61241	0.39638	0.83536	16.77
34	0.29790	0.14219	0.16876	0.12909	13.00
55	0.05446	0.01800	0.02765	0.01666	8.27
89	0.02395	0.00505	0.01411	0.00458	6.24
144	0.00509	0.00109	0.00243	0.00067	3.90
233	0.00215	0.00010	0.00125	0.00006	2.91
377	0.00046	0.00004	0.00021	0.00003	1.80
610	0.00019	0.00001	0.00011	0.00001	1.33
987	0.00004	0.00000	0.00002	0.00000	0.81

Table 6.1: Comparisons of the RMSE (%) for the 50 Randomly Generated Exchange Option Contracts

polynomial-4 is superior to polynomial-3 while $\sin^3$ is superior to $\sin^2$. Third, it is desirable to take into account of the additional regularity inherent in the function. For instance without periodization, the resulting RMSE using $N = 55$ can be as huge as $8.27/0.01666 \approx 496$ times than the corresponding periodized estimate using $\sin^3$-transformation.

Part B

In this part of the comparison, we relax the constraint that $K = 0$. For a non-zero strike price, the comparison will be harder since there do not exist closed-form solution for the spread options. For our purpose, the option estimate obtained from

the g.l.p. method using $N = 121{,}393$ is taken to be the "true" option value. This "true" option value is justified given the success that we observed for the exchange options in Part A,

Rather than generating the option contracts randomly as in Part A, we use the same set of parameter values as in Pearson [81]. Pearson considers 144 spread call option contracts with the parameter values comprise of the following: $S_{10} = 92, 96, 100, 104$, $S_{20} = 100$, $\sigma_1 = 10\%, 20\%, 30\%$, $\sigma_2 = 20\%$, $T = 1$ week, 1 month, 1 year, 5 years, $\rho_{12} = -0.5, 0, 0.5$, $\delta_1 = \delta_2 = 5\%$, $r = 10\%$.

The key to Pearson's approximation algorithm is to recognize that the double integrals (6.4.1) with $w_1 = w_2 = 1$ can be decomposed to

$$\begin{aligned} e^{rT}V_0 &= \int_0^\infty \int_0^\infty \max[S_{2T} - S_{1T} - K, 0] g(S_{2T}|S_{1T}) f(S_{1T}) dS_{2T} dS_{1T} \\ &= \int_0^\infty \int_{S_{1T}+K}^\infty (S_{2T} - S_{1T} - K) g(S_{2T}|S_{1T}) f(S_{1T}) dS_{2T} dS_{1T} \\ &= \int_0^\infty F(S_{1T}) f(S_{1T}) dS_{1T} \end{aligned} \tag{6.4.10}$$

where $g(S_{2T}|S_{1T})$ is the conditional density of S_{2T} given S_{1T}, $f(S_{1T})$ is the marginal density of S_{1T} and

$$F(S_{1T}) = \int_{S_{1T}+K}^\infty (S_{2T} - S_{1T} - K) g(S_{2T}|S_{1T}) dS_{2T}.$$

The function $F(S_{1T})$ can be evaluated analytically so that the pricing of spread options is now reduced to the one-dimensional integration problem of the form (6.4.10). Unfortunately, $F(S_{1T})$ is a complicated function involving the cumulative normal density function. This implies that the integral (6.4.10) would still need to

be evaluated numerically. The method proposed in Pearson [81] is to approximate $F(S_{1T})$ by piecewise linear functions. Pearson shows that using $N = 100$ (which corresponds to approximating F with $2N = 200$ piecewise linear functions, using Pearson's terminology), the approximation yields a much higher accuracy compared to the approach suggested by Wilcox [106].

We will demonstrate that the method of g.l.p. is superior to Pearson's approximation algorithm. Based on 144 option contracts described earlier, Tables 6.2 and 6.3 depict the percentage relative pricing errors for Pearson's approximation and g.l.p. method with $\sin^3$-transformation. In both methods, we have assumed $N = 55$.[2] Although the function F is only approximated by 110 piecewise linear functions, Pearson's method appears to be reasonable. Pearson's method is most accurate in situations with positive correlation and low volatility of the first underlying asset relative to the second asset. The approximation starts to deteriorate as the volatility of the first underlying asset is greater than the volatility of the second asset, particularly with zero or negative correlation and deep out-of-money option contracts. In general, most of relative errors are less than 5%. On the other hand, the result from the method of g.l.p. is amazingly accurate, with most of the relative pricing errors within 0.05%! Even if we increase the number of piecewise linear functions in Pearson's algorithm to 10,000, i.e. $N = 5{,}000$, (see Table 6.4), the method of g.l.p. with $N = 55$ is still superior.

We now compute the RMSE using the same values of N as in Part A. The option contracts are still based on Pearson's examples except that we ignore those spread

[2]Note that the complexities of these two methods are proportional to N, however, they are not necessary comparable to each other for a given N.

call options with values less than 0.5. This eliminates most of the deep out-of-money option contracts. For the g.l.p. method, we consider the estimates with and without periodization. We also use the same sets of periodization transformation. Figure 6.1 plots log N against the computed log(RMSE). The graph indicates that Pearson's method is more efficient than the Monte Carlo integration that does not take into account the additional smoothness. However, once such additional regularity is factored in, the g.l.p. method yields a more superior convergence rate. Similar to Part A, the transformations based on polynomial-4 and $\sin^3$ are the most efficient.

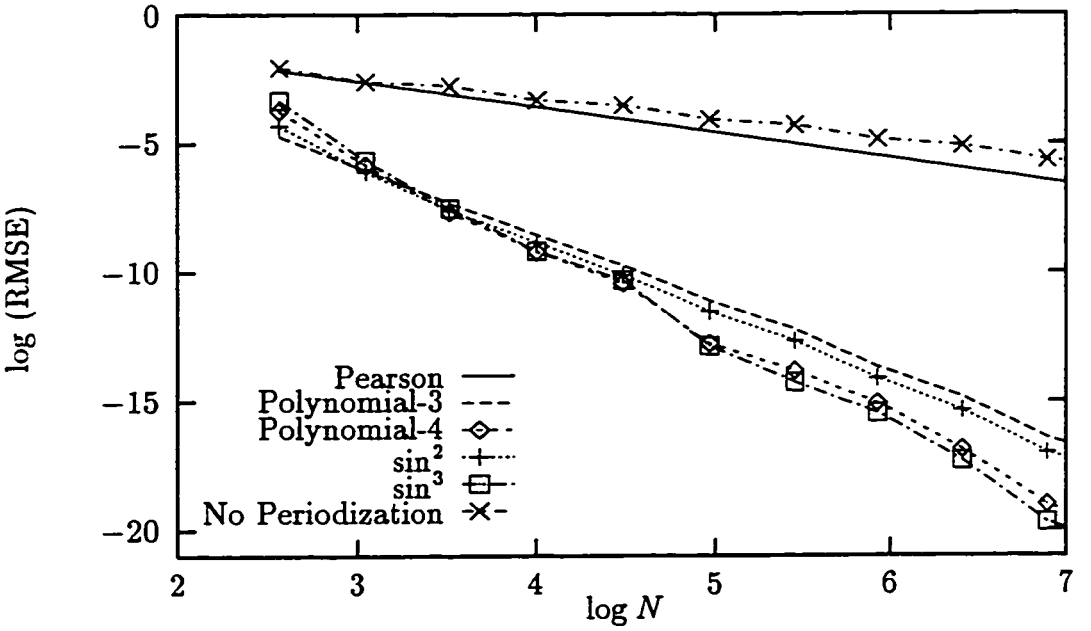


Figure 6.1: Comparisons of the RMSEs using g.l.p.and Pearson's Method

Before concluding this section, we should remark that our program implementing Pearson's algorithm does not reproduce exactly the results reported in Pearson [81]. The results in Exhibit 2 of Pearson [81] claim to have been obtained using

Volatility of First Underlying	Price of First Underlying	Correlation = -0.5				Correlation = 0				Correlation = 0.5			
		Time until Maturity				Time until Maturity				Time until Maturity			
		0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00
0.1	92	0.42	1.08	2.07	2.82	0.09	0.29	0.55	0.62	0.00	0.00	0.00	0.00
	96	2.16	2.21	2.48	3.03	0.71	0.70	0.70	0.69	0.00	0.00	0.00	0.00
	100	6.60	3.97	2.93	3.23	2.57	1.42	0.87	0.76	0.01	0.00	0.00	0.00
	104	14.79	6.45	3.42	3.44	6.13	2.47	1.06	0.84	0.02	0.01	0.00	0.00
0.2	92	0.85	1.68	2.79	3.64	0.38	0.89	1.47	1.58	0.06	0.24	0.39	0.11
	96	2.83	2.89	3.20	3.84	1.80	1.80	1.78	1.72	0.79	0.76	0.56	0.18
	100	6.92	4.58	3.65	4.04	5.27	3.18	2.12	1.87	3.40	1.71	0.77	0.26
	104	13.73	6.80	4.12	4.24	11.49	5.09	2.49	2.02	8.67	3.19	1.01	0.34
0.3	92	1.24	2.11	3.20	4.08	0.77	1.47	2.14	2.20	0.33	0.86	1.25	0.78
	96	3.18	3.24	3.58	4.25	2.53	2.52	2.48	2.36	1.98	1.92	1.60	0.93
	100	6.61	4.71	3.97	4.43	6.07	3.99	2.86	2.52	6.33	3.60	1.98	1.08
	104	11.89	6.54	4.38	4.60	11.90	5.89	3.25	2.67	14.45	6.01	2.41	1.24

Table 6.2: Relative Errors (%) of the Spread Call Option Prices Using Pearson's Algorithm with $N = 55$

Volatility of First Underlying	Price of First Underlying	Correlation = -0.5				Correlation = 0				Correlation = 0.5			
		Time until Maturity				Time until Maturity				Time until Maturity			
		0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00
0.1	92	-0.01	0.01	-0.01	-0.01	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
	96	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
	100	0.02	0.01	-0.01	-0.01	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
	104	0.03	0.02	-0.00	-0.01	-0.02	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
0.2	92	-0.01	-0.00	0.02	0.01	-0.00	0.00	-0.01	-0.01	0.00	0.00	0.00	0.00
	96	0.02	0.02	0.02	0.00	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00	0.00
	100	-0.00	-0.02	0.01	0.00	0.01	0.01	-0.00	-0.01	0.00	0.00	0.00	0.00
	104	0.03	-0.01	-0.01	-0.00	0.01	0.01	-0.00	-0.01	-0.06	0.00	0.00	0.00
0.3	92	-0.01	-0.03	0.00	-0.01	-0.02	0.01	0.01	0.00	-0.00	0.01	-0.00	-0.00
	96	0.03	0.02	0.01	-0.00	0.01	0.01	0.00	-0.00	-0.01	-0.01	-0.01	-0.00
	100	-0.03	0.03	0.02	0.00	0.01	-0.02	-0.00	-0.00	0.02	0.00	-0.01	-0.00
	104	0.04	-0.03	0.03	0.01	0.03	0.01	-0.01	-0.01	0.02	0.02	-0.01	-0.01

Table 6.3: Relative Errors (%) of the Spread Call Option Prices Using g.l.p. Method with $N = 55$ and $\sin^3$-Transformation

Volatility of First Underlying	Price of First Underlying	Correlation = -0.5 Time until Maturity				Correlation = 0 Time until Maturity				Correlation = 0.5 Time until Maturity			
		0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00
0.1	92	0.00	0.01	0.02	0.03	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
	96	0.02	0.02	0.03	0.03	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00
	100	0.07	0.04	0.03	0.04	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00
	104	0.16	0.07	0.04	0.04	0.07	0.03	0.01	0.01	0.00	0.00	0.00	0.00
0.2	92	0.01	0.02	0.03	0.04	0.00	0.01	0.02	0.02	0.00	0.00	0.00	0.00
	96	0.03	0.03	0.04	0.04	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.00
	100	0.08	0.05	0.04	0.04	0.06	0.04	0.02	0.02	0.04	0.02	0.01	0.00
	104	0.15	0.08	0.05	0.05	0.12	0.06	0.03	0.02	0.09	0.04	0.01	0.00
0.3	92	0.01	0.02	0.04	0.05	0.01	0.02	0.02	0.02	0.00	0.01	0.01	0.01
	96	0.04	0.04	0.04	0.05	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.01
	100	0.07	0.05	0.04	0.05	0.07	0.04	0.03	0.03	0.07	0.04	0.02	0.01
	104	0.13	0.07	0.05	0.05	0.13	0.07	0.04	0.03	0.15	0.07	0.03	0.01

Table 6.4: Relative Errors (%) of the Spread Call Option Prices Using Pearson's Algorithm with $N = 5{,}000$

$N = 100$. In our program, we need to increase N as large as 5,000 before getting virtually the same results (up to 2 decimal places). Tables 6.5 and 6.6 show a sample of the option estimates for $N = 100$ and $N = 5{,}000$ using our implemented version of Pearson's method.

6.4.2 Generalized Rainbow Options

So far we have only applied the g.l.p. to two-dimensional numerical problems. In this subsection, we evaluate its efficiency in higher dimensions. We consider the generalized rainbow options where the payoff function can be written as

$$g(\boldsymbol{S}_T) = \max[w_0 + w_1 S_{1T} + w_2 S_{2T} + \cdots + w_s S_{sT}, 0]$$

where the constants w_i, $0 \leq i \leq s$ are either positive or negative. When $w_0 = -K$, $w_1 = -1$ and $w_2 = 1$, the generalized rainbow options become the spread call options.

In our numerical examples, we consider a special case of the generalized rainbow options where $w_0 = -K$, $w_i = -1/(s-1)$ for $1 \leq i \leq s-1$ and $w_s = 1$. This particular option is similar to outperformance option which results in a positive payoff only if the designated asset outperforms the average of a portfolio of assets after adjusted for the strike price. We consider $s = 5$ and use parameter values: $S_{i0} = 100$, $\sigma_i = 20\%$, $i = 1, \ldots, 5$, $T =$ 1 year, and $K = 4$. Since these options do not have an analytic solution, an exhaustive Monte Carlo simulation was used to determine the appropriate values of the options. With 20 million simulation runs, the option value is 4.431856 with standard error 0.00058. This value is used as a

Volatility of First Underlying	Price of First Underlying	Correlation = -0.5				Correlation = 0				Correlation = 0.5			
		Time until Maturity				Time until Maturity				Time until Maturity			
		0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00
0.1	92	4.26	5.39	11.95	19.78	4.14	4.99	10.40	16.94	4.05	4.56	8.62	13.67
	96	1.49	3.04	10.12	18.53	1.26	2.56	8.52	15.62	0.98	1.99	6.65	12.26
	100	0.29	1.50	8.52	17.36	0.16	1.08	6.91	14.40	0.06	0.63	5.04	10.97
	104	0.03	0.65	7.13	16.26	0.01	0.38	5.55	13.27	0.00	0.14	3.74	9.81
0.2	92	4.54	6.19	14.87	24.98	4.30	5.53	12.49	20.69	4.09	4.76	9.46	15.23
	96	1.95	3.95	13.14	23.85	1.58	3.21	10.68	19.47	1.11	2.26	7.55	13.87
	100	0.59	2.35	11.58	22.78	0.34	1.66	9.09	18.33	0.10	0.84	5.93	12.63
	104	0.12	1.30	10.18	21.77	0.04	0.76	7.70	17.26	0.00	0.24	4.61	11.48
0.3	92	4.90	7.11	18.06	30.44	4.58	6.30	15.23	25.52	4.24	5.32	11.68	19.24
	96	2.44	4.96	16.41	29.44	2.01	4.08	13.51	24.41	1.46	2.96	9.86	18.00
	100	0.98	3.31	14.90	28.49	0.64	2.47	11.96	23.37	0.27	1.44	8.26	16.83
	104	0.32	2.12	13.53	27.58	0.14	1.39	10.57	22.38	0.02	0.61	6.89	15.75

Table 6.5: Spread Call Option Prices Using Pearson's Algorithm with $N = 100$

Volatility of First Underlying	Price of First Underlying	Correlation = -0.5				Correlation = 0				Correlation = 0.5			
		Time until Maturity				Time until Maturity				Time until Maturity			
		0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00	0.02	0.083	1.00	5.00
0.1	92	4.25	5.36	11.82	19.48	4.14	4.98	10.37	16.88	4.05	4.56	8.62	13.67
	96	1.48	3.00	9.99	18.23	1.25	2.55	8.49	15.56	0.98	1.99	6.65	12.26
	100	0.28	1.47	8.39	17.06	0.16	1.08	6.88	14.34	0.06	0.63	5.04	10.97
	104	0.03	0.63	7.01	15.96	0.01	0.37	5.52	13.21	0.00	0.14	3.74	9.81
0.2	92	4.52	6.13	14.65	24.49	4.29	5.51	12.39	20.52	4.08	4.75	9.44	15.22
	96	1.92	3.89	12.92	23.36	1.56	3.18	10.58	19.29	1.11	2.25	7.52	13.86
	100	0.57	2.29	11.36	22.30	0.33	1.63	8.99	18.15	0.10	0.83	5.91	12.61
	104	0.12	1.25	9.96	21.29	0.04	0.74	7.60	17.07	0.00	0.24	4.58	11.46
0.3	92	4.87	7.03	17.75	29.78	4.56	6.25	15.06	25.21	4.23	5.30	11.61	19.16
	96	2.40	4.87	16.10	28.78	1.98	4.02	13.34	24.10	1.44	2.93	9.77	17.90
	100	0.95	3.23	14.59	27.82	0.62	2.41	11.78	23.05	0.26	1.41	8.18	16.73
	104	0.30	2.05	13.21	26.91	0.14	1.35	10.39	22.06	0.02	0.59	6.80	15.64

Table 6.6: Spread Call Option Prices Using Pearson's Algorithm with $N = 5{,}000$

benchmark to compare with the results from the lattice points.

We use the published table for the 5-dimensional good lattice points. We consider 9 different values of N, which are 1069, 1543, 2129, 3001, 4001, 5003, 6007, 8191 and 10007. The corresponding optimal coefficients for the g.l.p. can be obtained from Hua and Wang [41]. Figure 6.2 summarizes the results. The 95% confidence limits from Monte Carlo methods are provided for benchmarking. We

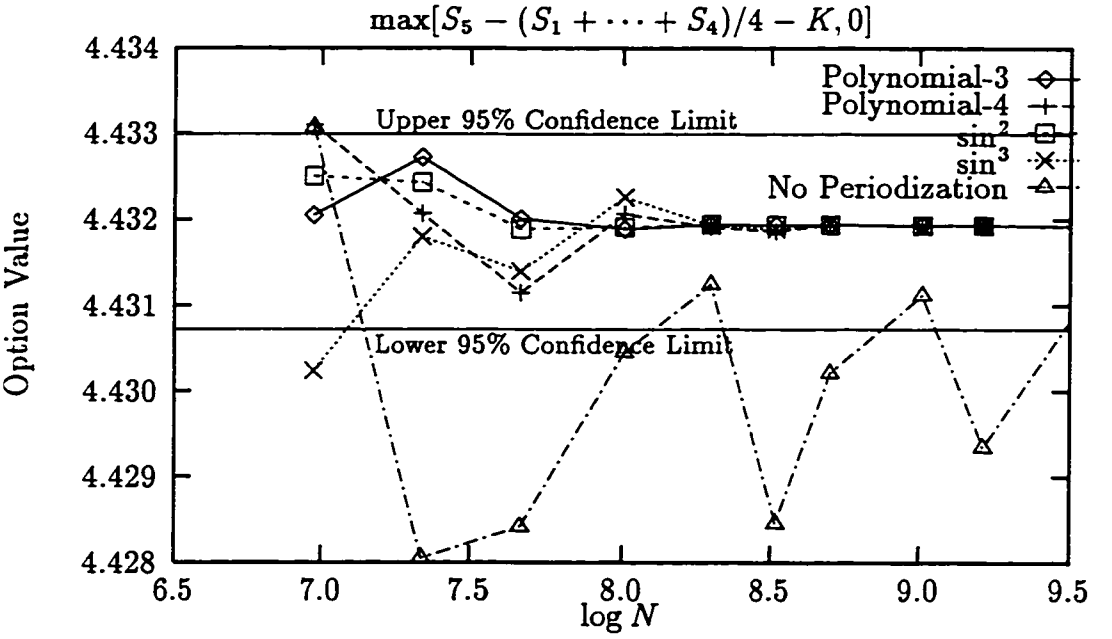


Figure 6.2: Comparisons of the RMSEs using g.l.p. with and without Periodization

observe that even with $s = 5$, the g.l.p. method still yields a very high degree of precision. All four transformations agree up to 4 decimal places (converging to 4.4319) for $N > 4000$. Straightforward application of the g.l.p. without periodizing the integrand results in a much higher error.

6.4.3 Path-Dependent Options: Lookback and Asian

In this subsection, we consider path-dependent options with values contingent on the realized asset prices at discretely sampled time points. The history of the asset prices becomes important in valuing this type of option. This is in contrast to the options considered in the last two subsections where the values of the options only depend on the terminal asset prices. For our numerical illustrations, we consider the path-dependent options that have the following payoff structures:

$$(a) \qquad g(\boldsymbol{S}) = \max\left[\max[S_0, S_1, \ldots, S_s] - K, 0\right]$$

$$(b) \qquad g(\boldsymbol{S}) = \max\left[K - \min[S_0, S_1, \ldots . S_s], 0\right]$$

$$(c) \qquad g(\boldsymbol{S}) = \max\left[\frac{\sum_{i=0}^{s} S_i}{s+1} - K, 0\right]$$

where S_i denotes the asset price at the i-th discretely sampled time points. Options of type (a) are known as fixed-strike lookback call options or high water mark options, options of type (b) are known as fixed-strike lookback put options, while options of type (c) are known as Asian options with arithmetic averaging. The high water mark options and Asian options are often embedded in equity-index annuities sold by insurance companies. The prices of discrete lookback options are expressed in terms of multivariate normal probabilities (see Heynen and Kat [36]) while the prices of Asian options are expressed in terms of the Bessel process (see German and Yor [28]). These are often too complicated to evaluate directly and hence simulation is a common pricing tool in these cases.

In this subsection, we demonstrate that the method of g.l.p. can be an efficient pricing tool for low-dimensional path-dependent options. For our numerical comparisons, we consider the following sets of parameter values:

$$\begin{aligned} S_0 &= 100 \\ r &= 10\% \\ \sigma &= 20\%,\ 30\%,\ 40\%. \end{aligned}$$

For the lookback options, we assume the times until maturity are 2 and 5 years with strike prices $K = 70, 100, 130$. For the Asian options, we assume the time until maturity is 7 years with strike prices $K = 90, 100, 110$. We further assume that the asset prices are sampled annually so that the time until maturity (in years) represents the dimensionality of the problem. We have deliberately chosen options that have long dated maturities and a small number of monitoring time points. These are typical features of the options embedded in equity-indexed annuities. For the 2-year options, we consider g.l.p. with $N = 64$, 256, 512. For the 5-year options, we take $N = 256$, 512, 4096 and for the 7-year options we assume $N = 512, 4096, 8192$. The optimal coefficients for these g.l.p. are determined using the searching algorithm outlined in Subsection 6.2.2. Haber [31] also published tables of low-dimensional g.l.p., where N is a multipler of 2 or 3, from a pool of randomly selected g.l.p., rather searching through the entire choices of the one-parameter g.l.p.. Therefore, Haber's g.l.p. is only a sub-optimal choice, whereas ours is optimal.

For each option contract and each point set, we generate 10 independent option estimates using the technique described in Subsection 6.2.4. An estimate of the

standard error of the option estimate based on these independent replications is computed. The resulting standard error is then compared to that obtained from the Monte Carlo methods using the same number of points. Tables 6.7, 6.8, 6.9, 6.10 and 6.11 summarize the results for these path-dependent options. For the g.l.p., we report 3 sets of results. Two of these sets use periodization techniques with polynomial-3 and $\sin^1$ transformations while the third set of results does not involve any periodization.

For the 2-year lookback options, we observe that a significant reduction of standard errors is achieved even for a point set as low as 640. This is consistent with the results reported in Subsection 6.4.1 which used Fibonacci numbers in constructing the 2-dimensional g.l.p.. For the 5-year lookback options and 7-year Asian options, we still observe a substantial improvement over standard Monte Carlo methods, although the magnitude of improvement is not as favourable as the 2-dimensional case. A much larger point set is required for 5-dimensional and 7-dimensional examples in order to achieve the same level of accuracy as the 2-dimensional cases. For instance, let consider the option contract with $\sigma = 30\%$ and $K = 100$. In this case, the efficiency ratio for the 2-year high water mark option with 5120 points and $\sin^1$ transformation is 1115. On the other hand, by merely increasing the dimensions from 2 to 5, the efficiency ratio with the same number of points diminishes to 17.4. This represents a significant decline of $1115/17.4 \approx 64$ times in efficiency. Another conclusion can be drawn from these experiments is that periodization, in general, speeds up the convergence of the g.l.p. methods.

It should be pointed out that in our comparisons, we have used crude Monte

Carlo method without any variance reduction techniques. In the case of Asian options, a more efficient simulation technique is to use the corresponding Asian option with geometric averaging as control variates. This leads to a substantial reduction in the magnitude of the standard errors. Such efficiency is not exploited in our comparisons since if control variate works well for the Monte Carlo methods, it also works well for g.l.p. methods. Hence, the efficiency ratio would likely remain the same when both methods incorporated the control variates.

6.5 Conclusion

In this chapter, we show that good lattice points can be used to exploit the additional smoothness exhibited in low-dimensional financial instruments. This method compares favorably to other numerical techniques. Its superiority is documented in the numerical examples based on spread options, a particular case of the generalized rainbow options and path-dependent options including high water mark options, lookback put options and Asian options. Evidently, the method of g.l.p. is not confined to these applications. A similar approach can be used to compute the hedging parameters such as the delta or gamma. We would expect the same level of efficiency in these situations.

σ	K	Random	Std. error	Good Lattice Points[a] Periodization polynomial-3	Periodization $\sin^1$	No Periodization
Total points $= 10 \times N = 640$						
0.2	70	49.09	0.96	48.30 (62.8)	48.30 (70.9)	47.94 (3.3)
	100	24.53	0.96	23.73 (78.4)	23.74 (89.4)	23.38 (3.3)
	130	9.24	0.71	9.15 (45.4)	9.15 (53.4)	8.85 (2.7)
0.3	70	54.95	1.50	54.06 (54.5)	54.06 (63.1)	53.49 (3.0)
	100	30.39	1.50	29.50 (61.4)	29.50 (71.5)	28.93 (3.0)
	130	15.99	1.24	15.75 (44.4)	15.75 (44.4)	15.24 (2.6)
0.4	70	61.27	2.16	60.24 (49.2)	60.25 (60.0)	59.44 (2.8)
	100	36.71	2.16	35.68 (51.9)	35.69 (64.0)	34.88 (2.8)
	130	23.11	1.91	22.63 (43.2)	22.64 (53.1)	21.89 (2.5)
Total points $= 10 \times N = 2560$						
0.2	70	48.27	0.48	48.31 (292.6)	48.31 (363.0)	48.28 (7.0)
	100	23.71	0.48	23.75 (341.7)	23.75 (423.1)	23.72 (7.0)
	130	9.11	0.35	9.16 (265.1)	9.16 (350.4)	9.12 (5.5)
0.3	70	54.05	0.74	54.09 (281.3)	54.09 (365.9)	54.02 (5.8)
	100	29.49	0.74	29.53 (311.3)	29.53 (407.3)	29.45 (5.8)
	130	15.72	0.60	15.77 (262.6)	15.77 (262.6)	15.68 (4.9)
0.4	70	60.25	1.04	60.28 (236.9)	60.28 (306.5)	60.11 (4.8)
	100	35.68	1.04	35.72 (250.3)	35.72 (323.8)	35.55 (4.8)
	130	22.63	0.91	22.67 (205.3)	22.67 (265.7)	22.49 (4.3)
Total points $= 10 \times N = 5120$						
0.2	70	48.57	0.34	48.31 (1029.9)	48.31 (1343.7)	48.28 (8.0)
	100	24.01	0.34	23.75 (1154.3)	23.75 (1542.6)	23.72 (8.0)
	130	9.29	0.25	9.16 (780.1)	9.16 (729.9)	9.14 (6.0)
0.3	70	54.55	0.53	54.09 (992.9)	54.09 (1033.9)	54.04 (6.6)
	100	29.99	0.53	29.53 (1082.8)	29.53 (1115.0)	29.48 (6.6)
	130	16.06	0.43	15.77 (859.7)	15.77 (859.7)	15.73 (5.5)
0.4	70	60.95	0.74	60.27 (977.3)	60.28 (1250.4)	60.20 (5.5)
	100	36.39	0.74	35.71 (1021.7)	35.71 (1294.0)	35.64 (5.5)
	130	23.15	0.65	22.66 (812.3)	22.66 (864.0)	22.59 (4.8)

[a]The value in bracket is the ratio of the standard errors from Monte Carlo method to g.l.p. methods.

Table 6.7: 2-Year High Water Mark Options with Annual Monitorings

σ	K	Random	Std. error	Good Lattice Points[a]		
				Periodization		No Periodization
				polynomial-3	$\sin^1$	
Total points $= 10 \times N = 640$						
0.2	70	0.28	0.06	0.23 (14.4)	0.22 (17.4)	0.23 (1.4)
	100	4.68	0.33	5.13 (40.4)	5.13 (45.7)	5.25 (4.0)
	130	29.24	0.33	29.69 (32.5)	29.69 (38.2)	29.81 (4.0)
0.3	70	1.62	0.20	1.65 (42.8)	1.65 (49.8)	1.71 (2.7)
	100	9.20	0.52	10.21 (81.8)	10.21 (76.5)	10.39 (4.4)
	130	33.77	0.52	34.77 (62.2)	34.77 (62.2)	34.96 (4.4)
0.4	70	3.85	0.33	4.23 (36.2)	4.23 (48.7)	4.34 (3.7)
	100	13.90	0.68	15.39 (91.0)	15.38 (83.7)	15.63 (5.0)
	130	38.46	0.68	39.95 (84.3)	39.95 (85.4)	40.20 (5.0)
Total points $= 10 \times N = 2560$						
0.2	70	0.31	0.03	0.23 (41.8)	0.23 (45.4)	0.21 (3.7)
	100	5.24	0.17	5.12 (202.0)	5.12 (194.2)	5.10 (8.4)
	130	29.80	0.17	29.68 (172.1)	29.68 (167.7)	29.66 (8.4)
0.3	70	1.81	0.10	1.66 (103.3)	1.65 (110.3)	1.63 (6.3)
	100	10.26	0.27	10.20 (333.2)	10.20 (302.1)	10.17 (10.0)
	130	34.82	0.27	34.76 (258.6)	34.76 (258.6)	34.73 (10.0)
0.4	70	4.36	0.17	4.22 (200.8)	4.22 (182.2)	4.20 (8.3)
	100	15.36	0.35	15.39 (344.0)	15.39 (331.9)	15.37 (11.4)
	130	39.92	0.35	39.96 (280.9)	39.96 (292.6)	39.93 (11.4)
Total points $= 10 \times N = 5120$						
0.2	70	0.26	0.02	0.23 (60.7)	0.23 (88.1)	0.23 (4.0)
	100	5.14	0.12	5.12 (430.9)	5.12 (472.8)	5.13 (9.4)
	130	29.70	0.12	29.68 (380.4)	29.68 (415.6)	29.69 (9.4)
0.3	70	1.70	0.07	1.65 (154.1)	1.65 (172.8)	1.66 (6.4)
	100	10.17	0.19	10.20 (631.4)	10.20 (677.1)	10.21 (12.4)
	130	34.73	0.19	34.76 (591.4)	34.76 (591.4)	34.77 (12.4)
0.4	70	4.23	0.12	4.22 (515.2)	4.22 (354.7)	4.23 (8.9)
	100	15.28	0.24	15.39 (711.8)	15.39 (636.0)	15.41 (13.7)
	130	39.85	0.24	39.95 (615.2)	39.95 (581.6)	39.97 (13.7)

[a]The value in bracket is the ratio of the standard errors from Monte Carlo method to g.l.p. methods.

Table 6.8: 2-Year Fixed-Strike Lookback Put Options with Annual Monitorings

σ	K	Random	Std. error	Good Lattice Points[a] Periodization polynomial-3	Periodization $\sin^1$	No Periodization
Total points $= 10 \times N = 2560$						
0.2	70	66.79	0.89	65.70 (8.0)	65.70 (8.7)	65.67 (4.5)
	100	48.60	0.89	47.49 (9.0)	47.49 (9.7)	47.48 (4.5)
	130	33.46	0.84	32.38 (9.5)	32.39 (9.7)	32.39 (4.1)
0.3	70	77.65	1.46	75.50 (9.6)	75.50 (9.9)	75.87 (3.1)
	100	59.45	1.46	57.29 (10.4)	57.28 (10.8)	57.68 (3.1)
	130	45.62	1.40	43.64 (11.0)	43.65 (11.1)	43.91 (3.0)
0.4	70	90.19	2.24	86.85 (12.4)	86.87 (12.5)	87.78 (2.4)
	100	71.99	2.24	68.64 (13.3)	68.66 (13.5)	69.58 (2.4)
	130	59.13	2.17	56.00 (14.0)	56.03 (14.3)	56.85 (2.3)
Total points $= 10 \times N = 5120$						
0.2	70	65.92	0.61	65.53 (16.2)	65.54 (16.1)	65.82 (4.7)
	100	47.72	0.61	47.33 (18.5)	47.34 (18.3)	47.62 (4.7)
	130	32.62	0.58	32.24 (14.4)	32.24 (14.5)	32.56 (3.9)
0.3	70	76.03	1.00	75.43 (16.3)	75.43 (16.0)	76.11 (3.5)
	100	57.84	1.00	57.22 (17.8)	57.23 (17.4)	57.91 (3.5)
	130	44.06	0.95	43.48 (17.5)	43.49 (17.3)	44.19 (3.2)
0.4	70	87.60	1.51	86.83 (24.7)	86.86 (23.8)	88.05 (3.0)
	100	69.40	1.51	68.63 (26.9)	68.65 (26.0)	69.85 (3.0)
	130	56.63	1.46	55.89 (25.1)	55.92 (25.3)	57.16 (2.9)
Total points $= 10 \times N = 40960$						
0.2	70	65.69	0.22	65.51 (34.1)	65.51 (43.8)	65.48 (8.1)
	100	47.49	0.22	47.31 (42.6)	47.31 (54.1)	47.28 (8.1)
	130	32.40	0.20	32.25 (55.0)	32.25 (69.5)	32.22 (7.1)
0.3	70	75.75	0.35	75.45 (36.0)	75.45 (41.6)	75.36 (5.0)
	100	57.56	0.35	57.26 (41.4)	57.26 (46.7)	57.16 (5.0)
	130	43.80	0.34	43.55 (42.1)	43.55 (47.4)	43.45 (4.8)
0.4	70	87.32	0.54	86.87 (47.9)	86.87 (57.5)	86.71 (3.4)
	100	69.12	0.54	68.67 (53.0)	68.67 (63.0)	68.52 (3.4)
	130	56.39	0.53	55.97 (71.6)	55.97 (88.1)	55.81 (3.3)

[a]The value in bracket is the ratio of the standard errors from Monte Carlo method to g.l.p. methods.

Table 6.9: 5-Year High Water Mark Options with Annual Monitorings

σ	K	Random	Std. error	Good Lattice Points[a]		
				Periodization		No Periodization
				polynomial-3	$\sin^1$	
Total points $= 10 \times N = 2560$						
0.2	70	0.60	0.05	0.56 (6.2)	0.56 (4.3)	0.57 (3.4)
	100	5.47	0.16	5.54 (9.7)	5.53 (9.4)	5.60 (6.0)
	130	23.67	0.16	23.75 (5.8)	23.74 (5.7)	23.80 (6.0)
0.3	70	3.25	0.13	3.25 (7.5)	3.24 (8.1)	3.29 (5.3)
	100	11.77	0.26	11.96 (6.3)	11.97 (6.0)	11.94 (6.3)
	130	29.97	0.26	30.17 (5.3)	30.18 (5.2)	30.14 (6.3)
0.4	70	7.23	0.20	7.30 (11.7)	7.31 (9.9)	7.31 (4.7)
	100	18.14	0.33	18.41 (9.3)	18.41 (8.8)	18.43 (6.7)
	130	36.34	0.33	36.62 (7.2)	36.62 (7.3)	36.63 (6.7)
Total points $= 10 \times N = 5120$						
0.2	70	0.59	0.03	0.58 (7.8)	0.58 (8.3)	0.59 (2.4)
	100	5.53	0.11	5.58 (13.7)	5.58 (13.2)	5.59 (4.2)
	130	23.73	0.11	23.78 (8.1)	23.78 (7.8)	23.78 (4.2)
0.3	70	3.25	0.09	3.26 (11.6)	3.25 (12.8)	3.26 (3.7)
	100	11.87	0.18	11.96 (12.1)	11.96 (10.7)	11.94 (9.4)
	130	30.07	0.18	30.16 (8.8)	30.16 (8.3)	30.13 (9.4)
0.4	70	7.25	0.14	7.32 (10.8)	7.32 (9.5)	7.31 (6.2)
	100	18.32	0.23	18.42 (7.6)	18.41 (7.1)	18.38 (7.3)
	130	36.52	0.23	36.62 (6.7)	36.62 (6.3)	36.58 (7.3)
Total points $= 10 \times N = 40960$						
0.2	70	0.58	0.01	0.56 (15.2)	0.56 (14.6)	0.57 (2.7)
	100	5.59	0.04	5.57 (19.1)	5.57 (19.0)	5.57 (5.6)
	130	23.79	0.04	23.77 (12.0)	23.77 (13.8)	23.76 (5.6)
0.3	70	3.29	0.03	3.26 (15.6)	3.26 (15.7)	3.25 (3.7)
	100	11.96	0.06	11.95 (23.5)	11.95 (29.3)	11.94 (7.5)
	130	30.16	0.06	30.15 (17.6)	30.15 (23.4)	30.14 (7.5)
0.4	70	7.34	0.05	7.32 (25.0)	7.32 (27.2)	7.31 (7.9)
	100	18.40	0.08	18.42 (36.7)	18.42 (46.1)	18.42 (10.0)
	130	36.59	0.08	36.62 (24.3)	36.62 (32.8)	36.61 (10.0)

[a]The value in bracket is the ratio of the standard errors from Monte Carlo method to g.l.p. methods.

Table 6.10: 5-Year Fixed-Strike Lookback Put Options with Annual Monitorings

σ	K	Random	Std. error	Good Lattice Points[a] Periodization polynomial-3	Periodization $\sin^{-1}$	No Periodization
Total points $= 10 \times N = 5120$						
0.2	90	28.53	0.35	28.08 (6.1)	28.08 (5.8)	28.06 (4.1)
	100	24.15	0.34	23.71 (7.6)	23.71 (7.1)	23.70 (3.7)
	110	20.14	0.32	19.76 (7.8)	19.77 (7.4)	19.77 (3.5)
0.3	90	30.25	0.55	29.63 (8.2)	29.64 (7.6)	29.65 (2.6)
	100	26.58	0.53	26.02 (8.0)	26.04 (7.6)	26.03 (2.5)
	110	23.29	0.52	22.81 (7.1)	22.83 (6.6)	22.80 (2.4)
0.4	90	32.72	0.81	31.89 (8.2)	31.91 (7.9)	31.95 (2.0)
	100	29.60	0.80	28.85 (7.5)	28.88 (7.1)	28.90 (1.9)
	110	26.79	0.79	26.09 (7.0)	26.12 (6.6)	26.18 (1.8)
Total points $= 10 \times N = 40960$						
0.2	90	28.30	0.12	28.08 (11.6)	28.08 (11.2)	28.09 (4.5)
	100	23.94	0.12	23.72 (11.5)	23.72 (11.0)	23.73 (4.1)
	110	19.98	0.11	19.77 (11.1)	19.77 (9.2)	19.77 (3.7)
0.3	90	29.97	0.19	29.63 (15.7)	29.63 (13.6)	29.63 (3.1)
	100	26.32	0.19	26.00 (15.3)	26.00 (12.5)	25.99 (2.9)
	110	23.06	0.18	22.76 (17.6)	22.76 (13.7)	22.75 (2.8)
0.4	90	32.31	0.28	31.82 (20.3)	31.83 (17.9)	31.81 (2.5)
	100	29.22	0.28	28.75 (17.8)	28.75 (14.3)	28.75 (2.4)
	110	26.46	0.27	26.01 (14.5)	26.01 (10.8)	26.02 (2.3)
Total points $= 10 \times N = 81920$						
0.2	90	28.22	0.09	28.10 (23.8)	28.10 (26.2)	28.08 (11.0)
	100	23.86	0.08	23.74 (28.4)	23.74 (30.5)	23.72 (8.2)
	110	19.90	0.08	19.78 (29.0)	19.78 (30.6)	19.76 (6.6)
0.3	90	29.86	0.13	29.65 (26.0)	29.65 (29.7)	29.60 (6.3)
	100	26.22	0.13	26.02 (25.5)	26.02 (28.6)	25.97 (5.8)
	110	22.97	0.13	22.78 (18.5)	22.78 (19.8)	22.73 (5.6)
0.4	90	32.18	0.20	31.86 (19.7)	31.86 (20.2)	31.77 (5.1)
	100	29.10	0.19	28.80 (16.7)	28.80 (16.5)	28.71 (5.1)
	110	26.35	0.19	26.06 (15.4)	26.06 (15.0)	25.98 (4.9)

[a]The value in bracket is the ratio of the standard errors from Monte Carlo method to g.l.p. methods.

Table 6.11: 7-Year Asian Call Options with Annual Fixings

Chapter 7

Efficient Techniques for Simulating Through Trees

The aim of this chapter is to show how low discrepancy sequences can be used to improve convergence when simulating path-dependent derivative securities. We consider binomial (or multinomial) trees where payoffs of the derivative securities depend on the path taken through the tree. Even with a recombining tree, the number of distinct paths grows exponentially with the number of time steps. It is therefore impossible, in term of the computational time, to evaluate all the outcomes when the number of time steps is very large, say greater than 25. One way to handle this situation is to use Monte Carlo simulation. In this chapter, we discuss how to use low discrepancy sequences to deal with these types of problems. The conventional low discrepancy sequences, however, may not be as efficient as the random sequences when the dimension is large. In this case, we need to enforce additional uniformity on the classical low discrepancy sequences to recover the

efficiency of the low discrepancy sequences. The resulting "refined" low discrepancy sequences also have the added advantage that when enough paths are sampled, the estimate converges exactly to the true value. This is in contrast to the Monte Carlo simulation where the convergence is only probabilistic.

The layout of this chapter is as follows: Section 7.1 outlines the problem inherent in valuing path-dependent derivative securities and also describes the Monte Carlo method for simulating through the trees. Section 7.2 discusses how to enhance the convergence using a refinment of low discrepancy sequences. Section 7.3 provides an explicit construction method for generating such sequences. Section 7.4 compares the efficiency of simulating through the trees using the proposed sequences to the random and classical low discrepancy sequences. Section 7.5 concludes the chapter.

7.1 Monte Carlo Pathwise Valuation

During the last two decades researchers have devoted considerable effort to the development of efficient numerical procedures for pricing derivative securities (or contingent claims). Very often the variables underlying derivative securities are the prices of traded securities or other factors such as interest rates. Examples include a stock option whose value is contingent on the price of a stock, or a bond option whose value is contingent on the value of the bond which in turn depends on the evolution of future interest rates.

When analytic expressions are not available, a common approach to the valuation of derivative securities is to model the underlying variable(s) in a discrete-time arbitrage-free framework. For instance, a continuous-time asset price process is

often approximated by an arbitrage-free binomial (see Cox, Ross and Rubinstein [18]) or trinomial (see Boyle [9]) lattice model of price evolution. The discrete-time models become the basis for direct computation of derivative securities. The backward induction method, which involves rolling backward at each node of the lattice, is a powerful numerical tool for obtaining the price of the derivative security.

In addition to modeling the movement of an asset, the lattice models are frequently used to model the evolution of interest rates. Examples of these models are Ho and Lee [40], Black, Derman and Toy [7], Black and Karasinski [8], Heath, Jarrow and Morton [35], and extended Vasicek (see Hull and White [43]). Interest-rate contingent claims are similarly valued using backward induction technique.

Although the backward induction technique is fast and efficient, it can be computationally demanding in certain circumstances, particularly when the payoffs of the derivative securities depend on the history of the underlying state variable(s) or when movements of the underlying state variable(s) are not recombining. Examples of the first case are

- path-dependent options such as Asian option whose payoff depends on the average of the asset prices, or lookback options whose payoff is a function of the extreme asset price achieved during the option's life.

- Mortgage-backed securities (MBS) or index amortization swaps (IAS). An MBS is a fixed-rate debt security where the principal may be paid off prior to maturity. The amount of prepayment is usually assumed to depend on the prevailing level of interest rates. An IAS is a fixed-for-floating interest rate swap where the notional principal is reduced according to a prespecified

prepayment schedule. The amount of prepayment again depends on the prevailing interest rate level. For both securities, the values at a particular time depend on the interest rate level and the cumulative amount of prepayments. In other words, the history of the interest rates must be known in order to price these types of securities. This leads to the path-dependency nature of the securities.

A popular example of the second case is the Heath, Jarrow and Morton model where the movements of the interest rates are, in general, not recombining. This results in a so-called bushy tree. For a binomial (or trinomial) lattice that is recombining, there are only $s+1$ (or $2s+1$) nodes after s time steps. On the other hand, for a non-recombining binomial (or trinomial) tree, there are 2^s (or 3^s) branches after s time steps. This exponential growth of the branches limits the efficiency of the backward induction method.

Several numerical techniques have been proposed for overcoming this problem. These include Hull and White [42] and Ho [39]. Hull and White combine the techniques of forward induction, backward induction and interpolation. Ho proposes the linear path space method, a schematic algorithm which generates the paths according to their importance.

Another approach is to rely on simulation to sample the scenarios or paths. Each scenario is typically simulated as follows: Suppose we have constructed an arbitrage-free b-nomial model. At each node of the tree, the state variable can evolve to one of the b states (or branches). For convenience, we label these branches as $0, 1, \ldots, b-1$. Suppose further that the risk-neutral probability of moving to

branch j is b_j where $b_j \geq 0$ and $\sum_{j=0}^{b-1} b_j = 1$. We also assume the branching probabilities are independent of time and state. This constraint can easily be relaxed. At time 0, the initial value of the underlying state variable is always known so that no simulation is required. At time 1, the underlying state variable could move to one of the b branches. This is resolved by drawing a random point $x \in [0, 1)$ so that the branching is determined according to the following rule:

$$\left.\begin{array}{ll} \text{if } x < b_0, & \text{then move to branch 0} \\ \text{if } x < b_0 + b_1, & \text{then move to branch 1} \\ \qquad\vdots & \\ \text{if } x \geq 1 - b_{b-1}, & \text{then move to branch } b-1. \end{array}\right\} \tag{7.1.1}$$

See Figure 7.1 for a generic branching of the nodes. We repeat the above procedure for subsequent time steps until we have simulated an entire path. The discounted value of the cash flow of the derivative security along the simulated path is computed. We denote the resulting quantity as the *pathwise value*. The estimate of the derivative security is obtained by averaging over all the simulated pathwise values. The convergence of this method is guaranteed by the law of large numbers. We label this technique as *Monte Carlo pathwise valuation* (MCPV).

7.2 Low Discrepancy Pathwise Valuation

The last section discussed Monte Carlo pathwise valuation. We noted that each simulated path with s time step requires s random numbers or a random point x from an s-dimensional unit-cube. In this section, we describe another method of

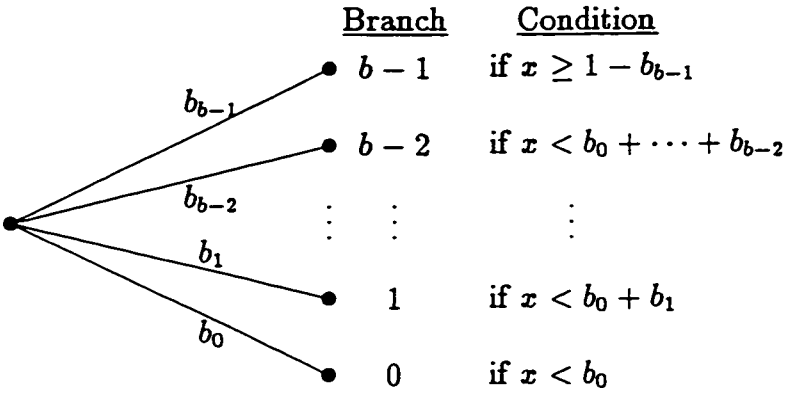


Figure 7.1: Branching from the Node

sampling the paths using low discrepancy sequences. We denote this method as *low discrepancy pathwise valuation* (LDPV). We also show that under certain conditions, low discrepancy pathwise valuation is consistent with backward induction in the sense that both techniques yield exactly the same estimate.

In the rest of this section, we establish the condition under which LDPV is equivalent to backward induction. We assume that we have constructed an arbitrage-free b-nomial model with s time steps. The model can be either recombining or non-recombing. We also assume that the branching movements are equal-probable; i.e. the risk-neutral probability of branching is $1/b$ for each branch. This is not a very restrictive condition since many models fulfil this constraint. For instance, the Black, Derman and Toy [7] model is a binomial interest rate lattice which assumes that the risk-neutral probability of the rate going up or down is 1/2.

Proposition 7.1 *Under the above assumptions, when paths are generated from a* $(0, m, s)$*-net in base* b*, for* $m \geq s$*, LDPV is consistent with backward induction; i.e.*

both methods yield the same value.

This proposition follows directly from the properties of the $(0, m, s)$-net but we spell out the proof for completeness. Before giving the proof of the above proposition, we first give the following two lemmas. These two lemmas are elementary but are included so that the proof to Proposition 7.1 will be more transparent.

Lemma 7.1 *Any point* $\boldsymbol{x} = (x_1, x_2, \ldots, x_s)$ *sampled from the following hypercube*

$$\hat{E} = \prod_{i=1}^{s} \left[\frac{a_i}{b}, \frac{a_i + 1}{b} \right), \tag{7.2.1}$$

for integers $0 \leq a_i < b$, $1 \leq i \leq s$, *leads to the same set of paths or scenarios.*

Proof By definition, $\boldsymbol{x} \in \hat{E}$ implies

$$\begin{aligned} \frac{a_1}{b} \leq \quad & x_1 \quad < \frac{a_1 + 1}{b}, \\ & \vdots \\ \frac{a_s}{b} \leq \quad & x_s \quad < \frac{a_s + 1}{b}. \end{aligned}$$

Since we have assumed the probability of each branching is $1/b$, the decision rule (7.1.1) with $b_j = 1/b$, $0 \leq j \leq b - 1$, and for $\boldsymbol{x}$ satisfying (7.2.1) would result in moving to branch a_1 at time 1, to branch a_2 at time 2, ..., and to branch a_s at time s. Hence this leads to the same set of paths or scenarios. □

Lemma 7.2 *Suppose* $[0, 1)^s$ *is partitioned into hypercubes of the form (7.2.1) and a point is drawn from each hypercube to generate a corresponding scenario. The*

value of the derivative security by averaging over b^s scenarios is consistent with backward induction.

Proof This is obvious by noting that the backward induction is equivalent to computing the pathwise values over b^s distinct paths and there are b^s hypercubes of the form (7.2.1) in $[0,1)^s$. □

Proof Proof of Proposition 7.1. We first consider the case where $m = s$. By definition, every elementary interval in base b of volume b^{-s} of a $(0, s, s)$-net in base b must contain a single point. In particular, the elementary interval (or hypercube) of the form

$$\hat{E} = \prod_{i=1}^{s} \left[\frac{a_i}{b}, \frac{a_i + 1}{b} \right), \tag{7.2.2}$$

which corresponds to setting $d_1 = \cdots = d_s = 1$ in Definition 3.1 of Chapter 3, must contain a point. Note that this particular elementary interval is exactly the same as the hypercube (7.2.1). Hence it follows from Lemmas 7.1 and 7.2 that LDPV is equivalent to backward induction.

We now consider $m > s$. In this case, it is sufficient to show that each elementary interval in base b of volume b^{-s} (as defined in (7.2.2)) contains the same number of points so that each distinct scenario is generated the same number of times. This is proved by noting that any (t, m, s)-net in base b is also a (u, m, s)-net in base b for integers $t \leq u \leq m$; see Proposition 3.1 of Chapter 3. Hence an $(0, m, s)$-net in base b is also an $(m - s, m, s)$-net in base b. Consequently, each elementary interval of volume b^{-s} contains exactly b^{m-s} points and this completes the proof. □

Remarks:

1. To minimize the computations, we always assume $m = s$, the minimum value of m for which LDPV is consistent with backward induction.

2. Corollary 4.21 of Niederreiter [68] states that for $m \geq 2$, a $(0, m, s)$-net in base b can only exist if $s \leq b + 1$. This implies that when $s > b + 1$, it is impossible to compute the pathwise values from a $(0, m, s)$-net in base b. This is unfortunate since in most problems, the number of time steps is much larger than the number of branches.

3. Proposition 7.1 is only a sufficient condition, not a necessary condition. To illustrate, let consider the following two point sets each with 8 elements:

A	B
(0, 0, 0)	(0, 0, 0)
(1/8, 1/8, 5/8)	(1/8, 1/8, 1/8)
(2/8, 4/8, 4/8)	(2/8, 4/8, 4/8)
(3/8, 5/8, 1/8)	(3/8, 5/8, 5/8)
(4/8, 2/8, 6/8)	(4/8, 2/8, 6/8)
(5/8, 3/8, 3/8)	(5/8, 3/8, 7/8)
(6/8, 6/8, 2/8)	(6/8, 6/8, 2/8)
(7/8, 7/8, 7/8)	(7/8, 7/8, 3/8)

Note that these two point sets are $(1, 3, 3)$-nets in base 2. To verify, we just have to show that each elementary interval of the following form, which has

volume 2^{-2}, contains exactly two points.

$$\hat{E}_1 = \left[\frac{a_1}{4}, \frac{a_1+1}{4}\right) \times [0,1) \times [0,1), \quad 0 \le a_1 < 4$$

$$\hat{E}_2 = [0,1) \times \left[\frac{a_2}{4}, \frac{a_2+1}{4}\right) \times [0,1), \quad 0 \le a_2 < 4$$

$$\hat{E}_3 = [0,1) \times [0,1) \times \left[\frac{a_3}{4}, \frac{a_3+1}{4}\right), \quad 0 \le a_3 < 4$$

$$\tilde{E}_4 = \left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times [0,1), \quad 0 \le a_1 < 2,\ 0 \le a_2 < 2$$

$$\tilde{E}_5 = \left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times [0,1) \times \left[\frac{a_3}{2}, \frac{a_3+1}{2}\right), \quad 0 \le a_1 < 2,\ 0 \le a_3 < 2$$

$$\tilde{E}_6 = [0,1) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times \left[\frac{a_3}{2}, \frac{a_3+1}{2}\right), \quad 0 \le a_2 < 2,\ 0 \le a_3 < 2.$$

However, these two point sets are not $(0,3,3)$-nets in base 2. If these two point sets are $(0,3,3)$-nets, then every elementary interval of the following with volume 2^{-3} must contain a single point.

$$\tilde{E}_1 = \left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times \left[\frac{a_3}{2}, \frac{a_3+1}{2}\right), \quad 0 \le a_i \le 1,\ 1 \le i < 4$$

$$\tilde{E}_2 = \left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times \left[\frac{a_2}{4}, \frac{a_2+1}{4}\right) \times [0,1), \quad 0 \le a_1 < 2,\ 0 \le a_2 < 3$$

$$\tilde{E}_3 = \left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times [0,1) \times \left[\frac{a_3}{4}, \frac{a_3+1}{4}\right), \quad 0 \le a_1 < 2,\ 0 \le a_3 < 4$$

$$\tilde{E}_4 = \left[\frac{a_1}{4}, \frac{a_1+1}{4}\right) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times [0,1), \quad 0 \le a_1 < 4,\ 0 \le a_2 < 2$$

$$\tilde{E}_5 = \left[\frac{a_1}{4}, \frac{a_1+1}{4}\right) \times [0,1) \times \left[\frac{a_3}{2}, \frac{a_3+1}{2}\right), \quad 0 \le a_1 < 4,\ 0 \le a_3 < 2$$

$$\tilde{E}_6 = [0,1) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times \left[\frac{a_3}{4}, \frac{a_3+1}{4}\right), \quad 0 \le a_2 < 2,\ 0 \le a_3 < 4$$

$$\tilde{E}_7 = [0,1) \times \left[\frac{a_2}{4}, \frac{a_2+1}{4}\right) \times \left[\frac{a_3}{2}, \frac{a_3+1}{2}\right), \quad 0 \le a_2 < 4,\ 0 \le a_3 < 2.$$

To show that these two point sets are not $(0,3,3)$-nets, it is sufficient to just find one elementary interval $\tilde{E}_i$ that does not have exactly one point. For instance, let consider the elementary interval $\tilde{E}_2$ with $a_1 = a_2 = 0$. In this case, either point set A or B has two points in this particular elementary interval, hence violating the condition for being a $(0,3,3)$-net. Consequently, these two point sets are only $(1,3,3)$-nets.

Despite that these two point sets are not $(0,3,3)$-nets in base 2. LDPV using point set A is nevertheless consistent with backward induction while using point set B is not. To see this, we first deal with point set A. Observe that every elementary interval in base 2 of the form $\tilde{E}_1$ contains a single point for point set A. This implies that LDPV using this point set will be consistent with backward induction.

Now we consider point set B. In this case, each elementary intervals of the

form

$$[0,1/2) \times [0,1/2) \times [0,1/2),$$
$$[0,1/2) \times [1/2,1) \times [1/2,1),$$
$$[1/2,1) \times [0,1/2) \times [1/2,1),$$
$$[1/2,1) \times [1/2,1) \times [0,1/2),$$

contains 2 points while none of the elementary intervals of the form

$$[0,1/2) \times [0,1/2) \times [1/2,1),$$
$$[0,1/2) \times [1/2,1) \times [0,1/2),$$
$$[1/2,1) \times [0,1/2) \times [0,1/2),$$
$$[1/2,1) \times [1/2,1) \times [1/2,1)$$

contains any point. Hence, that LDPV using this point set will not be consistent with backward induction.

The consequence of the third remark is that when $t > 0$, there is no assurance that LDPV using (t,s,s)-net in base b is consistent with backward induction. To ensure such consistency, we must impose an additional constraint on the classical (t,m,s)-net and (t,s)-sequence. We denote the resulting sequence as a *refined* (t,m,s)-net and a *refined* (t,s)-sequence.

Definition 7.1 *A refined (t,m,s)-net in base b is a (t,m,s)-net in base b satisfying the additional condition that for $m \geq s$, every elementary interval in base b of the*

form (7.2.2) contains exactly b^{m-s} points.

Definition 7.2 *An infinite sequence of points $\{\boldsymbol{x}_n\} \in [0,1)^s$ is a refined (t,s)-sequence in base b if for all $k \geq 0$ and $m > s$, the finite sequence $\boldsymbol{x}_{kb^m}$, ..., $\boldsymbol{x}_{(k+1)b^m-1}$ forms a refined (t,m,s)-net in base b.*

It follows from these two definitions that the refined nets and sequences have stronger combinatorial structure than the classical nets and sequences. In the trivial case where $t = 0$, a refined $(0,m,s)$-net is equivalent to a $(0,m,s)$-net and a refined $(0.s)$-sequence is equivalent to a $(0,s)$-sequence. For $t \neq 0$, another situation where the classical (t,m,s)-net is also a refined net is when $t = m - s$. To illustrate the difference between a refined (t,m,s)-net and a classical $(m-s,m,s)$-net when $t \neq m - s$, we consider the same point set A discussed earlier. In this example, point set A is a refined $(1,3,3)$-net since this point set not only satisfies the classical $(1,3,3)$-net, it also satisfies an additional constraint that each elementary interval of the form $\tilde{E}_1$ contains exactly a point. On the other hand, a $(0,3,3)$-net not only requires that every elementary interval of the form $\tilde{E}_1$ contains exactly a single point, all elementary interval of the form $\tilde{E}_j$, $2 \leq j \leq 7$ must also contain a point. When $b = 2$, the refined (t,s)-sequence in base 2 corresponds to the low discrepancy sequence which satisfies "Property A" of Sobol' [94].

Proposition 7.2 *An LDPV using a refined (t,m,s)-net in base b, for $m \geq s$, is consistent with backward induction.*

Proof This follows immediately from Definition 7.1 and Lemma 7.2. □

Before proceeding to the next section, we remark that both MCPV and LDPV are discontinuous. To see this, we consider a binomial lattice with recombining states. Let $S(i,j)$ denote the value of the underlying state variable at time step i after j an up-jumps. For simplicity, we also assume the probability of an up-jump or a down-jump is independent of time and states. In other words, at time step i, $S(i,j)$ can evolve to $S(i+1,j)$ with probability b_0 or to $S(i+1,j+1)$ with probability $b_1 = 1 - b_0$ at time step $i+1$.

Suppose we are interested in valuing a discrete-time Asian option with geometric averaging over s reset time points. The pathwise valuation can be described as

$$\text{pathwise valuation estimate} = \frac{e^{-rT}}{N} \sum_{n=1}^{N} \left[\prod_{i=1}^{s} S_n \left(i, \sum_{j=1}^{i} \chi_{\{x_{nj} \geq b_1\}} \right) \right]^{\frac{1}{s}} \qquad (7.2.3)$$

where N is the total number of sampled paths, $\boldsymbol{x}_n = (x_{n1}, \ldots, x_{ns}) \in [0,1)^s$ denotes the n-th term of the sequence, $\{S_n(i, \sum_{j=1}^{i} \chi_{\{x_{nj} \geq b_1\}});\ 1 \leq i \leq s\}$ denotes the n-th simulated path and

$$\chi_{\{x_{nj} \geq b_1\}}) = \begin{cases} 1 & \text{if } x_{nj} \geq b_1, \\ 0 & \text{otherwise.} \end{cases}$$

For MCPV, $\boldsymbol{x}_n$ is drawn from a random number generator while for LDPV, $\boldsymbol{x}_n$ is obtained from the low discrepancy sequences. The discontinuity of the pathwise valuation arises from the inherent characteristic function $\chi_{\{x_{nj} \geq b_1\}}$. The boundary where the characteristic function is defined is, however, parallel to the coordinate axes. This implies that the variation of the function is not infinite. Consequently

the problem of discontinuity discussed in Chapter 4 is not of particular concern in this case.

7.3 Constructions of Refined (t, m, s)-Nets and Refined (t, s)-Sequences

The previous section states the condition under which LDPV is consistent with backward induction. In this section, we provide an explicit construction method for the refined sequences and nets. First, recall that the i-th component of the n-th term of the low discrepancy sequences can be generated from

$$\begin{aligned} x_{ni} &= \frac{(q_{ni1}q_{ni2}\cdots q_{niR})_b}{b^R} \\ &= \frac{q_{ni1}b^{R-1}+\cdots+q_{niR}}{b^R} \\ &= \sum_{r=1}^{R}\frac{q_{nir}}{b^r} \end{aligned} \tag{7.3.1}$$

where

$$\boldsymbol{Q}_{ni} = \boldsymbol{C}_i\boldsymbol{A}_n.$$

The matrices $\boldsymbol{Q}_{ni}$, $\boldsymbol{C}_i$ and $\boldsymbol{A}_n$ are defined in Chapter 3, Section 3.2.

Since we can always obtain a (t, m, s)-net in base b from a (t, s)-sequence in base b, we just discuss the construction of (t, s)-sequences. The general construction principle for generating these sequences is given in Niederreiter [65] while an explicit

algorithm, based on formal Laurent series expansion, for finding the appropriate values c_{ijr} such that the resulting (t, s)-sequence in base b has small t is provided in Niederreiter [66]. This explicit construction method is summarized in the following theorem:

Theorem 7.1 *Let b be an arbitrary prime power and let $p_1, \ldots, p_s \in \mathbf{F}_b[x]$ be pairwise coprime where $s \geq 1$ is arbitrary and $deg(p_i) = e_i \geq 1$ for $1 \leq i \leq s$. For $j \geq 1$ and $1 \leq i \leq s$, let $g_{ij} \in \mathbf{F}_b[x]$ with $\gcd(g_{ij}, p_i) = 1$ and*

$$\lim_{j \to \infty} (je_i - \deg(g_{ij})) = \infty \qquad \text{for } 1 \leq i \leq s. \tag{7.3.2}$$

For $0 \leq k < e_i$, $1 \leq i \leq s$, and $j \geq 1$, now consider the expansion

$$\frac{x^k g_{ij}(x)}{p_i(x)^j} = \sum_{r=w}^{\infty} a_i(j, k, r) x^{-r-1}, \tag{7.3.3}$$

by which the elements $a_i(j, k, r) \in \mathbf{F}_b$ are determined. Here $w \leq 0$ may depend on i, j, k. If the elements $c_{ijr} \in \mathbf{F}_b$ are defined by

$$c_{ijr} = a_i(q + 1, u, r) \qquad \text{for } 1 \leq i \leq s,\ j \geq 1,\ r \geq 0, \tag{7.3.4}$$

where[1]

$$j - 1 = qe_i + u \tag{7.3.5}$$

[1] *or* $q = \lfloor (j - 1 - u)/e_i \rfloor$.

with integers $q = q_{ij}$ *and* $u = u_{ij}$ *satisfying* $0 \leq u < e_i$, *then the sequence*

$$\boldsymbol{x}_n = (x_{n1}, \ldots, x_{ns}) \in [0,1)^s, \qquad n \geq 0,$$

is a (t, s)*-sequence in base* b *with*

$$t = \sum_{i=1}^{s} (e_i - 1).$$

There are several important insights from this theorem. First, to minimize the quantity t for fixed s and b, the degrees of the polynomials $p_1, \ldots, p_s$ have to be chosen as small as possible. This is achieved by choosing $p_1, \ldots, p_s$ as the first s monic irreducible polynomials over $\mathbf{F}_b$, where $\deg(p_i) \leq \deg(p_j)$, $1 \leq i \leq j \leq s$. Second, the problem of determining the elments c_{ijr} is now reduced to finding the elements $a_i(j,k,r)$ as given by relation (7.3.4). Third, for fixed i, j, k, the coefficient $a_i(j,k,r)$ becomes a function of r and satisfy a linear recurrence relation with characteristic polynomial $p_i(x)^j$. Fourth, the calculation of $a_i(j,k,r)$ can further be simplified by a convenient choice of the polynomials $g_{ij}(x)$.

In Niederreiter [66], a special case where $g_{ij}(x) = 1$, for $i, j \geq 1$, is considered so that identity (7.3.3) becomes

$$\frac{x^k}{p_i(x)^j} = \sum_{r=w}^{\infty} a_i(j,k,r) x^{-r-1}.$$

The procedure for solving $a_i(j,k,r)$, and hence $c_i(j,k,r)$, can be summarized in the following 5-step algorithm. (Note that we assume at most b^R points will ever be

generated so that we only need to obtain the generator matrix C_i with dimension up to $R \times R$.)

I Choose a suitable monic polynomial $p_i(x) \in \mathbf{F}_b[x]$ and let the degree of $p_i(x)$ be $e_i \geq 1$. Set $j \leftarrow 0$, $q \leftarrow -1$, and $u \leftarrow e_i$.

II Increment j. If $u = e_i$, do Step III; otherwise go to Step IV.

III Increment q, set $u \leftarrow 0$ and calculate

$$b(x) = p_i(x)^{q+1} = x^m - b_{m-1}x^{m-1} - \cdots - b_0$$

to obtain a polynomial of degree $m = e_i(q+1)$. Then set

$$a_i(q+1, 0, r) = 0 \qquad \text{for } 0 \leq r \leq m-2$$

$$a_i(q+1, 0, r) = 1 \qquad \text{for } r = m-1$$

$$a_i(q+1, 0, r) = \sum_{l=1}^{m} b_{m-l} a_i(q+1, 0, r-l) \qquad \text{for } m \leq r \leq R + e_i - 2$$

IV For $0 \leq r < R$, set

$$c_{ijr} = a_i(q+1, 0, r+u). \tag{7.3.6}$$

V If $j = R$, stop; otherwise increment u and go to Step II.

Equation (7.3.6) is equivalent to (7.3.4) since for fixed q, we have $a_i(j, u, r) =$

$a_i(j, u+1, r-1)$.

As pointed out by Bratley, Fox and Niederreiter [14], one problem arises from the above implementation is so-called the "leading-zeros" phenomenon. In Step III, we set $a_i(j,0,r) = 0,\ 0 \le r \le e_i(q+1) - 2$. This implies that the initial values c_{ijr} are also zeros. For instance if $j = 1$, we have $c_{i1r} = 0$ for $0 \le r \le e_i - 2$. This leads to

$$d_{n1} = \sum_{r=0}^{R-1} c_{i1r} a_r(n) = 0$$

for $n < b^{e_i - 1}$. Consequently there are too many points, particular those near the initial segment of the sequences, clustered at the origin.

Several solutions have been suggested by Bratley, Fox and Niederreiter [14]. The simplest approach is to throw away a certain number of initial terms of the sequence. Naturally, the number to be skipped should be a power of base b. Furthermore, this power should be at least equal to e_s, where e_s is the maximum degree of the polynomials used to obtain the generating matrix $\boldsymbol{C}_s$.

Another solution is to modify the initial values $a_i(j,0,r),\ 0 \le r \le e_i(q+1) - 2$. Bratley, Fox and Niederreiter [14] discuss a general method for defining these initial values and in [15] a computer program is implemented by setting

$$a_i(j,0,r) = 0 \qquad 0 \le r < e_i q$$

$$a_i(j,0,r) = 1 \qquad e_i q \le r \le e_i q - 2$$

in Step III.

We now consider an alternative method for assigning these initial values. These values are selected so that the resulting sequence is a refined (t, s)-sequence. Let $k_n = (k_{n1}, \ldots, k_{ns})$, for integers $0 \leq k_{ni} < b$, denote the branching process corresponds to $x_n = (x_{n1}, \ldots, x_{ns})$. Therefore, at time 1 branch k_{n1} is realized, at time 2 branch k_{n2} is realized, ..., and at time s branch k_{ns} is realized. Since we have assumed the probability of branching is $1/b$ for each branch, the decision rule (7.1.1) implies that $x_{ni} \in [k_{ni}/b, (k_{ni} + 1)/b)$. From (7.3.1), it follows that

$$\frac{k}{b} \leq x_{ni} < \frac{k+1}{b}$$

if and only if $q_{ni1} = k$. In other words, $(k_{n1}, \ldots, k_{ns})$ belongs to the path generated from $(x_{n1}, \ldots, x_{ns})$ if and only if

$$k_{ni} = q_{ni1}$$

for $1 \leq i \leq s$.

Suppose $\{x_n\}$ is a (t, s)-sequence in base b so that $x_{jb^s}, \ldots, x_{(j+1)b^s-1}$ forms a (t, s, s)-net in base b. Then for $jb^s \leq n < (j+1)b^s$, the base b expansion corresponding to n can be expressed as

$$n \equiv (\tilde{a}_{R-1} \cdots \tilde{a}_s a_{s-1} \cdots a_0)_b,$$

for fixed digits $\tilde{a}_s \cdots \tilde{a}_{R-1}$ and arbitrary digits $a_0 \cdots a_{s-1}$.

Therefore, the path $(k_{n1}, \ldots, k_{ns})$ belongs to $(x_{n1}, \ldots, x_{ns})$ if and only if the

following system of equations is satisfied:

$$a_0 c_{i10} + \cdots + a_{s-1} c_{i1\,s-1} + \tilde{a}_s c_{i1s} + \cdots + \tilde{a}_{R-1} c_{i1\,R-1} = q_{ni1} = k_{ni}$$

for $1 \le i \le s$. Since $\tilde{a}_{R-1} \cdots \tilde{a}_s$ are fixed, the above system of equations reduces to

$$a_0 c_{i10} + \cdots + a_{s-1} c_{i1\,s-1} = \tilde{k}_{ni}$$

where $\tilde{k}_{ni} = k_{ni} - \tilde{a}_s c_{i1s} - \cdots - \tilde{a}_{R-1} c_{i1\,R-1}$. From elementary algebra, this system of equations yields a unique solution if and only if the following condition holds:

$$\det \begin{vmatrix} c_{110} & c_{111} & \cdots & c_{11\,s-1} \\ c_{210} & c_{211} & \cdots & c_{21\,s-1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{s10} & c_{s11} & \cdots & c_{s1\,s-1} \end{vmatrix} \neq 0 \qquad (7.3.7)$$

Hence we have the following theorem, which generalizes Theorem 1 of Sobol' [94]:

Theorem 7.2 *A (t, s)-sequence in base b is also a refined (t, s)-sequence in base b if and only if condition (7.3.7) is satisfied.*

We remark that the condition in Theorem 1 of Sobol' [94] has the determinant of the matrix in (7.3.7) being set to 1. This is consistent with the above theorem since Sobol' only considers $b = 2$.

7.4 Empirical Studies

In this section, we compare the efficiency of both MCPV and LDPV. For LDPV, we also examine the impact of using the refined low discrepancy sequences. Two models are used in our comparisons. Subsection 7.4.1 considers a binomial model which discretizes the continuous-time asset price while Subsection 7.4.2 deals with a trinomial lattice that models the evolution of short rates. Since each node of the tree either has 2 or 3 branches, we consider the (t, s)-sequences in bases 2 and 3. We use the implementation given in Bratley, Fox and Niederreiter [14] to generate these sequences. The refined (t, s)-sequences in bases 2 and 3 are generated similarly except that the generator matrices are constructed so as condition (7.3.7) is satisfied.

We evaluate the efficiency of the proposed methods by comparing the root-mean-squared relative error (RMSE). This measure is defined as

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left(\frac{\hat{C}_j - C_j}{C_j}\right)^2}$$

where $\hat{C}_j$ and C_j denote respectively, the simulated value and the backward induction value for the j-th derivative security contract. In our empirical studies, we consider $M = 50$ and the parameter values for each derivative security contract are randomly generated. This measure of performance avoids the situation where the result is influenced by a particular set of parameter values.

7.4.1 Binomial Lattice

In this subsection, we consider simulating a binomial lattice using Cox, Ross and Rubinstein [18] model. In one time step $\Delta t = T/s$, where T is the time until maturity of the derivative security and s is the number of discretization time points, the asset price moves up by a proportional amount u with probability of "up-jump" b_1, or down by a proportional amount d with probability of "down-jump" $b_0 = 1 - b_1$. From standard no-arbitrage arguments, the parameters u, d and b_1 are related according to

$$u = \frac{1}{d} = e^{\sigma\sqrt{\Delta t}}, \tag{7.4.1}$$

and

$$b_1 = \frac{e^{r\sqrt{\Delta t}} - d}{u - d} \tag{7.4.2}$$

where σ is the annualized volatility of the rate of return of the asset and r is the continuously compounded annual interest rate. The condition $ud = 1$ ensures that the binomial lattice is recombining.

As discussed in Section 7.1, MCPV is a common tool for valuing path-dependent options such as the discrete-time Asian option with arithmetic averaging. When the payoff of the Asian option depends on the geometric average of the asset prices, Panjer [79] develops a simple and efficient method which yields an exact pricing in a recombining binomial lattice. Hence the geometric average option with arbitrary discrete reset time points can be priced using Panjer's method that is consistent

with backward induction.

Our primary objective is to demonstrate the efficiency of LDPV. Hence we consider the discrete-time geometric Asian option so that for any number of averaging time points, the simulated values from LDPV and MCPV using (7.2.3) can be compared to the exact values from Panjer's method. In this case, the number of averaging time points represents the dimensions of the problem.

The parameter values for each Asian option contract are generated as follows: the strike price is fixed at 100, the initial asset price is uniformly distributed between 50 and 150, the annual volatility is uniformly distributed between 10% and 60%, the expiration date is uniformly distributed between 6 months and 3 years and and the annual interest rate is uniformly distributed between 5% and 15%.

Once the required parameter values are obtained, we construct the binomial lattice (or CRR model). Panjer's algorithm is then used to determine the option price. If the option price falls below 0.5, the set of parameter values is discarded and is replaced by another randomly generated set until the option price is at least 0.5. This is because very low option prices may lead to less reliable estimates of RMSE.

We divide our comparison into two cases where $b_0 = b_1 = 1/2$ and $p \neq 1/2$. Naturally, we would expect LDPV using the refined nets is most effective for $b_0 = b_1 = 1/2$. For arbitrary set of parameter values, the probability p determined from (7.4.2) need not be $1/2$. A simple way of enforcing $b_1 = 1/2$ is by manipulating the parameter value r. Rather than letting r be generated randomly, we determine r

so that it satisfies

$$r = \frac{\log((u+d)/2)}{\sqrt{\Delta t}}$$

where u and d are defined in (7.4.1). The relation ensures that the resulting b_1 is $1/2$.

In Part A, the option contracts considered are always confined to $b_1 = 1/2$. In Part B, we relax this constraint and repeat the comparison. It should be noted that the condition $b_0 = b_1 = 1/2$ is not restrictive since for an arbitrary set of input parameter values, the continuous-time asset price process can always be discretized such that the probability of branching is equally probable. See Jarrow and Turnbull [44] and He [34] for details.

Part A

We first consider $s = 15$ and randomly generate 50 sets of parameter values subject to the constraint $b_1 = 1/2$. For LDPV, a refined $(t, 15, 15)$-net in base 2 and a conventional $(t, 15, 15)$-net in base 2 are generated. From these two large nets, the RMSEs at $N = 2^m$, $7 \leq m \leq 15$ based on the 50 randomly generated option contracts are computed. The entire procedure is repeated using one additional refined and conventional $(t, 15, 15)$-nets that are generated by skipping an initial part of the sequence. For MCPV, a random sequence of 2^{15} elements are generated and the RMSEs at the same number of points as the nets are also computed. Table 7.1 reports the simulated results.

In general, we observe that LDPV using the adjusted nets is most effective

	LDPV ($b = 2$)				MCPV
	classical nets		refined nets		
m	skip = 0	skip = 32,768	skip = 0	skip = 32,768	
7	32.704	10.074	4.788	2.814	20.284
8	11.439	5.569	3.047	2.234	17.934
9	3.731	1.411	3.026	3.153	12.690
10	1.510	0.668	0.950	0.706	8.775
11	0.315	0.235	0.501	0.150	6.631
12	0.315	0.235	0.196	0.183	4.373
13	0.072	0.251	0.084	0.056	3.743
14	0.150	0.138	0.066	0.066	2.559
15	0.003	0.003	0.000	0.000	1.195

Table 7.1: RMSE (%) Based on 50 Randomly Generated Asian Option Contracts with 15 Reset Points and $b_0 = b_1 = 1/2$

while MCPV is the least. What is even more interesting is the "initial sequence" phenomenon exhibited in LDPV. The program in Bratley, Fox and Niederreiter [14] recommend throwing away the first 4,096 terms of the sequences. In our simulation results, we ignore the initial 32,768 elements for the skipped sequences. The number chosen is arbitrary, as long as it is a power of the base. Throwing away an initial portion of the sequence does not affect the asymptotic discrepancy of the sequence. However, it does have an important implication on the efficiency, particularly when only a small number of point set is considered. To illustrate, let consider the simulation results for $m = 7$. Without skipping the initial segment of the sequence, the RMSE using the first 2^7 points is 32.70%. This value is more than 3 times as big as the corresponding measure from $(t, 7, 15)$-net drawn after 32,768-th terms. The impact of the "initial sequence" phenomenon diminishes as N increases and becomes insignificant for large N. For instance when $N = 2^{15}$, the RMSEs using

skipped and unskipped sequences are essentially the same.

	LDPV ($b = 2$)		MCPV
m	classical nets	refined nets	
7	36.248	7.742	25.011
8	12.409	6.027	17.275
9	12.940	6.488	11.491
10	4.477	2.886	8.630
11	3.369	0.679	5.292
12	3.789	1.024	3.468
13	2.676	1.222	3.001
14	1.841	0.481	2.294
15	1.982	0.611	1.705

Table 7.2: RMSE (%) Based on 50 Randomly Generated Asian Option Contracts with 30 Reset Points and $b_0 = b_1 = 1/2$

The LDPV using the refined nets not only enhances the effectiveness of the underlying method, the "initial sequence" phenomenon has also been dampened somewhat. For instance with $N = 2^7$, the RMSE using the adjusted net without skipping only results in an $4.788/2.814 \approx 1.7$ increase when compared to the skipped sequence. Furthermore, the adjusted net achieves an order of improvement $10.074/2.814 \approx 3.6$ and $32.704/4.788 \approx 6.8$ times compared to the unadjusted nets with and without skipping.

In general, the RMSE declines as N increases, although it does not decrease monotonically. This is due to the random fluctuation exhibited when the same sequence is used repeatedly to evaluate all the option contracts. When $N = 2^{15}$, the RMSE for the classical net is nonzero while the adjusted net is zero. This is to be anticipated since it follows from Proposition 7.2 that LDPV using the refined

$(t, 15, 15)$-net in base 2 is consistent with backward induction when $s = 15$.

We now repeat the above exercise by doubling the dimensions so that we are valuing average options with 30 reset points. Since it is impractical to consider 2^{30} paths, where LDPV using the refined $(t, 30, 30)$-net in base 2 coincides with backward induction, we compute RMSE using the same number of point sets as in the previous case. The efficiency of the underlying methods therefore depends on how well the small subset of the sampled paths represents the entire 2^{30} paths. The results are presented in Table 7.2 where the nets have been generated by ignoring the initial 32,768 terms. By doubling the dimension, we observe a deterioration of the LDPV using the classical nets. In fact, At $N = 2^7, 2^9, 2^{12}, 2^{15}$, the RMSEs from the classical nets are larger than the corresponding values from the MCPV. On the other hand, the LDPV using the refined nets yields a much higher efficiency relative to both LDPV using the classical nets and MCPV.

An attempt has also been made to carry out the LDPV so that each option contract uses a distinct net. This reduces the variability of the RMSE. The findings are similar. The refined nets are still the most superior.

Part B

So far the examples in Tables 7.1 and 7.2 have restricted to the special case $b_0 = b_1 = 1/2$ where the valuation using the refined (t, m, s)-nets is most efficient. It is therefore of interest to evaluate its efficiency even when b_1 is not constrained to be equal to one half. We again consider $s = 30$. The RMSE (%) based on 50 randomly generated option contracts with $b_1 \in (0, 1)$ are depicted in Table 7.3.

Even though the refined nets are not designed specifically for this situation, we still observe a significant improvement over the classical nets and MCPV. For instance, when $N = 2^{10}$, the RMSE computed from the refined nets is 2.53%. This results in $5.212/2.526 \approx 2.1$ and $7.216/2.526 \approx 2.9$ times more efficient that the corresponding values from LDPV using classical nets and MCPV.

	LDPV ($b = 3$)		MCPV
m	classical nets	refined nets	
7	36.294	7.716	23.655
8	11.418	5.093	17.163
9	12.048	5.356	10.019
10	5.212	2.526	7.216
11	4.113	1.513	6.884
12	3.984	0.999	4.190
13	2.584	1.201	2.704
14	1.760	0.492	1.670
15	1.847	0.614	1.626

Table 7.3: RMSE (%) Based on 50 Randomly Generated Asian Option Contracts with 30 Reset Points and no Restriction on the Probability of Branching

7.4.2 Trinomial Lattice

In this subsection, we consider valuing interest-rate securities on a lattice. In particular, we use the extended Vasicek [103] model proposed by Hull and White [43]. The diffusion process for the short rate can be described as

$$dr = [\theta(t) - ar]dt + \sigma dz$$

where the parameter σ is the volatility of the short rates, the parameter a determines the rate of reversion and dz is the Wiener process. The parameter $\theta(t)$ is time-dependent and is determined so as the trinomial lattice fits exactly to the initial term structure. Unlike the CRR model, the branching probabilities of the trinomial model are time-dependent and state-dependent.

In our empirical studies, we consider the valuation of coupon bonds. The value of the coupon bond is easily priced from the current term structure or from the constructed lattice model using backward induction. The exact price serves as a benchmark for comparing to the simulated values from MCPV and LDPV. Similar to the previous subsection, we compute the RMSE based on 50 randomly generated coupon bonds. For all the randomly generated bonds, the face value of the bond is fixed at 100 and the bond has 5-year maturity with semi-annual coupons. The annual coupon rate is generated so that it is uniformly distributed between 0% and 15%. Since the bonds mature in 5 years, we construct the interest rate lattices that extend to the same time horizon. We also assume that each bond is associated with a randomly generating term structure. The term structure is determined by first randomly generate the instantaneous rate and two other zero-coupon rates for years 3 and 5. We refer these rates as the anchored rates and denote them by r_0, r_3 and r_5 respectively. The zero-coupon rates for other durations are determined by interpolating from the two nearest anchored rates.

Generating the term structure in this manner allows us to have a better control on the shape of the term structure assumption. In our examples, r_0 is uniformly distributed between 5% and 8%, r_3 is uniformly distributed between 8% and 11%,

and r_5 is uniformly distributed between 7% and 13%. When the randomly generated anchored rates satisfy $r_0 < r_3 < r_5$, we have an increasing term structure. Similarly, when $r_0 < r_3 > r_5$, we have a term structure that increases linearly from r_0 to r_3 in year 3, the term structure then decreases linearly to r_5 in year 5. In calibrating the trinomial to the term structure, we also assume the parameter values a and σ of the diffusion process are fixed at 0.1 and 0.014, respectively.

	LDPV ($b = 3$)		MCPV
m	classical nets	refined nets	
4	0.277	0.216	0.736
5	0.082	0.075	0.400
6	0.051	0.041	0.213
7	0.026	0.018	0.123
8	0.015	0.011	0.072
9	0.006	0.005	0.047

Table 7.4: RMSE (%) Based on 50 Randomly Generated Coupon Bonds with 30 Discretization Time Steps

For each pair of coupon bond and term structure assumptions, we construct a trinomial model with 30 discretization time steps. Hence, a stream of cash-flows occurs at time steps 3, 6, ..., 30. These cash flows are discounted from the generated set of interest rate paths. The paths from LDPV are determined using both $(t, m, 30)$-nets and refined $(t, m, 30)$-net in base 3 by skipping the first 3^9 terms. Table 7.4 provides a comparison of the RMSE using LDPV and MCPV with point sets $N = 3^m$, $4 \leq m \leq 9$. It should be noted that for this particular application, the LDPV using the adjusted nets may not be the most efficient since the branching probabilities at each node are not exactly 1/3. Hence, even if 3^{30}

paths are generated, the simulated value from the refined $(t, 30, 30)$-net in base 3 need not match exactly to the value from backward induction. Despite this fact, the results in Table 7.4 indicate considerable improvement over MCPV is achieved while only a marginal improvement over the classical nets. For instance when $N = 2^8$, the RMSEs for the classical nets and MCPV are roughly 1.4 and 6.5 times larger than the corresponding value from the adjusted nets.

7.5 Conclusion

In this chapter, we discussed ways to improve path simulation through a tree using low discrepancy sequences. It turns out that (t, m, s)-nets and (t, s)-sequences can be made more suitable for this purpose by introducing an additional constraint. These refined nets and sequences have a more regular combinatorial structure than standard nets and sequences. We discussed the properties of these refined nets and sequences and showed how to construct them.

To simulate path-dependent options, we observe a gain in efficiency. We applied our approaches to Asian options in the standard Cox, Ross and Rubinstein model. We also considered simulating bond prices on a trinomial interest rate model. In both cases, we observed a significant improvement using the refined low discrepancy sequences over classical low discrepancy sequences (and also over standard Monte Carlo method). We expect that these refined sequences would have the same desirable efficiency in the case of other applications.

Chapter 8

Summary and Future Research

The final chapter is divided into two distinct sections. The first section has attempted to provide a few general guidelines for the application of Monte Carlo and quasi-Monte Carlo methods. Naturally, it is difficult to be very precise when one is discussing problems in the abstract rather than specific examples. The next section of this chapter summarizes some of the possible extensions to this thesis.

8.1 Guidelines

The general problem we have in mind is the evaluation of a multi-dimensional integral. When no special knowledge of the function is available, it is often very useful to estimate the integral using standard Monte Carlo method. From the estimated sample variance of the estimate, we can obtain information on the number of simulation trials needed to attain a given accuracy level.

A natural question concerns the relative efficiency of the Monte Carlo and the

LD methods. In several finance applications, LD methods seem to outperform standard Monte Carlo method. However, the relative performance of the two methods depends on

- the smoothness of the integrands,
- the effective dimensions,
- and the number of points used to evaluate the function.

Furthermore, in the case of the LD method, the convergence will also depend on the properties of the low discrepancy sequences used. To fully exploit the advantages of the LD method, one should study the properties of the integrand and where possible consider using some of the approaches proposed in this thesis. For instance, rather than using conventional low discrepancy sequences, Chapter 7 concludes that we can improve the efficiency of the LD method by using a refined version of low discrepancy sequences.

Broadly speaking, ways of improving the LD methods can be summarized as follows:

1. Have a good understanding of the underlying problem. It will always be the case that the more you know about the problem at hand, the more can be exploited in enhancing the simulation techniques.
2. **Dimension reduction.** Identify the most important input dimensions and reformulate the problem so that the variation is concentrated in the first few dimensions. To illustrate, let us consider simulating the following two

functions:

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{s} w_i x_i \quad \text{and} \quad f_2(\boldsymbol{x}) = \sum_{i=1}^{s} w_{s-i+1} x_i,$$

where the weights w_i are increasing constants in i and $\boldsymbol{x} \in [0,1)^s$. Although these two functions have the same values and Monte Carlo method would yield the same level of efficiency, the effectiveness of the LD method, on the other hand, can vary. Due to the particular arrangement of the weights, the higher dimensions of the first integrand f_1 are of greater importance than the lower dimensions. The opposite observation holds for integrand f_2. Since telescopic low discrepancy sequences have greater uniformity in the initial dimensions, we would expect that these sequences would be more effective applying to f_2 than to f_1. This example illustrates that by simply reordering the weights, an improvement to LD method can be achieved. Other methods of reducing the dimensionality of the problem include the principal component technique (see Acworth, Broadie and Glasserman [1]) and the Brownian bridge discretization method (see Moskowitz and Caflisch [61] and Caflisch, Morokoff and Owen [16]).

3. **Continuity**. If the underlying function has unbounded variation due to discontinuity, consider the generalized smoothing technique discussed in Chapter 5.

4. **Additional Smoothness.** If the underlying function has additional regularity and when the dimension is small, (say $s < 10$), consider using the lattice

points (see Chapter 6) instead of the low discrepancy point sets.

5. **Termination Criteria.** In situations where constructing a probabilistic error bound is desired, the randomization method (for low dimension, say $s < 10$) proposed by Owen [74] or the partial randomization method (for high dimension) discussed in Chapter 4 can be applied.

6. **Variance reduction techniques.** Many of the traditional variance reduction techniques such as antithetic sampling, control variates, and importance sampling can be used in conjunction with the above methods for enhancing LD method.

7. If (t, s)-sequence in base b is used, a more accurate estimate is obtained at N that is proportional to the power of the base.

8.2 Future Work

This section summarizes some of the possible extensions to this thesis. These topics tend to be either extensions of the methodologies introduced in the thesis or applications of quasi-random sequences to specific practical problems in finance and actuarial science. Some of these topics are natural extensions of material in Chapters 3, 4, 5,6 and 7 and we present them along these lines. We then discuss the applications. Since most of the proposed future research suggested in this chapter are very exploratory, we can only give broad ideas and not details.

Chapter 3

The research question in Chapter 3 is to explore methods of constructing a wider class of (t, m, s)-nets using ideas from combinatorial design. In the study of (t, m, s)-nets and (t, s)-sequences, a central issue is the determination of the parameter values t, m, s, b for which these nets or sequences exist. Recall Theorem 3.1 of Chapter 3 states that for any $N > 1$, the discrepancy of the first N points of a (t, s)-sequence in base b satisfies

$$D_N^* \leq C(t, s, b)\frac{\log^s N}{N} + \mathcal{O}\left(\frac{b^t \log^{s-1} N}{N}\right)$$

where $C(t, s, b) = \frac{b^t}{s!} \cdot \frac{b-1}{2\lfloor b/2 \rfloor} \cdot \left(\frac{\lfloor b/2 \rfloor}{\log b}\right)^s$. This result indicates that a sequence with a small discrepancy bound is obtained by minimizing the coefficient $C(t, s, b)$ for a given dimension s. For fixed s, there are certain combinatorial constraints on the parameters b and t which must be satisfied for a (t, s)-sequence in base b to exist. For instance, from Niederreiter's [66] general construction principle, the minimum t for which a (t, s)-sequence in base b exists grows at a rate of $\mathcal{O}(s \log s)$ in dimension s when $b < s$. On the other hand, to obtain a (t, s)-sequence with $t = 0$, the base b must necessarily be increased to at least s. Another situation for which a $(0, s)$-sequence exists is to allow variable bases. This corresponds to the generalized (t, s)-sequences defined in Section 3.3.1 of Chapter 3 with $b_i, 1 \leq i \leq s$ being mutually coprime and $t_i = 0$, $1 \leq i \leq s$. Hence the coefficient of the discrepancy bound for the Halton sequence is

$$C^{\text{Halton}}(s, \boldsymbol{b}) = \prod_{i=1}^{s} \frac{b_i - 1}{2 \log b_i}.$$

The coefficient is minimized by taking $b_i, 1 \leq i \leq s$ to be the s smallest primes. As noted earlier that for conventional (t, s)-sequences, holding b fixed with $b < s$ can only be achieved at the expense of increasing t at a rate of $\mathcal{O}(s \log s)$ while enforcing $t = 0$ requires $b \geq s$. In order to reduce the growth of t while keeping the bases at reasonable magnitude, one idea is to construct a generalized $(\boldsymbol{t}, s)$-sequence. Unfortunately, the only known construction of a generalized $(\boldsymbol{t}, s)$-sequence is the special case where $t_i = 0$ and $b_i, 1 \leq i \leq s$ being mutually coprime. For more general bases, it appears that such sequences do not exist since there is no analog of the Chinese Remainder Theorem when the bases are not pairwise relatively prime.

Nevertheless, a generalized $(\boldsymbol{t}, \boldsymbol{m}, 3)$-net in base $\boldsymbol{b}$ with $t_i = 0, m_i = 1$, $1 \leq i \leq 3$ and $b_1 = b_2 = 2, b_3 = 3$ does exist. In this case, we just have to find a set of 12 points for which all the following 3 sets of generalized elementary intervals contain a single point.

1. $\left[\frac{a_1}{2}, \frac{a_1+1}{2}\right) \times \left[\frac{a_2}{2}, \frac{a_2+1}{2}\right) \times \left[\frac{a_3}{3}, \frac{a_3+1}{3}\right)$ $0 \leq a_1 < 2, 0 \leq a_2 < 2,$ $0 \leq a_3 < 3.$
2. $\left[\frac{a_1}{4}, \frac{a_1+1}{4}\right) \times [0,1) \times \left[\frac{a_3}{3}, \frac{a_3+1}{3}\right)$ $0 \leq a_1 < 4, 0 \leq a_3 < 3.$
3. $[0,1) \times \left[\frac{a_2}{4}, \frac{a_2+1}{4}\right) \times \left[\frac{a_3}{3}, \frac{a_3+1}{3}\right)$ $0 \leq a_2 < 4\ , 0 \leq a_3 < 3.$

An example of a point set which jointly satisfies all the above conditions is

$(0,0,0)$	$(1/2, 13/16, 1/3)$	$(1/16, 7/8, 8/9)$
$(1/4, 1/4, 2/3)$	$(3/4, 5/16, 1/9)$	$(9/16, 3/4, 1/27)$
$(1/8, 1/2, 4/9)$	$(5/8, 1/16, 7/9)$	$(3/8, 3/8, 10/27)$
$(3/8, 9/16, 2/9)$	$(7/8, 1/8, 5/9)$	$(7/8, 5/8, 19/27)$

This exercise indicates that a generalized $(\boldsymbol{t}, \boldsymbol{m}, s)$-net can exist even if the bases are not relatively coprime. This therefore provides hope that generalized $(\boldsymbol{t}, \boldsymbol{m}, s)$-nets or generalized $(\boldsymbol{t}, s)$-sequences also exist for other general bases.

We now sketch a possible approach to construct a generalized sequence or net using the equivalency between a net and an orthogonal array. The work of Mullen [63] and Niederreiter [67] indicate that there exists a $(t, t+2, s)$-net in base b if and only if there exists an orthogonal array $(b^{t+2}, s, b, 2)$ of index b^t; i.e. an $s \times b^{t+2}$ orthogonal array of size b^{t+2} (columns), s factors (rows), b levels and strength 2 and any $2 \times b^{t+2}$ submatrix contains all possible 2×1 columns with the same frequency b^t. However, an analogous equivalence does not exist between $(t, t+k, s)$-nets in base b and orthogonal arrays (b^{t+k}, s, b, k) for $k > 2$.

More recently, Lawrence [55] and Schmid [62] independently have defined a new family of combinatorial objects which they have denoted as "generalized orthogonal arrays" and "ordered orthogonal arrays" respectively. These generalizations enable them to obtain a general equivalency between the (t, m, s)-nets in base b and the orthogonal arrays. More importantly, this combinatorial characterization also provides a new method of constructing (t, m, s)-nets in an arbitrary base b and have resulted in many new nets that have smaller values of t for a given dimension s.

Strictly speaking, the particular type of the orthogonal arrays used in their generalizations is known as the symmetrical orthogonal arrays. The orthogonal arrays are symmetric in the sense that the "levels" of the orthogonal arrays are the same for each row. The equivalence results between the nets and the orthogonal arrays imply a delicate relationship between the "levels" of the orthogonal arrays

and the base of the (t, m, s)-nets. The conventional orthogonal arrays, on the other hand, have been generalized to allow variable "levels" for the orthogonal arrays. The resulting orthogonal arrays are known as asymmetrical orthogonal arrays and were first introduced in Rao [85, 86]. In recent years, several constructions of these orthogonal arrays have been proposed by Wang and Wu [104], Wu, Zhang and Wang [107] and Dey and Midha [21]. Since the asymmetrical orthogonal arrays allow variable "levels", a natural question is whether a similar generalization for the asymmetrical orthogonal arrays would be possible to establish an equivalency between the asymmetrical orthogonal arrays and the generalized $(t, \boldsymbol{m}, s)$-nets.

Chapter 4

In this case, we plan to extend the randomization idea to more general types of sequences. In chapter 4, we have considered scrambling a $(0, s)$-sequence and have discussed its advantages over the classical $(0, s)$-sequence. Nevertheless, the same randomization procedure can be applied to any general (t, s)-sequence in base b, for $t > 0$. We plan to investigate the efficiency of these randomized (t, s)-sequences for $t > 0$.

Chapter 5

In Chapter 5, we have provided a general framework for smoothing a characteristic function. There are many derivative securities which contain discontinuities. Interesting examples include binary options and barrier options. We would like to explore the application of the generalized smoothing technique to these types of options.

Chapter 6

In Chapter 6, we have described the method of good lattice points and have applied it to evaluate particular types of financial derivatives. In the case of certain low dimensional problems, it appears to work very well. Preliminary analysis, however, indicates that it does not work so well in the case of higher dimensional problems (say $s \geq 10$). One possible solution is to consider a hybrid method by using good lattice points for the first few important dimensions and standard or quasi-Monte Carlo for the remaining dimensions. A similar "mixed sequence" which combines both low discrepancy sequence and (pseudo)random sequence has also been proposed by Spanier [97] and Ökten [73]. Another possible research topic is to investigate of role of different periodization schemes. Our preliminary analysis also indicates that as the dimensions get larger, the effectiveness of the method of g.l.p. critically depends on the periodization method.

Chapter 7

Cash-flow testing or solvency testing are important issues faced by actuaries. Very

often they are required to sample interest rate scenarios from a binomial or trinomial tree in projecting the cash flows. In practice, they can only afford to sample a small subset of the scenarios. Hence they are constantly seeking for an efficient algorithm to accomplish this task. The technique discussed in Chapter 7 therefore offers an alternative. We plan to apply this technique to more realistic actuarial problems.

The methods described in this thesis can be applied to a set of different practical problems. There is now considerable interest in measuring risk at the portfolio level and in particular market risk and credit risk. The techniques suggested in this thesis should prove useful in developing more efficient solutions for these problems.

There are also several areas in actuarial science where these methods could be applied. Preliminary work suggests that the low discrepancy approach provides an effective way of simulating a representative set of future interest rate scenarios. These types of calculations are required for dynamic solvency calculations. We also plan to explore the applications of quasi-Monte Carlo methods to some of the classical problems in risk theory (see Dickson and Waters [22]).

In summary, the methods developed in this thesis can be used as a springboard for future research on the construction of more efficient nets and sequences. There also appears to be considerable scope for the application of these methods to a range of practical problems.

Bibliography

[1] P. Acworth, M. Broadie, and P. Glasserman. A comparison of some Monte Carlo and quasi-Monte Carlo methods for option pricing. In Niederreiter, Hellekale, Larcher and Zinterhof [70].

[2] T. J. Aird and J. R. Rice. Systematic search in high dimensional sets. *S.I.A.M Journal of Numerical Analysis*, **14**(2):296–312, 1977.

[3] I.A. Antonov and V.M. Saleev. An economic method of computing LP_τ-sequences. *U.S.S.R Computational Mathematics and Mathematical Physics*, **19**:252–256, 1979.

[4] N.S. Bahvalov. On approximation calculation of multiple integrals. *Vestnik Moskovskogo Universiteta, Seriya Matematiki, Mehaniki, Astronomii, Fiziki, Himii*, 4:3–18, 1959 (Russian).

[5] J. Beck. A two-dimensional van aardenne-ehrenfefst theorem in irregularities of distribution. *Compositio Math.*, **72**:269–339, 1989.

[6] M. Beckers and A. Haegemans. Transformations of integrands for lattice rules. In T.O. Espelid and A. Genz, editors, *Numerical Integration: Recent*

Development, Software and Applications, pages 329–340. NATO ASI Series, Kluwer Academic Publishers, Boston, 1992.

[7] F. Black, E. Derman, and W. Toy. A one-factor model of interest rates and its applications to Treasury bond options. *Financial Analysts Journal*, **46**(1):33–39, 1990.

[8] F. Black and P. Karasinski. Bond and option pricing when short rates are lognormal. *Financial Analysts Journal*, **47**(5):52–59, 1991.

[9] P.P. Boyle. A lattice framework for options pricing with two state variables. *Journal of Financial and Quantitative Analysis*, **23**:1–12, 1988.

[10] P.P. Boyle, M. Broadie, and P. Glasserman. Monte Carlo methods for security pricing. *Journal of Economics Dynamics and Control*, **21**:1267–1321, 1997.

[11] P.P. Boyle, J. Evnine, and S. Gibbs. Numerical evaluation of multivariate contingent claims. *Review of Financial Studies*, **2**(2):241–250, 1989.

[12] E. Braaten and G. Weller. An improved low discrepancy sequence for multidimensional quasi-Monte Carlo integration. *Journal of Computational Physics*, **33**:249–258, 1979.

[13] P. Bratley and B.L. Fox. Algorithm 659: Implementing Sobol's quasi random sequence generator. *ACM Transactions on Mathematical Software*, **14**:88–100, 1988.

[14] P. Bratley, B.L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, **2**(3):195–213, 1992.

[15] P. Bratley, B.L. Fox, and H. Niederreiter. Algorithm 738: Programs to generate Niederreiter's low-discrepancy sequences. *ACM Transactions on Mathematical Software*, **20**(4):494–495, 1994.

[16] R.E. Caflisch, W. Morokoff, and A. Owen. Valuation of morgaged-backed securities using Brownian bridges to reduce effective dimension. *Journal of Computational Finance*, **1**(1):27–46, 1997.

[17] K.L. Chung. An estimate concerning the Kolmogoroff limit distribution. *Transactions of the American Mathematical Society*, **67**:36–50, 1949.

[18] J.C. Cox, S.A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, **7**:229–263, 1979.

[19] R. Cranley and T.N.L. Patterson. Randomization of number theoretic methods for multiple integration. *S.I.A.M Journal of Numerical Analysis*, **23**:904–914, 1976.

[20] R.B. D'Agostino and M.A. Stephens, editors. *Goodness-of-Fit Techniques*. Marcel Dekker Inc, New York, 1986.

[21] A. Dey and C.K. Midha. Construction of some asymmetrical orthogonal arrays. *Statistics & Probability Letters*, **28**:211–217, 1996.

[22] D.C.M. Dickson and H.R. Waters. Ruin problems: Simulation or calculations. *British Actuarial Journal*, **2**(3):727–740, 1996.

[23] K.T. Fang and Y. Wang. *Numer-theoretic Methods in Statistics.* Chapman & Hall, London, 1994.

[24] H. Faure. Discrépance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, **41**:337–351, 1982.

[25] H. Faure. Good permutation for extreme discrepancy. *Journal of Number Theory*, **42**:47–56, 1992.

[26] P. Fox. Algorithm 647: Implementation and relative efficiency of quasi-random sequence generators. *ACM Transactions on Mathematical Software*, **12**(4):362–376, 1986.

[27] S. Galanti and A. Jung. Low-discrepancy sequences: Monte Carlo simulation of option prices. *Journal of Derivatives*, **5**(1):63–83, 1997.

[28] H. German and M. Yor. Bessel process, Asian options and perpetuities. *Mathematical Finance*, **3**:349–375, 1993.

[29] B. Goldman, H. Sosin, and M.A. Gatto. Path-dependent options: Buy at the low, sell at the high. *Journal of Finance*, **34**:1111–1127, 1979.

[30] F. Gray. *Pulse Code Communications.* U.S. Patent 2632058, 1953.

[31] S. Haber. Parameters for integrating periodic functions of several variables. *Mathematics of Computation*, **41**(163):115–129, 1983.

[32] J.H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, **2**:84–90, 1960.

[33] J.H. Halton and G.B. Smith. Algorithm 247: Radical-inverse quasi-random point sequence. *Algorithms*, **7**(12):701, 1964.

[34] H. He. Convergence from discrete- to continuous-time contingent claims prices. *Review of Financial Studies*, **3**(4):523–546, 1990.

[35] D. Heath, R. Jarrow, and A. Morton. Bond pricing and the term structure of interest rates: A discrete time approximation. *Journal of Financial and Quantitative Analysis*, **25**(4):419–440, 1990.

[36] R.C. Heynen and H.M. Kat. Lookback options with discrete and partial monitoring of the underlying price. *Applied Mathematical Finance*, **2**:273–284, 1995.

[37] F.J. Hickernell. The mean square discrepancy of randomized nets. *ACM Transactions on Modeling and Computer Simulation*, **6**:274–296, 1996.

[38] E. Hlawka. Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. *Annali di Matematica Pura Applicata*, **54**:324–334, 1961.

[39] T.S.Y. Ho. Managing illiquid bonds and the linear path space. *Journal of Fixed Income*, **2**(1):80–93, 1992.

[40] T.S.Y. Ho and S. Lee. Term structure movements and pricing interest rate contingent claims. *Journal of Finance*, **41**(5):1011–1029, 1986.

[41] L.K. Hua and Y. Wang. *Applications of Number Theory to Numerical Analysis*. Springer-Verlag, Berlin, 1981.

[42] J. Hull and A. White. Efficient procedures for valuing European and American path-dependent options. *Journal of Derivatives*, **1**(1):21–32, 1993.

[43] J. Hull and A. White. One-factor interest rate models and the valuation of interest rate derivative securities. *Journal of Financial and Quantitative Analysis*, **28**:225–254, 1993.

[44] R. Jarrow and S. Turnbull. *Derivative Securities*. South-Western, Ohio, 1996.

[45] S. Joe. Randomization of lattice rules for numerical multiple integration. *Journal of Computational and Applied Mathematics*, **31**:299–304, 1990.

[46] C. Joy, P.P. Boyle, and K.S. Tan. Quasi-Monte Carlo methods in numerical finance. *Management Science*, **42**(6):926–938, 1996.

[47] P. Keast. Optimal parameters for multidimensional integration. *S.I.A.M Journal of Numerical Analysis*, **10**(5):831–838, 1973.

[48] J. Keifer. On large deviation of the empiric d.f. of vector chance variables and a law of the iterated logarithm. *Pacific Journal Math.*, **11**:649–660, 1961.

[49] D.E. Knuth, editor. *The Art of Computer Programming : Seminumerical Algorithms*, volume **2**. Addison-Wesley, 2nd edition, 1981.

[50] J.F. Koksma. Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. *Mathematica B (Zutphen)*, **11**:7–11, 1942/1943.

[51] N.M. Korobov. The approximate computation of multiple integrals. *Doklady Akademii Nauk SSSR*, **124**:1207–1210 (Russian), 1959.

[52] N.M. Korobov. Properties and calculation of optimal coefficients. *Doklady Akademii Nauk SSSR*, **132**:1009–1012 (Russian), 1959. English Translation: *Soviet Mathematics Doklady*, 1, 696–700.

[53] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences.* John Wiley & Sons, New York, 1974.

[54] J.P. Lambert. A sequence well dispersed in the unit square. *Proceedings of the American Mathematical Society*, **103**(2):383–388, 1988.

[55] M. Lawrence. A combinatorial characterization of (t, m, s)-nets in base b. *Journal of Combinatorial Designs*, 4(4):275–293, 1996.

[56] C. Lécot. An algorithm for generating low discrepancy sequences on vector computers. *Parallel Computing*, **11**:113–116, 1989.

[57] P. L'Ecuyer. Uniform random number generation: Simulation and modeling. *Annals of Operations Research*, **53**:77–120, 1994.

[58] W. Margrabe. The value of an option to exchange one asset for another. *Journal of Finance*, **33**:177–186, 1978.

[59] W.J. Morokoff and R.E. Caflisch. Quasi-random sequences and their discrepancies. *S.I.A.M Journal Scientific Computing*, **15**:1251–79, 1994.

[60] W.J. Morokoff and R.E. Caflisch. Quasi-Monte Carlo integration. *Journal of Computational Physics*, **122**(2):218–230, 1995.

[61] B. Moskowitz and R.E. Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathematical and Computer Modelling*, **23**(8-9):37–54, 1996.

[62] G.L. Mullen and W.Ch. Schmid. An equivalence between (t, m, s)-nets and strongly orthogonal hypercubes. *Journal of Combinatorial Theory: Series A*. **76**:164–174, 1996.

[63] G.L. Mullen and G. Whittle. Point sets with uniformity properties and orthogonal hypercubes. *Monatshefte fur Mathematik*, **113**:265–273. 1992.

[64] H. Niederreiter. Existence of good lattice points in the sense of Hlawka. *Monatshefte fur Mathematik*, **86**:203–219, 1978.

[65] H. Niederreiter. Point sets and sequences with small discrepancy. *Monatshefte fur Mathematik*, **104**:273–337, 1987.

[66] H. Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal Number Theory*, **30**:51–70. 1988.

[67] H. Niederreiter. Orthognoal arrays and other combinatorial aspects in the uniform point distributions in unit cubes. *Discrete Mathmatics*, **106/107**:361–367, 1992.

[68] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. S.I.A.M., Philadelphia, 1992.

[69] H. Niederreiter. Improved error bounds for lattice rules. *Journal of Comlexity*, **9**:60–75, 1993.

[70] H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, editors. *Monte and quasi-Monte-Carlo Methods 1996 : proceedings of a conference at the University of Salzburg, Austria, July 9-12, 1996*, volume **127** of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1998.

[71] H. Niederreiter and P.J-S. Shiue, editors. *Monte and quasi-Monte-Carlo Methods in Scientific Computing: proceedings of a conference at the University of Nevada, Las Vegas, Nevada, USA, June 23-25, 1994*, volume **106** of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1996.

[72] S. Ninomiya and S. Tezuka. Toward real-time pricing of complex financial derivatives. *Applied Mathematical Finance*, **3**:1–20, 1996.

[73] G. Ökten. A probabilistic result on the discrepancy of a hybrid-Monte Carlo sequence and applications. *Monte Carlo Methods and Applications*, **2**(4):255–270, 1996.

[74] A.B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In Niederreiter and Shiue [71, p. 299–317].

[75] A.B. Owen. Monte Carlo variance of scrambled equidistribution quadrature. *S.I.A.M Journal of Numerical Analysis*, **34**(5):1884–1910, 1997.

[76] A.B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, **25**(4):1541–1562, 1997.

[77] A.B. Owen and D. Tavella. Scrambled nets for value at risk calculations. In S. Grayling, editor, *VaR: Understanding and Applying Value-at-Risk.* RISK, London, 1997.

[78] G. Pagès. Van der Corput sequences, Kakutani transforms and one-dimensional numerical integration. *Journal of Computational and Applied Mathematics*, **44**:21–39, 1992.

[79] H.H. Panjer. Exact pricing of geometric average options. Technical report, IIPR 95-04, University of Waterloo, 1995.

[80] S.H. Paskov and J.F. Traub. Faster valuation of financial derivatives. *Journal of Portfolio Management*, **22**(1):113–120, 1995.

[81] N.D. Pearson. An efficient approach for pricing spread options. *Journal of Derivatives*, **3**(1):76–91, 1995.

[82] M.J.D. Powell and J. Swann. Weighted uniform sampling – a Monte Carlo technique for reducing variance. *Journal of the Institute of Mathematics and its Applications*, **2**:228–236, 1966.

[83] W. Press and S. Teukolsky. Quasi-(that is, sub-) random numbers. *Computers in Physics*, **Nov/Dec**:76–79, 1989.

[84] W. Press, S. Teukolsky, T.V. William, and P.F. Brian. *Numerical Recipes in C.* Cambridge University Press, New York, 2nd edition, 1992.

[85] C.R. Rao. Factorial experiments derivable from combinational arrangements of arrays. *Journal of the Royal Statistical Society (supp)*, **9**:128–139, 1947.

[86] C.R. Rao. Some combinatorial problems of arrays and applications to design of experiments. In J.N. Srivastava, editor, *A Survey of Combinatorial Theory*, pages 349–359. North-Holland, Amsterdam, 1973.

[87] K.F. Roth. On irregularities of distribution. *Mathematika*, **1**:73–9, 1954.

[88] A.I. Saltykov. Tables for computing multiple integrals by the method of optimal coefficients. *Zh. Vychisl. Mat. i Mat. Fiz.*, **3**:181–186, 1963. English translation: *U.S.S.R Computational Mathematics and Mathematical Physics*, 3:235–242, 1963.

[89] P. Schatte. Priviate Communication, 1997.

[90] W.M. Schmidt. Irregularities of distribution. *Acta Arithmetics*, **21**:45–50, 1972.

[91] A. Sidi. A new variable transformation for numerical integration. In H. Brass and G. Hämmerlin, editors, *Numerical Integration V*, pages 359–373. Birkhaäuser Verlag, Switzerland, 1993.

[92] I.H. Sloan and S. Joe. *Lattice Methods For Multiple Integration.* Oxford University Press, New York, 1995.

[93] I.M. Sobol'. The distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R Computational Mathematics and Mathematical Physics*, **7**(4):86–112, 1967.

[94] I.M. Sobol'. Uniformly distributed sequences with an additional uniform property. *U.S.S.R Computational Mathematics and Mathematical Physics*, **16**:236–242, 1976.

[95] I.M. Sobol'. On the systematic search in a hypercube. *S.I.A.M Journal of Numerical Analysis*, **16**:790–793, 1979.

[96] I.M. Sobol', V.I. Turchaninov, Yu.L. Levitan, and B.V. Shukhman. Quasirandom seqeuence generators. Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, 1992.

[97] J. Spanier. Quasi-Monte Carlo methods for particle transport problems. In Niederreiter and Shiue [71].

[98] J. Spanier and E.H. Maize. Quasi-random methods for estimating integrals using relatively small samples. *S.I.A.M. Review*, **36**:18–44, 1994.

[99] A.H. Stroud. *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1970.

[100] R. Stulz. Options on the minimum or the maximum of two risky assets: analysis and applications. *Journal of Financial Economics*, **10**:161–185, 1982.

[101] S. Tezuka. *Uniform Random Numbers: Theory and Practice.* Kluwer Academic Publishers, Norwell, Massachusetts, 1995.

[102] I.F. Šargin. A lower estimate for the error of quadrature formulas for certain class of functions. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi*

Fiziki, **3**:370–6, 1963 (Russian). English translation: *U.S.S.R. Computational Mathematics and Mathematical Physics*, 3:489–497.

[103] O.A. Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, **5**:177–188, 1977.

[104] J.C. Wang and C.F.J. Wu. An approach to the construction of asymmetrical orthogonal arrays. *Journal of American and Statistical Association*, **86**:450–456, 1991.

[105] B.E. White. On optimal extreme-discrepancy point sets in the square. *Numerical Mathematics*, **27**:157–164, 1977.

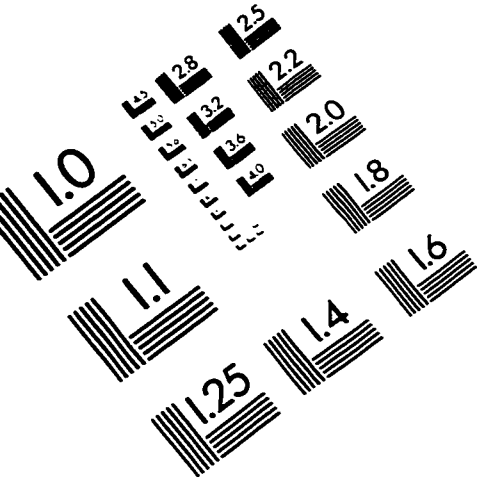
[106] D. Wilcox. Energy futures and options: Spread options in energy markets. Goldman Sachs & Co., New York, 1990.

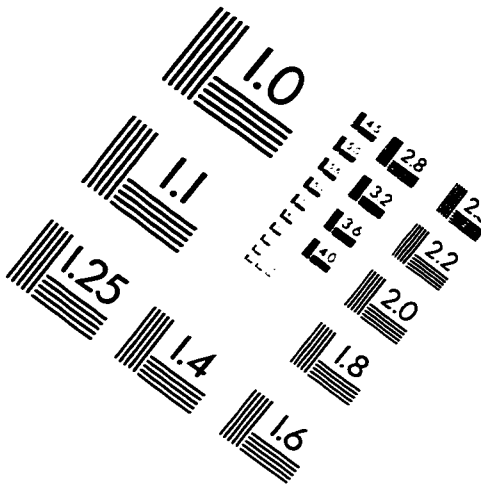
[107] C.F.J. Wu, R. Zhang, and R. Wang. Construction of asymmetrical orthogonal arrays of the type $OA(s^k, s^m(s^{r_1})^{n_1} \cdots (s^{r_i})^{n_i})$. *Statistica Sinica*, **2**:203–219. 1992.

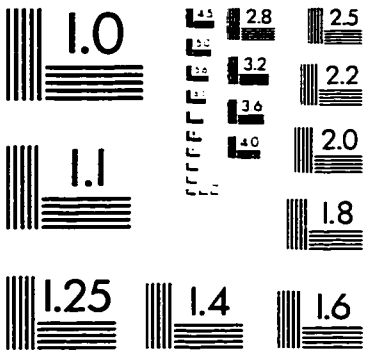
[108] S.K. Zaremba. Some applications of multidimensional integration by parts. *Annales Polonici Mathematici*, **21**:85–96, 1968.

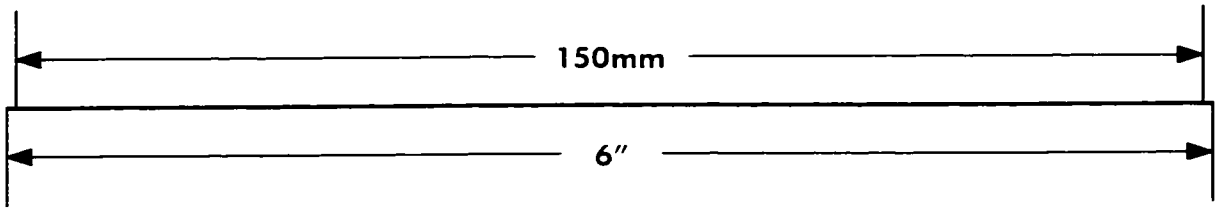
[109] S.K. Zaremba. La methode des "Bons Treillis" pour le calcul des integrales multiples. In S.K. Zaremba, editor, *Applications of Number Theory to Numerical Analysis*, pages 39–119. Academica Press, New York, 1972.

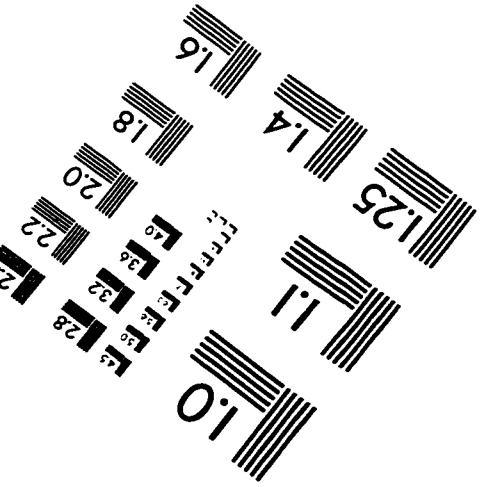
IMAGE EVALUATION
TEST TARGET (QA-3)

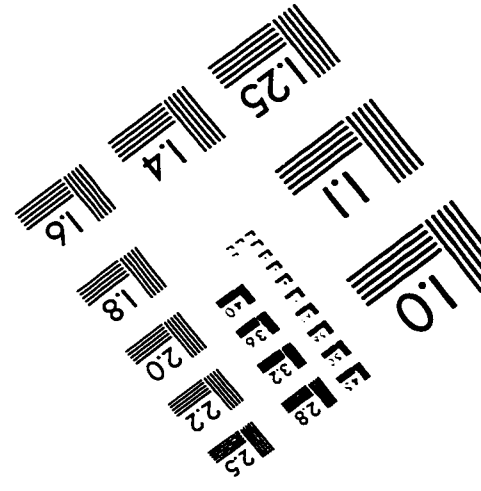












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989