Applications of randomized low discrepancy sequences to the valuation of complex securities

Ken Seng Tan\textsuperscript{a,\,*}, Phelim P. Boyle\textsuperscript{b}

\textsuperscript{a}Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
\textsuperscript{b}Centre for Advanced Studies in Finance, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Abstract

This paper deals with a recent modification of the Monte Carlo method known as quasi-random Monte Carlo. Under this approach, one 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 answers than random sequences. One of the main drawbacks of the use of low discrepancy sequences is that there is no obvious method of computing the standard error of the estimate. This means that in performing the calculations, there is no clear termination criterion for the number of points to use. We address this issue here and consider a partial randomization of Owen's technique for overcoming this problem. The proposed method can be applied to much higher dimensions where it would be computationally infeasible for Owen's technique. The efficiency of these procedures is compared using a particular derivative security. The exact price of this security can be calculated very simply and so we have a benchmark against which to test our calculations. We find that our procedures give promising results even for very high dimensions. Statistical tests are also conducted to support the confidence statement drawn from these procedures. © 2000 Elsevier Science B.V. All rights reserved.

\textsuperscript{*}Corresponding author. Tel.: (519) 888-4567 x 5552; fax: (519) 746-1875.
E-mail addresses: kstan@uwaterloo.ca (K.S. Tan), pboyle@uwaterloo.ca (P.P. Boyle).
1. Introduction

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 computational finance. 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 et al. (1997) 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 finance area. In modern financial economics, security prices are modeled 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. Boyle et al. (1997) describe the method as follows:

Monte Carlo becomes increasingly attractive compared to other methods of numerical integration as the dimension of the problem increases. Consider the integral of the function \( f(x) \) over the \([s]\)-dimensional unit hypercube. The simple (or crude) Monte Carlo estimate of the integral is equal to the average value of the function \( f \) over \( n \) points selected at random from unit hypercube.
From the strong law of numbers this estimate converges to the true value of the integral as \( n \) tends to infinity. In addition, the central limit theorem assures us that the standard error of the estimate tends to zero as \( 1/\sqrt{n} \). Thus the error convergence rate is independent of the dimension of the problem and this is the dominant advantage of the method over classical numerical integration approaches. The only restriction on the function \( f \) is that it should be square integrable and this is a relatively mild restriction.

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 for speeding up the convergence have been proposed. These techniques are known as variance reduction techniques. Recently, so-called quasi-Monte Carlo methods\(^1\) or 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 low discrepancy sequences. The monograph by Niederreiter (1992) provides an excellent discussion of these sequences. Applications of low discrepancy sequences to finance problems are discussed by Boyle et al. (1997), Caflisch et al. (1997), Joy et al. (1996), Ninomiya and Tezuka (1996) and Paskov and Traub (1995). There are differences in the relative efficiency of LD methods versus standard Monte Carlo for 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 (1995) report good results using Sobol' sequences while Ninomiya and Tezuka (1996) conclude generalized Niederreiter sequences are superior. For more general 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 et al. (1992) and Morokoff and Caflisch (1994, 1995). The advantages of LD methods for finance applications appear to stem from the smoothness of the integrand in many applications and the fact that the effective dimension in finance applications is often lower than the nominal 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.

---

\(^1\) This is an unfortunate misnomer. There is nothing random about this method. Nevertheless, the name quasi-Monte Carlo is widely used to describe the evaluation of integrals using low discrepancy sequences. We prefer to call this approach the low discrepancy (LD) method.
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 the standard error of the estimate is readily available. The purpose of the present paper is to investigate a modification of the technique proposed by Owen (1996) 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 paper is as follows. In the next section we introduce the concept of discrepancy. We describe different types of point sets and sequences which have the property of low discrepancy. Section 3 describes a technique developed by Owen which provides a probabilistic error bound for quasi-Monte Carlo by randomizing the points in such a way to preserve the low discrepancy property. In Section 3.1, we propose a simplification to Owen’s randomization technique that is feasible for very high dimensional problems. Section 4 discusses two different ways of obtaining the variance of the estimated values from the randomized nets (or sequences). Section 5 describes the class of complex derivative securities that we will use for our numerical calculations. The numerical comparisons are conducted in Section 6. We divide the comparisons into two parts. The first part (see Section 6.1) deals with low-dimensional examples so that the efficiency of our proposed randomization can be assessed by comparing it to Owen’s randomization technique. Statistical tests are also provided to validate our studies. The second part (see Section 6.2) applies the proposed method to higher dimensional examples. Section 7 concludes the paper.

2. Basic concepts related to discrepancies

The concept of discrepancy is useful in discussing how well points are dispersed throughout a region. There are several types of discrepancy measure. In this paper, we just consider one particular type known as the star discrepancy which we will just term discrepancy. We first give an intuitive description of this concept and then a more formal definition. Suppose we have an $s$-dimensional unit hypercube and a set of points scattered throughout this region. The discrepancy can be thought of as the greatest absolute difference between the continuous uniform probability distribution and a discrete probability distribution, taken over all possible subcubes of $[0,1)^s$ containing the origin. Let $V$ be a subcube of $[0,1)^s$ containing the origin and let $v(V)$ be its volume. The

\footnote{The discrete distribution is obtained by taking the ratio of the number of points in $V$ to the total number of points $N$.}
discrepancy $D_N^s$ of the sequence $\{X_n\}$ of $N$ points is defined as

$$D_N^s = \sup_{V \subseteq [0,1]^s} \left| \frac{\text{Number of points in } V}{N} - v(V) \right|. $$

The expectation of the discrepancy of a random sequence can be shown to be bounded by $(\log(\log N)) N^{-1/2}$. It is possible to construct sequences for which the discrepancy is smaller than that of a random sequence. Sequences satisfying such criteria are known as low discrepancy sequences. Typically, a sequence (or point set) is said to be quasi-random or low discrepancy if the discrepancy is $\mathcal{O}((\log N)^s N^{-1})$ (or $\mathcal{O}((\log N)^s-1 N^{-1})$). Examples of low discrepancy sequences are given by Halton (1960), Sobol’ (1967), Faure (1982) and Niederreiter (1987).

The importance of discrepancy can be seen from the integration error bound given by the Koksma–Hlawka inequality, which states that

$$\int_{[0,1]^s} f(X) dX - \frac{1}{N} \sum_{n=1}^N \hat{f}(X_n) \leq V(f) \cdot D_N^s,$$

where $V(f)$ is the bounded variation in the sense of Hardy and Krause, $\{X_n\}$ is a sequence of $N$ points in $[0,1]^s$ with discrepancy $D_N^s$, and $\hat{f}(X_n)$ is the functional value evaluated at $X_n$. In essence, the Koksma–Hlawka inequality separates the integration error bound into two components. These correspond to the smoothness of the integrand and the uniformity of the sequence used in evaluating the function. The Koksma–Hlawka inequality implies that the sequence with the smaller discrepancy results in smaller error bounds. This suggests we should use a sequence with the smallest possible discrepancy in evaluating the function in order to achieve the smallest possible error bound. Low discrepancy methods strive to attain this goal. The Koksma–Hlawka inequality asserts that if we use low discrepancy sequences as the integration nodes in Monte Carlo integration, the absolute integration error is $\mathcal{O}((\log N)^s N^{-1})$ for infinite sequences and $\mathcal{O}((\log N)^s-1 N^{-1})$ for finite point sets. For large $N$, these rates of convergence are considerably faster than the standard Monte Carlo methods whose error bound is $\mathcal{O}(N^{-1/2})$.

The Koksma–Hlawka inequality provides a theoretical justification for the application of low discrepancy sequences in Monte Carlo integration. In practice, it is not very useful. First, the two factors $D_N^s$ and $V(f)$ are extremely hard to compute. Second, even if these factors can be estimated, the bound is not sharp and is only correct asymptotically. As argued by Caflisch and Morokoff (1994), the number of points, $N$, at which the low discrepancy sequence starts to exhibit asymptotic behaviour grows exponentially with dimension $s$. In actual applications, we normally will be quite far away from the asymptotic behaviour and hence there is a need for a more feasible way of characterizing the error bound. This issue is addressed in this paper.
Many of the low discrepancy sequences, cited above, can be considered as special cases of \((t, m, s)\)-nets and \((t, s)\)-sequences. We now give a brief introduction to these concepts. See Niederreiter (1992, Chapter 4) for a detailed discussion. We start by giving formal definitions. Later we provide a more intuitive discussion using diagrams.

**Definition 1.** An elementary interval in base \(b\) is an interval \(E\) in \([0, 1)^s\) of the form

\[
E = \bigcup_{j=1}^{s} \left[ \frac{a_j}{b^{d_j}}, \frac{a_j + 1}{b^{d_j}} \right)
\]

with \(d_j \geq 0\), \(0 \leq a_j < b^{d_j}\) and \(a_j, d_j\) are integers.

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

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

Another definition of the nets introduced by Owen (1997a) is as follows:

**Definition 4.** Let \(s, m, t, b, \lambda\) be integers with \(s \geq 1, m \geq 0, 0 \leq t \leq m, b \geq 2\), and \(1 \leq \lambda < b\). A sequence \(\{X_n\}\) of \(\lambda b^m\) points is called a \((\lambda, t, m, s)\)-net in base \(b\) if every elementary interval in base \(b\) of volume \(b^{t-m}\) contains \(\lambda b^t\) points of the sequence and no elementary interval in base \(b\) of volume \(b^{t-1}\) 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 \(\{X_n\}\) is a \((t, s)\)-sequence in base \(b\), then \(X_{kb^{m-1}+1}, \ldots, X_{kb^m+\lambda b^r}\) is a \((\lambda, t, m, s)\)-net in base \(b\) for integers \(k \geq 0\) and \(1 \leq \lambda < b\).

Sobol’ (1967) describes how to construct \((t, m, s)\)-nets and \((t, s)\)-sequences in base 2. Faure (1982) provides a construction of the \((0, m, s)\)-net and \((0, s)\)-sequence in a prime base greater than or equal to \(s\). Niederreiter (1987) 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 \(s\).
We now illustrate how the uniformity of a sequence is maintained when the sequence satisfies the net property by considering \((0, s)\)-sequences in base \(b\). We use the construction due to Faure (1982) to generate the required sequence. Details of the generation algorithm can be found in Joy et al. (1996) and Fox (1986). 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

\[ X_{kh^n+1}, \ldots, X_{(k+1)b^n} \]

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 us consider \(k = 8\) and \(m = 3\) in sequence (1). The resulting sequence is therefore a \((0, 3, 2)\)-net in base 2 with \(2^3 = 8\) elements. The theory of nets guarantees 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

\[ \left[ \frac{a_1}{2^{d_1}}, \frac{(a_1 + 1)}{2^{d_1}} \right] \times \left[ \frac{a_2}{2^{d_2}}, \frac{(a_2 + 1)}{2^{d_2}} \right] \]

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 (2) 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 \((0, 3), (3, 0), (1, 2), (2, 1)\), 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 \(b\). Fig. 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 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 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 Fig. 2. The 8 points in the first set are denoted by the “●”, while the subsequent 8 points are denoted by “×”. 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 (1)). 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 Fig. 2.

The above phenomenon can be explained as follows: consider the lower right panel of Fig. 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 exactly one point, a $(0, 4, 2)$-net in base 2 is formed as shown in Fig. 2.
A similar phenomenon holds when the dimension of the sequence is greater than two. 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, Fig. 3 plots the first and second coordinates of a $(0, 4, 7)$-net in base 7. These points appear to be uniformly dispersed throughout the unit-square. Fig. 4 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.
Fig. 3. Orthogonal projection of a $(0, 4, 7)$-net.

Fig. 4. Orthogonal projection of a seven-dimension random points.

In Fig. 5 we plot nine pairs of the orthogonal projections of $(0, 3, 19)$-net in base 19. These nine pairs were selected at random. By merely increasing the dimensions from 7 to 19, the orthogonal projection reveals an interesting characteristic of nets. The graph clearly suggests that undesirable correlation exists between these points. Such patterns have also been pointed out by Morokoff...
and Caflisch (1994) and Boyle et al. (1997). As argued by Morokoff and Caflisch (1994), the phenomenon is due to the large prime base used to generate \((0,m,s)\)-nets and can be explained by Fig. 6. The grid shows subsets of the elementary intervals with volume \(19^{-3}\) for the subsets of the points in the first panel of Fig. 5. 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 Fig. 6. 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 Fig. 5. Such patterns, however, are not unique to the particular construction due to Faure. Braaten and Weller (1979) pointed out the existence of such problems in Halton
sequences and Morokoff and Caflisch (1994) examined this phenomenon in various types of low discrepancy sequences.

3. Randomization

In this section we describe the scrambled (or randomized) \((t,m,s)\)-nets or \((t,s)\)-sequences proposed by Owen (1996). 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 (1976) first introduced the technique in the context of number-theoretic methods. Braaten and Weller (1979) randomly permute the Halton sequence. Joe (1990) randomizes lattice rules. Faure (1992) provides an optimal permutation for a one-dimensional low discrepancy sequence.

Our interpretation of Owen’s randomization technique may be described as follows: Suppose \(\{A\} \) is a \((t,m,s)\)-net or a \((t,s)\)-sequence in base \(b\). Let \(A_n = (A_{n1}, A_{n2}, \ldots, A_{ns})\) denote the \(n\)th term in the net (or sequence). Each component of \(A_n\) 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 \( \{A\} \) is a net (or sequence) \( \{X\} \) with components \( X_n = (X_{n1}, \ldots, X_{ns}) \) defined in terms of \( a_{nij} \) as follows:

\[
X_{ni} = \sum_{j=1}^{\infty} x_{nij} b^{-j} = \frac{\pi_1(a_{n11})}{b} + \frac{\pi_{12}(a_{n12})}{b^2} + \cdots + \frac{\pi_{1s}(a_{n1s})}{b^s} + \cdots.
\]

The functions \( \pi \) are random permutation of the digits \( \{0,1, \ldots, b-1\} \) and are mutually independent with each permutation function uniformly distributed over \( b! \) permutations. The first digit \( x_{n1} \) in the base \( b \) expansion of the scrambled sequence \( X_{ni} \) is obtained by permuting \( a_{n1} \) using the randomly chosen permutation function \( \pi_1 \). The second digit \( x_{n2} \) is obtained by permuting \( a_{n2} \) using the appropriate randomly generated permutation function \( \pi_{12} \), which depends on the value of first digit, i.e., \( a_{n1} \). In general, the permutation function applied to the \( k \)th digit \( a_{nk} \) depends on the first \( k-1 \) values \( a_{nij}, j = 1, \ldots, k-1 \).

The scrambled sequences \( \{X_n\} \) defined above not only inherit the equidistribution properties of the un scrambled sequences \( \{A_n\} \), each individual point of the sequence is also uniformly distributed on \([0,1)^s\). This implies that the sample estimate \( (1/N)\sum_{n=1}^{N} f(X_n) \) is an unbiased estimator for \( \int f(X)dX \). These two properties follow from the following two Propositions of Owen (1996).

**Proposition 1.** If \( \{A_n\} \) is a \((t,m,s)\)-net in base \( b \), then \( \{X_n\} \) is a \((t,m,s)\)-net in base \( b \) with probability 1. If \( \{A_n\} \) is a \((t,s)\)-sequence in base \( b \), then \( \{X_n\} \) is a \((t,s)\)-sequence in base \( b \) with probability 1.

**Proposition 2.** Let \( \{X_n\} \) be a randomized \((t,m,s)\)-net or \((t,s)\)-sequence in base \( b \) as described in Eq. (3). Then each element \( \{X_n\} \) has the uniform distribution on \([0,1)^s\). That is, for any Lebesgue measurable \( \mathcal{G} \subseteq [0,1)^s \), \( P(X_n \in \mathcal{G}) = \lambda_s(\mathcal{G}) \), the \( s \)-dimensional Lebesgue measure of \( \mathcal{G} \).

As a corollary to Proposition 1, if \( \{A_n\} \) is a \((\lambda,t,m,s)\)-net in base \( b \), then \( \{X_n\} \) is also a \((\lambda,t,m,s)\)-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 (1997a,b) and Hickernell (1996). For instance, Owen (1997b) shows that for a sufficiently smooth integrand \( f \); i.e., the mixed partial derivative \( g(x) = \partial^s f(x) / \partial x^s \) is Lipschitz continuous which means that \( |g(x) - g(x^*)| \leq B|x - x^*|^{\beta} \) for some \( B \geq 0, \beta \in (0,1)^s \) and \( \|z\| \) is the Euclidean norm of \( z \), the variance of the randomized net is of order \( N^{-3}(\log N)^{s-1} \) as \( N = \lambda b^m \rightarrow \infty \). Thus the integration errors are of order \( N^{-3/2}(\log N)^{(s-1)/2} \) in probability, which compares favourably to the rate \( N^{-1}(\log N)^{s-1} \) 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 \( N^{-3/2}(\log N)^{(s-1)/2} \) 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.

Theorem 3 of Owen (1997a) (or Theorem 1 of Owen, 1997b) 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 (1996) is the worst case over integrands of bounded variation. By averaging over permutations of a net, Hickernell has shown that the worst case over such functions attains a rate of \( N^{-1}(\log N)^{(s-1)/2} \). 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 \( N \approx s^t \). 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 dimension of the problems is considerably less than the nominal dimension. 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 et al., 1997; Joy et al., 1996; Ninomiya and Tezuka, 1996 and Paskov and Traub, 1995) even for very high-dimensional problems, which seems to contradict the findings in other fields (see e.g. Bratley et al., 1992; Fox, 1986).

3.1. Partial randomization

The implementation of Owen’s randomization procedure can be represented graphically in Fig. 7. It is convenient to describe Owen’s technique in terms of levels of randomization. In the first level of randomization, we require only one particular permutation function \( \pi_t \) to permute the first digit in the base \( b \) expansion of \( A_n \). In the second level of randomization, there are \( b \) possible choices of
permutation functions \{\pi_{i, a_{n1} = 0}, \pi_{i, a_{n1} = 1}, \ldots, \pi_{i, a_{n1} = b - 1}\}, depending on the value of \(a_{n1}\). 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 \(a_{n_k}\) is determined by the \(k - 1\) values \(\{a_{n_j}, j = 1, \ldots, k - 1\}\). 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 $k^*$, Fig. 7 indicates that scrambling a $s$-dimensional net (or sequence) requires $s(1 + b + b^2 + \cdots + b^{k^*-1}) = s(b^{k^*} - 1)/(b - 1)$ 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 (1996) and Owen and Tavella (1997) are based on low-dimensional problems (such as 6 and 10).

In this paper, 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

$$\pi_{ia_{nij}} \cdots a_{nk-1} = \pi_{ia_{nij}} \cdots a_{nk} = \pi_{ia_{nij}} \cdots a_{nk}.$$ 

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 $s(b^{k-1} - 1)/(b - 1)$ 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 dimension 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 $A_{ni}$ 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 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 in which 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 place. 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.

### 3.2. Orthogonal projections

We now examine the consequence of randomizing the nets. We apply our proposed partial randomization of level 2 to the same sets of points reported in Fig. 6. These correspond to the first panel in Fig. 5. Fig. 8 shows subsets of the elementary intervals after randomization. The randomized procedure effectively destroys the regular structure displayed in Fig. 6 while retaining the property that each elementary interval still contains a single point. Fig. 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.
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 (1997a) discusses two different approaches to estimate the variance of the estimate, $\sigma_f^2$. The simplest way of estimating $\sigma_f^2$ is by replication. A classical net (or sequence) $\{A_n, n = 1, \ldots, N\}$ is chosen and is scrambled independently $r$ times using $r$ independent sets of permutation functions $\pi$ so that the $r$ estimates of the underlying value can be used to estimate $\sigma_f^2$. Suppose $\hat{f}_j$ denotes the resulting estimate from the $j$th set of the independent permutation functions, then the unbiased estimate of $\sigma_f^2$ is given by

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

where $\bar{f} = \sum_{j=1}^{r} \hat{f}_j / r$, while the unbiased estimate of the pooled estimate $\bar{f}$ is given by

$$\hat{\sigma}_f^2 = \frac{\hat{\sigma}_f^2}{r} = \frac{1}{r(r - 1)} \sum_{j=1}^{r} (\hat{f}_j - \bar{f})^2.$$

Using the estimated $\hat{\sigma}_f^2$, 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 $\hat{\sigma}_I^2$ (or $\hat{\sigma}_F^2$) 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 $\{A_n, n = 1, \ldots, b^m\}$. Suppose the $(0, m, s)$-net is divided equally into $b$ subsets of points as follows:

$$
\underbrace{A_0, A_{b^m-1}, A_{b^m+1}, \ldots, A_{2b^m-1}, \ldots, A_{(b-1)b^m+1}, \ldots, A_{b^m}}_{(0, m-1, s)\text{-net}}
$$

so that each subset (or netlet) becomes a $(0, m-1, s)$-net with $b^{m-1}$ 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 us now denote $\hat{f}_j$ as the estimate obtained from the $j$th internal replicate of $(0, m - 1, s)$-net. Then the sample variance of the estimate $\hat{f}$ can be obtained from (5) 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 $\hat{\sigma}_f^2$. Normally, $\lambda$ is 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 $\hat{\sigma}_f^2$. 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 $\mathcal{O}(\log(rN)^{-1/2}/rN)$. 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. As a result, the variance estimated using $r$ netlets overstates the true sampling variance. This implies that the confidence statement constructed from this variance estimate is conservative. This intuition is misleading for some functions, but it does hold for those that satisfy the conditions in Owen (1997a, Remark 5). The numerical calculations performed in Section 6 are also consistent with the above argument that the variance estimates are conservative.

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 solution for any finite dimension. One of the simplest such problems in finance is the European average option where the average is taken to be the geometric average. The payoff of the call option at maturity can be represented by

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

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 a 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 also 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.

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.

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 analyse 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 Section 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, \(f_j\), is computed from the \(s\) independent standardized normal variates \(\{e_1, \ldots, e_s\}\) while the second estimate, \(f^A_j\), is obtained from \(\{-e_1, \ldots, -e_s\}\). The overall estimate of the option price for this particular simulation trial is given by the average of these two estimates. Hence,

\[
f_j = \frac{f_j + f^A_j}{2}
\]

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 et al. (1992).

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.

Suppose \(\{\pi_{a_{m-1}}, \ldots, \pi_{a_1, \ldots, a_{m-1}}\}\) denotes the randomly generated permutation functions for randomized technique \(p\) where \(p \in \{\text{partial-I, partial-II, partial-}k^*\}\). By construction, we have

\[
\pi_{id_{m-1}}^{\text{partial-II}} = \pi_{id_{m-1}}^{\text{partial-II}}
\]

for \(2 \leq j < k^*\) and

\[
\pi_{id_{m-1}}^{\text{partial-I}} = \pi_{id_{m-1}}^{\text{partial-I}}
\]
for \( 1 \leq j < k^* \). 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

\[
\pi_{i_{\text{partial-II}}} = \pi_{i_{\text{partial-k^*}}}.
\]

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.

6.1.1. 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.

Fig. 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.

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 Fig. 10 where each option contract uses a different (and independent) scrambled net. Different scrambled nets are obtained by using independent sets 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
6.1.2. Part B

We now proceed to the second part where we analyse 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 construct confidence intervals around the estimated option value, hence providing a practical error bound where it would not be possible under the classical LD methods.

full randomization technique seems to be consistently better than the other two scrambled nets.
To facilitate our hypothesis testing, we consider a 10-dimensional at-the-money option contract using the parameter values given in Section 5. For each randomization technique, a single large scrambled net consisting of $1000 \times 11^3$ 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 1000 option estimates estimating the same underlying value. This enables us to use 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. This is only a small sample of the statistical tools that can be used to test the normality assumption. See D’Agostino and Stephens (1986) for comprehensive treatment of this topic.

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

$$
E[f_{K(j)}] = \mu + \sigma \gamma_j
$$

where $\gamma_j \approx \Phi^{-1}[(j - 3/8)/(r + 1/4)]$ and $\Phi^{-1}$ is the inverse function of the standard normal cumulative density function. From (6), it follows that if $f_1, \ldots, f_r$ are i.i.d. $N(\mu, \sigma)$, a plot of $f_{K(j)}$ against $\gamma_j$ would yield approximately a straight line. Such a plot is known as the normal probability plot.

Fig. 11 provides the required plots for the three types of randomization techniques based on $r = 1000$ option estimates. These plots support the normality 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

$$
\rho_{QQ} = \frac{\sum_{j=1}^r (f_{K(j)} - \bar{f})(\gamma_{(j)} - \bar{\gamma})}{\sqrt{\sum_{j=1}^r (f_{K(j)} - \bar{f})^2} \sqrt{\sum_{j=1}^r (\gamma_{(j)} - \bar{\gamma})^2}}
$$

where $\bar{f} = \sum_{j=1}^r f_{K(j)}/r$, $\bar{\gamma} = \sum_{j=1}^r \gamma_{(j)}/r$ and $r = 1000$ in this example. This statistic measures the straightness of the rankit plot and provides 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–Shenton (Bowman and Shenton, 1975) test for normality.

The sample skewness can be estimated as

$$
\text{skewness} = \frac{\sum_{j=1}^r (f_{K(j)} - \bar{f})^3}{\bar{\sigma}^3}/r
$$
and the sample kurtosis as

\[ \text{kurtosis} = \frac{\sum_{j=1}^{r} (\hat{f}_j - \bar{f})^4}{\hat{\sigma}^4}, \]

where \( \hat{\sigma} \) is the sample standard deviation.

A statistic which tests the symmetrization and the tail of the population can be constructed as follows:

\[ \text{Bowman-Shenton} = r \left[ \frac{(\text{skewness})^2}{6} + \frac{(\text{kurtosis} - 3)^2}{24} \right]. \quad (8) \]

Under the null hypothesis that the population distribution is normal, this test statistic converges to a chi-square distribution with two degrees of freedom as
Table 1
Test statistics based on 1000 internal replicates

<table>
<thead>
<tr>
<th>Test statistic</th>
<th>Partial-(k^*)</th>
<th>Partial-II</th>
<th>Partial-I</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\rho_{QQ})</td>
<td>0.9994</td>
<td>0.9995</td>
<td>0.9995</td>
</tr>
<tr>
<td>Skewness</td>
<td>-0.0001</td>
<td>-0.0211</td>
<td>0.0747</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.9150</td>
<td>2.8300</td>
<td>2.8581</td>
</tr>
<tr>
<td>Bowman–Shenton</td>
<td>0.3014</td>
<td>1.2781</td>
<td>1.7696</td>
</tr>
</tbody>
</table>

the number of observations becomes very large. The null hypothesis is rejected for large values of the test statistic.

Table 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. Moreover, this deterioration is not sufficient for us to reject the normal hypothesis. For instance, the critical point for the Bowman–Shenton statistic at the 10% significance level with \(N = 800\) sample size is 4.32 and \(N = \infty\) is 4.61 (chi-square distribution with two degrees of freedom). Since the Bowman–Shenton test 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.

6.1.3. Part C

We now consider the estimation of the standard errors based on the at-the-money option example used in Part B. For each scrambling method, a \((\lambda, 0, 3, 10)\)-net in base 11 is generated so that \(\lambda\) 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 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 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.
Table 2
Comparison of the average standard errors and the percentage of the violations based on at-the-money option

<table>
<thead>
<tr>
<th>λ</th>
<th>Std. error</th>
<th>%</th>
<th>Ratio</th>
<th>%</th>
<th>Ratio</th>
<th>%</th>
<th>Ratio</th>
<th>%</th>
<th>Ratio</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.145</td>
<td>6.0</td>
<td>1.8</td>
<td>5.4</td>
<td>9.7</td>
<td>2.0</td>
<td>9.0</td>
<td>1.5</td>
<td>8.5</td>
<td>2.0</td>
</tr>
<tr>
<td>8</td>
<td>0.136</td>
<td>5.9</td>
<td>1.8</td>
<td>5.8</td>
<td>9.6</td>
<td>1.7</td>
<td>9.0</td>
<td>0.9</td>
<td>8.5</td>
<td>1.6</td>
</tr>
<tr>
<td>9</td>
<td>0.129</td>
<td>4.9</td>
<td>1.8</td>
<td>5.0</td>
<td>9.7</td>
<td>1.0</td>
<td>9.0</td>
<td>0.6</td>
<td>8.5</td>
<td>1.1</td>
</tr>
<tr>
<td>10</td>
<td>0.123</td>
<td>5.5</td>
<td>1.8</td>
<td>4.2</td>
<td>9.7</td>
<td>1.1</td>
<td>9.1</td>
<td>0.4</td>
<td>8.6</td>
<td>0.3</td>
</tr>
<tr>
<td>11</td>
<td>0.118</td>
<td>4.8</td>
<td>1.8</td>
<td>4.6</td>
<td>9.7</td>
<td>0.7</td>
<td>9.1</td>
<td>0.4</td>
<td>8.5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

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 $\lambda$. 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 result in a slight decrease in the efficiency of the underlying method. For instance, Table 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 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 \( \lambda \) 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 not to 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 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.

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 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.
Table 3
Comparison of the average standard errors and the percentage of the violations using 1000 randomly generated option contracts

<table>
<thead>
<tr>
<th>Monte Carlo</th>
<th>Randomized ($\lambda$, 0, 3, 10)-nets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Crude</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Std. error</td>
</tr>
<tr>
<td>7</td>
<td>0.153</td>
</tr>
<tr>
<td>8</td>
<td>0.144</td>
</tr>
<tr>
<td>9</td>
<td>0.136</td>
</tr>
<tr>
<td>10</td>
<td>0.130</td>
</tr>
<tr>
<td>11</td>
<td>0.125</td>
</tr>
</tbody>
</table>

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.

6.2.1. 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 5. Three types of LD sequences – classical $(0, s)$-sequences in base $b$ using the generation algorithm by Faure (1982) 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 integers $\lambda$ and $m$ satisfying $10,000 \leq N \leq 200,000$. The results (in log-log scale) are plotted in Fig. 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 functions.
Similar to Part A of the last subsection, we repeat the above calculations so that each option contract uses an independent scrambled sequence generated from an independent set of permutation functions. 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 Fig. 13. For comparison, the results from Monte Carlo methods with and without antithetic sampling are also illustrated.

Comparing Figs. 12 and 13, we see that using different independent sets of sequences have substantially 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 \( N \approx s^\alpha \), which is a lot larger than the number of points used in our studies. However, as discussed in Section 3, the benefits of the LD methods can appear even for \( N \ll s^\alpha \). 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 Fig. 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.

Fig. 13. Relative efficiency with increasing dimensions (each option contract uses different 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 this point any further increments in \(N\) actually increase 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 leads to slightly higher integration errors. This explains the small RMSE observed at \(\log 367^2 \approx 11.8\) at the bottom panel of Fig. 13 since \(b = 367\) is the required prime base used to generate \((0, 365)\)-sequences.

### 6.2.2. 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 the 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

Comparison of the average standard errors using 1000 randomly generated sets of option contracts \((s = 100)\)

<table>
<thead>
<tr>
<th>(\lambda)</th>
<th>Monte Carlo</th>
<th>Randomized ((\lambda, 0, 3, 100))-Nets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Crude</td>
<td>Antithetic %</td>
</tr>
<tr>
<td></td>
<td>Std. Error</td>
<td>Ratio %</td>
</tr>
<tr>
<td>7</td>
<td>0.052</td>
<td>5.9</td>
</tr>
<tr>
<td>8</td>
<td>0.049</td>
<td>6.0</td>
</tr>
<tr>
<td>9</td>
<td>0.047</td>
<td>4.9</td>
</tr>
<tr>
<td>10</td>
<td>0.045</td>
<td>4.5</td>
</tr>
<tr>
<td>11</td>
<td>0.043</td>
<td>5.3</td>
</tr>
</tbody>
</table>
Table 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 $1.4( \approx 2/\sqrt{2})$, 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.

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 paper 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 (1996) 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 paper is that it is possible to modify the randomization technique due to Owen so that it can be applied effectively to high dimensional problems.

Acknowledgements

The authors are grateful to Adam Kolkiewicz, the referees of this journal and the Editor Michael Selby for constructive comments on earlier versions. Earlier versions of this paper were presented at the 1997 AFIR Conference in Australia and the 1997 Econometric Society Meetings in Chicago. The authors are grateful for the comments received. Ken Seng Tan acknowledges research support from the Natural Sciences and Engineering Research Council of Canada and the Society of Actuaries. Phelim P. Boyle thanks the Social Science and Humanities Research Council of Canada for research support.
References

Bowman, K.O., Shenton, L.R., 1975. Omnibus test contours for departures from normality based on $\sqrt{b_1}$ and $b_2$. Biometrika 62 (2), 243–250.


