Chapter 6

Quasi- Monte Carlo Multiple Integration

Introduction

In some sense, this chapter fits within Chapter 4 on variance reduction; in some sense it is stratification run wild. Quasi-Monte Carlo methods are purely deterministic, numerical analytic methods in the sense that they do not even attempt to emulate the behaviour of *independent* uniform random variables, but rather cover the space in d dimensions with fewer gaps than independent random variables would normally admit. Although these methods are particularly when evaluating integrals in moderate dimensions, we return briefly to the problem of evaluating a one-dimension integral of the form

$$\int_0^1 f(x)dx.$$

The simplest numerical approximation to this integral consists of choosing a point x_j in the interval $[\frac{j}{N}, \frac{j+1}{N}], j = 0, 1, ..., N-1$, perhaps the midpoint of the interval, and then evaluating the average

$$\frac{1}{N}\sum_{j=0}^{N-1} f(x_j). \tag{6.1}$$

If the function f has one continuous derivative, such a numerical method with N equally or approximately equally spaced points will have bias that approaches 0 at the rate $1/N$ because, putting $M = \sup\{|f'(z)|; 0 < z < 1\}$,

$$\int_{j/N}^{(j+1)/N} f(x)dx - \frac{1}{N}f(x_j) \leq \frac{1}{N^2}M \tag{6.2}$$

and so summing both sides over j gives

$$|\int_0^1 f(x)dx - \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)| \leq \frac{1}{N}M.$$

We will refer to the error in the numerical integral in this case

$$\varepsilon_N = |\int_0^1 f(x)dx - \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)|$$

as $O(N^{-1})$ which means that the sequence of errors ε_N satisfies

$$\limsup_{N\to\infty} N^{-1}\varepsilon_N < \infty$$

or intuitively that the errors are bounded by a constant times N^{-1}.

If the function f is known to have bounded derivatives of second or third order, then integrals can be approximated to an even higher degree of precision. For example various numerical quadrature formulae permit approximating an integral of the form $\int_0^1 f(x)w(x)dx$ with a weighted average of N points

$$\sum_{j=1}^{N} w_j f(x_j) \tag{6.3}$$

in such a way that if $f(x)$ is a polynomial of degree $2N-1$ or less, the approximation is *exact*. Here the function $w(x)$ is typically some density such as the uniform, exponential or normal density and the optimal placement of the points x_j as well as the weights w_j depends on $w(x)$. Of course a smooth function can be closely approximated with a polynomial of high degree and so numerical quadrature formulae of the form (6.3) permit approximating a one-dimension integral arbitrarily closely provided that the function is sufficiently smooth, i.e. it has bounded derivatives of sufficiently high order. We should note that in this case, the weights w_j and the points x_j are both deterministic. By contrast, the Monte Carlo integral

$$\widehat{\theta}_{MC} = \frac{1}{N}\sum_{i=1}^{N} f(U_i)$$

with N points places these points at random or pseudo-random locations, has zero bias but the standard deviation of the estimator $\sqrt{var(\widehat{\theta}_{MC})}$ is a constant multiple of $1/\sqrt{N}$. The Central Limit theorem assures us that

$$N^{1/2}(\widehat{\theta}_{MC} - \int_0^1 f(x)dx)$$

converges to a normal distribution which means that the error is order (in probability) $N^{-1/2}$. Note that there is a change in our measure of the size of an error, since only the variance or standard deviation of a given term in the sequence of errors is bounded, not the whole sequence of errors ε_N. In particular if a pseudo-random estimator $\widehat{\theta}$ satisfies

$$E(\widehat{\theta} - \int_0^1 f(x)dx)^2 = O(N^{-2k})$$

then we say that the error is $O_P(N^{-k})$ where O_P denotes "order in probability". This is clearly a weaker notion than $O(N^{-k})$. Even the simplest numerical integral (6.1) has a faster rate of convergence then that of the Monte Carlo integral with or without use of the variance reduction techniques of Chapter 4. This is a large part of the reason numerical integration is usually preferred to Monte Carlo methods in one dimension, at least for smooth functions, but it also indicates that for regular integrands, there is room for improvement over Monte Carlo in higher dimensions as well.

The situation changes in 2 dimensions. Suppose we wish to distribute N points over a uniform lattice in some region such as the unit square. One possible placement is to points of the form

$$\left(\frac{i}{\sqrt{N}}, \frac{j}{\sqrt{N}}\right), i, j = 1, 2, ... \sqrt{N}$$

assuming for convenience of notation that $\sqrt{N}$ is integer. The distance between adjacent points is of order $1/\sqrt{N}$ and by an argument akin to (6.2), the bias in a numerical integral is order $1/\sqrt{N}$. This is the now same order as the standard deviation of a Monte Carlo integral, indicating that the latter is already, in two dimensions, competitive. When the dimension $s \geq 3$, a similar calculation shows that the standard deviation of the Monte-Carlo method is strictly smaller order than the error of a numerical integral with weights at lattice points. Essentially, the placement of points on a lattice for evaluating a $d-$dimensional integral is far from optimal when $d \geq 2$. Indeed various deterministic alternatives called quasi-random samples provide substantially better estimators especially for smooth functions of several variables. Quasi-random samples are analogous to equally spaced points in one dimension and are discussed at length by Niederreiter (1978), where it is shown that for sufficiently smooth functions, one can achieve rates of convergence close to the rate $1/N$ for the one-dimensional case.

We have seen a number of methods designed to reduce the dimensionality of the problem. Perhaps the most important of these is conditioning, which can reduce an $d-$dimensional integral to a one-dimensional one. In the multidimensional case, variance reduction has an increased importance because of the high variability induced by the dimensionality of crude methods. The other variance reduction techniques such as regression and stratification carry over to the multivariable problem with little change, except for the increased complexity of determining a reasonable stratification in such problems.

Errors in numerical Integration

We consider the problem of numerical integration in d dimensions. For $d = 1$ classical integration methods, like the trapezoidal rule, are weighted averages of the value of the function at equally spaced points;

$$\int_0^1 f(u)du \approx \sum_{n=0}^{m} w_n f(\frac{n}{m}), \tag{6.4}$$

where $w_0 = w_m = 1/(2m)$, and $w_n = 1/m$ for $1 \leq n \leq m-1$. The trapezoidal rule is exact for any function that is linear (or piecewise linear between grid-points) and so we can assess the error of integration by using a linear approximation through the points $(\frac{j}{m}, f(\frac{j}{m}))$ and $(\frac{j+1}{m}, f(\frac{j+1}{m}))$. Assume

$$\frac{j}{m} < x < \frac{j+1}{m}.$$

If the function has a continuous second derivative, we have by Taylor's Theorem that the difference between the function and its linear interpolant is of order $O(x-\frac{j}{m})^2$, i.e.

$$f(x) = f(\frac{j}{m}) + (x-\frac{j}{m})m[f(\frac{j+1}{m}) - f(\frac{j}{m})] + O(x-\frac{j}{m})^2.$$

Integrating both sides between $\frac{j}{m}$ and $\frac{j+1}{m}$, notice that

$$\int_{j/m}^{(j+1)/m} \{f(\frac{j}{m}) + (x-\frac{j}{m})m[f(\frac{j+1}{m}) - f(\frac{j}{m})]\}dx = \frac{f(\frac{j+1}{m}) + f(\frac{j}{m})}{2m}$$

is the area of the trapezoid and the error in the approximation is

$$O(\int_{j/m}^{(j+1)/m} (x-\frac{j}{m})^2) = O(m^{-3}).$$

Adding these errors of approximation over the m trapezoids gives $O(m^{-2})$. Consequently, the error in the trapezoidal rule approximation is $O(m^{-2})$, provided that f has a continuous second derivative on $[0,1]$.

We now consider the multidimensional case, $d \geq 2$. Suppose we evaluate the function at all of the $(m+1)^d$ points of the form $(\frac{n_1}{m}, \ldots, \frac{n_s}{m})$ and use this to approximate the integral. The classical numerical integration methods use a Cartesian product of one-dimensional integration rules. For example, the d-fold Cartesian product of the trapezoidal rule is

$$\int_{[0,1]^d} f(\mathbf{u})d\mathbf{u} \approx \sum_{n_1=0}^{m} \cdots \sum_{n_s=0}^{m} w_{n_1} \cdots w_{n_s} f(\frac{n_1}{m}, \ldots, \frac{n_s}{m}), \tag{6.5}$$

where $[0,1]^d$ is the closed s-dimensional unit cube and the w_n are as before. The total number of nodes is $N = (m+1)^s$. From the previous error bound it follows that the error is $O(m^{-2})$, provided that the second partial derivatives of f are continuous on $[0,1]^d$. We know that the error cannot be smaller because when the function depends on only one variable and is constant in the others, the one-dimensional result is a special case. In terms of the number N of nodes or function evaluations, since $m = O(N^{1/d})$, the error is $O(N^{-2/d})$, which, with increasing dimension d, changes dramatically. For example if we required $N = 100$ nodes to achieve a required precision in the case $d = 1$, to achieve the same precision for a $d = 5$ dimensional integral using this approach we would need to evaluate the function at a total of $100^d = 10^{10}$= *ten billion nodes.* As the dimension increases, the number of function evaluations or computation required for a fixed precision increases exponentially. This phenomena is often called the "curse of dimensionality", exorcised in part at least by quasi or regular Monte Carlo methods.

The ordinary Monte Carlo method based on simple random sampling is free of the curse of dimensionality. By the central limit theorem, even a crude Monte Carlo estimate for numerical integration yields a probabilistic error bound of the form $O_P(N^{-1/2})$ in terms of the number N of nodes (or function evaluations) and this holds under a very weak regularity condition on the function f. The remarkable feature here is that this order of magnitude does not depend on the dimension d. This is true even if the integration domain is complicated. *Note however that the definition of "O" has changed from one that essentially considers the worst case scenario to O_P which measures the average or probabilistic behaviour of the error.*

Some of the oft-cited deficiencies of the Monte Carlo method limiting its usefulness are:

1. There are only probabilistic error bounds (there is no guarantee that the expected accuracy is achieved in a particular case -an alternative approach would optimize the "worst-case" behaviour);

2. Regularity of the integrand is not exploited even when it is available. The probabilistic error bound $O_P(N^{-1/2})$ holds under a very weak regularity condition but no extra benefit is derived from any additional regularity or smoothness of the integrand. For example the estimator is no more precise if we know that the function f has several continuous derivatives. In cases when we do not know whether the integrand is smooth or differentiable, it may be preferable to use Monte Carlo since it performs reasonably well without this assumption.

3. Genuine Monte Carlo is not feasible anyway since generating truly independent random numbers is virtually impossible. In practice we use pseudo-random numbers to approximate independence.

Theory of Low discrepancy sequences

The quasi-Monte Carlo method places attention on the objective, approximating an integral, rather than attempting to imitate the behaviour of independent uniform random variates. Quasi-random sequences of low discrepancy sequences would fail all of the tests applied to a pseudo-random number generate except those testing for uniformity of the marginal distribution because the sequence is, by construction, autocorrelated. Our objective is to approximate an integral using a average of the function at N points, and we may adjust the points so that the approximation is more accurate. Ideally we would prefer these sequences to be self-avoiding, so that as the sequence is generated, holes are filled. As usual we will approximate the integral with an average;

$$\int_{[0,1]^d} f(\mathbf{u})d\mathbf{u} \approx \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x_n}). \tag{6.6}$$

Quasi Monte-Carlo is able to achieve a deterministic error bound $O((logN)^d/N)$ for suitably chosen sets of nodes and for integrands with a relatively low degree of regularity, much better than the rate $O(N^{-1/2})$ achieved by Monte Carlo methods. Even smaller error bounds can be achieved for sufficiently regular integrands. There are several algorithms or quasi-Monte-Carlo sequences which give rise to this level of accuracy.

Suppose, as with a crude Monte Carlo estimate, we approximate the integral with (6.6) with $\mathbf{x_1}, \ldots, \mathbf{x_N} \in [0,1]^d$. The sequence $\mathbf{x_1}, \ldots, \mathbf{x_N}$,... is deterministic (as indeed are the pseudo-random sequences we used for Crude Monte-Carlo), but they are now chosen so as to guarantee a small error. Points are chosen so as to achieve the maximal degree of *uniformity* or a *low degree of discrepancy* with a uniform distribution. A first requirement for a low discrepancy sequence is that we obtain convergence of the sequence of averages so that:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x_n}) = \int_{[0,1]^d} f(\mathbf{u})\mathbf{du},$$

and this should hold for a reasonably large class of integrands. This suggests that the most desirable sequences of nodes $\mathbf{x_1}, \ldots, \mathbf{x_N}$ are “evenly distributed” over $[0,1]^d$. Various notions of discrepancy have been considered as quantitative measures for the deviation from the uniform distribution but we will introduce only one here, the so-called “star-discrepancy”. The star discrepancy is perhaps the more natural one in statistics, since it measures the maximum difference between the empirical cumulative distribution function of the points $\{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ and the uniform distribution of measure on the unit cube. Suppose we construct

$$\widehat{F}_N(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} I(\mathbf{x_n} \le \mathbf{x}),$$

the empirical cumulative distribution function of the points $\mathbf{x_1}, \ldots, \mathbf{x_N}$, and compare it with

$$F(\mathbf{x}) = F(x_1, ...x_d) = \min(1, x_1 x_2 ... x_d) \text{ if all } x_i \geq 0$$

the theoretical uniform distribution on $[0,1]^d$. While any measure of the difference could be used, the star discrepancy is simply the Kolmogorov-Smirnov distance between these two cumulative distribution functions

$$D_N^* = \sup_{\mathbf{x}} |\widehat{F}_N(\mathbf{x}) - F(\mathbf{x})| = \sup_B |\frac{\text{\# of points in B}}{N} - \lambda(B)|,$$

where the supremum is taken over all rectangles B of the form $[0,x_1] \times [0,x_2] \times ... \times [0,x_d]$ and where $\lambda(B)$ denotes the Lebesgue measure of B in $\mathcal{R}^d$.

It makes intuitive sense that we should choose points $\{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ such that the discrepancy is small for each N. This intuition is supported by a large number of theoretical results, at least in the case of smooth integrands with smooth partial derivatives. The smoothness is measured using $V(f)$, a "total variation" in the sense of Hardy and Krause, intuitively the length of the monotone segments of f. For a one dimensional function with a continuous first derivative it is simply

$$V(f) = \int_0^1 |f'(x)| dx.$$

In higher dimensions, the Hardy Krause variation may be defined in terms of the integral of partial derivatives;

Definition 48 *Hardy and Krause Total Variation*

If f is sufficiently differentiable then the variation of f on $[0,1]^d$ in the sense of Hardy and Krause is

$$V(f) = \sum_{k=1}^{s} \sum_{1 \square i_1 < \cdots < i_k \square s} V^{(k)}(f; i_1, \ldots, i_k), \qquad (6.7)$$

where

$$V^{(k)}(f; i_1, \ldots, i_k) = \int_0^1 \cdots \int_0^1 \left| \frac{\partial^s f}{\partial x_{i_1} \cdots \partial x_{i_k}} \right|_{x_j = 1, j \neq i_1, \ldots, i_k} dx_{i_1} \cdots dx_{i_k}. \qquad (6.8)$$

The precision in our approximation to an integral as an average of function values is closely related to the discrepancy measure as the following result shows. Indeed the mean of the function values differs from the integral of the function by an error which is bounded by the product of the discrepancy of the sequence and the measure $V(f)$ of smoothness of the function.

Theorem 49 *(Koksma-Hlawka inequality)*

If f has bounded variation $V(f)$ on $[0,1]^d$ in the sense of Hardy and Krause, then, for any $\mathbf{x_1},\dots,\mathbf{x_N} \in [0,1]^d$, we have

$$|\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x_n}) - \int_{I^s} f(\mathbf{u})\mathbf{du}| \leq V(f)D_N^*. \tag{6.9}$$

We do not normally use this inequality as it stands since the evaluation of the error bound on the right hand side requires determining $V(f)$, typically a very difficult task. However this bound allows a separation between the regularity properties of the integrand and the degree of uniformity of the sequence. We can guarantee a reasonable approximation for any function f with bounded total variation $V(f)$ by ensuring that the discrepancy of the sequence D_N^* is small. For this reason, the discrepancy is central to quasi-Monte Carlo integration. Sequences with small star discrepancy are called low-discrepancy sequences. In fact since a variety of sequences exist with discrepancy of order

$$\frac{(\log N)^d}{N}$$

as $N \to \infty$, the term “low-discrepancy” is often reserved for these.

Examples of low discrepancy sequences

Van der Corput Sequence.

In the one dimensional case the best rate of convergence is $O(N^{-1}\log N)$, $N \geq 2$. It is achieved, for example, by the **van der Corput sequence**, obtained by reversing the digits in the representation of some sequence of integers in a given base. Consider one-dimensional case $d = 1$ and base $b = 2$. Take the base b representation of the sequence of natural numbers;

$$1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, ...$$

and then map these into the unit interval $[0, 1]$ so that the integer $\sum_{k=0}^{t} a_k b^k$ is mapped into the point $\sum_{k=0}^{t} a_k b^{-k-1}$. These binary digits are mapped into (0,1) in the following three steps;

1. Write n using its binary expansion. e.g. $13 = 1(8) + 1(4) + 0(2) + 1(1)$ becomes 1101.

2. Reverse the order of the digits. e.g. 1101 becomes 1011.

3. Determine the number that this is the binary decimal expansion for. e.g. $1011 = 1(\frac{1}{2}) + 0(\frac{1}{4}) + 1(\frac{1}{8}) + 1(\frac{1}{16}) = \frac{11}{16}$.

Thus 1 generates 1/2, 10 generates $0(\frac{1}{2}) + 1(\frac{1}{4})$, 11 generates $1(\frac{1}{2}) + 1(\frac{1}{4})$ and the sequence of positive integers generates the points. The intervals are recursively split in half in the sequence $1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, ...$ and the points are fairly evenly spaced for any value for the number of nodes N, and perfectly spaced if N is of the form $2^k - 1$. The star discrepancy of this sequence is

$$D_N^* = O(\frac{\log N}{N})$$

which matches the best that is attained for infinite sequences.

The Halton Sequence

This is simply the multivariate extension of the Van der Corput sequence. In higher dimensions, say in d dimensions, we choose d distinct primes, $b_1, b_2, ...b_d$ (usually the smallest primes) and generate, from the same integer m , the d components of the vector using the method described for the Van der Corput sequence. For example, we consider the case $d = 3$ and use bases $b_1 = 2$, $b_2 = 3, b_3 = 5$ because these are the smallest three prime numbers. The first few vectors , $(\frac{1}{2}, \frac{1}{3}, \frac{1}{5}), (\frac{1}{4}, \frac{2}{3}, \frac{2}{5}), (\frac{3}{4}, \frac{1}{9}, \frac{3}{5}), ...$are generated in the table below.

m	repres	first	repres.	second	repres	third
	base 2	component	base 3	comp	base 5	comp
1	1	1/2	1	1/3	1	1/5
2	10	1/4	2	2/3	2	2/5
3	11	3/4	10	1/9	3	3/5
4	100	1/8	11	4/9	4	4/5
5	101	5/8	12	7/9	10	1/25
6	110	3/8	20	2/9	11	6/25
7	111	7/8	21	5/9	12	11/25
9	1000	1/16	22	8/9	13	16/25
10	1001	9/16	100	1/27	14	21/25

Figure 6.1 provides a plot of the first 500 points in the above Halton sequence of dimension 3.

There appears to be greater uniformity than a sequence of random points would have. Some patterns are discernible on the two dimensional plot of the first 100 points, for example see Figures 6.2 and 6.3.

These figures can be compared with the plot of 100 pairs of independent uniform random numbers in Figure 6.4, which seems to show more clustering and more holes in the point cloud.

These points were generated with the following function for producing the Halton sequence.

```
function x=halton(n,s)
%x has dimension n by s and is the first n terms of the halton sequence of
%dimension s.
p=primes(s*6); p=p(1:s); x=[];
```

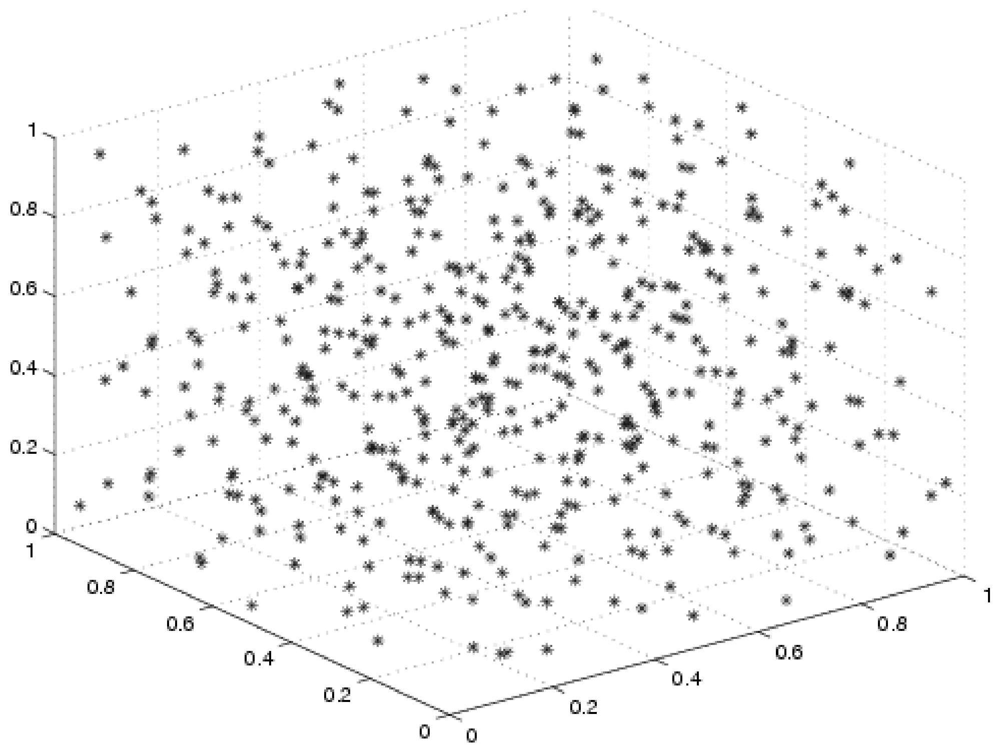

Figure 6.1: 500 Points from a Halton sequence of dimension 3

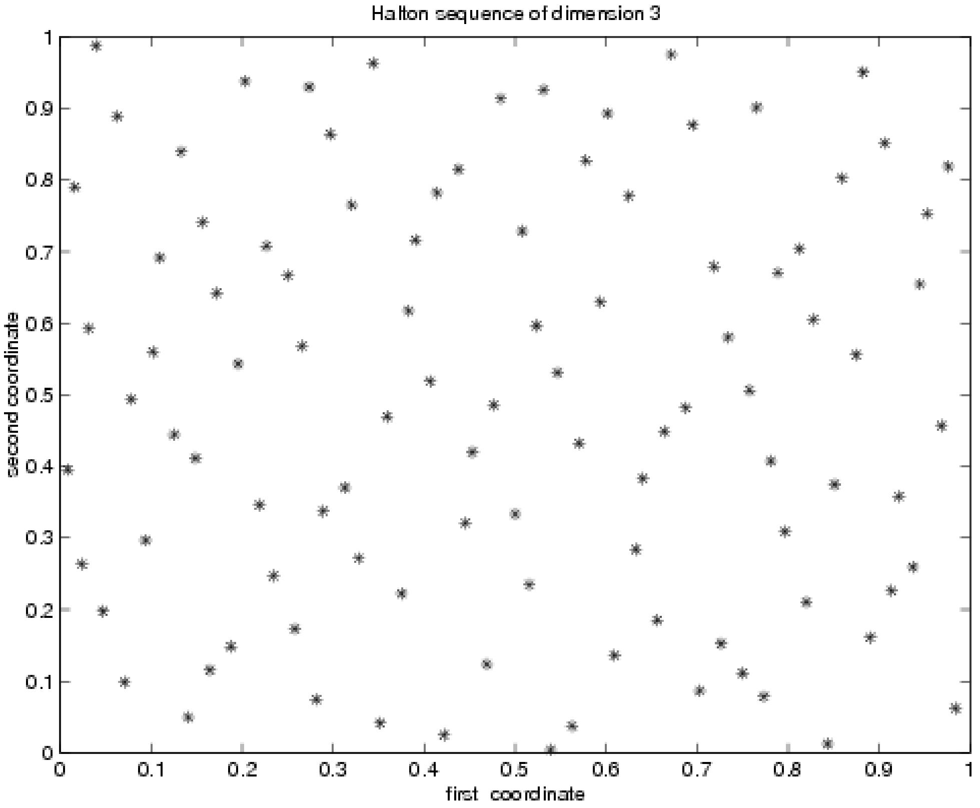

Figure 6.2: The first and second coordinate of 100 points from the Halton sequence of dimension 3

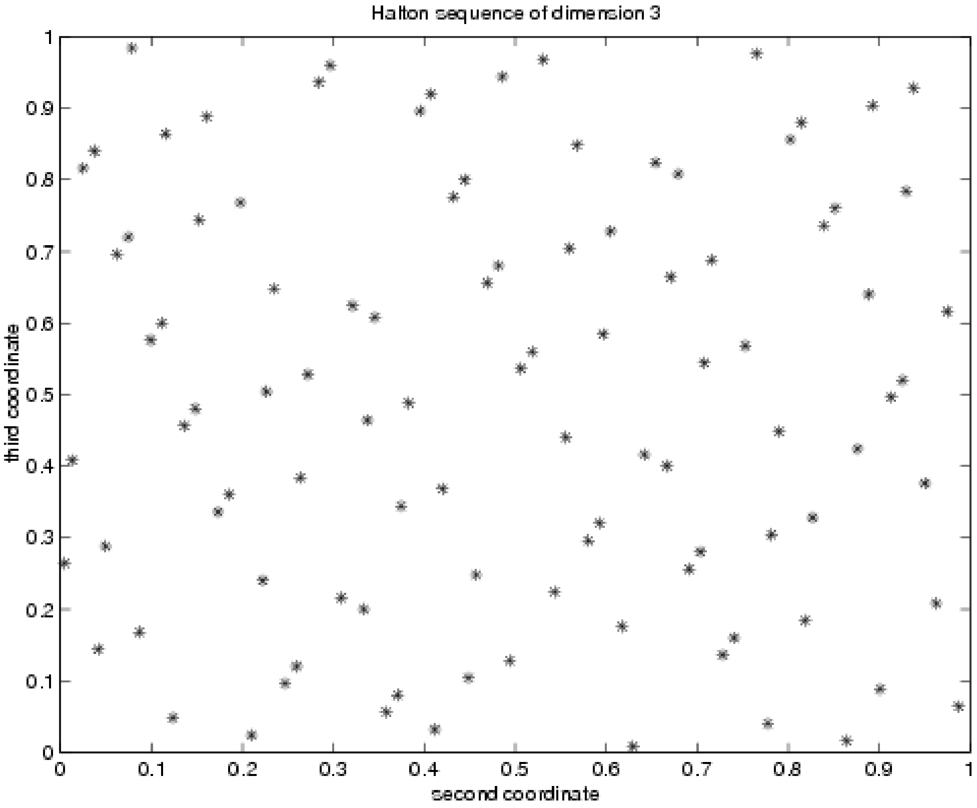

Figure 6.3: The second and third coordinate of 100 points from the Halton sequence of dimension 3

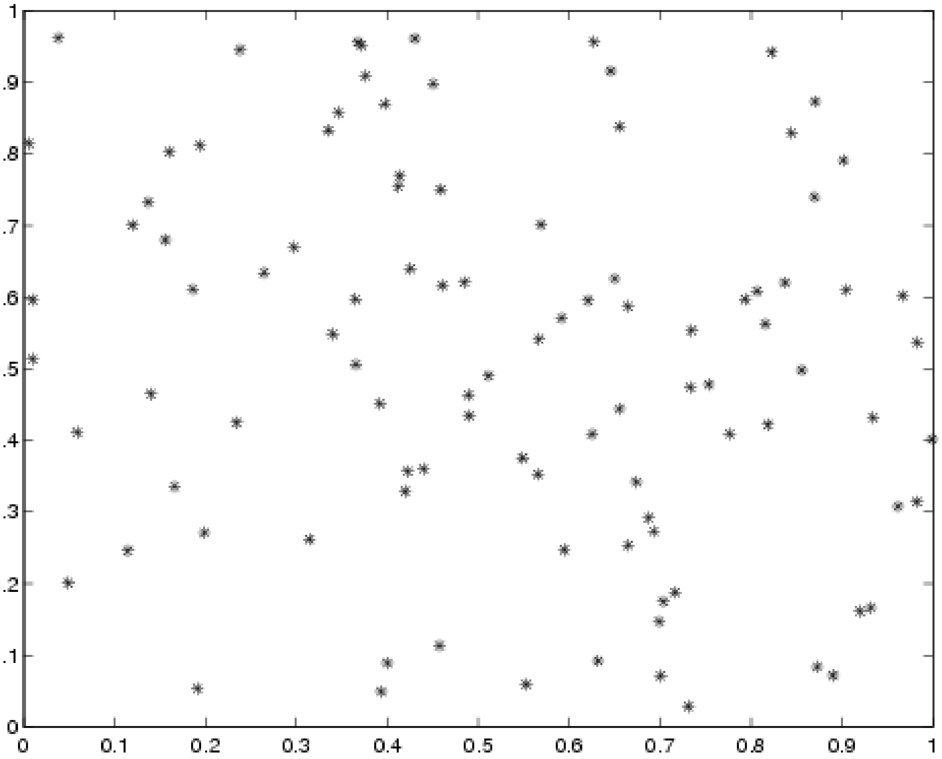

Figure 6.4: 100 independent $U[0,1]$ pairs

```
for i=1:s
 x=[x (corput(n,p(i)))’];
end
function x=corput(n,b)
% converts integers 1:n to from van der corput number with base b
m=floor(log(n)/log(b));
n=1:n;                A=[];
for i=0:m
 a=rem(n,b);      n=(n-a)/b;
A=[A ;a];
end
x=((1./b’).^(1:(m+1)))*A;
```

The Halton sequence is a genuine low discrepancy sequence in the sense that

$$D_N^* = O(\frac{(\log N)^d}{N})$$

and the coverage of the unit cube is reasonably uniform for small dimensions. Unfortunately the notation $O()$ hides a constant multiple, one which, in this case, depends on the dimension d. Roughly (Niedereiter, 1992), this constant is asymptotic to d^d which grows extremely fast in d. This is one indicator that for large d, the uniformity of the points degrades rapidly, largely because the relative sparseness of the primes means that the $d'th$ prime is very large for d large. This results in larger holes or gaps in that component of the vector than we would like. This is evident for example in Figure6.5 where we plot the last two coordinates of the Halton sequence of dimension 15.

The performance of the Halton sequence is considerably enhanced by permuting the coefficients a_k prior to mapping into the unit interval as is done by the Faure sequence.

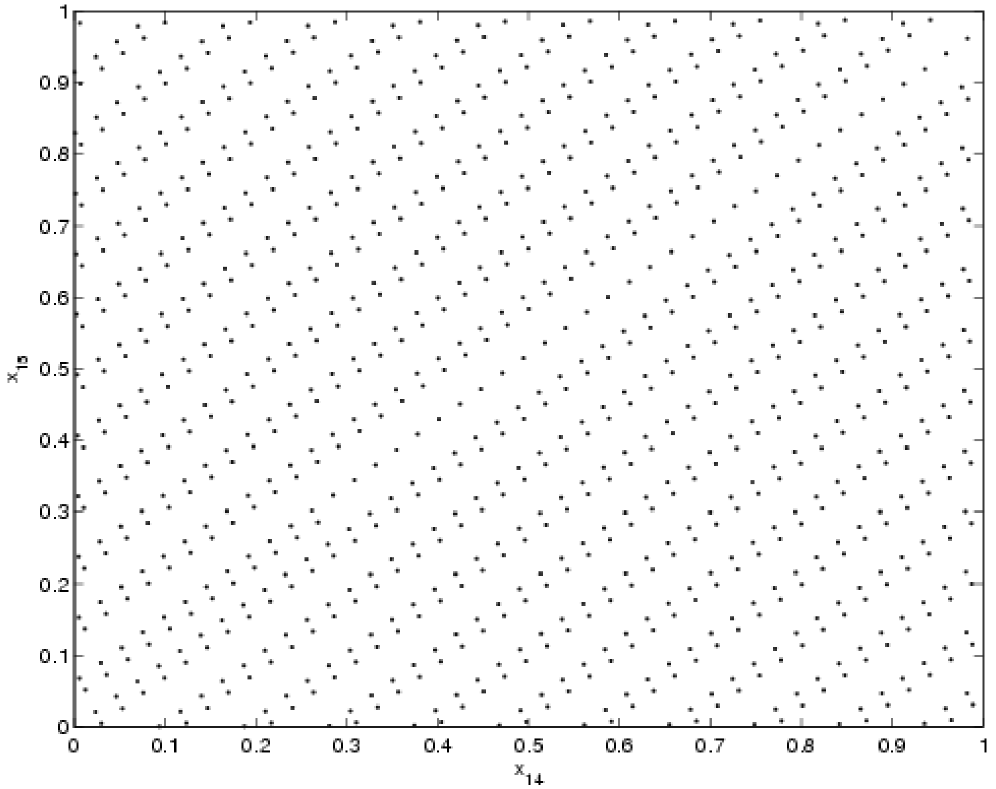

Figure 6.5: The 14'th and 15'th coordinates of the first 1000 of a Halton sequence $d = 15$

Faure Sequence

The **Faure** sequence is similar to the Halton sequence in that each dimension is a permutation of a van der Corput sequence; however, the same prime is used as the base b for each of the components of the vector, and is usually chosen to be the smallest prime greater than or equal to the dimension (Fox, 1996).

In the Van der Corput sequence we wrote the natural numbers in the form $\sum_{k=0}^{t} a_k b^k$ which was then mapped into the point $\sum_{k=0}^{t} a_k b^{-k-1}$in the unit interval. For the Faure sequence we use the same construction but we use different permutations of the coefficients a_k for each of the coordinates. In particular in order to generate the i'th coordinate we generate the point

$$\sum_{k=0}^{t} c_k b^{-k-1}$$

where

$$c_k = \sum_{m=k}^{t} \binom{m}{k} (i-1)^{m-k} a_m \bmod b$$

Notice that only the last $t-k+1$ values of a_i are used to generate c_k. For example consider the case $d=2, b=2$. Then the first 10 Faure numbers are

$$\begin{array}{cccccccccc} 0 & 1/2 & 1/4 & 3/4 & 1/8 & 5/8 & 3/8 & 7/8 & 1/16 & 9/16 \\ 0 & 1/2 & 3/4 & 1/4 & 5/8 & 1/8 & 3/8 & 7/8 & 15/16 & 7/16 \end{array}$$

The first row corresponds to the Van der Corput numbers and the second row of obtained from the first by permuting the values with the same denominator.

The Faure sequence has better regularity properties than does the Halton sequence above particularly in high dimensions. However the differences are by no means evident from a graph when the dimension is moderate. For example we plot in Figure 6.6 the 14'th and 15'th coordinates of 1000 points from the Faure sequence of dimension $d = 15$ for comparison with Figure 6.5.

Other suggestions for permuting the digits in a Halton sequence include using only every l'th term in the sequence so as to destroy the cycle.

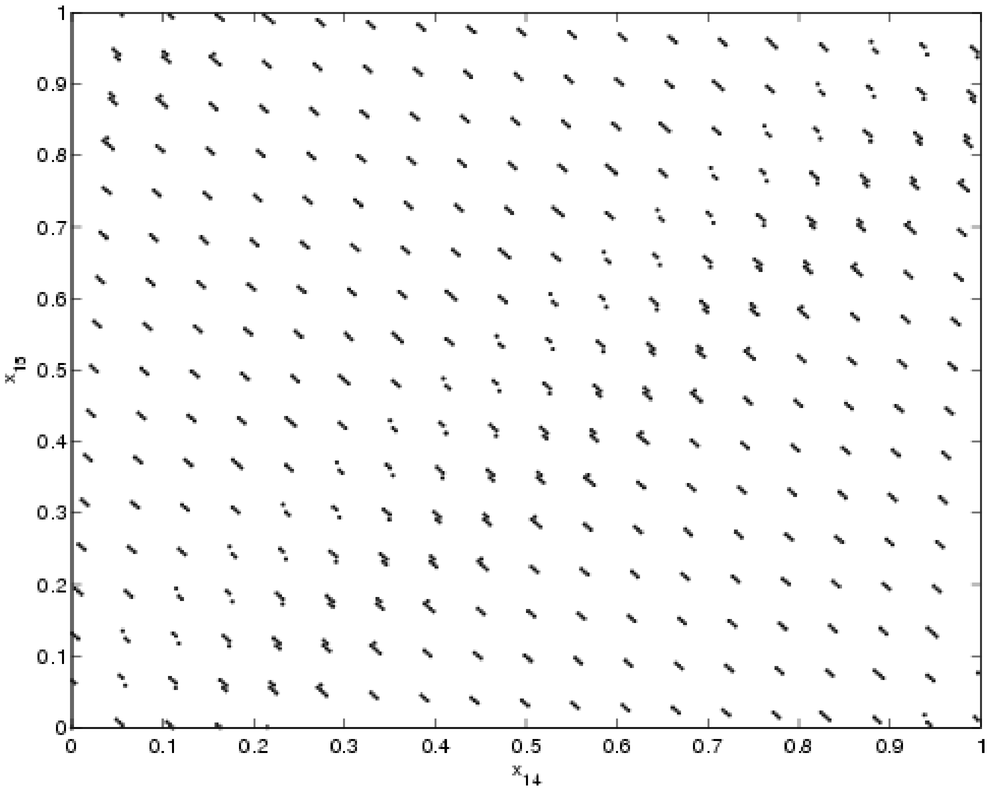

Figure 6.6: The last two coordinates of the first 1000 Faure points of dimension $d = 15$.

In practice, in order to determine the effect of using one of these low discrepancy sequences we need only substitute such a sequence for the vector of independent uniform random numbers used by a simulation. For example if we wished to simulate a process for 10 time periods, then value a call option and average the results, we could replace the 10 independent uniform random numbers that we used to generate one path by an element of the Halton sequence with $d = 10$.

Suppose we return briefly to the call option example treated in Chapter 3. The true value of this call option was around 0.4615 according to the Black-Scholes formula. If however we substitute the Van der Corput sequence for the sequence of uniform random numbers,

```
mean(fn(corput(100000,2)))
```

we obtain an estimate of 0.4614 very close to the correct value. I cannot compare these estimators using the notion of efficiency that we used there, however, because these low-discrepancy sequences are not random and do not even attempt to emulate random numbers. Though unable to compare performance with the variance of an estimator, we can look at the Mean squared error (see for example Figure 6.8). which shows a faster rate of convergence for Quasi Monte Carlo equivalent to variance reduction in excess of 100). Galanti & Jung (1997), report that the Faure sequence suffers from the problem of start-up and especially in high-dimensions and the Faure numbers can exhibit clustering about zero. In order to reduce this problem, Faure suggests discarding the first $b^4 - 1$ points.

Sobol Sequence

The Sobol sequence is generated using a set of so-called *direction numbers* $v_i = \frac{m_i}{2^i}, i = 1, 2,$ where the m_i are odd positive integers less than 2^i. The values of m_i are chosen to satisfy a recurrence relation using the coefficients of a *primitive polynomial in the Galois Field of order 2.* A primitive polynomial is irreducible (i.e. cannot be factored into polynomials of smaller degree) and does not divide the polynomial $x^r + 1$ for $r < 2^p - 1$. For example the polynomial $x^2 + x + 1$ has no non-trival factors over the *Galois Field of order 2* and it does divide $x^3 + 1$ but not $x^r + 1$ for $r < 3$. Corresponding to a primitive polynomial

$$z^p + c_1 z^{p-1} + ...c_{p-1}z + c_p$$

is the recursion

$$m_i = 2c_1 m_{i-1} + 2^2 c_2 m_{i-2} + ... + 2^p c_p m_{i-p}$$

where the addition is carried out using binary arithmetic. For the Sobol sequence, we then replace the binary digit a_k by $a_k v_k$.

In the case $d = 2$, the first 10 Sobol numbers are, using irreducible polynomials $x + 1$ and $x^3 + x + 1$

0	1/2	1/4	3/4	3/8	7/8	1/8	5/8	5/16	13/16
0	1/2	1/4	3/4	1/8	5/8	3/8	7/8	11/16	3/16

Again we plot the last two coordinates for the first 1000 points from a Sobol sequence of dimension $d = 15$ in Figure 6.7 for comparison with Figures 6.5 and 6.6.

Although there is a great deal of literature espousing the use of one quasi-Monte Carlo sequence over another, most results from a particular application and there is not strong evidence at least that when the dimension of the problem is moderate (for example $d \leq 15$) it makes a great deal of difference whether we use Halton, Faure or Sobol sequences. There is evidence that the starting values for the Sobol sequences have an effect on the speed of convergence, and that Sobol sequences can be generated more quickly than Faure Moreover neither the Faure nor Sobol sequence provides a “black-box” method because both are

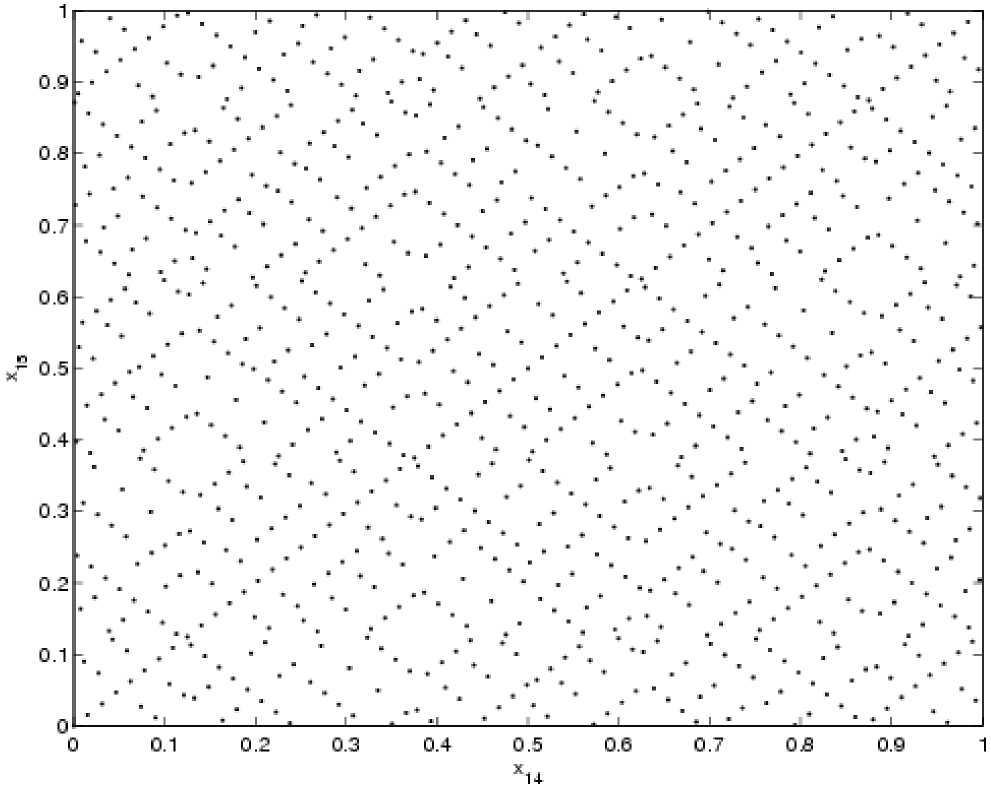

Figure 6.7: The last two coordinates of the first 1000 points from a Sobel sequence of dimension 15

sensitive to intitialization. I will not attempt to adjudicate the considerable literature on this topic here, but provide only a fragment of evidence that, at least in the kind of example discussed in the variance reduction chapter, there is little to choose between the various methods. Of course this integral, the discounted payoff from a call option as a function of the uniform input, is a one-dimensional integral so the Faure, Halton and Van der Corput sequences are all the same thing in this case. In Figure 6.8 we plot the (expected) squared error as a function of sample size for $n = 1, ..., 100000$ for crude Monte Carlo (the dashed line) and the Van der Corput sequence. The latter, although it oscillates somewhat, is substantially better at all sample sizes, and its mean squared error is equivalent to a variance reduction of around 1000 by the time we reach $n = 100,000$. The different slope indicates an error approaching zero at rate close to n^{-1} rather than the rate $n^{-1/2}$ for the Crude Monte Carlo estimator. The Sobol sequence, although highly more variable as a function of sample size, appears to show even more rapid convergence along certain subsequences.

The Sobol and Faure sequences are particular cases of $(t, s) - nets$. In order to define then we need the concept of an elementary interval.

Elementary Intervals and Nets

Definition: elementary interval

An elementary interval in base b is n interval E in I^s of the form

$$E = \prod_{j=1}^{s} {}^{\square} \frac{a_j}{b^{d_j}}, \frac{(a_j + 1)}{b^{d_j}} \Big), \tag{6.10}$$

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

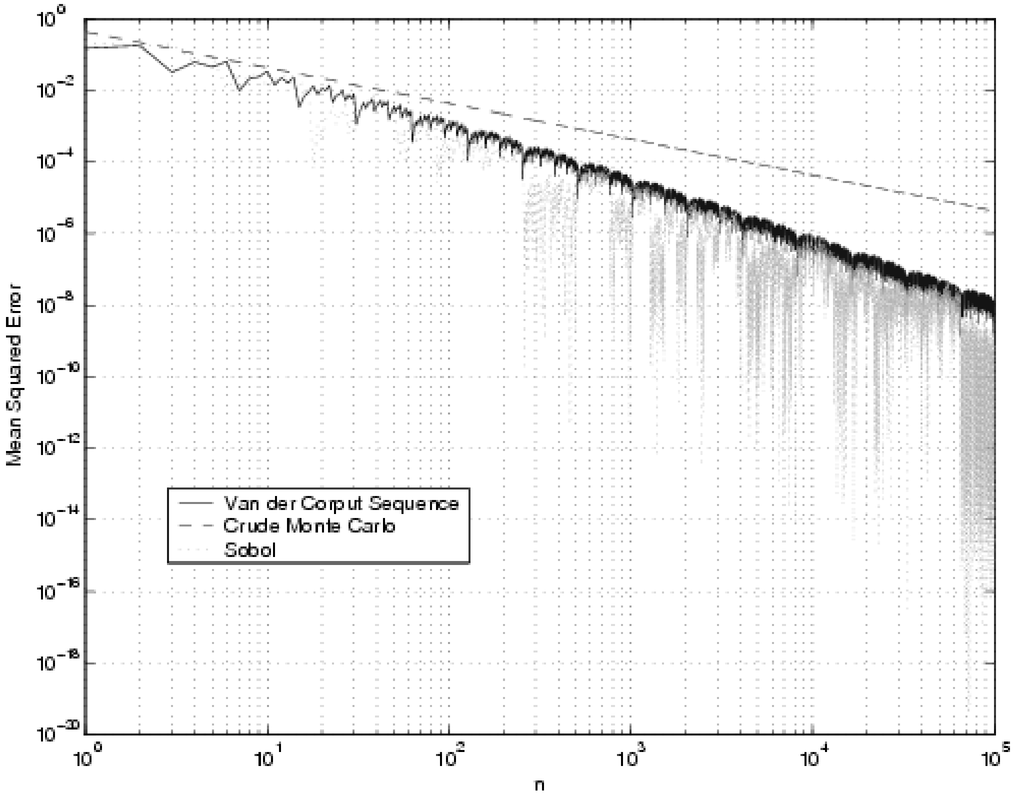

Figure 6.8: (Expected) squared error vs. sample size in the estimation of an Call option price for Crude MC and Van der Corput sequence.

Definition: (t, m, s) - net

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

Definition: (t, s) - sequence

An infinite sequence of points $\{\mathbf{x_i}\} \in I^s$ is a (t,s)-sequence in base b if for all $k \geq 0$ and $m > t$, the finite sequence $\mathbf{x_{kb^m}}, \ldots, \mathbf{x_{(k+1)b^{m-1}}}$ forms a (t,m,s) - net in base b.

It is known that for a (t, s)-sequence in base b the low discrepancy is ensured:

$$D_N^* \leq C\frac{(\log N)^s}{N} + O(\frac{(\log N)^{s-1}}{N}). \tag{6.11}$$

Special constructions of such sequences for $s \geq 2$ have the smallest discrepancy that is currently known (H. Niederreiter, 1992, *Random Number Generation and Quasi-Monte Carlo Methods*). j

The thesis of K.S. Tan (1998) provides a thorough investigation into various improvements in Quasi-Monte Carlo sampling, as well as the evidence of the high efficiency of these methods when valuing Rainbow Options in high dimensions. Papageorgiou and Traub (1996) tested what Tezuka called generalized Faure points. They concluded that these points were superior to Sobol points for the model problem. Particularly important for financial computation, a reasonably small error could be achieved with few evaluations. For example, just 170 generalized Faure points were sufficient to achieve an error of less than one part in a hundred for a 360 dimensional problem. See also Traub and Wozniakowski (1994) and Paskov and Traub (1995).

In summary, Quasi-Monte Carlo frequently generates estimates superior to Monte-Carlo methods in many problems of low or intermediate effective dimension. If the dimension d is large, but a small number of variables determine most of the variability in the simulation, then we might expect Quasi Monte-Carlo methods to continue to perform well. The price we pay for the smaller error often associated with quasi Monte-Carlo methods and other numerical techniques, one that often a consequence of any departure from a crude simulation of the process, is a feel for the *distribution* of various functionals of the process of interest, as opposed to generating a single precise estimate. The theory supporting low-discrepancy sequences, both the measures of discrepancy themselves and the variation measure $V(f)$ are both tied, somewhat artificially, to the axes. For example if $f(x)$ represents the indicator function of a square with sides parallel to the axes in dimension $d = 2$, then $V(f) = 0$. However, if we rotate this rectangle by 45 degrees, the variation becomes infinite indicating that functions with steep isoclines at a 45 degree angle to the exes may be particularly difficult to integrate using Quasi Monte Carlo.

Problems

1. Use 3-dimensional Halton sequences to integrate the function

$$\int_0^1 \int_0^1 \int_0^1 f(x, y, z) dx dy dz$$

where $f(x, y, z) = 1$ if $x < y < z$ and otherwise $f(x, y, z) = 0$. Compare your answer with the true value of the integral and with crude Monte Carlo integral of the same function.

2. Use your program from question 1 to generate 50 points uniformly distributed in the unit cube. Evaluate the Chi-squared statistic χ^2_{obs} for a test that these points are independent uniform on the cube where we divide the cube into 8 subcubes, eacc having sides of length 1/2. Carry out the test by finding $P[\chi^2 > \chi^2_{obs}]$ where χ^2 is a random chi-squared variate with the appropriate number of degrees of freedom. This quantity $P[\chi^2 > \chi^2_{obs}]$ is usually referrred to as the "significance probability" or "p-value" for

the test. If we suspected *too much uniformity* to be consistent with assumption of independent uniform, we might use the other tail of the test, i.e. evaluate $P[\chi^2 < \chi^2_{obs}]$. Do so and comment on your results.