CHAPTER 30

NUMERICAL INTEGRATION

1. **Value a European call option using a Monte Carlo simulation. Use the simulation to estimate the value of an integral, as opposed to simulating the random walk for the asset.**

Let us suppose that we have d assets following correlated random walks. The risk-neutral value of these assets at a time t can be written as

$$S_i(T) = S_i(t)e^{\left(r-D_i-\frac{1}{2}\sigma_i^2\right)(T-t)+\sigma_i\phi_i\sqrt{T-t}},$$

in terms of their initial values at time t. The random variables ϕ_i are Normally distributed and correlated. We can now write the value of our European option as

$$e^{-r(T-t)}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\text{Payoff}(S_1(T),\ldots,S_d(T))\,p(\phi_1,\ldots,\phi_d)$$
$$\times\, d\phi_1\ldots d\phi_d,$$

where $p(\phi_1,\ldots,\phi_d)$ is the probability density function for d correlated Normal variables with zero mean and unit standard deviation. We don't need to know p explicitly as long as we generate numbers from its distribution.

To value the option we must generate suitable Normal variables. The first step is to generate uncorrelated variables and then transform them into correlated variables. Use Box–Muller and then Cholesky. The option value is then estimated by the average of the payoff over all the randomly generated numbers.

The following is a code fragment for calculating the value of a European option in `NDim` assets using `NoPts` points. The interest rate is `IntRate`, the dividend yields are `Div(i)`, the volatilities are `Vol(i)`, time to expiry `Expiry`. The initial values of the assets are `Asset(i)`. The Normally-distributed variables are the `x(i)` and the `S(i)` are the lognormally distributed future asset values.

```
a = Exp(-IntRate * Expiry) / NoPts
suma = 0
For k = 1 To NoPts
For i = 1 To NDim
If test = 0 Then
    Do
        y = 2 * Rnd() - 1
        z = 2 * Rnd() - 1
        dist = y * y + z * z
    Loop Until dist < 1
x(i) = y * Sqr(-2 * Log(dist) / dist)
test = 1
Else
x(i) = z * Sqr(-2 * Log(dist) / dist)
test = 0
End If
Next i
For i = 1 To NDim
    S(i) = Asset(i) * Exp((IntRate - Div(i) - 0.5 * Vol(i) * _
                Vol(i)) * Expiry + Vol(i) * x(i) * Sqr(Expiry))
Next i
term = Payoff(S(1), S(2), S(3), S(4), S(5))
suma = suma + term
Next k
Value = suma * a
```

2. **Repeat the above using Halton sequences.**

   Simply replace the two VBA functions `RND()` with Halton numbers. Just make sure that the bases for each dimension, each `i`, are different primes. (And you'll need two bases for each dimension because two `RND()`s are used.)

3. **Calculate Halton sequences for bases 2, 3 and 4. Compare the results (you may wish to plot one base against another.) What do you notice? How should this affect the bases you choose for a multi-factor problem?**

   The Halton sequence is a sequence of numbers $h(i; b)$ for $i = 1, 2, \cdots$. The integer b is the base. The numbers all lie between zero and one and are constructed as follows. First choose your base. Let us choose 2. Now write the positive integers in ascending order in base 2, i.e. 1, 10, 11, 100, 101, 110, 111 etc. The Halton sequence base 2 is the reflection of the positive integers in the decimal point i.e.

Integers base 2	Halton sequence base 2	Halton number base 10
1	$1 \times \frac{1}{2}$	0.5
10	$0 \times \frac{1}{2} + 1 \times \frac{1}{4}$	0.25
11	$1 \times \frac{1}{2} + 1 \times \frac{1}{4}$	0.75
100	$0 \times \frac{1}{2} + 0 \times \frac{1}{4} + 1 \times \frac{1}{8}$	0.125
$\cdots$	$\cdots$	$\cdots$

Generally, the integer n can be written as

$$i = \sum_{j=1}^{m} a_j b^j,$$

in base b, where $0 \leq a_j < b$. The Halton numbers are then given by

$$h(i; b) = \sum_{j=1}^{m} a_j b^{-j-1}.$$

The following is an algorithm for calculating Halton numbers of arbitrary base; the `n`th term in a Halton sequence of base `b` is given by `Halton(n, b)`.

```
Function Halton(n, b)
Dim n0, n1, r As Integer
Dim h As Double
Dim f As Double
    n0 = n
    h = 0
    f = 1 / b
While (n0 > 0)
    n1 = Int(n0 / b)
    r = n0 - n1 * b
    h = h + f * r
    f = f / b
    n0 = n1
Wend
Halton = h
End Function
```

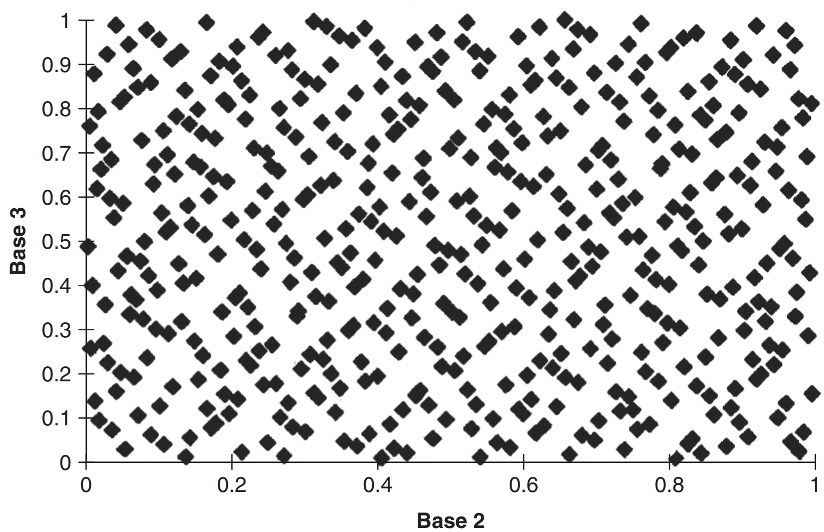

Figure 30.1 Halton points in two dimensions—bases 2 and 3.

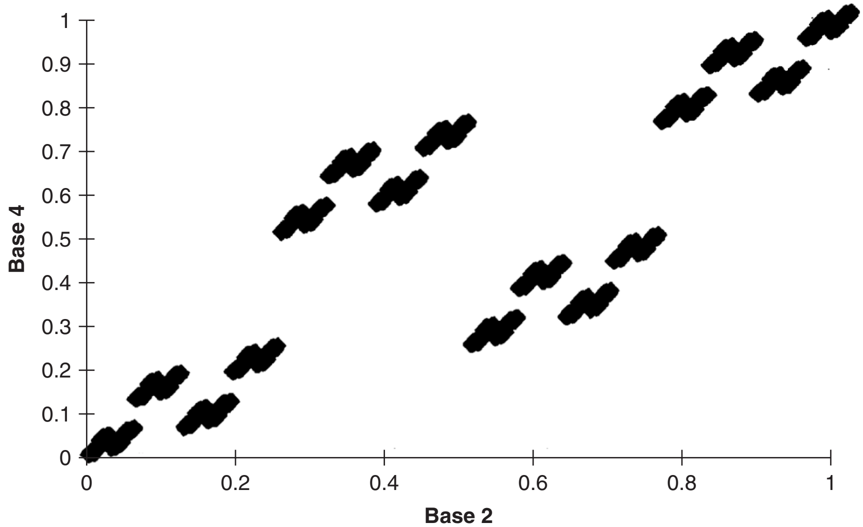

Figure 30.2 Halton points in two dimensions—bases 2 and 4.

Figure 30.1 shows the distribution of points in the Halton sequence for bases 2 and 3. Figure 30.2 shows the distribution of points in the Halton sequence for bases 2 and 4. We can see that the sequence for bases 2 and 3 ‘fills out’ the space and will be a suitable set of quasi-random numbers to use to value an integral. However, the sequence for bases 2 and 4 has extreme clumping of points and would give a very inaccurate answer for the integral because large regions of the space have no points in at all.

When distributing numbers in two dimensions, we choose, for example, the Halton sequence of bases 2 and 3 and calculate our integrand at the points $(h(i, 2), h(i, 3))$ for $i = 1, \cdots, N$. The bases of the two sequences we choose should be prime numbers to prevent this clumping effect from occurring.