1 Quasi-Monte Carlo Methods

Quasi-Monte Carlo simulation is often described as the deterministic version of the Monte Carlo simulation. It uses the uniformly distributed modulo 1 sequences (other names are low-discrepancy sequences, quasi-Monte Carlo sequences) to simulate the problem, instead of pseudorandom numbers used by Monte Carlo.

Definition 1 *A sequence $x_1, x_2, ...$ in $[0,1)$ is said to be uniformly distributed modulo 1 (u.d. mod 1) if*

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} 1_{[a,b)}(x_n) = b - a$$

for any $0 \leq a < b \leq 1$.

This definition is generalized to higher dimensions in the following way. Put $I = [0,1)$ and $I^s = [0,1)^s$. For $a, b \in I^s$ such that $a_i \leq b_i, i = 1, ..., s$, define the $s-$ dimensional interval $[a,b)$ as $\prod_{i=1}^{s} [a_i, b_i)$.

Definition 2 *A sequence $x_1, x_2, ...$ in I^s is said to be u.d. mod 1 if*

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} 1_{[a,b)}(x_n) = \prod_{i=1}^{s} (b_i - a_i)$$

for any $0 \leq a < b \leq 1$.

Theorem 1 *A sequence $x_1, x_2, ...$ in I^s is u.d. mod 1 iff*

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} f(x_n) = \int_{I^s} f(x)dx$$

for all Riemann integrable functions f on I^s.

Example 1 *The van der Corput sequence and its generalization, the Halton sequence, is a well-known low-discrepancy sequence. The van der Corput sequence $\phi_2(n), n = 1, 2, ...$ in base 2 is defined as follows:*

n	*base 2*	*inversion*	*van der Corput seq*
0	0	0.0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
...			

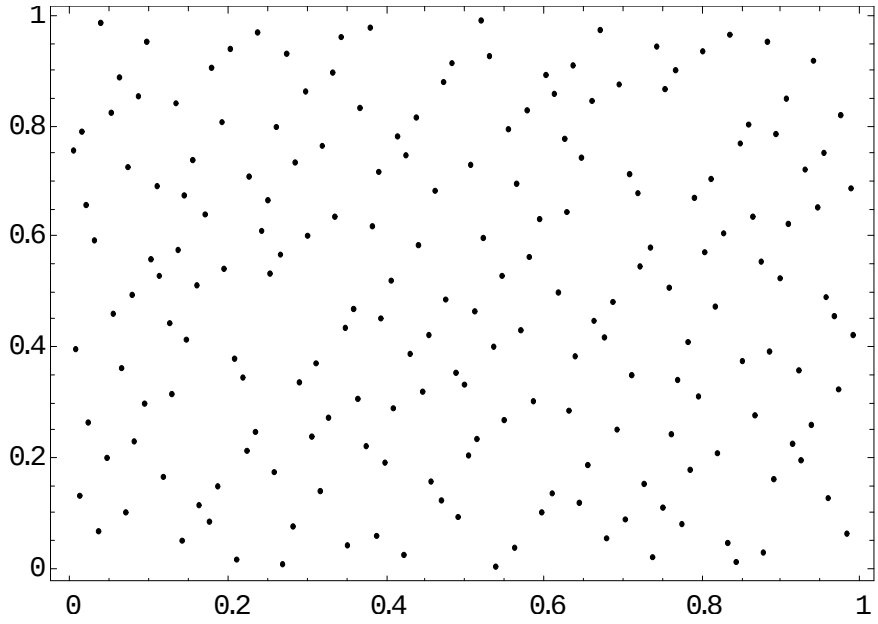
Formally, for any base b, if

$$n = (a_k \cdots \ a_1 a_0)_b = a_0 + a_1 b + ... + a_k b^k$$

then

$$\phi_b(n) = (.a_0 a_1 \cdots a_k)_b = \frac{a_0}{b} + \frac{a_1}{b^2} + ... + \frac{a_k}{b^{k+1}} \tag{1}$$

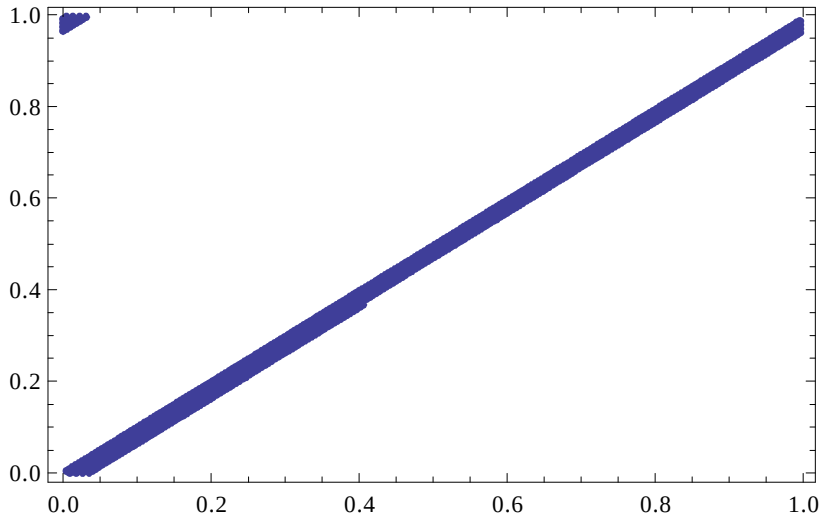
The Halton sequence is a generalization of the van der Corput sequence to higher dimensions. The $s-$ dimensional Halton sequence in the bases $b_1, ..., b_s$ is defined as $(\phi_{b_1}(n), ..., \phi_{b_s}(n))_{n=1}^{\infty}$. The van der Corput sequences are u.d. mod 1 sequences for any base. The Halton sequence is u.d. mod 1 if its bases $b_1, ..., b_s$ are relatively prime. Usually, these numbers are chosen simply as prime numbers, i.e., b_n is taken as the nth prime number. The following is a plot of 2-dimensional Halton vectors when the bases are 2 and 3.



Halton sequence in bases 2 and 3.

Remark 1 *The Halton sequence, however, exhibits poor uniformity in high dimensions. The following is a plot of the first 1000 Halton vectors in 2-*

dimensions, when the bases are the 49th and 50th prime numbers; 227, 229.



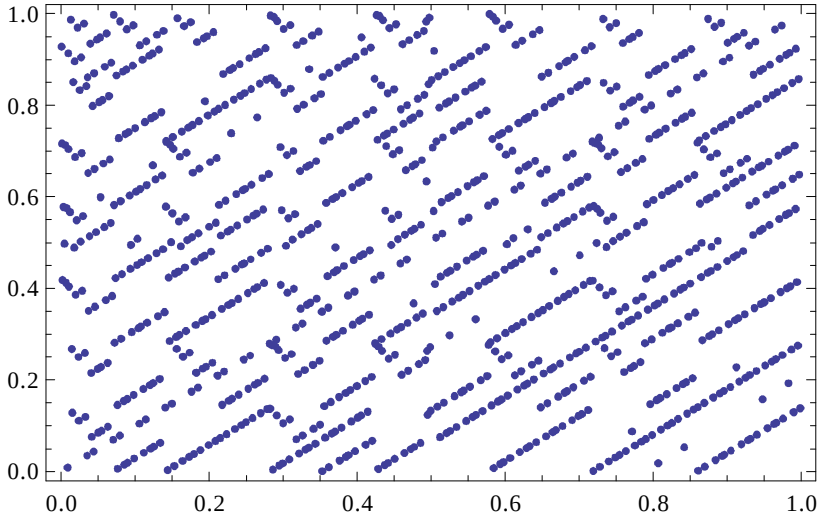
Halton sequence in bases 227 and 229.

Starting with Braaten and Weller in 1979, several researchers introduced permutations to scramble the digits of the van der Corput sequences that make up the Halton sequence, in order to improve the uniformity of the Halton sequence. These sequences are called scrambled Halton, or generalized Halton sequences. The scrambled van der Corput sequence generalizes equation (1) as

$$\phi_b(n) = \frac{\sigma(a_0)}{b} + \frac{\sigma(a_1)}{b^2} + ... + \frac{\sigma(a_k)}{b^{k+1}}$$

where σ is a permutation on the digit set $\{0, ..., b-1\}$. The scrambled Halton sequence is obtained from scrambled van der Corput sequences in the usual way. There are several permutations published in the literature; see Ökten & Shah [3] for details. One of these permutations is by Faure, and the corresponding scrambled van der Corput sequence for the same bases we considered before, 227

and 229, produce the following plot:



Faure scrambled Halton sequence in bases 227 and 229.

Remark 2 *Other examples of u.d. mod 1 sequences are the Sobol', Faure, and Niederreiter sequences. Descriptions of the first two can be found in Glasserman. Also see Joy et. al [4] for a description of the Faure sequence (the paper is on Blackboard).*

Remark 3 *Read the Mathematica file* ***Halton and scrambled Halton sequences*** *for more.*

1.1 Numerical integration

Consider the problem of estimating

$$\theta = \int_{I^s} f(x)dx.$$

In both Monte Carlo and quasi-Monte Carlo, we estimate the integral by averages of the form:

$$\theta_N = \frac{1}{N}\sum_{n=1}^{N} f(x_n).$$

In Monte Carlo, x_n are pseudorandom numbers, and $\theta_N \to \theta$ as $N \to \infty$ almost surely. The error $\theta_N - \theta$ is a normal random variable with zero mean and variance $\left(\int_{I^s} f^2(x)dx - \theta^2\right)/N = \left(\int_{I^s}(f(x)-\theta)^2dx\right)/N$.

In quasi-Monte Carlo, x_n come from an $s-$dimensional u.d. mod 1 sequence, and $\theta_N \to \theta$ as $N \to \infty$. Now we discuss the error of this approximation, $\theta_N - \theta$.

1.1.1 Discrepancy

Given a sequence $x_1, ..., x_N$ in $[0,1)^s$, its extreme discrepancy is

$$D_N(x_n) = \sup_{[a,b)} \left| \frac{1}{N} \sum_{n=1}^{N} 1_{[a,b)}(x_n) - \lambda_s([a,b)) \right|.$$

If the supremum is taken over all intervals of the form $[0,a)$ then we obtain the definition of star-discrepancy

$$D_N^*(x_n) = \sup_{[0,a)} \left| \frac{1}{N} \sum_{n=1}^{N} 1_{[0,a)}(x_n) - \lambda_s([0,a)) \right|.$$

Theorem 2 *For any finite set P consisting of N points,*

$$D_N^*(P) \leq D_N(P) \leq 2^s D_N^*(P).$$

Theorem 3 *$D_N(x_n) \to 0$ iff $\{x_n\}$ is a u.d. mod 1 sequence.*

Sequences with best known bounds for star-discrepancy satisfy $O((\log N)^s/N)$. (The term "low-discrepancy" sequence is in fact used for these sequences.) Different sequences have different constants multiplying the $(\log N)^s/N$ term, and whether one sequence is "better" than the other is sometimes settled by comparing these constants. However, this way of comparing sequences is not quite useful, as we will discuss later.

Theorem 4 (Koksma-Hlawka inequality) *If f has bounded variation $V(f)$ in the sense of Hardy and Krause over $[0,1]^s$, then, for any $x_1, ..., x_N \in [0,1)^s$, we have*

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_{I^s} f(x)dx \right| \leq V(f) D_N^*(x_n) \tag{2}$$

This inequality shows that the quasi-Monte Carlo error bound is $O((\log N)^s/N)$, and smaller error is possible if we can make $V(f)$ or $D_N^*(x_n)$ smaller. However, there are certain limitations of the Koksma-Hlawka inequality.

1.1.2 Drawbacks of Koksma-Hlawka inequality

- For a given f, the inequality can be loose;
- In general, $V(f)$ and $D_N^*(x_n)$ are not known exactly, and need to be numerically estimated, if we want to use the upper bound of (2) as a measure of integration error;
- Variation in the sense of Hardy and Krause is restrictive and difficult to work with.

Example 2 *A simple function like*

$$f(x,y) = \begin{cases} 1, x \geq y \\ 0, x < y \end{cases}$$

is not of finite bounded variation in the sense of Hardy and Krause in the unit square

1.1.3 Bounded variation in the sense of Hardy and Krause

$f(x) = f(x_1, ..., x_s)$ real valued function on $[0,1]^s$. P a partition of $[0,1]^s$, i.e., is a set of s finite sequences

$$P : \nu_0^{(j)}, \nu_1^{(j)}, ..., \nu_{m(j)}^{(j)} \text{ for } j = 1, ..., s$$

with $0 = \nu_0^{(j)} \leq \nu_1^{(j)} \leq ... \leq \nu_{m(j)}^{(j)} = 1$. For such a partition operators define Δ_j as the jth difference operator

$$\Delta_j f(\cdot....\cdot, \upsilon_i^{(j)}, \cdot....\cdot) = f(\cdot....\cdot, \upsilon_{i+1}^{(j)}, \cdot....\cdot) - f(\cdot....\cdot, \upsilon_i^{(j)}, \cdot....\cdot).$$

and let

$$\Delta_{j_1...j_p} = \Delta_{j_1}...\Delta_{j_p}$$

The variation of f in the sense of **Vitali** is

$$V^s(f) = \sup_P \sum_{i_1=0}^{m(1)-1} ... \sum_{i_s=0}^{m(s)-1} \left| \Delta_{1,...,s} f(\upsilon_{i_1}^{(1)}, ..., \upsilon_{i_s}^{(s)}) \right|$$

The variation of f in the sense of **Hardy and Krause** is

$$V(f) = \sum_{k=1}^{s} \sum_{1 \leq i_1 < i_2 < ... < i_k \leq s} V^k(f; i_1, ..., i_k)$$

where $V^k(f; i_1, ..., i_k)$ is the variation in the sense of Vitali of the restriction of f to the k-dimensional face

1.2 (t, m, s) nets and (t, s) sequences

Definition 3 *Let $s \geq 1, b \geq 2$ be integers. An elementary interval in base b is a subinterval of $[0,1)^s$ of the form*

$$\prod_{j=1}^{s} \left[\frac{a_j}{b^{d_j}}, \frac{a_j + 1}{b^{d_j}} \right)$$

where integers a_j, d_j satisfy $d_j \geq 0$ and $0 \leq a_j < b^{d_j}$.

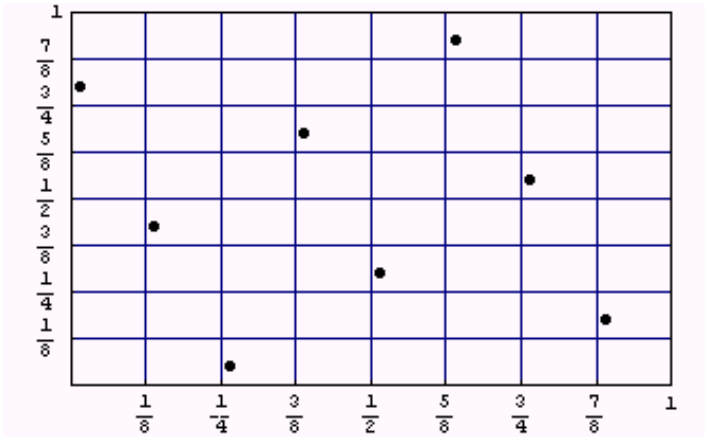


Figure 1: $(0, 3, 2)$-net in base 2

Example 3 *Elementary intervals in base $b = 3$ and $s = 1$ are:*

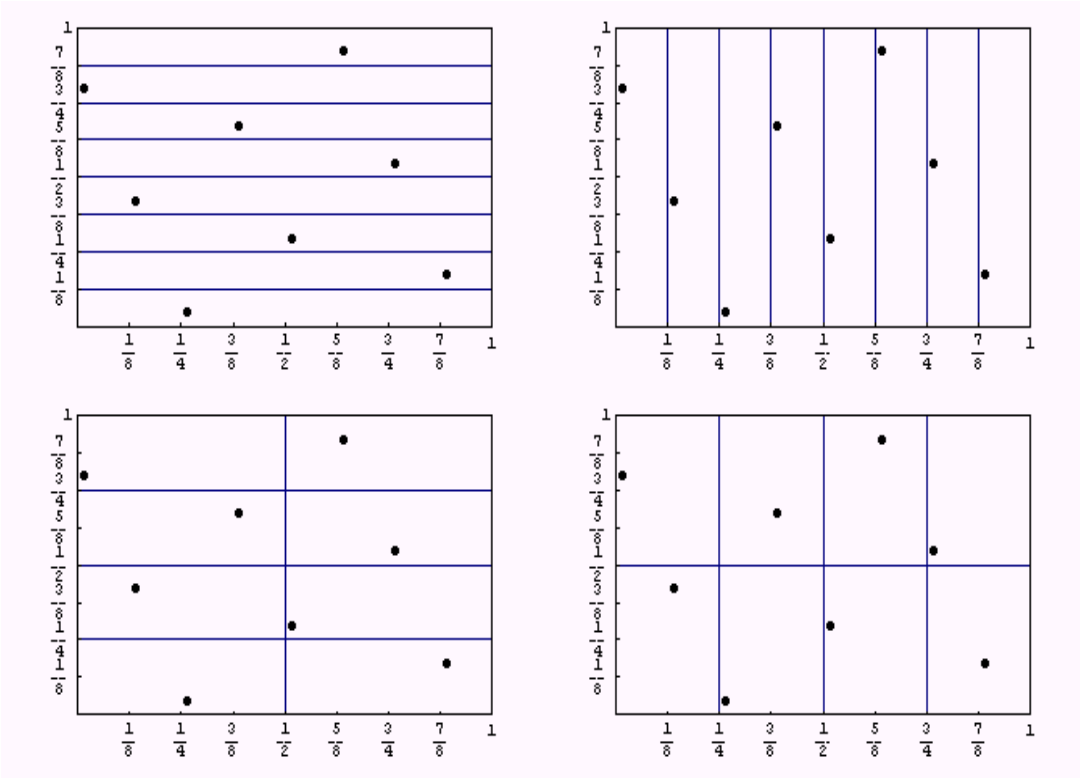
$$\begin{aligned} d_1 &= 0 \Rightarrow [0, 1) \\ d_1 &= 1 \Rightarrow [0, 1/3), [1/3, 2/3), [2/3, 1) \\ d_1 &= 2 \Rightarrow [0, 1/9), [1/9, 2/9), [2/9, 3/9), ... \\ &\cdots \end{aligned}$$

Example 4 *Elementary intervals in base $b = 2$ and $s = 2$ are:*

$$\begin{aligned} d_1 &= 0 \ \& \ d_2 = 0 \Rightarrow [0, 1) \times [0, 1) \\ d_1 &= 1 \ \& \ d_2 = 1 \Rightarrow [0, 1/2) \times [0, 1/2) \text{ and } [1/2, 1) \times [1/2, 1) \text{ and } [0, 1/2) \times [1/2, 1) \\ &\quad \text{and } [1/2, 1) \times [0, 1/2) \\ d_1 &= 1 \ \& \ d_2 = 2 \Rightarrow [0, 1/2) \times [0, 1/4) \text{ and } [0, 1/2) \times [1/4, 3/4) \ \cdots \\ &\quad \text{and } [1/2, 1) \times [0, 1/4) \text{ and } [1/2, 1) \times [1/4, 3/4) \ \cdots \end{aligned}$$

Definition 4 *For $0 \leq t \leq m$, a finite sequence of b^m points in $[0, 1)^s$ is a (t, m, s)-net in base b if every elementary interval in base b of volume b^{t-m} contains exactly b^t points of the sequence.*

Example 5 *First figure plots a $(0, 3, 2)$-net in base 2, and the next figure plots the net together with the elementary intervals of area 2^{-3} to illustrate that exactly one point (there are eight points total and eight elementary intervals, so the "right" proportion is one point for each interval) falls in every elementary interval.*



Elementary intervals of area 1/8

Definition 5 *An infinite sequence of points* $q_1, q_2, ...$ *is called a* (t, s)*-sequence in base* b *if the finite sequence* $q_{kb^m+1}, ..., q_{(k+1)b^m}$ *is a* (t, m, s)*-net in base* b *for all* $k \geq 0$ *and* $m \geq t$.

An example of a $(0, s)$ sequence is the Faure sequence. If we take a finite point set from the Faure sequence as follows $\{q_k : jb^m \leq k \leq (j+1)b^m\}$, then we obtain a $(0, m, s)-$net. Here b is the base of the sequence, which depends on the dimension s. Read the *Mathematica* file **Faure and scrambled Faure sequences**, for a description of nets and sequences, and the scrambled Faure sequence, which we will discuss next. A detailed investigation of the construction of nets and sequences is given by Niederreiter [1].

2 Randomized quasi-Monte Carlo methods

The objective is to develop a method that will enable the use of statistical methods to analyze error in the context of u.d. mod 1 sequences. The idea is as follows: Let β be a low-discrepancy sequence. *Randomize* β in order to obtain *independent* copies $\beta_1, ..., \beta_m$. If the problem is to estimate $\theta = \int_{I^s} f(x)dx$,

then each β_i gives an estimate, θ_i. If the randomization is done properly, we will estimate θ using the sample mean

$$\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m}\theta_i$$

and estimate error, simply by,

$$Var(\hat{\theta}) = \frac{\sum_{i=1}^{m}(\theta_i - \theta)^2}{m-1}.$$

Here is a more technical description. In randomized quasi-Monte Carlo, we consider a family of s-dimensional sequences β_u indexed by the random parameter u. This in turn yields a family of quadrature rules

$$Q(\beta_u) = \frac{1}{N}\sum_{n=1}^{N} f(\beta_u^{(n)})$$

and the main research problem is the study of the discrepancy of the sequence β_u, and the expected value and variance of $Q(\beta_u)$. In general, the quadrature rules indexed by the random parameter u are unbiased, i.e.,

$$E[Q(\beta_u)] = I$$

and I is estimated by the sample mean

$$\frac{Q(\beta_{u_1}) + ... + Q(\beta_{u_m})}{m}.$$

The random parameter u will typically have uniform distribution on $(0,1)^s$, and $u_1, ..., u_m$ are independent samples from u.

Some of the RQMC methods are:

1. Scrambled nets and sequences
2. Random shifting
3. Digit scrambling, and linear scrambling of nets and sequences
4. Random-start Halton sequences

For a survey of these methods and their use in computational finance see Ökten & Eastman [2]. Here I will explain the random shifting approach, which is pretty simple to implement.

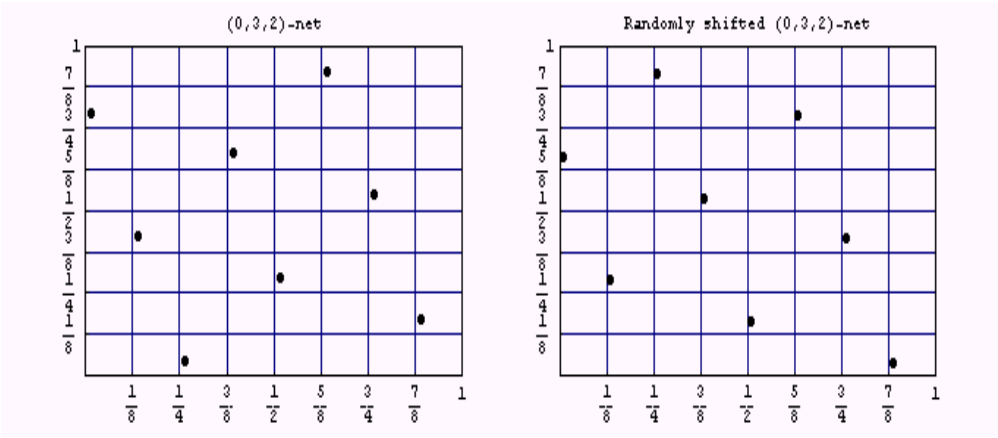


Figure 2: A (0,3,2)-net and its randomly shifted version

2.1 Random shifting

Consider the first six Halton vectors in dimension 2, using bases 2 and 3:

$$\left[\begin{array}{c} 1/2 \\ 1/3 \end{array}\right], \left[\begin{array}{c} 1/4 \\ 2/3 \end{array}\right], \left[\begin{array}{c} 3/4 \\ 1/9 \end{array}\right], \left[\begin{array}{c} 1/8 \\ 4/9 \end{array}\right], \left[\begin{array}{c} 5/8 \\ 7/9 \end{array}\right], \left[\begin{array}{c} 3/8 \\ 2/9 \end{array}\right]$$

To obtain an independent randomization of this set, generate two random numbers u_1, u_2 from $U(0,1)$, say, $u_1 = 0.7$ and $u_2 = 0.1$. Add the vector $u = \left[\begin{array}{c} u_1 \\ u_2 \end{array}\right]$ to each vector above, componentwise, and take the fractional part of the sum. For example, the first vector becomes

$$\left[\begin{array}{c} \{1/2 + 0.7\} \\ \{1/3 + 0.1\} \end{array}\right] = \left[\begin{array}{c} 0.2 \\ 0.4\bar{3} \end{array}\right]$$

(here $\{x\}$ denotes the fractional part of x). Similarly, add u to the rest of the vectors.

Here is an example, of a $(0, 3, 2)-$net and its shifted version. The shift is done by the random vector $u = (0.74, 0.62)$.

Theorem 5 *The discrepancy of a shifted low-discrepancy sequence satisfies*

1. $$D_N(\beta_u) \leq 2^s D_N(\beta),$$

   and the corresponding quadrature rules satisfy

2. $$E\left[Q(\beta_u)\right] = I,$$

3.

$$Var(Q(\beta_u)) = O\left(\frac{(\log N)^{2s}}{N^2}\right).$$

2.2 Linear Scrambled Faure sequence

The Faure sequence is an example of a *digital* (t, s) *sequence,* with $t = 0$. The digital sequences comprise a big portion of low-discrepancy sequences used in practice. Consider an s-dimensional digital sequence $q_1, q_2, ...,$ in the s-dimensional unit hypercube $[0, 1)^s$. To compute the nth term $q_n = (q_n^{(1)}, ..., q_n^{(s)})$ of the sequence, we first write n in its base p expansion: $n = a_l p^{l-1} + ... + a_2 p + a_1$. Here p is called the base of the digital sequence. In the case of the Faure sequence, the base is the smallest prime number greater than or equal to s. Let $d_n = (a_1, ..., a_l)^T$ be the (column) vector of the digits of n. Similarly, let $\phi(q_n^{(k)})$ denote the (column) vector of the digits of the real number $q_n^{(k)} \in [0, 1)$ in base p. For example, if the base p expansion of a real number is $\tilde{a}_{\tilde{l}}/p^{\tilde{l}} + ... + \tilde{a}_2/p^2 + \tilde{a}_1/p$ then $\phi(\cdot) = (\tilde{a}_1, ..., \tilde{a}_{\tilde{l}})^T$. The kth component of the nth term q_n is defined through its digit vector:

$$\phi(q_n^{(k)}) = \mathbf{C}^{(k)} d_n, \ k = 1, ..., s \tag{3}$$

where $\mathbf{C}^{(k)}$ is the kth *generator matrix* of the digital sequence, and the matrix-vector multiplication is done in modular arithmetic mod p.

The generator matrix for the Faure sequence is

$$\mathbf{C}^{(k)} = \mathbf{P}^{k-1}, \ k = 1, ..., s \tag{4}$$

where $\mathbf{P}$ is the $l \times l$ Pascal matrix[1] $\bmod p$: this is an upper triangular matrix where the ijth entry is the binomial term $\binom{j-1}{i-1} \bmod p$. For example, the 3×3 Pascal matrix is

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Powers of the Pascal matrix can be computed without matrix multiplication as follows. Define $\mathbf{P}(a)$ as the upper triangular matrix with the ijth entry equal to $a^{j-i}\binom{j-1}{i-1}$. Then, we have the following useful identity: $\mathbf{P}^k = \mathbf{P}(k)$ for any integer k (adopting the convention $0^0 = 1$, so that $\mathbf{P}^0 = \mathbf{P}(0) = \mathbf{I}$).

I now describe how to obtain randomly scrambled "copies" of the Faure sequence. Generate s lower triangular square matrices $\mathbf{L}^{(k)}, k = 1, ..., s$, whose non-diagonal entries are random integers between 0 and $p - 1$, and diagonal entries are random integers between 1 and $p - 1$. Also generate s vectors $g^{(k)}$ whose entries are random integers between 0 and $p - 1$. The dimension of the square matrices and the vectors is equal to l; the dimension of the vector d_n.

[1] In other literature such as combinatorics, the Pascal matrix is the transpose of the matrix defined here, without the $\bmod p$ reduction.

A scrambled Faure sequence is obtained by replacing the generator matrices of the Faure sequence (4) by

$$\mathbf{C}^{(k)} = \mathbf{L}^{(k)}\mathbf{P}^{k-1},\ k = 1, ..., s \tag{5}$$

and replacing (3) by

$$\phi(q_n^{(k)}) = \mathbf{C}^{(k)}d_n + g^{(k)},\ k = 1, ..., s. \tag{6}$$

As before, all operations are done in mod p. This scrambling method is called linear scrambling by Matoušek [6], who discusses this as well as other scrambling techniques. A detailed discussion of scrambled sequences and their implementation is also given by Hong & Hickernell [5].

The Mathematica file **Faure and scrambled Faure sequences** has an implementation of these sequences. I take the vector $g^{(k)}$ to be 0 in that implementation.

3 Simulating the GBM using quasi-Monte Carlo method

We want to generate N price paths $S^1, S^2, ..., S^N$, where each price path $S^j = \left(S^j(t_1), ..., S^j(t_n)\right)$ is a realization of $GBM(\mu, \sigma^2)$ at $0 = t_0 < t_1 < ... < t_n$, i.e.,

$$S^j(t_i) = S^j(t_{i-1}) \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t_i - t_{i-1}) + \sigma\sqrt{t_i - t_{i-1}}Z_{i,j}\right) \tag{7}$$

where Z_{ij} are i.i.d. standard normals and $i = 1, ..., n$ and $j = 1, ..., N$.

The number of time steps is n. This determines the dimension of the low-discrepancy sequence. Let $\phi(j)$, $j = 1, 2, ...$, be an $n-$ dimensional QMC sequence:

$$\phi(j) = (\phi_1(j), \phi_2(j), ..., \phi_n(j)), j = 1, 2, ...$$

Let's assume that to generate the normal distribution, we use an algorithm that takes one uniform random number and outputs one normal random number (e.g., the inverse transformation method), and let's denote this transformation by $M(u) = z$. Here is the algorithm for QMC generation of $GBM(\mu, \sigma^2)$:

for $j = 1, ..., N$
  for $i = 1, ..., n$
  $Z_{ij} = M(\phi_i(j))$
  $S^j(t_i) = S^j(t_{i-1}) \exp\left(\left(\mu - \sigma^2/2\right)(t_i - t_{i-1}) + \sigma\sqrt{t_i - t_{i-1}}Z_{i,j}\right)$
  end
end

4 Problems

1. Write a computer code for the Halton sequence. The input should be the dimension s, and n, and the output should be the nth Halton vector where

the bases are the first s prime numbers. Then, generate 1000 numbers from the standard normal distribution using the Moro's method and the Box-Muller method. What is the constructive dimension for each method? What is the dimension of the sequence you use for each method? Draw two histograms for each data set. Do they "look" right?

2. Write a computer code for the Faure sequence, and the scrambled Faure sequence (as explained in the Mathematica example).

   (a) Use scrambled Faure sequences to obtain 40 estimates for the price of a European call option with parameters: T = expiry = 1, K = exercise price = 50; $r = 0.1; \sigma = 0.3; S_0 = 50$. For each estimate, use $N = 10,000$ price paths. To price the option, you will simulate the GBM with $\mu = r$, and using 10 time steps $t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, ..., t_{10} = 10\Delta t = T$. Use the Box-Muller method in generating the stock price paths. (Note that the European call option can actually be estimated by directly generating the price at expiry, so there is no need to generate the complete price path. Nevertheless, this problem prepares you for the pricing of path dependent options.)

   (b) Do part (a) using Moro's method.

   (c) Compute the Black-Scholes-Merton price of the option, i.e., the exact option price.

   (d) Compare the accuracy of the estimates you obtained in parts (a) and (b) as follows: Each set of 40 estimates should be distributed according to the normal distribution whose mean is the true option price you found in part (c), and an unknown variance. Apply the Anderson-Darling statistic to test the data you obtained in parts (a) and (b), for this hypothesis. Consult Stephens' paper on Anderson-Darling statistic for critical points. Which one of the cases discussed in the paper applies to this problem? What are your conclusions for each data set?

   (e) Read the paper by Joy, Boyle, and Tan (on Blackboard), in particular, the footnote on page 930. Do you agree with the authors that the Box-Muller method is worse than Moro's method?

References

[1] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia, 1992.

[2] G. Ökten and W. Eastman. Randomized quasi-Monte Carlo methods in pricing securities. *Journal of Economic Dynamics & Control,* 28:2399–2426, 2004.

[3] G. Ökten, M. Shah, Y. Goncharov. Randomly permuted Halton sequences. Technical report.

[4] C. Joy, P. P. Boyle, K. S. Tan, Quasi-Monte Carlo Methods in Numerical Finance, *Management Science*, **42**, 926-938 (1996).

[5] H. S. Hong and F. J. Hickernell, Algorithm 823: Implementing scrambled digital sequences, *ACM Transactions on Mathematical Software* **29**, 95-109, (2003).

[6] J. Matoušek, On the L_2-Discrepancy for Anchored Boxes, *Journal of Complexity* **14**, 527-556, (1998).