Numerical Methods in Finance

Hardy Hulley,
Grant Lotter
and
Tom McWalter

Email address: tom@analytical.co.za

Abstract. A course on Numerical Methods, with particular emphasis on Monte Carlo methods with applications to finance. The notes are based on lectures given in the Programme for Advanced Mathematics of Finance at the University of the Witwatersrand (2000-2012) and at the University of Cape Town and University of Johannesburg thereafter.
Email: tom@analytical.co.za

Contents

Chapter 1

Random Number Generation

Anybody who considers purely arithmetical methods of producing random digits is, of course, in a state of sin.

— John von Neumann

Nothing in nature is random... A thing appears random only through the incompleteness of our knowledge.

— Spinoza

1.1. Introduction

In order to use Monte Carlo methods (for pricing derivatives, for example), we need to be able to generate random variates from a specified probability distribution. We distinguish between the following:

Random numbers. These are generated by some sort of physical process — like the intervals between clicks of a Geiger counter placed next to a radioactive element. Such random sequences suffer from two defects. Firstly, they are not easily generated (they certainly can't be generated by a computer, for example), and so they must be stored (on a CD-ROM, say). Secondly, all such sequences will be different, and so no technique which uses them will generate reproducible results.

Pseudo-random numbers. These are computer-generated numbers, which are designed to look random. But such sequences are deterministic, in the sense that any algorithm starting from the same initial conditions will generate exactly the same sequences of numbers. Furthermore, computer-generated sequences of pseudo-random numbers always cycle! Pseudo-random sequences have two advantages. Firstly, because every sequence generated by an algorithm under the same initial conditions is identical, numerical techniques which use such sequences will generate reproducible results. Secondly, there exist many statistical and analytical techniques which can be used to establish how well pseudo-random number generators perform; so we can easily develop an idea of the quality of such a sequence.

Quasi-random numbers. Quasi-random sequences are not random at all. Instead, they are rule-generated sequences designed to fill out a space in an optimally regular way. They often offer a better approach to integration than pseudo-random sequences, because they are easier to generate, and because they often lead to quicker convergence. However, in contradistinction to Monte Carlo techniques based upon pseudo-random sequences (where convergence properties are easily analyzed), understanding the behaviour of a method based on quasi-random numbers is more difficult.

In all that follows, when referring to "random numbers", we shall mean "pseudo-random numbers".

The properties of a random number generating algorithm can be grouped into three basic categories:
(1) *Structural properties.* An example is *period length.* As we shall see, all generators necessarily have a finite period — obviously we want this to be as long as possible.
(2) *Statistical properties.* These relate mainly to issues such as *goodness of fit* and *randomness.* Goodness of fit measures for a generator indicate how well the sample distribution of simulated variates fit the target distribution.
(3) *Computational properties.* Here one is mainly concerned with how quick a random number generator is, and its memory requirements. In the early days of computing, these were very important considerations, but they are less so now, due to the speed of modern computers. In most applications of random numbers (Monte Carlo integration, for example), the time taken to produce random numbers is negligible, compared with the time taken to process them.

The rest of this chapter will be concerned first with generators of uniform random variates, and then with algorithms for sampling from other distributions.

1.2. Generating Uniformly Distributed Variates

In this section, we will be concerned with the problem of generating $\mathscr{U}[0,1)$ values. In a sense, this problem is fundamental, because the most popular techniques for generating variates from non-uniform distributions rely upon the ability to generate uniform numbers.

We first note that, since a computer can only represent a number with a finite accuracy, it is typical to consider uniform generators of integers in a finite range, and then to construct approximate $\mathscr{U}[0,1)$ values from these. Suppose we have a generator capable of producing i.i.d. $\mathscr{U}\{0,1,\ldots,m-1\}$ values. Then, if x is the output from this generator, we record $u := \frac{x}{m}$ as the approximate $\mathscr{U}[0,1)$ number. Obviously, u is constrained to lie on a grid of mesh length $\frac{1}{m}$, but this isn't a problem if m is large enough.

The most common class of generator is based upon a choice of m, a function[1]

$$f : \mathbb{Z}_m^k \to \mathbb{Z}_m, \tag{1.1}$$

and a set of *seed values* (initial values) $x_1, \ldots, x_k$. The sequence is then generated by the recursive rule

$$x_{i+1} := f(x_{i-k+1}, \ldots, x_i), \tag{1.2}$$

for $i \geq k$. We can immediately prove the following result concerning the performance of generators of the form (1.2), *which implies that such generators must cycle if they run long enough!*

Lemma 1.1. (Generators cycle) *There exist* α, $\beta \in \mathbb{N}$ *such that* $x_{i+\beta} = x_i$, *for all* $i > \alpha$.

Proof. $\mathbb{Z}_m^k$ contains m^k elements. By repeatedly applying (1.2), we can construct the following sequence of elements of $\mathbb{Z}_m^k$:

$$\begin{aligned} s_1 &:= (x_1, x_2, \ldots, x_k) \\ s_2 &:= (x_2, x_3, \ldots, x_{k+1}) \\ &\vdots \\ s_{m^k} &:= (x_{m^k}, x_{m^k+1}, \ldots, x_{m^k+k-1}) \\ s_{m^k+1} &:= (x_{m^k+1}, x_{m^k+2}, \ldots, x_{m^k+k}). \end{aligned}$$

This sequence clearly contains $m^k + 1$ elements. Consequently, we must be able to pick two (different) elements from here which are equal. We can (without loss of generality) write these as

$$s_{\alpha-k+1} = (x_{\alpha-k+1}, x_{\alpha-k+2}, \ldots, x_\alpha)$$

[1] $\mathbb{Z}_m$ is the set of integer values modulo m, i.e., $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$.

and

$$s_{\alpha+\beta-k+1} = (x_{\alpha+\beta-k+1}, x_{\alpha+\beta-k+2}, \ldots, x_{\alpha+\beta}),$$

for some $k \leq \alpha < m^k + k$ and $1 \leq \beta \leq m^k + k - \alpha$. In that case, we clearly have the following sequence of equalities:

$$\begin{aligned}
x_{\alpha+1} &= f(x_{\alpha-k+1}, \ldots, x_\alpha) = f(x_{\alpha+\beta-k+1}, \ldots, x_{\alpha+\beta}) = x_{\alpha+\beta+1} \\
x_{\alpha+2} &= f(x_{\alpha-k+2}, \ldots, x_{\alpha+1}) = f(x_{\alpha+\beta-k+2}, \ldots, x_{\alpha+\beta+1}) = x_{\alpha+\beta+2} \\
&\vdots \\
x_{\alpha+n} &= f(x_{\alpha-k+n}, \ldots, x_{\alpha+n-1}) = f(x_{\alpha+\beta-k+n}, \ldots, x_{\alpha+\beta+n-1}) = x_{\alpha+\beta+n},
\end{aligned}$$

for $n \in \mathbb{N}$. We can rewrite this as $x_{i+\beta} = x_i$, for all $i > \alpha$. □

Definition 1.1. (Period of a generator) The smallest $\beta \in \mathbb{N}$ satisfying $x_{i+\beta} = x_i$, for some $\alpha \in \mathbb{N}$ and all $i > \alpha$, is called the *period* of the generator.

A very useful conclusion we can derive from the above result is that the period of (1.2) must always satisfy $1 \leq \beta \leq m^k$. One of the aims, in developing any random number generator, is to make its period both as long as possible, and independent of its seeds $x_1, \ldots, x_k$. (Why is independence of starting values important?)

The following existence result asserts that it is always possible to construct a generator which has the maximum theoretically possible period.

Lemma 1.2. (Existence of maximal generators) *There always exists a generator (1.2) whose period is equal to the theoretical maximum (i.e., for which $\beta = m^k$).*

Proof. We construct the generator iteratively. We set $(x_{k+1}, \ldots, x_{2k}) := (0, \ldots, 0)$, and for each $i \geq 2k$, let

$$x_{i+1} := \max\{x \in \mathbb{Z}_m : (x_{i-k+1}, \ldots, x_i, x) \text{ has not occurred since } (0, \ldots, 0)\}.$$

When this condition cannot be satisfied, set $x_{i+1} := 0$. □

Exercise 1.1. By using different values for m and k, convince yourself that the generator specified in the proof of the previous lemma does indeed have a period of maximal length. Can you provide further justification? Why is the period independent of the seeds?

The random number generator we have just constructed teaches us an important lesson. Consider the case where $k = 1$; i.e., the function (1.1) is of the form $f : \mathbb{Z}_m \to \mathbb{Z}_m$. Then, when seeded with x, the rule will produce the following sequence:

$$x, 0, m-1, m-2, \ldots, 2, 1, 0, m-1, \ldots$$

Clearly, we cannot regard the above as random. The lesson is that a generator with a long period can still perform poorly. So, in addition to determining that a given generator has an acceptable period, which is invariant under choice of initial conditions, we must also demonstrate that it does in fact produce i.i.d. variables from the specified distribution.

One last note concerning our rule from the above lemma. When we set $m := 2$, it becomes a *random bit generator*, and the sequences it produces are known as *de Bruijn sequences*. Unfortunately, de Bruijn sequences are difficult to analyze mathematically, and so haven't found much application.

1.2.1. Linear Congruential Generators. Historically, the most important class of generators are *linear congruential generators*, introduced by D.H. Lehmer in 1951.

Definition 1.2. (Linear congruential generators) We have $k = 1$, and $f : \mathbb{Z}_m \to \mathbb{Z}_m$ is given by

$$f(x) = (ax + c) \bmod m, \tag{1.3}$$

Consequently, the recurrence relation which specifies the generator is

$$x_{i+1} := (ax_i + c) \bmod m, \tag{1.4}$$

for all $i \geq 1$, and where x_1 is the seed. In the above, a is called the *multiplier*, c is called the *increment*, and m is called the *modulus*.

Because of the simplicity of these generators they can be analyzed mathematically, so that good choices can be made for the parameters a, c and m. For example, we can easily determine (simply be inspecting (1.4)) that the maximum possible period for this rule must be m. (So, we clearly want m to be large.) The following result characterizes those triples (a, c, m) for which the linear congruential generator achieves its theoretical maximum period.

Theorem 1.1. (Hull, Dobel condition) *The linear congruential generator specified by (a, c, m) has period m iff the following conditions hold*[2]*:*

(1) $(c, m) = 1$ *(i.e., c and m are relatively prime);*
(2) $p \mid (a - 1)$ *for every prime p such that $p \mid m$; and*
(3) $4 \mid (a - 1)$ *if* $4 \mid m$.

Proof. See [**10**, page 17]. □

Note that the conditions of this theorem are trivially satisfied by $(1, 1, m)$, for any m. Although this choice provides the maximum period, it does not

[2]Here the notation (x, y) is read as the "largest common factor between x and y", and $x|y$ is read as "x divides y without remainder" and means that $y \bmod x = 0$.

guarantee a good generator. We emphasize again that statistical tests should be used to choose between generators with good structural properties.

The linear congruential method is fast, requiring only a few operations per call — hence the ubiquity of such generators. However, as can be seen by the previous set of parameters, it has one fundamental flaw — namely, it is not free of sequential correlation on successive calls. If k random variables are used at a time to plot points in k-dimensional space (with each coordinate between 0 and 1), then the points will not tend to "fill up" the k-dimensional space — *instead they will all lie on at most* $m^{1/k}$ $(k-1)$-*dimensional "planes"*. And that is the best case. If the parameters are not chosen carefully, there could be far fewer than $m^{1/k}$ of these hyperplanes. If m is as bad as 32768, then the number of planes in $\mathbb{R}^3$ on which points lie will be no greater than $\sqrt[3]{32768} = 32$. Even if m is close to the machine's largest representable integer (i.e., $m \sim 2^{32}$), then the number of planes in $\mathbb{R}^3$ containing all points generated by the algorithm will be no greater than $\sqrt[3]{2^{32}} \approx 1600$. If you are concerned with a process where all the action happens in a small fraction of the volume, this discreteness of the planes can be very pronounced.

Example 1.1. (IBM `RANDU` Routine) IBM produced a linear congruential generator (called `RANDU`) for its mainframes, with $a = 65539$, $c = 0$ and $m = 2^{31}$. This infamous routine (which was widely copied onto other systems) placed all "randomly" generated points in 3-dimensional space onto only 15 planes! Using the seed $x_1 = 1$, we generate 10000 points in the sequence using recurrence relation (1.4) and these parameters. The following graphs show, the histogram (Figure 1.1) and two and three dimensional scatter plots using successive points as cordinates (Figure 1.2). The serial correlation can be seen clearly in the three dimensional plot.

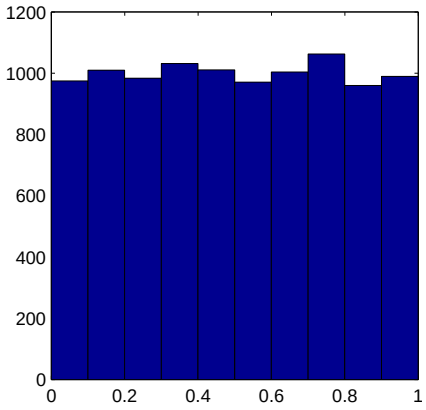


Figure 1.1. Histogram of LCG generated points using IBM `RANDU` parameters.

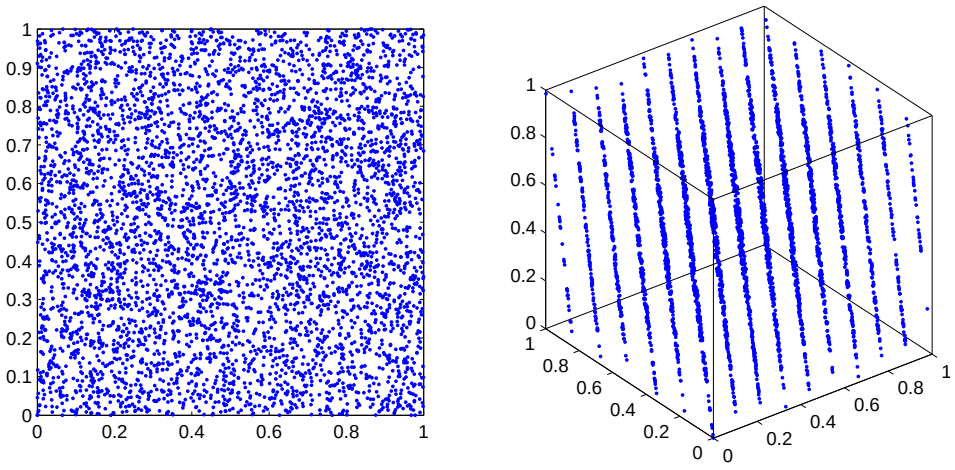


Figure 1.2. Two dimensional (left) and three dimensional (right) scatter plot of LCG generated points using IBM `RANDU` parameters.

1.3. Generating from other Distributions

1.3.1. The Inverse Transform Method. The general method for simulating a random variable having a continuous distribution, called the *inverse transform method*, is based upon the following result.

Theorem 1.2. (INVERSE TRANSFORM METHOD) *Let U be a $\mathscr{U}[0,1)$ random variable. For any continuous distribution function F, define the random variable X by*

$$X := F^{-1}(U).$$

Then X has distribution function F. (Here $F^{-1}(u) = \inf\{\, x : F(x) = u \,\}$.)

Proof. Let G be the probability distribution function of X. Then

$$\begin{aligned} G(a) &= \mathbb{P}[X \le a] \\ &= \mathbb{P}[F^{-1}(U) \le a]. \end{aligned}$$

Now, since F is monotone, $F^{-1}(U) \le a \Leftrightarrow F(F^{-1}(U)) \le F(a)$. Also, since $F(F^{-1}(U)) = U$, we have

$$\begin{aligned} G(a) &= \mathbb{P}[F(F^{-1}(U)) \le F(a)] \\ &= \mathbb{P}[U \le F(a)] \\ &= \int_0^{F(a)} 1 \, du = F(a), \end{aligned}$$

which is exactly what we had to prove. □

From this result it follows that we can generate a random variate X from F, by generating a U from $\mathscr{U}[0,1)$, and then setting $X := F^{-1}(U)$.

Example 1.2. (EXPONENTIAL RANDOM VARIABLE) The density function for an exponential random variable is given by

$$f(x) = \begin{cases} \lambda \exp(-\lambda x), & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

This random variable has mean $\frac{1}{\lambda}$ and variance $\frac{1}{\lambda^2}$. Its distribution function is given by

$$\begin{aligned} F(x) =& \mathbb{P}[X \leq x] \\ =& \int_0^x \lambda \exp(-\lambda y)\, dy \\ =& 1 - \exp(-\lambda x), \end{aligned}$$

for $x \geq 0$. Now, $F^{-1}(u)$ must be that x which satisfies

$$F(x) = 1 - \exp(-\lambda x) = u.$$

Consequently, $\exp(-\lambda x) = 1 - u$, so taking logs and solving for x yields:

$$x = -\frac{1}{\lambda} \log(1 - u).$$

So, if U is $\mathscr{U}[0,1)$, then

$$X := -\frac{1}{\lambda} \log(1 - U),$$

is exponentially distributed with mean $\frac{1}{\lambda}$ and variance $\frac{1}{\lambda^2}$. In fact, since U is uniformly distributed,

$$X := -\frac{1}{\lambda} \log U,$$

will also come from the desired distribution.

The applicability of the inverse transform method relies on finding a tractable expression for F, in the sense that we must be able to determine the inverse of this function. If we can't find an analytic expression for F^{-1}, we can still revert to a numerical inversion procedure. For example, we could determine $X := F^{-1}(U)$, by solving for X in $F(X) = U$ with the *bisection method*. Suppose that the distribution of X is concentrated on $[a, b]$, with F strictly increasing there, and suppose further that we want to determine X with an accuracy of ε.

(1) `Set` $x_{\texttt{low}} := a$, $x_{\texttt{high}} := b$, $X_{\texttt{old}} := b$;

(2) `Generate` $U \sim \mathscr{U}[0,1)$;

(3) `Set` $X_{\texttt{new}} := (x_{\texttt{low}} + x_{\texttt{high}})/2$ `and calculate` $F_{\texttt{new}} := F(X_{\texttt{new}})$;

(4) `If` $|X_{\texttt{new}} - X_{\texttt{old}}| \leq \varepsilon$ `then (return` $X := X_{\texttt{new}}$ `and stop) else (set` $X_{\texttt{old}} := X_{\texttt{new}}$`);`

(5) `If` $F_{\texttt{new}} < U$ `then (set` $x_{\texttt{low}} := X_{\texttt{new}}$`) else (set` $x_{\texttt{high}} := X_{\texttt{new}}$`);`

(6) `Go to step 3;`

This type of procedure slows down random number generation quite considerably, and it is only feasible when efficiency is not a major concern (e.g. when a relatively small sequence of variates is being generated). We illustrate another numerical method, namely *Newton's method,* for determining $\Phi^{-1}(u)$, the inverse of the cumulative distribution function for the standard normal density,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}z^2\right)\, dz.$$

This method quickly yields very accurate approximations and is often used to refine more primitive approximations (from say rational approximations).

Example 1.3. (Inverse normal via Newton) Recall that Newton's method attempts to solve the equation $g(x) = 0$ via a sequence of iterates,

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)},$$

with some specified starting point x_0. In our case we would like to solve the nonlinear equation $\Phi(x) = z$, for some fixed z, so that we have, in terms of previous notation, $g(x) = \Phi(x) - z$ and

$$x_{n+1} = x_n - \frac{\Phi(x_n) - z}{f(x_n)},$$

where $f(x)$ is the standard normal density function. Explicitly, we have

$$x_{n+1} = x_n + (z - \Phi(x_n)) \exp\left(-0.5x_n^2 + \log\left(\sqrt{2\pi}\right)\right).$$

The starting point

$$x_0 = \pm\sqrt{|-1.6\log(1.0004 - (1-2z)^2)|},$$

is recommended[3], the sign being decided as $z \geq 1/2$ or $z < 1/2$.

1.3.2. The Rejection Method. The *rejection method* is a powerful technique, invented by von Neumann[4], for generating random variates from a density function which is known and computable. Unlike the case with the inverse transform method, it doesn't require the corresponding distribution function to be computable (much less the inverse of that function!). Suppose

[3]Marsaglia, G., Zaman, A., and Marsaglia, J.C.W., (1994) Rapid evaluation of the inverse of the normal distribution function, *Statistics and Probability Letters* 19:259-266.

[4]von Neumann, J., (1951) Various techniques used in connection with random digits, *Applied Mathematics Series,* 12, National Bureau of Standards, Washington, D.C.

we have a method for generating random variables having density function g. Under certain conditions, we can use this as a basis for sampling from the continuous distribution having density f, by generating Y from g, and then accepting this value with a probability proportional to $\frac{f(Y)}{g(Y)}$. Specifically, let $c > 1$ be a constant satisfying

$$\frac{f(y)}{g(y)} \leq c,$$

for all y. The following algorithm then generates a random variable having density f.

(1) `Generate` Y `with density` g `and (independent)` $U \sim \mathscr{U}[0,1)$`;`

(2) `If` $U \leq \frac{f(Y)}{cg(Y)}$ `then (set` $X := Y$ `and stop) else (return to step 1);`

We now prove that this method actually works.

Theorem 1.3. (Validity of the rejection method) *The random variable generated by the rejection method has density function f.*

Proof. We compute and find

$$\begin{aligned}\mathbb{P}[X \leq x] &= \mathbb{P}\left[Y \leq x \;\middle|\; U \leq \frac{f(Y)}{cg(Y)}\right] \\ &= \frac{\mathbb{P}\left[Y \leq x \text{ and } U \leq \frac{f(Y)}{cg(Y)}\right]}{\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right]}.\end{aligned}$$

Recalling that the density function of a random variable $\mathscr{U}[0,1)$ is $h(u) = \mathbb{I}_{[0,1)}(u)$, defined by $h(u) = 1$ for $0 \leq u < 1$, and zero otherwise, and by the independence of Y and U, the joint density function of Y and U is

$$g(y)h(u) = \begin{cases} g(y), & \text{if } 0 \leq u < 1, \\ 0, & \text{otherwise.} \end{cases}$$

So, we have

$$\begin{aligned}\mathbb{P}[X \leq x] &= \frac{1}{\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right]} \int_{-\infty}^{x} \int_{0}^{\frac{f(y)}{cg(y)}} g(y)\, du\, dy \\ &= \frac{1}{c\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right]} \int_{-\infty}^{x} f(y)\, dy.\end{aligned}$$

But, if we let $x \to \infty$, then we get

$$1 = \frac{1}{c\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right]} \int_{-\infty}^{\infty} f(y)\, dy = \frac{1}{c\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right]},$$

from the fact that f is a density. Hence $c\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right] = 1$; and so, putting everything together,

$$\mathbb{P}[X \leq x] = \int_{-\infty}^{x} f(y)\, dy,$$

which proves the result. □

The way we "accept the value Y with probability $\frac{f(Y)}{cg(Y)}$" is by generating a $\mathscr{U}[0,1)$ random variable U, and then accepting Y if $U \leq \frac{f(Y)}{cg(Y)}$. Each iteration will (independently) result in an accepted value with probability $\mathbb{P}\left[U \leq \frac{f(Y)}{cg(Y)}\right] = \frac{1}{c}$, sometimes called the "unconditional acceptance probability". It follows that the number of iterations before acceptance has a geometric distribution with mean c.

Example 1.4. (Normal via exponential) To simulate a standard normal random variable Z (that is, one with mean 0 and variance 1), first note that $|Z|$ has probability density

$$f(x) = \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$$

for $0 \leq x < \infty$. We can generate unit normal random variables by first generating random numbers from the density g, using the rejection method, with g being the exponential density with mean 1. In other words we are taking $g(x) := \exp(-x)$ for $0 \leq x < \infty$. Now, consider the following:

$$\begin{aligned}
\frac{f(x)}{g(x)} &= \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2 - 2x}{2}\right) \\
&= \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2 - 2x + 1}{2} + \frac{1}{2}\right) \\
&= \sqrt{\frac{2e}{\pi}} \exp\left(-\frac{1}{2}(x-1)^2\right) \\
&\leq \sqrt{\frac{2e}{\pi}}.
\end{aligned}$$

Hence, we can take $c = \sqrt{\frac{2e}{\pi}}$, and so the conditional probability of accepting $x \sim g$ is

$$\frac{f(x)}{cg(x)} = \exp\left(-\frac{1}{2}(x-1)^2\right).$$

Figure 1.3 shows the probability density function for the absolute value of a standard normal random variate ($f(x)$) and the scaled density for an exponential random variate ($cg(x)$) as well as the conditional probability of accepting a variate generated from $g(x)$.

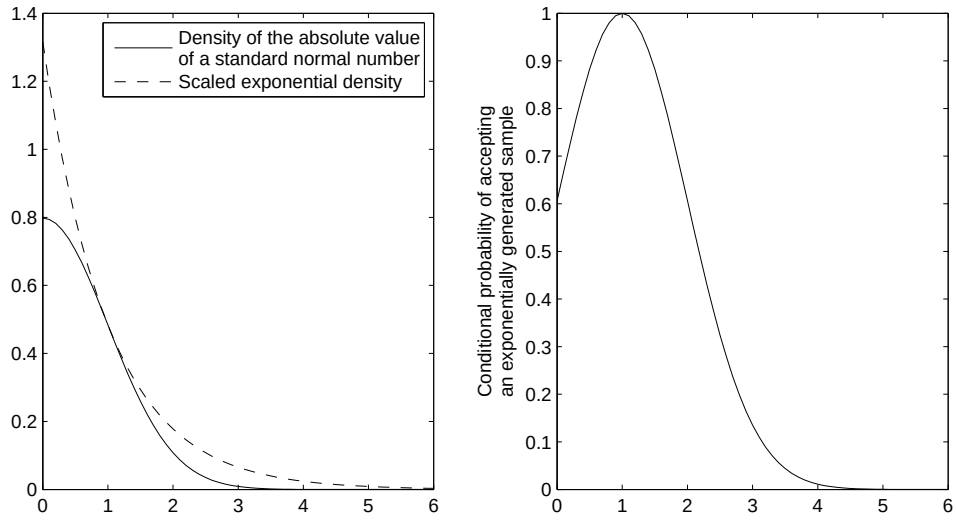


Figure 1.3. Density $f(x)$ and scaled density $g(x)$ (left) and conditional probability of acceptance (right).

So, the rejection method algorithm for simulating the absolute value of a unit normal random variable is:

(1) `Generate independent random variables` Y `and` U, `where` Y `is exponential with rate 1 and` U `is` $\mathscr{U}[0,1)$;

(2) `If` $U \le \exp\left(-\frac{1}{2}(Y-1)^2\right)$ `then (set` $X := Y$ `and stop) else (return to step 1);`

This algorithm is easily adapted for generating $\mathscr{N}(0,1)$ numbers:

(1) `Generate independent random variables` Y `and` U, `where` Y `is exponential with rate 1 and` U `is` $\mathscr{U}[0,1)$;

(2) `If` $U > \exp\left(-\frac{1}{2}(Y-1)^2\right)$ `then (return to step 1);`

(3) `Generate a` $\mathscr{U}[0,1)$ `random variable` V;

(4) `If` $V < \frac{1}{2}$ `then (set` $X := -Y$ `and stop) else (set` $X := Y$ `and stop);`

1.3.3. The Box–Muller Transform. The *Box–Muller transform* [**2**] is specifically for generating normally distributed random variables. Given two independent $\mathscr{U}[0,1)$ random numbers, it produces two independent $\mathscr{N}(0,1)$ random numbers. The algorithm is:

(1) `Generate independent random variables` U_1 `and` U_2 `from` $\mathscr{U}[0,1)$;

(2) `Set` $Z_1 := \cos(2\pi U_1)\sqrt{-2\log U_2}$ `and` $Z_2 := \sin(2\pi U_1)\sqrt{-2\log U_2}$;

To convert a $\mathcal{N}(0,1)$ random variable Z into a $\mathcal{N}(\mu,\sigma^2)$ random variable X, simply set $X := \mu + \sigma Z$ — you will prove that this is true in the next exercise.

Exercise 1.2. (Normal variates via Box-Muller)

(1) Prove that if U_1, $U_2 \sim \mathscr{U}[0,1)$ are independent random variables, then Z_1 and Z_2, given by the algorithm above, are independent $\mathcal{N}(0,1)$ random variables.

(2) The density function of a $\mathcal{N}\left(\mu,\sigma^2\right)$ random variable is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right).$$

Prove that if $Z \sim \mathcal{N}(0,1)$, then $\mu + \sigma Z \sim \mathcal{N}\left(\mu,\sigma^2\right)$. Thus to generate samples from $\mathcal{N}(\mu,\sigma^2)$ it will be enough to be able to generate samples from $\mathcal{N}(0,1)$.

1.3.4. Rational Approximations for Generating Normal Variates. As we have seen above, generating normal variates is made difficult by the fact that the distribution function is not invertible. Using Newton's method or the rejection method are time consuming, and using the Box-Muller transform requires the use of two independent uniform random numbers. The later method is certainly not applicable when using quasi-random sequences. For these reasons, methods that employ rational approximations of the inverse of the cumulative normal distribution function have become popular. The method of Moro [**14**] (see accompanying paper) has become the industry standard in this respect. For even more accuracy, one could use a rational approximation as a first guess to a Newton method (for example, this has been suggested by Glasserman [**5**]).

1.3.5. Generating from a Multivariate Normal Distribution. Multivariate normal distributions will be of great importance in the applications we shall be considering. We review some basic definitions and properties.

Definition 1.3. (Positive definite, Multivariate normal)

(1) A square matrix $A \in \mathbb{R}^{k\times k}$ is called *positive definite* if $\boldsymbol{x}^T A\boldsymbol{x} > 0$ for each $\boldsymbol{x} \in \mathbb{R}^k \setminus \{\mathbf{0}\}$.

(2) We characterize a k-dimensional normal distribution, $\mathcal{N}_k(\boldsymbol{\mu},\Sigma)$ by $\boldsymbol{\mu} \in \mathbb{R}^k$ and $\Sigma \in \mathbb{R}^{k\times k}$, a symmetric and positive definite matrix (the *covariance* matrix). $\mathcal{N}_k(\boldsymbol{\mu},\Sigma)$ has density on $\mathbb{R}^k$ given by

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^k|\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

where $|\Sigma|$ is the determinant of Σ. The *standard* k-dimensional normal, $\mathscr{N}_k(\mathbf{0}, I_k)$, then has density

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^k}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right).$$

Notice that this does indeed generalize the one-dimensional case.

(3) If $\boldsymbol{X}$ is multivariate random variable, i.e., $\boldsymbol{X} \sim \mathscr{N}_k(\boldsymbol{\mu}, \Sigma)$, then the jth component of $\boldsymbol{X}$ has distribution $\mathscr{N}(\mu_j, \sigma_j^2)$, where $\sigma_j^2 = \Sigma_{jj}$. The ith and jth components of $\boldsymbol{X}$ have covariance given by

$$\mathbb{C}\mathrm{ov}[X_i, X_j] = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] = \Sigma_{ij}.$$

The *correlation* between X_i and X_j is, as usual, given by

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j}.$$

We will sometimes choose to specify Σ_{ij} by specifying the correlation matrix (ρ_{ij}) and standard deviations σ_j.

The simplest approach to generating random vectors from $\mathscr{N}_k(\boldsymbol{\mu}, \Sigma)$ relies on finding a suitable decomposition of Σ into the product of two matrices. Such a decomposition is not unique, however. We will consider two ways in which it can be done.

1.3.5.1. *The Square Root Matrix.* Since Σ is symmetric, it can be expanded in what is known as the *spectral decomposition* of Σ,

$$\begin{aligned}\Sigma &= \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^T + \cdots + \lambda_k \boldsymbol{e}_k \boldsymbol{e}_k^T \\ &= P \Lambda P^T,\end{aligned}$$

where $P := (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k)$ is the matrix whose columns are the normalized eigenvectors of Σ, and Λ is the diagonal matrix of corresponding eigenvalues. The matrix

$$\begin{aligned}\Sigma^{\frac{1}{2}} &:= \sqrt{\lambda_1} \boldsymbol{e}_1 \boldsymbol{e}_1^T + \cdots + \sqrt{\lambda_k} \boldsymbol{e}_k \boldsymbol{e}_k^T \\ &= P \sqrt{\Lambda} P^T,\end{aligned}$$

is called the *square root* of Σ.

Exercise 1.3. (Properties of the square root matrix) Verify the following basic properties of the matrix square root.

- $\left(\Sigma^{\frac{1}{2}}\right)^T = \Sigma^{\frac{1}{2}}$;
- $\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} = \Sigma$;
- $\left(\Sigma^{\frac{1}{2}}\right)^{-1} = \frac{1}{\sqrt{\lambda_1}} \boldsymbol{e}_1 \boldsymbol{e}_1^T + \cdots + \frac{1}{\sqrt{\lambda_k}} \boldsymbol{e}_k \boldsymbol{e}_k^T =: P \Lambda^{-\frac{1}{2}} P^T$; and

- $\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}} = \Sigma^{-\frac{1}{2}}\Sigma^{\frac{1}{2}} = I$ and $\Sigma^{-\frac{1}{2}}\Sigma^{-\frac{1}{2}} = \Sigma^{-1}$, where we of course define $\Sigma^{-\frac{1}{2}} := \left(\Sigma^{\frac{1}{2}}\right)^{-1}$.

Now, if $\boldsymbol{Z} \sim \mathscr{N}_k(\mathbf{0}, I)$[5], then $\boldsymbol{X} := \boldsymbol{\mu} + \Sigma^{\frac{1}{2}}\boldsymbol{Z} \sim \mathscr{N}_k(\boldsymbol{\mu}, \Sigma)$. To prove this, note that $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu} + \Sigma^{\frac{1}{2}}\mathbb{E}[\boldsymbol{Z}] = \boldsymbol{\mu}$, while

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\boldsymbol{X}] &= \mathbb{E}\left[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T\right] \\ &= \mathbb{E}\left[\Sigma^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{Z}^T\Sigma^{\frac{1}{2}}\right] \\ &= \Sigma^{\frac{1}{2}}\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{Z}^T\right]\Sigma^{\frac{1}{2}} \\ &= \Sigma.\end{aligned}$$

1.3.5.2. *The Cholesky Decomposition.* If we can factorize Σ as $\Sigma = LL^T$ (clearly, $L := \Sigma^{\frac{1}{2}}$ is a special case of this), then

$$\boldsymbol{Z} \sim \mathscr{N}_k(\mathbf{0}, I) \Rightarrow \boldsymbol{X} := \boldsymbol{\mu} + L\boldsymbol{Z} \sim \mathscr{N}_k(\boldsymbol{\mu}, \Sigma);$$

The proof of this is similar to the case where $L := \Sigma^{\frac{1}{2}}$, above.

Cholesky[6] *decomposition* is a quick technique for factorizing symmetric positive definite matrices (which is the case for Σ). Solving $\Sigma = LL^T$ gives

$$L_{ii} = \sqrt{\Sigma_{ii} - \sum_{n=1}^{i-1} L_{in}^2},$$

and

$$L_{ji} = \frac{1}{L_{ii}}\left(\Sigma_{ij} - \sum_{n=1}^{i-1} L_{in}L_{jn}\right),$$

for $i = 1, \ldots, k$ and $j = i+1, \ldots, k$. The matrix L that results is lower triangular.

Example 1.5. (Cholesky for a bivariate normal) Suppose that Σ is a 2×2 covariance matrix given by

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix},$$

where $\sigma_j > 0$. You may verify that we have $\Sigma = LL^T$, where we write

$$L = \begin{bmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2 \end{bmatrix}.$$

[5]Generating a random vector $\mathscr{N}_k(\mathbf{0}, I)$ is the same thing as generating k independent random numbers from $\mathscr{N}(0,1)$. We already know how to do that.

[6]André-Louis Cholesky (1875-1918) was a French military engineer of Polish origin who specialized in geodesy and surveying. He surveyed much of North Africa.

Golub and van Loan, [**6**, Chapter 4], give a detailed derivation of the Cholesky decomposition for symmetric positive definite matrices, and also consider factorization schemes for more general matrices.

1.3.6. Multivariate Conditioning. Having shown how to generate multivariate normal distributed random variables using the Cholesky decomposition of the covariance matrix, we now present, without proof, a conditioning formula for multivariate normal random variables.

Lemma 1.3. (Conditioning Formula) *Let* $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}^{(1)} \\ \boldsymbol{X}^{(2)} \end{bmatrix} \in \mathbb{R}^k$ *be a vector, where* $\boldsymbol{X}^{(1)}$ *and* $\boldsymbol{X}^{(2)}$ *may also be vectors, and suppose that*

$$\boldsymbol{X} \sim \mathscr{N}_k \left(\begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \boldsymbol{\mu}^{(2)} \end{bmatrix}, \begin{bmatrix} \Sigma^{(11)} & \Sigma^{(12)} \\ \Sigma^{(21)} & \Sigma^{(22)} \end{bmatrix} \right),$$

where the vectors $\boldsymbol{\mu}^{(i)}$ *and the sub-matrices* $\Sigma^{(ij)}$ *have dimensions consistent with the vectors* $\boldsymbol{X}^{(i)}$*. If* $\Sigma^{(22)}$ *has full rank, then*

$$\begin{aligned} &\left(\boldsymbol{X}^{(1)} | \boldsymbol{X}^{(2)} = \boldsymbol{x}\right) \\ &\quad \sim \mathscr{N}_k \Big(\boldsymbol{\mu}^{(1)} + \Sigma^{(12)} \left(\Sigma^{(22)}\right)^{-1} \left(\boldsymbol{x} - \boldsymbol{\mu}^{(2)}\right), \Sigma^{(11)} - \Sigma^{(12)} \left(\Sigma^{(22)}\right)^{-1} \Sigma^{(21)} \Big). \end{aligned} \tag{1.5}$$

In other words, the formula above gives the distribution of $\boldsymbol{X}^{(1)}$ conditional on $\boldsymbol{X}^{(2)} = \boldsymbol{x}$. This relationship will be put to good use when we look at generating Wiener processes using bridging.

1.4. Testing Random Number Generators

1.4.1. Goodness of Fit Tests. So far our analysis of the performance of random number generators has been limited to the period of uniform generators (a 'structural property'). But, there is another important question we must ask when simulating random variables: "Does a sequence of numbers generated look like it conforms to the desired distribution?" A test designed to compare a sample of observed data with a hypothesized distribution to determine whether the distribution "fits" the sample is called a *goodness of fit test*. Here we examine two such tests — the *chi-square test* and the *Kolmogorov-Smirnov test*.

1.4.2. Some Results from Probability Theory.

Definition 1.4. (Gamma function) The *gamma function* is defined by

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t)\, dt,$$

for all $x > 0$.

Exercise 1.4. Prove that $\Gamma(n+1) = n!$ for each $n \in \mathbb{N}$.

Definition 1.5. (Chi-square distribution) The *chi-square density with d degrees of freedom* is

$$f_d(x) = \begin{cases} \frac{1}{\Gamma\left(\frac{d}{2}\right)} \left(\frac{1}{2}\right)^{\frac{d}{2}} x^{\frac{d}{2}-1} \exp\left(-\frac{1}{2}x\right), & \text{if } x > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where $d \in \mathbb{N}$. We denote the corresponding distribution function by χ^2_d.

Theorem 1.4. *Let $\boldsymbol{X} \sim \mathcal{N}_k(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \in \mathbb{R}^k$ and $\Sigma \in \mathbb{R}^{k \times k}$ is symmetric and positive definite. Then $(\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \sim \chi^2_k$.*

Proof. See [**8**, pages 172–173], for example. □

1.4.3. The Chi-Square Test. The chi-square test is one of the simplest available for testing if a given generator is producing random numbers conforming to a desired distribution. It is a very weak test, in the sense that while it may cause us to reject a hypothesis it will give us little reason to accept one. It is however an easily computed statistic, and a good first port of call. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_1 \in \mathbb{R}^k$ be an i.i.d. sample of random variates from some (unknown) distribution G. Suppose we want to test whether this sample fits a given distribution function[7] F. We start by constructing $d+1$ sets $A_1, \ldots, A_{d+1} \subseteq \mathbb{R}^k$, such that

- $\operatorname{supp}(F) = \bigcup_{i=1}^{d+1} A_i$; and
- $\forall 1 \leq i \neq j \leq d+1 \,.\, A_i \cap A_j = \emptyset$.

(Thus, $A_1, \ldots, A_{d+1}$ *partitions* the *support* of F.) For each $1 \leq i \leq d+1$, set $n_i := \#(\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\} \cap A_i) = \sum_{j=1}^n \mathbb{I}_{A_i}(\boldsymbol{X}_j)$, and let $\hat{p}_i := \mathbb{P}^F[A_i]$ and $p_i := \mathbb{P}^G[A_i]$ (the latter probabilities being unknown, of course). Clearly, if $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ fit F well, then we should have $\hat{p}_i \approx p_i$, for each $1 \leq i \leq d+1$. Consequently, the hypothesis we wish to test is

$$\mathscr{H}_0 \,:\, \forall 1 \leq i \leq d+1 \,.\, \hat{p}_i = p_i$$

This may be restated in words as

$$\mathscr{H}_0 : \text{the observed sample fits the distribution } F$$

The next result provides a mechanism for doing this.

Theorem 1.5. (The Pearson statistic tends towards chi-square) *The* Pearson statistic,

$$Q_d := \sum_{i=1}^{d+1} \frac{(n_i - np_i)^2}{np_i},$$

[7] In other words, we wish to test whether $G = F$.

has χ^2_d as its limiting distribution, as $n \to \infty$.

Proof. Look at [**13**, page 445] for an outline of a proof, in the case where $d = 1$. □

By this result, if it is the case that $\mathscr{H}_0$ is true, then we will have that

$$\hat{Q}_d := \sum_{i=1}^{d+1} \frac{(n_i - n\hat{p}_i)^2}{n\hat{p}_i},$$

is distributed approximately χ^2_d, for large enough n. A (one-sided) test of $\mathscr{H}_0$ at the α-th level of significance (or the $(1-\alpha)$-th confidence level) $(0 < \alpha < 1)$ comprises finding $x_{1-\alpha} \geq 0$, such that $\mathbb{P}[X \leq x_{1-\alpha}] = 1 - \alpha$ where X is a χ^2_d random variable. Then we reject $\mathscr{H}_0$ if $\hat{Q}_d > x_{1-\alpha}$. This is called the (one-sided) *chi-square goodness of fit test.*

Example 1.6. (Chi-square test for a uniform generator) The following example is maybe instructive. We consider the purely multiplicative generator[8] given by

$$x_{i+1} = 129140163x_i \pmod{2^{29}}.$$

We first generate 10 000 integers and convert them to numbers, (η_i), distributed in $[0, 1)$, by dividing each of the generated integers by 2^{29}. We then count the number of (η_i), m_j, which satisfy the inequality

$$\frac{(j-1)}{25} \leq \eta_i < \frac{j}{25}, \qquad \text{for } j = 1, 2, \ldots, 25.$$

In terms of our previous notation $d + 1 = 25$, so that the number of degrees of freedom equals twenty-four. The results are tabulated below:

392	423	386	396	425
386	400	393	416	363
411	389	385	363	441
437	387	385	399	416
396	406	405	415	385

We would like to test whether our generator is producing random numbers distributed uniformly in $[0, 1)$ at the $\alpha = 0.05$ level of significance. Notice that the expected number of (η_i) in each of the twenty-five subintervals, $\hat{p}_j n$, is exactly 400. We compute the sample statistic, $\hat{Q}_{24}$, and find that $\hat{Q}_{24} = 23.4$. Computing or looking up the value $x_{0.95}$ we find that $x_{0.95} \approx 36.4$. We cannot therefore, at this significance level, reject the

[8] To obtain a maximal period for such generators we need (as a necessary condition) to take the initial seed, x_1, to be relatively prime to the modulus.

hypothesis that the generator is producing $[0,1)$ uniformly-distributed numbers. Occasionally you will see it stated that one should simply 'reject if $\hat{Q}_d$ is large', since there is then considerable discrepancy between observed and theoretical frequencies. However, one might also be suspicious of anyone claiming to have a generator that consistently produces too small a statistic $\hat{Q}_d$ — particularly $\hat{Q}_d = 0$! To deal with this we could make use of the *two-sided* chi-square test: $\mathcal{H}_0$ at the α-th level of significance (or the $(1-\alpha)$-th confidence level) $(0 < \alpha < 1)$ comprises finding $x_{\frac{\alpha}{2}}$, $x_{1-\frac{\alpha}{2}} \geq 0$, such that $\mathbb{P}\left[X \leq x_{\frac{\alpha}{2}}\right] = \frac{\alpha}{2}$ and $\mathbb{P}\left[X \leq x_{1-\frac{\alpha}{2}}\right] = 1 - \frac{\alpha}{2}$, where X is a χ^2_d random variable. Then we reject $\mathcal{H}_0$ if $\hat{Q}_d < x_{\frac{\alpha}{2}}$ or $\hat{Q}_d > x_{1-\frac{\alpha}{2}}$. This is called the (two-sided) *chi-square goodness of fit test*. For our example, at the $\alpha = 0.10$ level of significance, we have $x_{0.05} = 13.8 < \hat{Q}_{24} < 36.4 = x_{0.95}$, still within bounds, so that our generator is still 'believable'.

1.4.4. The Kolmogorov-Smirnov Test. We again consider a sample $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \in \mathbb{R}^k$ of i.i.d. random variates. Again, we denote the (unknown) distribution function for the population from which this sample is drawn by G — which we now assume to be *continuous*. The *sample distribution function* (or *empirical distribution function*) G_n is defined by

$$G_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{\prod_{j=1}^{k}(-\infty, x_j]}(\boldsymbol{Y}_i),$$

for every $\boldsymbol{x} = (x_1, \ldots, x_k) \in \mathbb{R}^k$. Now, set

$$D_n := \sup\left\{ |G_n(\boldsymbol{x}) - G(\boldsymbol{x})| \,:\, \boldsymbol{x} \in \mathbb{R}^k \right\}.$$

D_n is then a random number that measures how far G_n deviates from G. Denote its distribution function by H_n. We now have the following result.

Theorem 1.6. (Glivenko-Cantelli Theorem: The fundamental theorem of statistics) *With notation as before we have*

$$\mathbb{P}\left[\lim_{n\to\infty} D_n = 0\right] = 1.$$

This tells us that G_n converges uniformly to G, with probability one. In particular, H_n converges to the *Heaviside function*[9] u_0. We now turn our attention to the limiting distribution of $\sqrt{n}D_n$.

[9]The (shifted) *Heaviside function* u_a is given by $u_a(x) = \mathbb{I}_{[a,\infty)}(x)$. This is the distribution function whose corresponding density is the *Dirac delta function* δ_a.

Theorem 1.7. *For each $x \in \mathbb{R}$ we have the following:*

$$\begin{aligned}\lim_{n\to\infty} H_n\left(\frac{x}{\sqrt{n}}\right) &= \lim_{n\to\infty} \mathbb{P}\left[D_n \le \frac{x}{\sqrt{n}}\right] \\ &= \begin{cases} 1 - 2\sum_{i=1}^{\infty}(-1)^{i-1}\exp\left(-2i^2x^2\right), & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases} \\ &=: H(x) \text{ (say).}\end{aligned}$$

Figure 1.4 shows the function $H(x)$, indicating some useful critical values. The important point about this result is that the limiting distribution H of $\sqrt{n}D_n$ does not depend on the distribution G from which the sample $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots$ is drawn (i.e., it is *distribution free*). This suggests that if we wish to test the hypothesis

$$\mathscr{H}_0 : \text{ the observed sample fits the distribution } F$$

for some given distribution function F, we may simply check whether $\sqrt{n}K_n$ is approximately distributed as H (for large enough n), where

$$K_n := \sup\left\{ |G_n(\boldsymbol{x}) - F(\boldsymbol{x})| : \boldsymbol{x} \in \mathbb{R}^k \right\}$$

is called the *Kolmogorov statistic*.

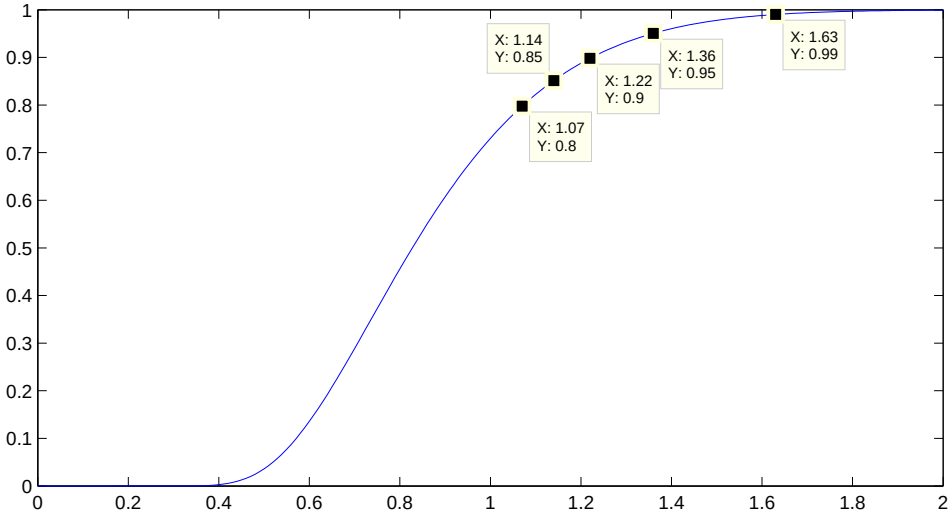


Figure 1.4. The Kolmogorov-Smirnov limiting distribution $H(x)$ showing some important critical values useful for hypothesis testing.

Formally, a one-sided test of $\mathscr{H}_0$ at the α-th level of significance ($0 < \alpha < 1$) comprises finding $x_{1-\alpha} \ge 0$, such that $\mathbb{P}[X \le x_{1-\alpha}] = 1 - \alpha$, where X is an H random variable. Then we reject $\mathscr{H}_0$ if $\sqrt{n}K_n > x_{1-\alpha}$. This is


Figure 1.5. Kolmogorov and Smirnov.

called the (one-sided) *Kolmogorov-Smirnov goodness of fit test.* One can, in the same way as before, set up a two sided test if one were suspicious that the observed fit was too good.

Example 1.7. (K-S Test of normal variates) Consider the situation where we generate $n = 50$ normal random variates using the Box-Muller transform applied to 50 uniform variates produced using the IBM `RANDU` linear congruential generator with initial seed $x_1 = 1$. Note, here we have used the first 25 numbers in the sequence for successive values of U_1 and the remaining 25 numbers for successive values of U_2 in the transform. Figure 1.6 shows the sample distribution G_n along with a standard normal distribution F. We test the hypothesis

$$\mathscr{H}_0 : \text{the observed sample } (G_n) \text{ fits the distribution } F$$

at the $\alpha = 10\%$ level of significance. In this case, from Figure 1.4, the critical value is $x_{1-\alpha} = 1.22$. The graph in Figure 1.6 also shows a region, of width $x_{1-\alpha}/\sqrt{n}$, around the standard normal distribution in which the sample distribution should lie in order to fail to reject the null hypothesis. Since we are dealing with a sample in $\mathbb{R}^1$, we can compute the Kolmogorov statistic as

$$K_n = \max_{1 \le i \le n} \left(F(Z_i) - \tfrac{i-1}{n}, \tfrac{i}{n} - F(Z_i) \right),$$

where $Z_1 \le Z_2 \le \ldots \le Z_n$ are the sorted (smallest to largest) normal samples generated using the procedure above. The value of the Kolmogorov statistic is computed to be $K_n = 0.18$ and thus, since $\sqrt{n}K_n = 1.27 > x_{1-\alpha}$, we reject the hypothesis.

1.4.5. Testing for Randomness. So far we've examined the issues of period and "goodness of fit" for a sample of pseudo-random numbers. However, there is still the question of "randomness". It is quite possible for a sequence to have maximal period, and to fit a target distribution very nicely, yet for the sequence to not "look" random at all. For example, if $Z_1, \ldots, Z_n$

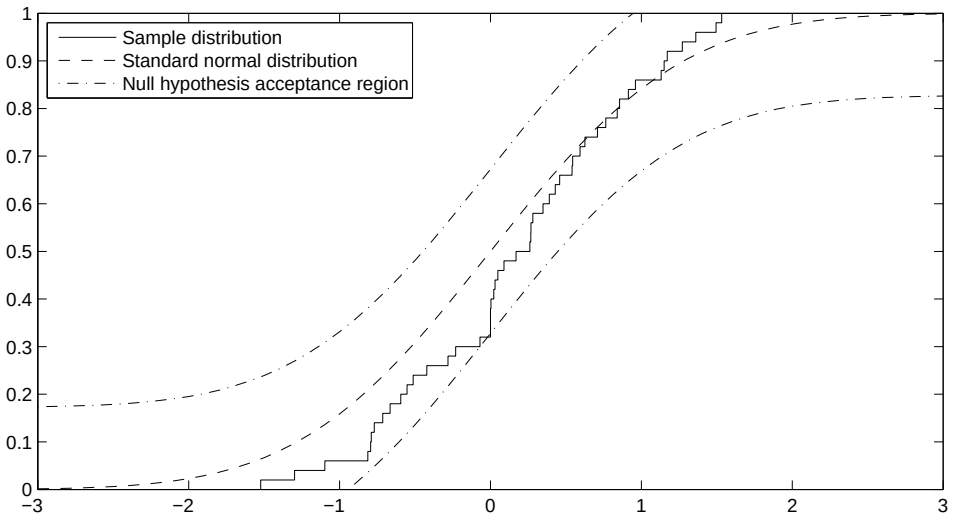


Figure 1.6. Graph showing the sample and standard normal distributions with the region of acceptance for a one-sided Kolmogorov Smirnov test at the $\alpha = 10\%$ level.

is a non-repeating sequence of numbers which fit the $\mathscr{N}(0,1)$ distribution, the same will be true of $Z_1, -Z_1, \ldots, Z_n, -Z_n$. Yet the latter is clearly not random — it displays a pattern! Naively, then, we may think of tests for randomness as tests for the non-existence of patterns.

There are a large number of tests for randomness in a sample of apparently uniformly distributed numbers. [**10**, §3.3.2] surveys a number of these, and points out that it is difficult to choose a best test from among them — some generators will pass certain tests, while failing others. There is, however, one test which is widely regarded as being strong. This is the *runs test*, which will be examined here.

1.4.6. The Runs Test. Let $u_1, \ldots, u_n$ be an i.i.d. sample of distinct[1] $\mathscr{U}[0,1)$ random numbers. A *run of length* r is a subsample $u_i, \ldots, u_{i+r-1}$ $(1 \leq i \leq n+1-r)$, such that $i > 1 \Rightarrow u_{i-1} > u_i$, $i < n+1-r \Rightarrow u_{i+r-1} > u_{i+r}$, and $u_i < \ldots < u_{i+r-1}$. Let R_r denote the number of runs of length r in the sample. For example, the sequence

$$0.2879\,|\,0.0065\;\;0.7342\;\;0.9349\,|\,0.4307\;\;0.8329\,|\,0.5220\,|\,0.3317\;\;0.5639$$
$$0.7001\;\;0.9344\,|\,0.0237\;\;0.1090\;\;0.8884\,|\,0.3439\;\;0.6725\,|\,0.5555\;\;0.9278$$

contains eight runs, with $R_1 = 2$, $R_2 = 3$, $R_3 = 2$, and $R_4 = 1$. The run boundaries have been demarcated using the symbol $|$.

It can be shown (see [**10**, pages 66-69] for details) that

$$\begin{aligned}\mathbb{E}\left[\sum_{r=p}^{n} R_r\right] &= \frac{(n+1)p}{(p+1)!} - \frac{p-1}{p!} \\ &\approx \frac{p}{(p+1)!}n,\end{aligned}$$

for $1 \leq p \leq n$, and consequently that the expected number of runs of length r is given by

$$\begin{aligned}\mathbb{E}\left[R_p\right] &= \frac{(n+1)(p^2+p-1)}{(p+2)!} - \frac{p-1}{p!} + \frac{p}{(p+1)!} \\ &\approx \frac{p^2+p-1}{(p+2)!}n,\end{aligned}$$

where the approximations follow when $p \ll n$. This, along with an analysis of the correlation of runs, forms the basis of the following theorem.

Theorem 1.8. *Let*

$$\boldsymbol{X} = \left[R_1, \ldots, R_5, \sum_{r=6}^{n} R_r\right]^T.$$

Then, in the limit as $n \to \infty$, $\boldsymbol{X}$ *is distributed* $\mathscr{N}_6(\boldsymbol{\mu}, \Sigma)$ *where*

$$\begin{aligned}\boldsymbol{\mu} &= \left[\mathbb{E}[R_1], \mathbb{E}[R_2], \mathbb{E}[R_3], \mathbb{E}[R_4], \mathbb{E}[R_5], \sum_{r=6}^{n} \mathbb{E}[R_r]\right]^T \\ &\approx n\left[\frac{1}{6}, \frac{5}{24}, \frac{11}{120}, \frac{19}{720}, \frac{29}{5040}, \frac{1}{840}\right]^T,\end{aligned}$$

[1]This is not a problem, since $\mathbb{P}[u_i = u_j] = 0$, for $1 \leq i \neq j \leq n$.

and

$$\Sigma \approx \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} & C_1^6 \\ C_{1,2} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} & C_2^6 \\ C_{1,3} & C_{2,3} & C_{3,3} & C_{3,4} & C_{3,5} & C_3^6 \\ C_{1,4} & C_{2,4} & C_{3,4} & C_{4,4} & C_{4,5} & C_4^6 \\ C_{1,5} & C_{2,5} & C_{3,5} & C_{4,5} & C_{5,5} & C_5^6 \\ C_1^6 & C_2^6 & C_3^6 & C_4^6 & C_5^6 & C^{6,6} \end{bmatrix},$$

where

$$C_{p,q} = \mathbb{C}\text{ov}[R_p, R_q] = C_p^q - C_p^{q+1},$$

with

$$C_p^q = \mathbb{C}\text{ov}\left[R_p, \sum_{r=q}^{n} R_r\right] = C^{p,q} - C^{p+1,q}$$

and

$$\begin{aligned} C^{p,q} &= \mathbb{C}\text{ov}\left[\sum_{r=p}^{n} R_r, \sum_{r=q}^{n} R_r\right] \\ &\approx n\left(\frac{\max(p,q)}{(\max(p,q)+1)!} + \frac{(p+q)(1-pq)+pq}{(p+1)!(q+1)!} - \frac{2(p+q)}{(p+q+1)!}\right). \end{aligned}$$

Proof. See [**18**], [**12**] and [**10**, pages 66-69] for details. □

Hence, by Theorem 1.4, $(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})$ must be distributed approximately χ_6^2, for large[2] n. So, in order to test a (long) sequence of numbers $u_1, \ldots, u_n$ for randomness, the runs test is as follows: select a level of significance, and determine whether the above statistic appears to be distributed χ_6^2 up to that level.

Performing the runs test requires the computation of Σ^{-1}. We can perform the runs test on the statistic $(\boldsymbol{X}-\boldsymbol{\mu})^T\hat{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})$, where

$$\hat{\Sigma}^{-1} \approx \frac{1}{n}\begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix},$$

is a commonly used approximation.

[2] It is often suggested that $n > 4000$ is reasonable.

Example 1.8. (RUNS TEST FOR `RANDU`) We apply the runs test to `RANDU`, the purely multiplicative congruential generator specified by

$$x_{n+1} = (2^{16} + 3)x_n \pmod{2^{31}},$$

with $n = 5000$, and various initial seeds. This generally 'bad' random number generator holds up surprisingly well, but requires a careful choice of initial seed. For instance, with seed $x_1 = 2$, we obtain the following runs vector, $\boldsymbol{X}$, from MATLAB:

$$\boldsymbol{X} = \begin{bmatrix} 815 \\ 1040 \\ 448 \\ 132 \\ 35 \\ 9 \end{bmatrix}.$$

Computing, we find that $(\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \approx 9.16$ (9.15 if $\hat{\Sigma}^{-1}$ is used). Consulting percentile values for the chi-square distribution (or, computing them using Matlab) we find that $\mathbb{P}[X \leq 12.59] = 0.95$, and $\mathbb{P}[X \leq 1.64] = 0.05$, so that the generator passes the two-sided test at the 10% level of significance. For a different choice of seed, $x_1 = 1$, we compute and obtain the statistic, $(\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \approx 19.25$ (19.22 if $\hat{\Sigma}^{-1}$ is used). So, we would, at the same level of significance, reject in this case.

Chapter 2

Generating Stock Prices

The idea for what was later called the Monte Carlo method occurred to me when I was playing solitaire during my illness. I noticed that it may be much more practical to get an idea of the probability of the successful outcome of a solitaire game (like Cranfield or some other where the skill of the player is not important) noticing what proportion comes out successfully, rather than to try to compute all the combinatorial possibilities which are of an exponentially increasing number so great that, except in very elementary cases, there is no way to estimate it. This is intellectually surprising, and if not exactly humiliation, it gives one a feeling of modesty about the limits of rational or traditional thinking. In a sufficiently complicated problem, actual sampling is better than examination of all the chains of possibilities.... One can write differential equations or integral differential equations for the "expected values," but to solve *them or even to try and get an approximate idea of the properties of the solution, is an entirely different matter. It seems to me that the name Monte Carlo contributed very much to the popularization of this procedure, It was named Monte Carlo because of the element of chance, the production of random numbers with which to play the suitable games.*

— Stanislaw Ulam, *Adventures of a Mathematician.*

2.1. Introduction

The most common model for the behaviour of the price of a non-dividend-paying stock is *geometric Brownian motion*. Let S_t be a random variable denoting the price at time $t \geq 0$ of some (non-dividend-paying) stock[1]. (We assume that S_0 is degenerate.) Within the framework of geometric Brownian motion, S_t is expressed as the solution of an *SDE (stochastic differential equation)*

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t, \qquad S_0 \text{ a constant.} \tag{2.1}$$

Here, μ is called the *drift rate*[2] of the stock price, and σ is called its *volatility*. W_t $(t \geq 0)$ is a stochastic process known as a *Brownian motion* or a *Wiener process*.

Definition 2.1. (Brownian motion, Wiener process) A stochastic process W_t $(t \geq 0)$ is called a (standard) *Brownian motion* or a *Wiener process* if

(1) $W_0 = 0$;

(2) For all $0 \leq t_1 < t_2 \leq t_3 < t_4$, $W_{t_4} - W_{t_3}$ and $W_{t_2} - W_{t_1}$ are independent[3];

(3) For all $0 \leq t_1 < t_2$, $W_{t_2} - W_{t_1} \sim \mathcal{N}(0, t_2 - t_1)$; and

(4) W has continuous trajectories.

Figure 2.1. Norbert Wiener.

The SDE (2.1) is amenable to analytic solution — this is a large part of its attraction as a model for stock price behaviour. To solve the SDE, we require one of the fundamental tools of stochastic analysis — *Itō's formula*.

[1] Such a time-indexed family random of variables forms a *stochastic process*. When we wish to refer to the entire process, we simply drop the subscript.

[2] That is jargon from stochastic differential equations. As we shall see, in a financial context, it is useful to think of μ as the expected (continuous-time) rate of return earned by an investor in the stock.

[3] In other words, a Brownian motion has *independent increments*.

Definition 2.2. (ITŌ PROCESSES) A stochastic process X is called an *Itō process* if it is the solution of an SDE

$$dX_t = a(t, X_t)\,dt + b(t, X_t)\,dW_t, \quad X_0 \text{ a constant}, \tag{2.2}$$

where W is a Brownian motion. In order for (2.2) to be well-defined, the processes $a(.,.)$ and $b(.,.)$ must also satisfy certain boundedness criteria — but we won't concern ourselves with such technicalities here.

Clearly, (2.2) is a generalization of (2.1), implying that geometric Brownian motion is an Itō process.[4]

Figure 2.2. Kiyoshi Itō, inventor of stochastic integration.

Theorem 2.1. (ONE-DIMENSIONAL ITŌ FORMULA) *Let X be an Itō process defined by (2.2) and $g \in C^{1,2}([0,\infty) \times \mathbb{R})$. Then the process $g(.,X)$ follows the SDE*

$$\begin{aligned} dg(t,X_t) &= \frac{\partial g}{\partial t}(t,X_t)\,dt + \frac{\partial g}{\partial x}(t,X_t)\,dX_t + \frac{1}{2}\frac{\partial^2 g}{\partial x^2}(t,X_t)(dX_t)^2 \\ &= \frac{\partial g}{\partial t}(t,X_t)\,dt + \frac{\partial g}{\partial x}(t,X_t)(a(t,X_t)\,dt + b(t,X_t)\,dW_t) \\ &\quad + \frac{1}{2}\frac{\partial^2 g}{\partial x^2}(t,X_t)(a(t,X_t)\,dt + b(t,X_t)\,dW_t)^2 \\ &= \left(\frac{\partial g}{\partial t}(t,X_t) + a(t,X_t)\frac{\partial g}{\partial x}(t,X_t) + \frac{1}{2}b^2(t,X_t)\frac{\partial^2 g}{\partial x^2}(t,X_t)\right)\,dt \\ &\quad + b(t,X_t)\frac{\partial g}{\partial x}(t,X_t)\,dW_t, \end{aligned}$$

and is also an Itō process. The above 'derivation' uses the heuristic rules $(dt)^2 = 0$, $dt\,dW_t = 0$ and $(dW_t)^2 = dt$.

[4] Set $a(.,.) := \mu X$ and $b(.,.) := \sigma X$.

Circumstances now force us to confront the dirty little secret of stochastic calculus: SDEs don't really exist! If we were to take (2.2) at face value, we would naturally be inclined to rewrite it as

$$\frac{dX_t}{dt} = a(t, X_t) + b(t, X_t)\frac{dW_t}{dt}, \qquad X_0 \text{ a constant.}$$

But, this would be nonsense, since sample paths of Brownian motions are *nowhere differentiable* (even though they are continuous)[5] — in other words, $\frac{dW_t}{dt}$ does not exist!

In truth, (2.2) is no more than a convenient shorthand for the *stochastic integral equation*

$$X_t = X_0 + \int_0^t a(s, X_t)\, ds + \int_0^t b(s, X_t)\, dW_s,$$

where the second integral above is called an *Itō integral* (the precise construction of which we leave for a more rigorous course on stochastic calculus). It is worth noting that one application of Itō's formula allows the easy computation of certain Itō integrals.

Example 2.1. (Using Itō's lemma)

(1) Let $g(t, x) := x$. Then $\frac{\partial g}{\partial t} = 0$, $\frac{\partial g}{\partial x} = 1$, and $\frac{\partial^2 g}{\partial x^2} = 0$. So,

$$dg(t, W_t) = dW_t.$$

In integral form, this gives

$$g(t, W_t) = g(0, W_0) + \int_0^t dW_s,$$

which shows that $\displaystyle\int_0^t dW_s = W_t$.

(2) Let $g(t, x) := \frac{1}{2}x^2$. Then $\frac{\partial g}{\partial t} = 0$, $\frac{\partial g}{\partial x} = x$, and $\frac{\partial^2 g}{\partial x^2} = 1$. So,

$$dg(t, W_t) = \frac{1}{2}\, dt + W_t\, dW_t,$$

In integral form, this gives

$$g(t, W_t) = g(0, W_0) + \int_0^t \frac{1}{2}\, ds + \int_0^t W_s\, dW_s,$$

which gives us

$$\int_0^t W_s\, dW_s = \frac{1}{2}W_t^2 - \frac{1}{2}t.$$

[5]Technically speaking, the correct terms are "non-differentiable *almost everywhere*" and "continuous *almost everywhere*".

Exercise 2.1. Demonstrate that

$$\int_0^t W_s^2\, dW_s = \frac{1}{3}W_t^3 - \int_0^t W_s\, ds,$$

using Itō's formula.

Exercise 2.2. (Arithmetic Brownian motion) Solve the SDE

$$dS_t = \mu\, dt + \sigma\, dW_t, \qquad S_0 \text{ a constant.}$$

This process is called *arithmetic Brownian motion* or *Brownian motion with drift*. The discipline of stochastic finance was initiated by the French mathematician Louis Bachelier, who in [**1**] was the first person to develop the mathematical theory of Brownian motion (five years before Einstein's famous paper of 1905!) In his thesis, Bachelier modelled stock prices as arithmetic Brownian motions. Unfortunately, as a stock price model, arithmetic Brownian motion suffers from a serious flaw. Can you think what it is? The economist Paul Samuelson introduced geometric Brownian motion in [**16**] as a remedy for this problem.

Let us return now to the problem of solving the SDE for geometric Brownian motion (2.1). To do so, first set $g(t,x) := \log x$. Then, $\frac{\partial g}{\partial t} = 0$, $\frac{\partial g}{\partial x} = \frac{1}{x}$ and $\frac{\partial^2 g}{\partial x^2} = -\frac{1}{x^2}$. Consequently, by Itō's formula,

$$\begin{aligned} dg(t,S_t) =& \frac{1}{S_t}\, dS_t + \frac{1}{2}\left(-\frac{1}{S_t^2}\right)(dS_t)^2 \\ =& \frac{1}{S_t}(\mu S_t\, dt + \sigma S_t\, dW_t) - \frac{1}{2S_t^2}\sigma^2 S_t^2\, dt \\ =& \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma\, dW_t. \end{aligned}$$

In integral form,

$$g(t,S_t) = g(0,S_0) + \int_0^t \left(\mu - \frac{1}{2}\sigma^2\right) ds + \int_0^t \sigma\, dW_s.$$

In other words, applying Example 2.1 (1),

$$\log S_t = \log S_0 + \left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t, \tag{2.3}$$

and so

$$S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right). \tag{2.4}$$

This is then the solution of (2.1). It can be shown (see Exercise 2.3) that $\mathbb{E}[S_t] = S_0 \exp(\mu t)$. This explains the earlier comment that one may regard the drift rate in (2.1) as the rate of return a share-holder in the stock

can expect. It is also possible to show that $\mathbb{V}\mathrm{ar}[S_t] = S_0^2 e^{2\mu t}(e^{\sigma^2 t} - 1)$, see Exercise 2.4 below for details.

Exercise 2.3. (Expected Value of Geometric Brownian Motion) By directly integrating the expression in (2.4) multiplied by the appropriate normal density function show that $\mathbb{E}[S_t] = S_0 \exp(\mu t)$.

Exercise 2.4. (Higher Moments Of Geometric Brownian Motion) Assuming the SDE given by (2.1), show that

$$\mathbb{E}[S_t^n] = S_0^n \exp\left(\left(n\mu + \frac{1}{2}n(n-1)\sigma^2\right)t\right),$$

for $n \in \mathbb{N}$ and S_0 a constant. Now, show that

$$\mathbb{V}\mathrm{ar}[S_t] = S_0^2 e^{2\mu t}\left(e^{\sigma^2 t} - 1\right).$$

For most of the examples in these notes we will regard geometric Brownian motion as the canonical paradigm for modelling stock prices. Before we continue, however, let us highlight some of the shortcomings of (2.1) as a description of stock price behaviour; *for it has many shortcomings, and they are important.*

- Since Brownian motions have continuous sample paths, it is clear that (2.4) forces the same to be true of S. This is at odds with observed reality. While market prices are generally continuous over time, they do occasionally "gap" (usually down). This phenomenon is especially prevalent during crashes, when prices can become very discontinuous.
- One of the major objections to (2.1) is the fact that σ is itself not a stochastic process. (Of course, making it stochastic would render the entire model less tractable — and consequently less attractive. This trade-off between expediency and accuracy is typical of all mathematical modelling.) In reality, price volatility (both historical and implied) is itself very "volatile". Typically, volatilities are negatively correlated with returns[6] — this is known as the leverage effect.
- From (2.3) and Definition 2.1, we can conclude that the log-stock prices are *normally* distributed[7] with mean $\log S_0 + \left(\mu - \frac{1}{2}\sigma^2\right)t$ and variance $\sigma^2 t$. Furthermore, we can conclude that the (continuously

[6] During crashes they balloon dramatically. Such events are catastrophic for option sellers who have not "bought their tails" — i.e., bought out-of-the-money puts to get some long volatility exposure.

[7] This is the reason why (2.1) is often referred to as the *log-normal model* for a stock price.

compounded) stock price returns are normally distributed

$$\log \frac{S_t}{S_0} \sim \mathscr{N}\left(\left(\mu - \frac{1}{2}\sigma^2\right)t, \sigma^2 t\right).$$

This distribution suffers from "narrow tails" — meaning that it tends to underestimate the likelihood of either large returns or large losses. Furthermore, this distribution is symmetric, which is not true in practice — markets, after all, tend to crash downwards, not upwards. Market-makers in options (those whose pricing models are implicitly built upon (2.1)) compensate for these shortcomings by applying different volatilities to the pricing of instruments with different strikes (on the same underlying asset)[8]. This is one of the most convincing explanations for the phenomenon known as the *volatility smile* (the non-constant function of *implied volatility* with respect to strike).

2.2. Generating Prices for a Single Stock

Since $W_t = W_t - W_0 \sim \mathscr{N}(0, t)$, (2.4) indicates that if $Z \sim \mathscr{N}(0, 1)$, then

$$S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma\sqrt{t}\, Z\right), \tag{2.5}$$

will be a random variable satisfying (2.1). Since we can already generate $\mathscr{N}(0, 1)$ random variables, we can thus also simulate stock prices.

Example 2.2. Consider a stock price described by (2.1), with a drift rate of $\mu = 20\%$, a volatility[9] of $\sigma = 45\%$, and a current price of $S_0 = \$100.00$. We can use (2.5) to simulate the distribution of stock prices after half a year ($t = \frac{1}{2}$). To do so, generate $Z_1, \ldots, Z_n \sim \mathscr{N}(0, 1)$, and compute the stock realisations

$$S_{t,i} := S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\sqrt{t} Z_i\right),$$

for $1 \leq i \leq n$. Figure 2.3 shows a histogram of the resulting prices. Note, in particular, its "log-normal" shape.

Note the conventions for quoting time (in units of a year), drift rate and volatility (as percentages) in the above example; and what these values mean, in terms of (2.4).

[8] Essentially, they tend to select larger volatilities for pricing out-of-the-money options — thus inflating prices, and thereby compensating for the fact that the log-normal distribution underestimates the chances of exercise for such instruments.

[9] These are reasonable parameter values for the equity market.

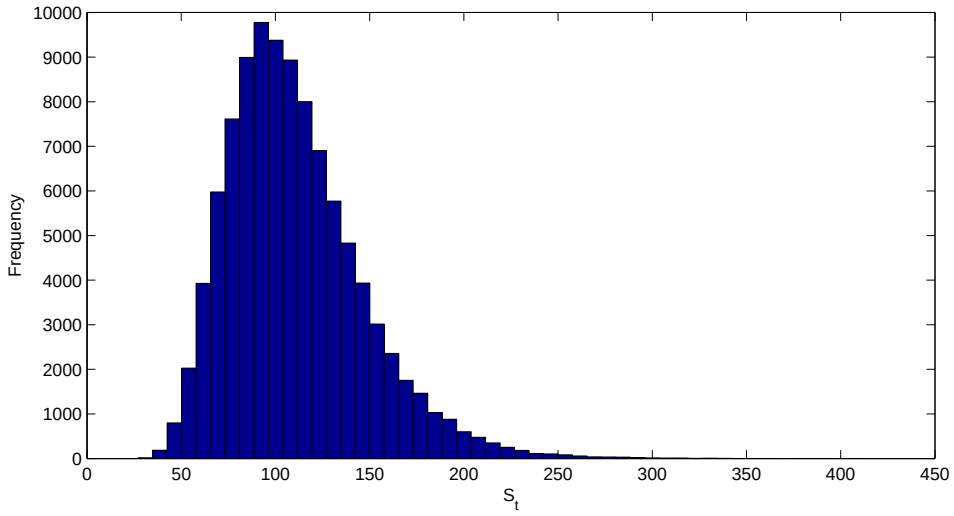


Figure 2.3. A histogram of stock prices in half a year from the present, for a stock with a current price of \$100.00, a drift rate of 20%, and a volatility of 45%. The number of sample prices generated was 100 000.

2.3. Generating Correlated Stock Prices

Let us now consider a family of k stocks, whose price processes are determined by the SDEs

$$dS_t^{(i)} = \mu_i S_t^{(i)}\,dt + \sigma_i S_t^{(i)}\,dW_t^{(i)}, \qquad S_0^{(i)} \text{ a constant}, \tag{2.6}$$

for $1 \le i \le k$, where $W^{(1)}, \ldots, W^{(k)}$ are Brownian motions.

In general, the price processes won't be independent; and whatever statistical dependence there is must arise from a dependence between the Brownian motions. However, statistical dependence between stock prices is conventionally expressed as the correlation structure of stock returns. In other words, the pairwise correlations

$$\rho_{ij} := \mathbb{C}\mathrm{orr}\left[\log\frac{S_{t+\Delta t}^{(i)}}{S_t^{(i)}}, \log\frac{S_{t+\Delta t}^{(j)}}{S_t^{(j)}}\right], \quad (\Delta t > 0),$$

for $1 \le i \ne j \le k$, are specified. The question we must ask is: how does this translate into a correlation structure on the Brownian motions? Clearly, if we are to simulate $\boldsymbol{S} = \left(S^{(1)}, \ldots, S^{(k)}\right)^T$, we must know the answer to this question.

Since (as we have already seen)

$$\log\frac{S_{t+\Delta t}^{(i)}}{S_t^{(i)}} \sim \mathcal{N}\left(\left(\mu_i - \frac{1}{2}\sigma_i^2\right)\Delta t, \sigma_i^2 \Delta t\right),$$

we have that

$$\mathbb{SD}\text{ev}\left[\log\frac{S^{(i)}_{t+\Delta t}}{S^{(i)}_t}\right]=\sigma_i\sqrt{\Delta t}.$$

Thus, using the short hand $\Delta W^{(i)}_t = W^{(i)}_{t+\Delta t} - W^{(i)}_t$, for $1 \le i \ne j \le k$ we have that

$$\begin{aligned}
\mathbb{C}\text{orr}\left[\log\frac{S^{(i)}_{t+\Delta t}}{S^{(i)}_t},\log\frac{S^{(j)}_{t+\Delta t}}{S^{(j)}_t}\right] =& \frac{\mathbb{C}\text{ov}\left[\log\frac{S^{(i)}_{t+\Delta t}}{S^{(i)}_t},\log\frac{S^{(j)}_{t+\Delta t}}{S^{(j)}_t}\right]}{\mathbb{SD}\text{ev}\left[\log\frac{S^{(i)}_{t+\Delta t}}{S^{(i)}_t}\right]\mathbb{SD}\text{ev}\left[\log\frac{S^{(j)}_{t+\Delta t}}{S^{(j)}_t}\right]}\\
=&\frac{1}{\sigma_i\sigma_j\Delta t}\mathbb{C}\text{ov}\left[\log\frac{S^{(i)}_{t+\Delta t}}{S^{(i)}_t},\log\frac{S^{(j)}_{t+\Delta t}}{S^{(j)}_t}\right]\\
=&\frac{1}{\sigma_i\sigma_j\Delta t}\mathbb{C}\text{ov}\left[\left(\mu_i-\frac{1}{2}\sigma_i^2\right)\Delta t+\sigma_i\Delta W^{(i)}_t,\right.\\
&\qquad\left.\left(\mu_j-\frac{1}{2}\sigma_j^2\right)\Delta t+\sigma_j\Delta W^{(j)}_t\right]\\
=&\frac{1}{\sigma_i\sigma_j\Delta t}\mathbb{C}\text{ov}\left[\sigma_i\Delta W^{(i)}_t,\sigma_j\Delta W^{(j)}_t\right]\\
=&\frac{1}{\Delta t}\mathbb{C}\text{ov}\left[\Delta W^{(i)}_t,\Delta W^{(j)}_t\right]\\
=&\frac{1}{\Delta t}\mathbb{SD}\text{ev}\left[\Delta W^{(i)}_t\right]\mathbb{SD}\text{ev}\left[\Delta W^{(j)}_t\right]\\
&\qquad\times\mathbb{C}\text{orr}\left[\Delta W^{(i)}_t,\Delta W^{(j)}_t\right]\\
=&\mathbb{C}\text{orr}\left[\Delta W^{(i)}_t,\Delta W^{(j)}_t\right].
\end{aligned}$$

In other words, correlations between share price returns are the same thing as correlations between Brownian motion increments. Now, let Σ be the correlation matrix determined by ρ_{ij} $(1 \le i \ne j \le k)$. Then, if $\boldsymbol{Z} = \left[Z^{(1)},\ldots,Z^{(k)}\right]^T \sim \mathscr{N}_k(\mathbf{0},\Sigma)$, the solution to the system of SDEs (2.6) — subject to the given correlation structure — is $\boldsymbol{S}_t = \left[S^{(1)}_t,\ldots,S^{(k)}_t\right]^T$, where

$$S^{(i)}_t = S^{(i)}_0\exp\left(\left(\mu_i-\frac{1}{2}\sigma_i^2\right)t+\sigma_i\sqrt{t}\,Z^{(i)}\right), \tag{2.7}$$

for $1 \le i \le k$.

Example 2.3. (Correlated stocks) Consider a pair of stocks, whose price processes $S^{(1)}$ and $S^{(2)}$ are specified by (2.6) (with $k = 2$). Suppose these processes exhibit drift rates of $\mu_1 = 15\%$ and $\mu_2 = 30\%$ (respectively), volatilities of $\sigma_1 = 35\%$ and $\sigma_2 = 50\%$ (respectively), and that their respective initial values are $S^{(1)}_0 = \$100.00$ and $S^{(2)}_0 = \$50.00$. Denote the

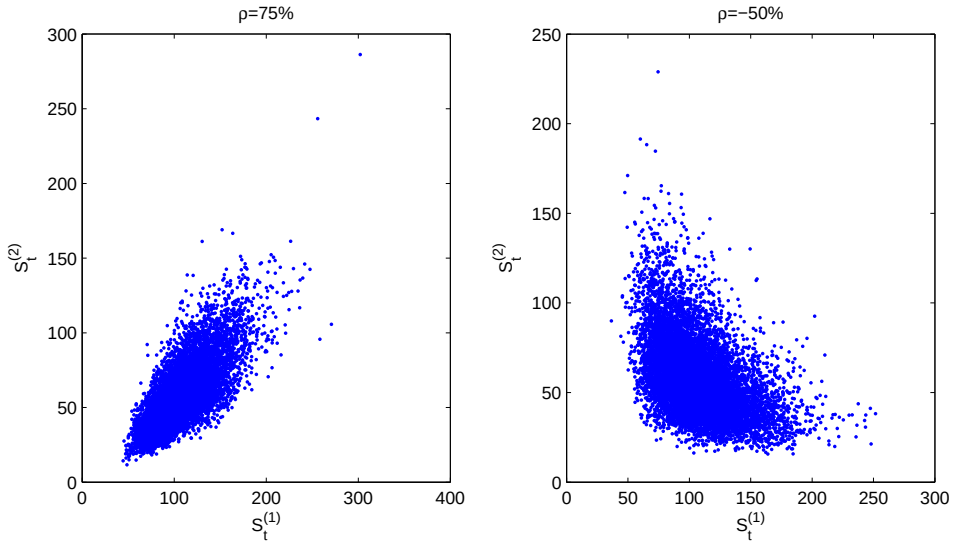


Figure 2.4. Scatter plot of simulated stock prices in half a year, for two stocks $S^{(1)}$ and $S^{(2)}$, whose returns have a 75% correlation (left) and a −50% correlation (right). The number of samples was 10 000.

correlation between their instantaneous returns by ρ. To simulate n realisations of the stock prices after half a year ($t = \frac{1}{2}$), generate

$$\boldsymbol{Z}_1 = \left[Z_1^{(1)}, Z_1^{(2)}\right]^T, \ldots, \boldsymbol{Z}_n = \left[Z_n^{(1)}, Z_n^{(2)}\right]^T \sim \mathcal{N}_2(\mathbf{0}, I),$$

and compute the Cholesky decomposition, $LL^T = \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Then set

$$\boldsymbol{X}_1 := L\boldsymbol{Z}_1, \ldots, \boldsymbol{X}_n := L\boldsymbol{Z}_n \sim \mathcal{N}_2(\mathbf{0}, \Sigma).$$

Finally, by (2.7),

$$\boldsymbol{S}_{t,i} = \begin{bmatrix} S_{t,i}^{(1)} \\ S_{t,i}^{(2)} \end{bmatrix} := \begin{bmatrix} S_0^{(1)} \exp\left(\left(\mu_1 - \frac{1}{2}\sigma_1^2\right) t + \sigma_1 \sqrt{t} X_i^{(1)}\right) \\ S_0^{(2)} \exp\left(\left(\mu_2 - \frac{1}{2}\sigma_2^2\right) t + \sigma_2 \sqrt{t} X_i^{(2)}\right) \end{bmatrix}$$

for $1 \le i \le n$, are the resulting stock price samples. Figure 2.4 shows scatter plots of the prices generated, for correlations of 75% and −50%.

2.4. Generating Stock Price Paths

In this section we consider the problem of generating a *sample path*, up to some future time $T > 0$, for the process determined by (2.1). Specifically, for some $N \in \mathbb{N}$, we want to simulate the sequence[1] $S_0, S_1, \ldots, S_N$, at time

[1]Note, stock price subscripts here are specified as indices, not times. From time to time we shall use this convention, indicating the times associated with the indices in a sperate time vector.

points $0 = t_0, t_1, \ldots, t_N = T$ with time increments $\Delta t_i = t_i - t_{i-1}$. By repeated application of (2.4), this sequence is given by

$$\begin{aligned} S_i &= S_{i-1} \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t_i + \sigma\sqrt{\Delta t_i} Z_i\right) \\ &= S_0 \prod_{j=1}^{i} \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t_j + \sigma\sqrt{\Delta t_j} Z_j\right) \\ &= S_0 \exp\left(\sum_{j=1}^{i}\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t_j + \sigma\sqrt{\Delta t_j} Z_j\right), \end{aligned} \tag{2.8}$$

for $1 \leq i \leq N$, and where $Z_1, \ldots, Z_N \sim \mathcal{N}(0,1)$ are independent random variables. The above approach for generating stock processes explicitly uses the property of independent increments for Brownian motions.

For reasons that will become apparent later in the text, it is sometimes advantageous to fix/generate the end points of the stock price process and use conditioning to fill in intermediate values. This can be achieved by generating Brownian bridges and then determining the corresponding geometric Brownian path by an application of (2.4).

2.4.1. Brownian Bridges. We start by considering the simple case of generating an intermediate value for a Wiener process conditional on knowing the start and end points. More specifically, we wish to compute W_m conditional on knowing $W_l = x$ and $W_r = y$, where $0 \leq t_l \leq t_m \leq t_r \leq T$.

As a result of the independent increments property of Brownian motion (Definition 2.1) we have

$$\begin{aligned} \mathbb{C}\mathrm{ov}[W_l, W_m] &= \mathbb{C}\mathrm{ov}[W_l, W_l + (W_m - W_l)] \\ &= \mathbb{C}\mathrm{ov}[W_l, W_l] + \mathbb{C}\mathrm{ov}[W_l, W_m - W_l] \\ &= t_l. \end{aligned}$$

Similarly, $\mathbb{C}\mathrm{ov}[W_l, W_r] = t_l$ and $\mathbb{C}\mathrm{ov}[W_m, W_r] = t_m$. We may, therefore, conclude that

$$\begin{bmatrix} W_m \\ W_l \\ W_r \end{bmatrix} \sim \mathcal{N}_3\left(0, \begin{bmatrix} t_m & t_l & t_m \\ t_l & t_l & t_l \\ t_m & t_l & t_r \end{bmatrix}\right).$$

We are now in the position to use the conditioning formula (1.5) to find the distribution of W_m given that $W_l = x$ and $W_r = y$. The mean is given

by

$$\begin{aligned}\mathbb{E}[W_m|W_l = x, W_r = y] &= 0 + \begin{bmatrix} t_l & t_m \end{bmatrix} \begin{bmatrix} t_l & t_l \\ t_l & t_r \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \frac{(t_r - t_m)x + (t_m - t_l)y}{t_r - t_l}, \end{aligned} \tag{2.9}$$

and the variance by

$$\begin{aligned}\mathbb{V}\mathrm{ar}[W_m|W_l = x, W_r = y] &= t_m - \begin{bmatrix} t_l & t_m \end{bmatrix} \begin{bmatrix} t_l & t_l \\ t_l & t_r \end{bmatrix}^{-1} \begin{bmatrix} t_l \\ t_m \end{bmatrix} \\ &= \frac{(t_m - t_l)(t_r - t_m)}{t_r - t_l}, \end{aligned} \tag{2.10}$$

since

$$\begin{bmatrix} t_l & t_l \\ t_l & t_r \end{bmatrix}^{-1} = \frac{1}{(t_r - t_l)t_l} \begin{bmatrix} t_r & -t_l \\ -t_l & t_l \end{bmatrix}.$$

Note that (2.9) is essentially a linear interpolation between the points (t_l, x) and (t_r, y).

Now that we know the conditional distribution of W_m given that $W_l = x$ and $W_r = y$, we can envisage procedures for generating the intermediate values W_i, $0 < i < N$ of a Brownian bridge with endpoints $W_0 = 0$ and W_N, for times $0 = t_0, t_1, \ldots, t_N = T$. Naively, one could just compute the updates W_i successively with a knowledge of W_{i-1} and W_N but this turns out not to be the best approach (for reasons to be revealed later).

A better approach is to find the time point (with index m) in the sequence that is closest to the middle of the time interval $[t_0, t_N]$ and then compute W_m given the values for W_0 and W_N. This procedure can then be repeated recursively for the left and right subintervals $[t_0, t_m]$ and $[t_m, t_N]$. Assuming that the time intervals $\Delta t_i = t_i - t_{i-1}$ are of equal length, the following recursive algorithm implements this approach for a single path.

`Function` $\boldsymbol{W}$ =`BrownianBridge(`$\boldsymbol{t}$`,`W_0`,`W_N`,`$\boldsymbol{Z}$`)`

(1) `If` $\boldsymbol{Z} = []$ `then return` $\boldsymbol{W} = []$`; otherwise`

(2) `Using` $\boldsymbol{t}$`, determine` N `and set` $m = \lfloor (N+1)/2 \rfloor$`;`

(3) `Set` $W_m = \frac{(t_N - t_m)W_0 + (t_m - t_0)W_N}{t_N - t_0} + \left(\frac{(t_m - t_0)(t_N - t_m)}{t_N - t_0}\right)^{1/2} Z_m$`;`

(4) `Set`
$\boldsymbol{W}_l$`=BrownianBridge(`$[t_0, \ldots, t_m]$`,`W_0`,`W_m`,`$[Z_1, \ldots, Z_{m-1}]$`);`
`and`
$\boldsymbol{W}_r$`=BrownianBridge(`$[t_m, \ldots, t_N]$`,`W_m`,`W_N`,`$[Z_{m+1}, \ldots, Z_{N-1}]$`);`

(5) `Concatenate` $\boldsymbol{W}_l$`,` W_m `and` $\boldsymbol{W}_r$ `to form` $\boldsymbol{W}$`.`

The above function is called with $\boldsymbol{t} := [t_0, t_1, \ldots, t_N]$; W_0 and W_N, which are the values for the Wiener process at times t_0 and t_N; and $\boldsymbol{Z} := [Z_1, \ldots, Z_{N-1}]$, with $Z_1, \ldots, Z_{N-1} \sim \mathcal{N}(0,1)$, which provides the normal random variates used to generate the Brownian increments. It returns the intermediate Brownian bridge values $\boldsymbol{W} := [W_1, \ldots, W_{N-1}]$.

The algorithm is stated in such a way that multiple paths may be implemented using vectorisation in `Matlab` — you should, however, take care to ensure the the vectors $\boldsymbol{t}$, $\boldsymbol{W}_0$ and $\boldsymbol{W}_N$ are of compatible size to the matrix of random variates passed, $\boldsymbol{Z}$. Of course, the function will now return a matrix of values $\boldsymbol{W}$ of the same size as $\boldsymbol{Z}$.

If the time increments in $\boldsymbol{t}$ are not equal, then step (2) in the algorithm above should be computed using a more sophisticated search for the index m such that t_m is the closest point to the middle of the interval $[t_0, t_N]$.

2.4.2. Generating Stock Price Paths with Bridges. Previously we used the independence of increments property for Wiener processes to generate a geometric Brownian motion sample path using (2.8) directly from the normal random variates. Since we have now generated the Wiener process using bridging, we are now required to use the direct relation (2.4) to generate stock price paths in terms of the Wiener paths.

Example 2.4. (Stock Paths generated using Brownian Bridges) Consider a stock price S with drift rate $\mu = 15\%$, volatility $\sigma = 20\%$ and initial price $S_0 = \$100.00$. Suppose we constrain the terminal price of the stock in $T = 5$ years to be $\$250.00$. At times $\boldsymbol{t} := [t_0, t_1, t_2, \ldots, t_{100}] = [0, 0.05, 0.1, \ldots, 5]$ we wish to compute a realization of the stock path as the sequence $S_0, S_1, S_2, \ldots, S_{100}$, which requires the computation of the Brownian bridge sequence $W_0, W_1, W_2, \ldots, W_{100}$. To ensure the correct terminal stock price we invert (2.4) and set

$$W_{100} = \frac{\log(S_{100}/S_0) - (\mu - \frac{1}{2}\sigma^2)T}{\sigma} \approx 1.3315$$

and generate $\boldsymbol{Z}$ consisting of $Z_1, \ldots, Z_{99} \sim \mathcal{N}(0,1)$. Then, performing the call `BrownianBridge(`$\boldsymbol{t}$`,`W_0`,`W_{100}`,`$\boldsymbol{Z}$`)` generates the bridged Wiener values $\boldsymbol{W} = [W_1, \ldots, W_{99}]$. These, in turn, can be used in conjunction with (2.4) to compute the intermediate stock prices. Figure 2.5 shows 5 sample paths generated using this procedure.

Previously, when generating correlated stock price paths, we computed the Cholesky decomposition L of the correlation matrix $\Sigma = LL^T$ and then generated correlated normal random numbers in order to compute the stock price increments.

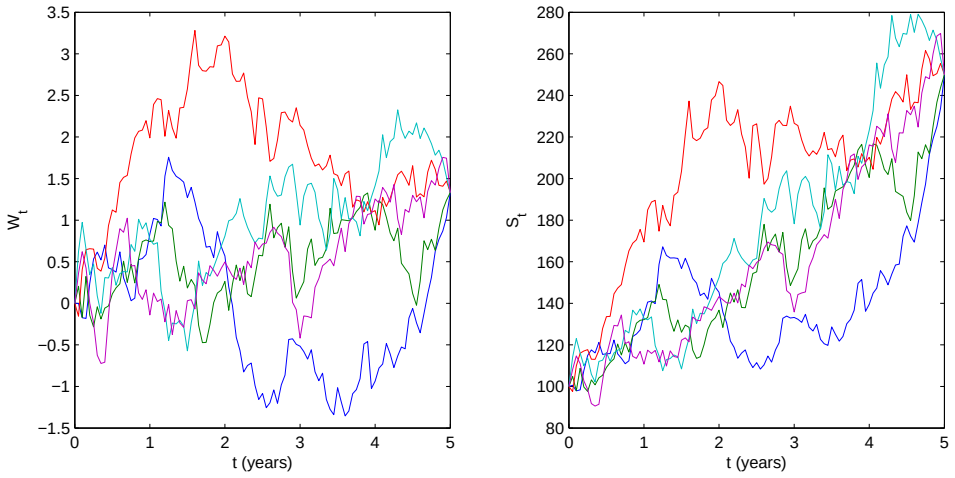


Figure 2.5. Five realizations of a Brownian bridge process (left) and corresponding stock price processes (right).

It is, unfortunately, rather complicated to include the correlation structure directly into the algorithm for computing Brownian bridges. It is, however, possible to generate independent Brownian bridges $W^{\perp}$ with the property $W_N = 0$ and then produce correlated bridges using $W = LW^{\perp}$ which have increments $\sim \mathcal{N}_k(0, \Sigma)$. These may then be used to generate correlated stock prices, with the slight complication that the drift of the stocks must be compensated for the fact that $W_N = 0$ was used. We shall not explore this further.

2.5. Generating Paths for General Itō Processes

The above approaches for generating stock price paths rely on the fact that we have a closed form solution for geometric Brownian motion. Unfortunately this model for stock prices is rather restrictive. We now explore schemes for generating paths from general Itō processes based on Itō-Taylor approximations. The two approximation schemes reviewed here are the *Euler-Maruyama method* and the *Milstein method*. (Kloeden and Platen [**9**, §10.2–3] offer a rigorous analysis of these and other higher order schemes.)

2.5.1. The Euler-Maruyama Method. Consider a stochastic process X defined by

$$dX_t = a(X_t)\,dt + b(X_t)\,dW_t, \qquad X_0 \text{ a constant}, \tag{2.11}$$

where $a, b \in C^2(\mathbb{R})$. X is clearly an Itō process, and generalizes the stock price process described by (2.1). Letting $g \in C^2(\mathbb{R})$, $s > t$, and applying

Itō's formula (in integral form), yields

$$\begin{aligned} g(X_s) &= g(X_t) + \int_t^s \left(a(X_r)\frac{dg}{dx}(X_r) + \frac{1}{2}b^2(X_r)\frac{d^2g}{dx^2}(X_r) \right) dr \\ &\quad + \int_t^s b(X_r)\frac{dg}{dx}(X_r)\, dW_r \\ &= g(X_t) + \int_t^s L^0 g(X_r)\, dr + \int_t^s L^1 g(X_r)\, dW_r, \end{aligned} \tag{2.12}$$

where we are writing

$$L^0 := a\frac{d}{dx} + \frac{1}{2}b^2\frac{d^2}{dx^2}, \quad \text{and} \quad L^1 := b\frac{d}{dx}.$$

Now, consider a small time increment Δt. Rewriting (2.11) in integral form and applying (2.12) to a and b gives

$$\begin{aligned} X_{t+\Delta t} &= X_t + \int_t^{t+\Delta t} a(X_s)\, ds + \int_t^{t+\Delta t} b(X_s)\, dWs \\ &= X_t + \int_t^{t+\Delta t} \left(a(X_t) + \int_t^s L^0 a(X_r)\, dr + \int_t^s L^1 a(X_r)\, dW_r \right) ds \\ &\quad + \int_t^{t+\Delta t} \left(b(X_t) + \int_t^s L^0 b(X_r)\, dr + \int_t^s L^1 b(X_r)\, dW_r \right) dW_s \\ &= X_t + a(X_t)\int_t^{t+\Delta t} ds + b(X_t)\int_t^{t+\Delta t} dW_s + R(t, t+\Delta t) \\ &= X_t + a(X_t)\Delta t + b(X_t)\Delta W_t + R(t, t+\Delta t) \qquad (2.13) \\ &\approx X_t + a(X_t)\Delta t + b(X_t)\Delta W_t, \qquad (2.14) \end{aligned}$$

where we write

$$\begin{aligned} R(t, t+\Delta t) &:= \int_t^{t+\Delta t} \left(\int_t^s L^0 a(X_r)\, dr + \int_t^s L^1 a(X_r)\, dW_r \right) ds \\ &\quad + \int_t^{t+\Delta t} \left(\int_t^s L^0 b(X_r)\, dr + \int_t^s L^1 b(X_r)\, dW_r \right) dW_s \\ &= \int_t^{t+\Delta t}\int_t^s L^0 a(X_r)\, dr\, ds + \int_t^{t+\Delta t}\int_t^s L^1 a(X_r)\, dW_r\, ds \\ &\quad + \int_t^{t+\Delta t}\int_t^s L^0 b(X_r)\, dr\, dW_s + \int_t^{t+\Delta t}\int_t^s L^1 b(X_r)\, dW_r\, dW_s. \end{aligned}$$

In this derivation, (2.13) follows from Example 2.1 (1).

The *Euler-Maruyama approximation scheme* is the sequence of random variables $\hat{X}_0, \hat{X}_1, \hat{X}_2, \ldots$, at times $t_i = i\Delta t$ for $i \geq 0$, generated from (2.14)

by setting

$$\hat{X}_i := \begin{cases} X_0, & \text{if } i = 0, \\ \hat{X}_{i-1} + a\left(\hat{X}_{i-1}\right)\Delta t + b\left(\hat{X}_{i-1}\right)\Delta W_t, & \text{otherwise.} \end{cases}$$

In the case where $b \equiv 0$, this method becomes the classic *Euler method* for approximating solutions of the ODE (*ordinary differential equation*)

$$\frac{dX}{dt} = a(X).$$

It can be demonstrated that the Euler-Maruyama technique has *weak order of convergence* 1, and *strong order of convergence* $\frac{1}{2}$.

Definition 2.3. (Weak and strong order of convergence) An approximation $\hat{X}_0$, $\hat{X}_1$, $\hat{X}_2, \ldots$, of a stochastic process X, at times $t_i = i\Delta t$, is said to have *weak order of convergence* γ if, for small enough Δt, there exists a constant C, such that

$$\left|\mathbb{E}[\hat{X}_i] - \mathbb{E}\left[X_i\right]\right| \leq C(\Delta t)^\gamma,$$

for every $i \geq 0$. It is said to have *strong order of convergence* γ if, for small enough Δt, there exists a constant C, such that

$$\mathbb{E}\left[\left|\hat{X}_i - X_i\right|\right] \leq C(\Delta t)^\gamma,$$

for every $i \geq 0$.

2.5.2. The Milstein Method. If we use (2.12) to expand the last term in the expression for $R(t, t+\Delta t)$, (assuming a, $b \in C^4(\mathbb{R})$), we get

$$\begin{aligned} &\int_t^{t+\Delta t}\int_t^s L^1 b(X_r)\, dW_r\, dW_s \\ &\quad = \int_t^{t+\Delta t}\int_t^s L^1 b(X_t)\, dW_r\, dW_s \\ &\qquad + \int_t^{t+\Delta t}\int_t^s \left(\int_t^r L^0 L^1 b(X_q)\, dq + \int_t^r L^1 L^1 b(X_q)\, dW_q\right) dW_r\, dW_s. \end{aligned}$$

The first term above can be evaluated as

$$\begin{aligned}
&\int_t^{t+\Delta t}\int_t^s L^1 b(X_t)\,dW_r\,dW_s \\
&\quad = L^1 b(X_t)\int_t^{t+\Delta t}\int_t^s dW_r\,dW_s \\
&\quad = b(X_t)\frac{db}{dx}(X_t)\int_t^{t+\Delta t}(W_s - W_t)\,dW_s \qquad (2.15)\\
&\quad = b(X_t)\frac{db}{dx}(X_t)\left(\frac{1}{2}\left(W_{t+\Delta t}^2 - W_t^2 - \Delta t\right) - W_t\Delta W_t\right) \qquad (2.16)\\
&\quad = \frac{1}{2}b(X_t)\frac{db}{dx}(X_t)\left((W_{t+\Delta t} + W_t)(W_{t+\Delta t} - W_t) - 2W_t\Delta W_t - \Delta t\right)\\
&\quad = \frac{1}{2}b(X_t)\frac{db}{dx}(X_t)\left((2W_t + \Delta W_t)\Delta W_t - 2W_t\Delta W_t - \Delta t\right)\\
&\quad = \frac{1}{2}b(X_t)\frac{db}{dx}(X_t)\left((\Delta W_t)^2 - \Delta t\right),
\end{aligned}$$

Here (2.15) and (2.16) follow from Example 2.1, (1) and (2), respectively. So, we can rewrite (2.13) as

$$\begin{aligned}
X_{t+\Delta t} &= X_t + a(X_t)\Delta t + b(X_t)\Delta W_t \\
&\qquad + \frac{1}{2}b(X_t)\frac{db}{dx}(X_t)\left((\Delta W_t)^2 - \Delta t\right) + \bar{R}(t, t+\Delta t) \qquad (2.17)\\
&\approx X_t + a(X_t)\Delta t + b(X_t)\Delta W_t + \frac{1}{2}b(X_t)\frac{db}{dx}(X_t)\left((\Delta W_t)^2 - \Delta t\right), \qquad (2.18)
\end{aligned}$$

where

$$\begin{aligned}
\bar{R}(t, t+\Delta t) := &\int_t^{t+\Delta t}\int_t^s L^0 a(X_r)\,dr\,ds + \int_t^{t+\Delta t}\int_t^s L^1 a(X_r)\,dW_r\,ds \\
&+ \int_t^{t+\Delta t}\int_t^s L^0 b(X_r)\,dr\,dW_s \\
&+ \int_t^{t+\Delta t}\int_t^s\int_t^r L^0 L^1 b(X_q)\,dq\,dW_r\,dW_s \\
&+ \int_t^{t+\Delta t}\int_t^s\int_t^r L^1 L^1 b(X_q)\,dW_q\,dW_r\,dW_s.
\end{aligned}$$

The *Milstein approximation scheme* is the sequence of random variables $\hat{X}_0, \hat{X}_1, \hat{X}_2, \ldots$, at times $t_i = i\Delta t$ for $i \geq 0$, generated from (2.18) by setting

$$\hat{X}_i := \begin{cases} X_0, & \text{if } i = 0, \\ \left.\begin{aligned} &\hat{X}_{i-1} + a\left(\hat{X}_{i-1}\right)\Delta t + b\left(\hat{X}_{i-1}\right)\Delta W_t \\ &\quad + \tfrac{1}{2}b\left(\hat{X}_{i-1}\right)\tfrac{db}{dx}\left(\hat{X}_{i-1}\right)\left((\Delta W_t)^2 - \Delta t\right), \end{aligned}\right\} & \text{otherwise.} \end{cases}$$

Once again, this reduces to the classic Euler method for ODEs, when $b \equiv 0$. The Milstein scheme[2] has better convergence properties than the Euler-Maruyama technique: its weak order of convergence is 1, while its strong order of convergence is also 1.

The expressions (2.13) and (2.17) are called *Itō-Taylor expansions* of X. They are computed by applying (2.12) iteratively to the terms of the SDE (2.11).

Example 2.5. Consider again a stock price S described by (2.1), with drift rate $\mu = 20\%$, volatility $\sigma = 15\%$ and current price $S_0 = \$100.00$. In order to simulate the sequence $S_0, S_1, S_2, \ldots, S_{50}$, at times $t_i = i\Delta t$ for $0 \leq i \leq 50$ and $\Delta t = 0.1$, first generate a random sequence $Z_1, \ldots, Z_{50} \sim \mathcal{N}(0, 1)$. Using (2.8) for an exact (discrete) sample path, set

$$S_i := S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right) i\Delta t + \sigma\sqrt{\Delta t}(Z_1 + \ldots + Z_i)\right),$$

for $1 \leq i \leq 50$. To approximate the same path using the Euler-Maruyama method, set

$$S_i := \begin{cases} S_0, & \text{if } i = 0, \\ S_{i-1} + \mu S_{i-1}\Delta t + \sigma S_{i-1}\sqrt{\Delta t}\, Z_i, & \text{otherwise,} \end{cases}$$

for $0 \leq i \leq 50$. Using the Milstein method, the approximate sample path is

$$S_i := \begin{cases} S_0, & \text{if } i = 0, \\ \left.\begin{array}{c} S_{i-1} + \mu S_{i-1}\Delta t + \sigma S_{i-1}\sqrt{\Delta t}\, Z_i \\ +\frac{1}{2}\sigma^2 S_{i-1}\left(Z_i^2 - 1\right)\Delta t, \end{array}\right\} & \text{otherwise,} \end{cases}$$

for $0 \leq i \leq 50$. Figure 2.6 plots approximate samples paths generated by the Euler-Maruyama and Milstein schemes against a sample path corresponding with the exact stock price process.

[2]Be warned that there are other more complex numerical schemes for SDE's due to Milstein (also transliterated as *Milshtein*) which are also referred to as *the* 'Milstein scheme'.

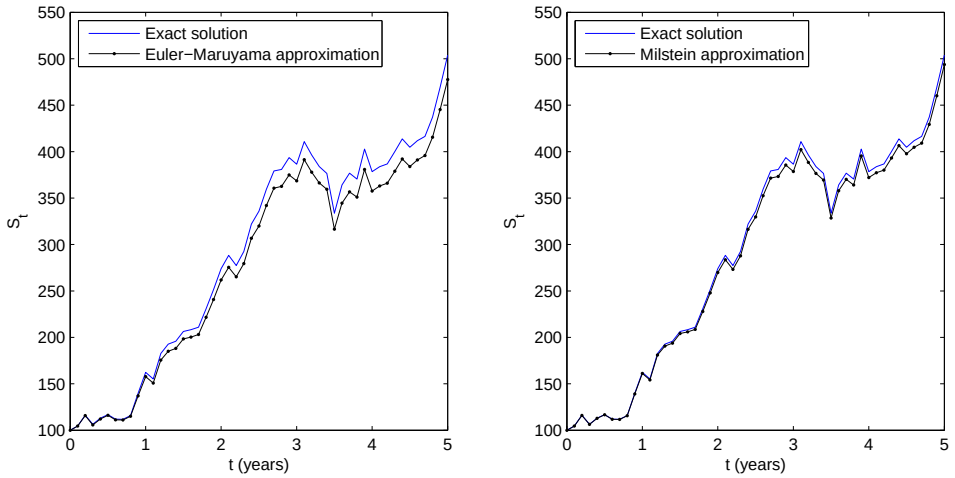


Figure 2.6. A sample path for the price of a stock over a five year period. The stock price was generated using an exact solution, the Euler-Maruyama scheme and the Milstein scheme.

Chapter 3

Monte Carlo Integration

> *...these [Monte Carlo] investigations share some of the features of ordinary experimental work in that they are susceptible to both statistical and systematic errors With regard to these matters we believe that papers should meet much the same standards as are normally required for experimental investigations. We have in mind the inclusion of estimates of statistical error descriptions of experimental conditions i.e. parameters of the calculation relevant details of apparatus, program design comparisons with previous investigations discussion of systematic errors etc. Only if these are provided will the results be trustworthy guides to improved theoretical understanding.*
>
> — Wood and Erpenbeck, Ann. Rev. Phys. Chem. **27** 319 (1976).

3.1. Introduction

Multiple quadrature rules for numerical integration are very effective in low dimensions. However, when we want to evaluate

$$I_A(f) := \int_A f(\boldsymbol{x})\, d\boldsymbol{x},$$

where $A \subseteq \mathbb{R}^k$, when k is large then multiple quadrature techniques take too long, and are impractical. (This is known as the *curse of dimensionality*.) In such high dimensional situations, *Monte Carlo integration* techniques

offer a real alternative. For instance consider *Simpson's rule* for integrating functions of k variables. The error in Simpson's rule with n nodal points behaves asymptotically as $n^{-4/k}$ (for sufficiently well behaved integrands). Since, as we shall see, the error in Monte Carlo integration is asymptotic to $n^{-1/2}$, in dimensions $k < 8$ Simpson's rule would be better than Monte Carlo integration, but in dimensions $k > 8$ Monte Carlo performs much better. The theoretical physicist Alan Sokal has put it as follows:[1]

> *It [Monte Carlo] is still a bad method; with an error proportional to $n^{-1/2}$ it is difficult to achieve more than* 4 *or* 5 *digits accuracy. But numerical integration in high dimension is very difficult; though Monte Carlo is bad all other known methods are worse.*

Exercise 3.1. (ESTIMATING π THE HARD WAY) Figure 3.1 shows part of the *Large Magellanic Cloud*, as seen from the Hubble Space Telescope. I counted 133 stars in the the circle, and 169 in the square (including those in the circle). Estimate π. You may assume that the stars in the Large Magellanic Cloud are uniformly distributed.

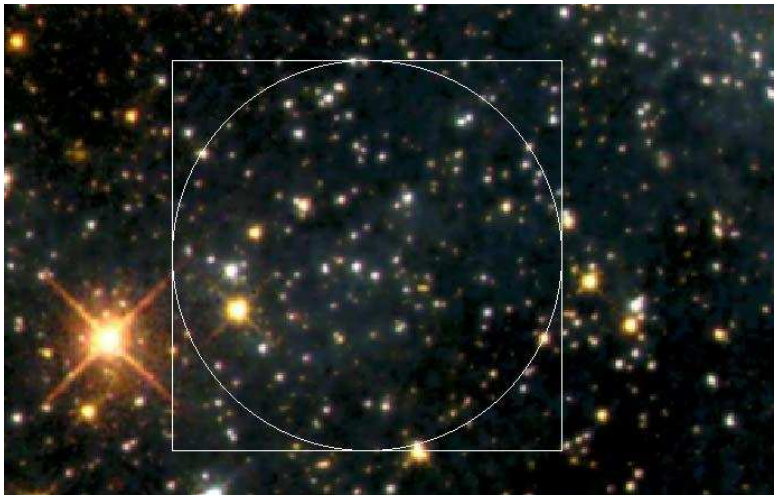


Figure 3.1. π in the sky.

One advantage that Monte Carlo integration techniques enjoy over classical quadrature formulae is that they do not require any smoothness assumptions on the integrand: almost sure continuity is enough.

[1] SOKAL, ALAN D., (1996) Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms, in *Lectures at the Cargèse Summer School on Functional Integration: Basics and Applications.*

3.2. Some Results from Probability Theory

The following results from probability theory are necessary for analyzing the properties of Monte Carlo integration algorithms.

Theorem 3.1. (Strong Law of Large Numbers) *For any infinite sequence of pairwise i.i.d. random variables $X_1, X_2, \ldots$, with finite mean $\mu = \mathbb{E}[X_i]$ and finite variance, we define $S_n := X_1 + \cdots + X_n$. Then, almost surely, we have*

$$\frac{S_n}{n} \to \mu,$$

as $n \to \infty$.

Proof. (Optional) For the proof of this result under the stated hypotheses see [**4**, page 56], for example. We will however prove a quick and dirty version of the result under the additional hypothesis that the fourth moment of one (and so of all) of the X_i exists, i.e., $\mathbb{E}\left[|X_i|^4\right] < \infty$, for each $i \in \mathbb{N}$. Without loss of generality we may take $\mu = 0$. By the $\mathscr{L}^4$ Chebyshev-Markov inequality we have, for each $\varepsilon > 0$,

$$\mathbb{P}[|S_n|] > n\varepsilon] \leq \frac{\mathbb{E}\left[S_n^4\right]}{(n\varepsilon)^4}.$$

Expanding the expectation, $\mathbb{E}\left[S_n^4\right]$, and making use of the independence of the $\{X_i\}$ together with the fact that $\mathbb{E}[X_i] = 0$ yields:

$$\mathbb{E}\left[S_n^4\right] = n\mathbb{E}\left[X_1^4\right] + \binom{4}{2}\binom{n}{2}\mathbb{E}\left[X_1^2 X_2^2\right].$$

Thus we have $\sum \mathbb{P}[S_n/n > \varepsilon] \leq \sum C/n^2 < \infty$, so that by the first Borel-Cantelli lemma we deduce that $S_n/n \to 0$ a.s. as $n \to \infty$. □

Definition 3.1. (Convergence in distribution) For a sequence of random variables $\{X_i\}$, and a random variable X, we write F_i for the distribution function of X_i, F for the distribution function of X. If $I = (a, b]$ then we write $F(I) = F(b) - F(a)$. We say that $\{X_i\}$ converges *in distribution* to X if, for every interval of continuity I of F, we have

$$F_i(I) \to F(I), \quad \text{as } i \to \infty.$$

The notation

$$X_i \xrightarrow{\mathscr{D}} X,$$

means that the sequence $\{X_i\}$ converges to X in distribution.

Theorem 3.2. (Central Limit Theorem) *Let $X_1, X_2, \ldots$, be i.i.d., with finite mean $\mu = \mathbb{E}[X_i]$ and finite variance $\sigma^2 = \mathbb{V}\mathrm{ar}[X_i] \in (0, \infty)$. If $S_n := X_1 + \cdots + X_n$, then*

$$\frac{S_n - n\mu}{\sigma\sqrt{n}} \xrightarrow{\mathscr{D}} Z,$$

as $n \to \infty$*, where* $Z \sim \mathcal{N}(0,1)$*.*

3.3. Crude Monte Carlo Integration

Let us consider the general problem of evaluating the integral

$$I_A(fw) := \int_A f(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x}, \tag{3.1}$$

where $A \subseteq \mathbb{R}^k$ and w is a probability density function with[2] $\mathrm{supp}(w) \subseteq A$. Then

$$I_A(fw) = \mathbb{E}[f(\boldsymbol{X})],$$

where $\boldsymbol{X} \sim w$ is a random variable. This can be estimated by

$$\hat{I}_{A,n}(fw) := \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{X}_i),$$

where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are independent random variates drawn from w. The justification for the above estimator is the strong law of large numbers — recall that if $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots,$ are i.i.d random variables with common probability density function w then $f(\boldsymbol{X}_1), f(\boldsymbol{X}_2), \ldots,$ will also be i.i.d random variables, with $\mathbb{E}[f(\boldsymbol{X}_i)] = I_A(fw)$. The estimator, $\hat{I}_{A,n}(fw)$, is an *unbiased estimator* of the original integral, $I_A(fw)$, since $\mathbb{E}\left[\hat{I}_{A,n}(fw)\right] = I_A(fw)$.

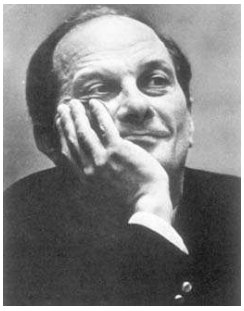


Figure 3.2. Stan Ulam and John von Neumann, Fathers of Monte Carlo methods.

For large n, the central limit theorem tells us that

$$\frac{n\hat{I}_{A,n}(fw) - nI_A(fw)}{\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]\sqrt{n}} \sim \mathcal{N}(0,1),$$

which gives us

$$\frac{\hat{I}_{A,n}(fw) - I_A(fw)}{\frac{\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]}{\sqrt{n}}} \sim \mathcal{N}(0,1),$$

[2]Recall that the *support* of a function w is defined to be closure of the set where the function is non-zero: $\mathrm{supp}(w) = \overline{\{\, \boldsymbol{x} : w(\boldsymbol{x}) \neq 0 \,\}}$.

and so

$$\hat{I}_{A,n}(fw) \sim \mathscr{N}\left(I_A(fw), \frac{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})]}{n}\right).$$

Hence,

$$\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] = \frac{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})]}{n}.$$

Now, from the properties of the normal distribution, we can deduce that

$$\hat{I}_{A,n}(fw) \in \left[I_A(fw) - \frac{3\,\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]}{\sqrt{n}}, I_A(fw) + \frac{3\,\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]}{\sqrt{n}}\right],$$

with virtual certainty[3], for large n. In this case, the error in our estimate of the integral is (with virtual certainty) less than $\frac{3\,\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]}{\sqrt{n}}$. Thus the Monte Carlo convergence rate is $\frac{1}{\sqrt{n}}$ (in a probabilistic sense). To estimate the value of this maximum likely error, set

$$\widehat{\mathbb{V}\mathrm{ar}}_n[f(\boldsymbol{X})] := \frac{1}{n-1}\sum_{i=1}^{n}(f(\boldsymbol{X}_i) - \hat{I}_{A,n}(fw))^2,$$

where the choice of $n-1$ in the denominator (as opposed to n) ensures that we have an unbiased estimator of $\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})]$. Then

$$\widehat{\mathbb{SD}\mathrm{ev}}_n[f(\boldsymbol{X})] := \sqrt{\widehat{\mathbb{V}\mathrm{ar}}_n[f(\boldsymbol{X})]} \approx \mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})],$$

and we can quote

$$\frac{3\,\widehat{\mathbb{SD}\mathrm{ev}}_n[f(\boldsymbol{X})]}{\sqrt{n}},$$

as an estimate of how much error there is in the approximation $\hat{I}_{A,n}(fw)$. Once again this is based on the notion that

$$\hat{I}_{A,n}(fw) \in \left[I_A(fw) - \frac{3\,\widehat{\mathbb{SD}\mathrm{ev}}_n[f(\boldsymbol{X})]}{\sqrt{n}}, I_A(fw) + \frac{3\,\widehat{\mathbb{SD}\mathrm{ev}}_n[f(\boldsymbol{X})]}{\sqrt{n}}\right],$$

with virtual certainty, when n is large.

Example 3.1. Consider estimating the value of $I_{[0,\pi]}(\sin x) := \int_0^\pi \sin x\,dx$. (Of course, we wouldn't want to do this in practice, since we know that the value of this integral is 2.) We then have

$$\begin{aligned} I_{[0,\pi]}(\sin x) &= \pi \int_0^\pi \frac{1}{\pi}\sin x\,dx \\ &= \pi \int_0^\pi \sin x\, w(x)\,dx \\ &= \pi\,\mathbb{E}[\sin U], \end{aligned}$$

[3]Since $\hat{I}_{A,n}(fw)$ is normally distributed, "virtual certainty" translates into 99.7% certainty, in this case.

where $w(x) := \frac{1}{\pi}\mathbb{I}_{[0,\pi]}(x)$ is the $\mathscr{U}[0,\pi]$ density, and U is a random variable from this distribution. This can be estimated with

$$\pi \hat{I}_{[0,\pi],n}\left(\frac{1}{\pi}\sin x\right) := \pi \frac{1}{n}\sum_{i=1}^{n} \sin U_i,$$

where $U_1, \ldots, U_n$ are independently generated from $\mathscr{U}[0,\pi]$. The performance of this estimator is displayed in Figure 3.3.

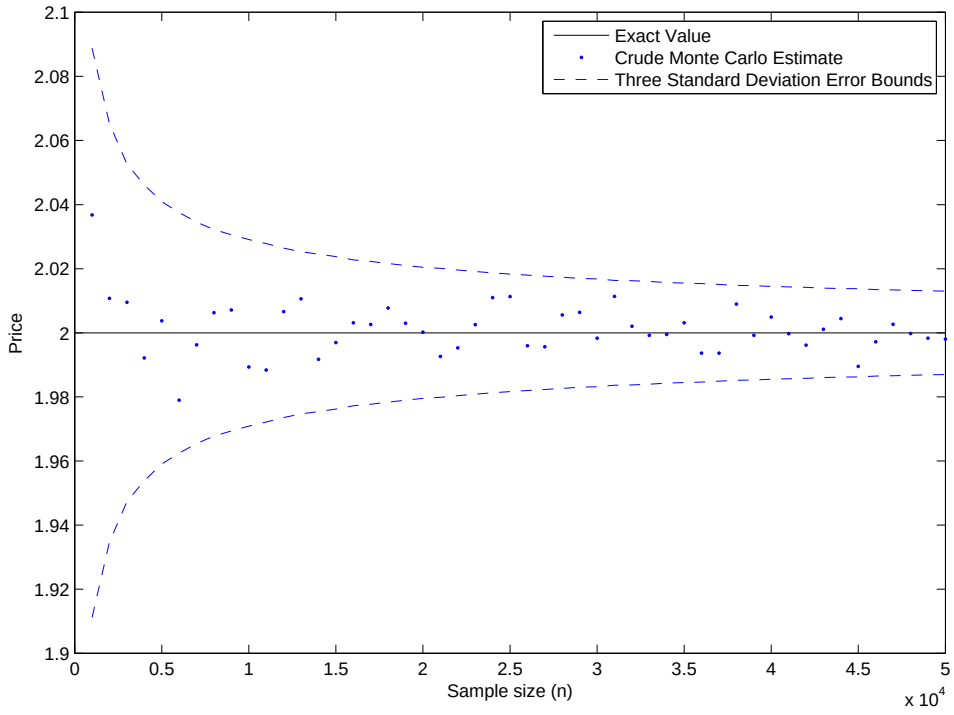


Figure 3.3. Crude Monte Carlo estimates of $\int_0^\pi \sin x\, dx$, with estimated three standard deviation error bounds, plotted against sample size.

3.4. Option Pricing as an Integration Problem

Let us now consider the problem of pricing a European call option on a non-dividend paying stock, whose price process is specified by (2.1). Suppose the option has strike price K and maturity $T > 0$, and that the continuously-compounded risk-free interest rate is r (which is assumed to be constant) with an associated numeraire asset $\beta(t) = e^{rt}$.

We can price the instrument above by applying *martingale pricing theory*. Now, the mantra of martingale pricing theory is:

> *In a complete market, the price of a contingent claim on some underlying asset is the expectation of the numeraire denominated (discounted) payoff of that claim, under the risk-neutral measure.*

To apply this approach we must first construct the *risk-neutral* stock price process $\tilde{S}$. This is a version of the discounted stock price process which has the property that it is a martingale[4], i.e., we require

$$\mathbb{E}\left[\left.\frac{\tilde{S}_t}{\beta(t)}\right| \tilde{S}_u\right] = \frac{\tilde{S}_u}{\beta(u)}, \tag{3.2}$$

for $u < t$.

By a direct application of conditional expectation on the discounted stock price, under the 'real-world' measure, we have

$$\begin{aligned}\mathbb{E}\left[\left.\frac{S_t}{\beta(t)}\right| S_u\right] &= \frac{e^{\mu(t-u)}}{\beta(t)} S_u \\ &= \frac{\beta(u)e^{\mu(t-u)}}{\beta(t)} \frac{S_u}{\beta(u)} \\ &= e^{(\mu-r)(t-u)} \frac{S_u}{\beta(u)}.\end{aligned}$$

Thus, we see that the only way that this could be a martingale is if we had $\mu = r$. In order to have the exponential term in the last expression evaluate to one, so that the martingale property (3.2) is obeyed, we can define $\tilde{S}$ to be the 'drift-adjusted' version of S. To do this we define a new Brownian motion process

$$d\tilde{W}_t = dW_t + \frac{\mu - r}{\sigma}\, dt,$$

where $\frac{\mu-r}{\sigma}$ is the market price of risk, then our drift adjusted SDE is given by

$$d\tilde{S}_t = r\tilde{S}_t\, dt + \sigma\tilde{S}_t\, d\tilde{W}_t, \qquad \tilde{S}_0 := S_0 \text{ a constant.}$$

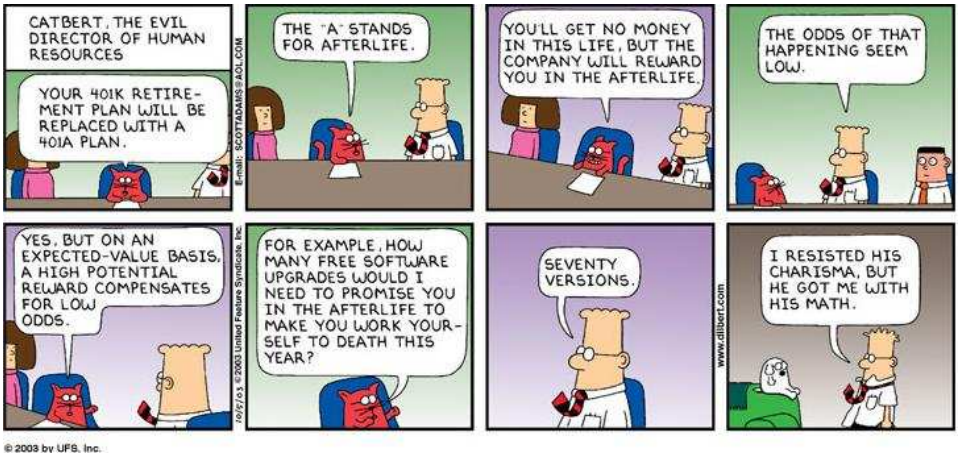
This drift adjusted SDE is a martingale under the measure which makes $\tilde{W}$ a martingale (standard Wiener process), the so-called risk-neutral measure. From now on, we shall assume that expectations are taken under this measure.

Then, by (2.4),

$$\tilde{S}_t = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right) t + \sigma\tilde{W}_t\right),$$

for all $t \geq 0$.

[4]We shall say no more on martingales; they will however be a dominant theme of the Stochastic Calculus course.



In our particular example, we have obtained a model of assets under the risk-neutral measure by the simple device of replacing the drift with the risk-free interest rate.

Applying martingale pricing theory, the call option price is given by the discounted pay-off of the claim using the risk-neutral stock price process:

$$\begin{aligned} c(S_0, r, \sigma, T, K) &= \mathbb{E}\left[\beta(T)^{-1}\left(\tilde{S}_T - K\right)^+\right] \\ &= \mathbb{E}\left[\exp(-rT)\left(S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\tilde{W}_T\right) - K\right)^+\right] \\ &= \int_{-\infty}^{\infty} f(z)w(z)\,dz, \end{aligned}$$

where

$$f(z) := \exp(-rT)\left(S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}z\right) - K\right)^+,$$

and

$$w(z) := \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right),$$

is the density of a $\mathcal{N}(0,1)$ random variable. If we generated n i.i.d. samples $Z_1, \ldots, Z_n$ from $\mathcal{N}(0,1)$, the expression

$$\hat{c}_n(S_0, r, \sigma, T, K) := \frac{1}{n}\sum_{i=1}^{n} f(Z_i),$$

would provide a Monte Carlo estimate of the option price. As before, for large n, the central limit theorem tells us that

$$\hat{c}_n(S_0, r, \sigma, T, K) \sim \mathcal{N}\left(c(S_0, r, \sigma, T, K), \frac{\mathbb{V}\mathrm{ar}[f(Z_i)]}{n}\right).$$

Hence, we can estimate that with virtual certainty (99.7% of the time)

$$\hat{c}_n(\cdot) \in \left[c(\cdot) - \frac{3\,\widehat{\mathbb{SD}\mathrm{ev}}_n[f(Z_i)]}{\sqrt{n}}, c(\cdot) + \frac{3\,\widehat{\mathbb{SD}\mathrm{ev}}_n[f(Z_i)]}{\sqrt{n}} \right].$$

Example 3.2. (European call via Monte Carlo) Consider an at-the-money European call on a stock whose current price is \$100.00, and whose volatility is 30%. Suppose the option has a two year maturity, and that the risk-free interest rate is 8.5%. In Figure 3.4, the performance of the crude Monte Carlo method for pricing this instrument is compared with the *Black-Scholes* price for a call,

$$\begin{aligned} \mathrm{BS}(S_0, \sigma, T, r, K) = S_0 \Phi \left(\frac{\log(S_0/K) + \left(r + \frac{1}{2}\sigma^2\right) T}{\sigma\sqrt{T}} \right) \\ - e^{-rT} K \Phi \left(\frac{\log(S_0/K) + \left(r - \frac{1}{2}\sigma^2\right) T}{\sigma\sqrt{T}} \right), \end{aligned}$$

where Φ denotes the standard normal cumulative distribution function.

Example 3.3. (Path dependency: Asian options) In our previous example the payoff was determined by the terminal stock price, S_T, i.e., it did not depend on the trajectory S_t for $t \in (0, T)$. As a result, in our simulation of paths, we needed only to generate values of S_T. We now examine an example of a particularly simple *path-dependent* option, namely an *Asian option.* These options have payoffs which depend on the *average* value of the underlying over the period from inception until maturity. As before, the fair price of the option is given as the risk-neutral expectation of the discounted payoff. In the case of an arithmetic Asian call option, which averages the stock at N points along the stock path, this is given by

$$c^{\mathrm{arith}}(\tilde{\boldsymbol{S}}, r, \sigma, T, K) = \mathbb{E}\left[\beta(T)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} \tilde{S}_i - K \right)^+ \right]$$

where $\tilde{\boldsymbol{S}} = [\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_N]$ corresponds to the stock prices at times $0 = t_0 < t_1 < \cdots < t_N = T$. The (multi-dimensional) Monte Carlo estimate for this price may be computed by generating

$$\boldsymbol{Z}_1 = \left[Z_1^{(1)}, \ldots, Z_1^{(N)} \right]^T, \ldots, \boldsymbol{Z}_n = \left[Z_n^{(1)}, \ldots, Z_n^{(N)} \right]^T \sim \mathscr{N}_N(\boldsymbol{0}, I)$$

and computing the sample expectation

$$\hat{c}_n^{\mathrm{arith}}(\tilde{\boldsymbol{S}}, r, \sigma, T, K) := \frac{1}{n} \sum_{j=1}^{n} f^{\mathrm{arith}}(\boldsymbol{Z}_j),$$

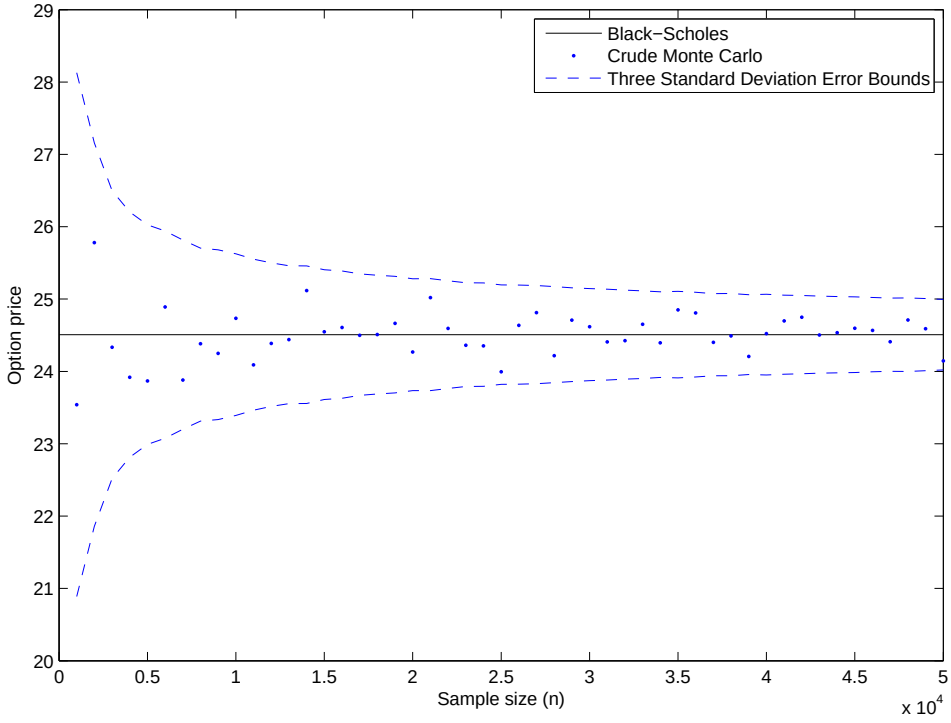


Figure 3.4. Crude Monte Carlo estimates of the price of a European call, with estimated three standard deviation error bounds, plotted against sample size. The theoretically correct *Black-Scholes* price ($24.51) is also included. The pricing parameters were a current stock price of $100.00, a risk-free rate of 8.5%, a stock price volatility of 30%, an option maturity of two years, and a strike price of $100.00.

where

$$f^{\text{arith}}(\boldsymbol{Z}) = \exp(-rT)\left(\frac{1}{N}\sum_{i=1}^{N}\tilde{S}_i(\boldsymbol{Z}) - K\right)^+$$

and the corresponding (risk-neutral) stock price paths are generated using the procedure of Section 2.4:

$$\tilde{S}_i(\boldsymbol{Z}) = S_0 \exp\left(\sum_{k=1}^{i}\left(r - \frac{1}{2}\sigma^2\right)\Delta t_k + \sigma\sqrt{\Delta t_k}Z^{(k)}\right),$$

for $1 \leq i \leq N$. Again, error bounds may be computed using the fact that

$$\hat{c}_n^{\text{arith}}(\tilde{\boldsymbol{S}}, r, \sigma, T, K) \sim \mathcal{N}\left(c^{\text{arith}}(\tilde{\boldsymbol{S}}, r, \sigma, T, K), \frac{\mathbb{V}\text{ar}[f^{\text{arith}}(\boldsymbol{Z})]}{n}\right).$$

3.5. Introducing Term Structure

In the above we have assumed that the short rate r is constant. This means that the price, at time s, of a zero-coupon bond maturing at time $t > s$, with unit value, is

$$B(s,t) = e^{-r(t-s)}.$$

Suppose, however, that at time 0 we observe a collection of bond prices $B(0,t)$, indexed by maturity t. We can then introduce a deterministic[5], but time varying, risk-free rate $r(s)$ given by

$$r(s) = -\frac{\partial}{\partial t} \log B(0,t)\bigg|_{t=s},$$

which means that

$$B(0,t) = \exp\left(-\int_0^t r(s)\,ds\right).$$

We then use similar arguments as made previously to price options consistent with the observed term structure of bond prices. The risk-neutral stock price process $\tilde{S}$ is described by the SDE

$$d\tilde{S}_t = r(t)\tilde{S}_t\,dt + \sigma\tilde{S}_t\,d\tilde{W}_t, \qquad \tilde{S}_0 := S_0 \text{ a constant},$$

where $\tilde{W}$ is a standard Wiener process under the risk-neutral measure. This has a general solution given by

$$\tilde{S}_t = S_0 \exp\left(\int_0^t r(s)\,ds - \frac{1}{2}\sigma^2 t + \sigma\tilde{W}_t\right)$$

for $t \geq 0$.

Then the sequence $\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_N$, $N \in \mathbb{N}$, at times $0 = t_0, t_1, \ldots, t_N = T$ with time increments $\Delta t_i = t_i - t_{i-1}$ can be simulated by repeated application of

$$\begin{aligned}\tilde{S}_i &= \tilde{S}_{i-1} \exp\left(\int_{t_{i-1}}^{t_i} r(s)\,ds - \frac{1}{2}\sigma^2\Delta t_i + \sigma\sqrt{\Delta t_i}Z_i\right) \\ &= \tilde{S}_{i-1}\frac{B(0,t_{i-1})}{B(0,t_i)} \exp\left(-\frac{1}{2}\sigma^2\Delta t_i + \sigma\sqrt{\Delta t_i}Z_i\right) \qquad (3.3)\end{aligned}$$

for $1 \leq i \leq N$, and where $Z_1, \ldots, Z_N \sim \mathcal{N}(0,1)$ are independent random variables.

To price options, the same approach as before is used, except that the option payoff, in terms of the risk-neutral process $\tilde{S}$, is discounted using the factor $B(0,T)$.

[5]It is possible to extend this reasoning to stochastic short rates.

3.6. Estimating the "Greeks"

The *Greeks* are the measures of sensitivity of the price of a derivative instrument to changes in the values of its various pricing parameters. Being able to compute them is just as important as being able to calculate the price of the instrument itself, because of the key role they play in risk management and hedging. We shall explore three different Monte Carlo approaches to sensitivity computation, namely finite differences, pathwise derivatives and the likelihood ratio method.

3.6.1. Finite Difference Estimates. In the (usual) case where we do not have closed form expressions for the Greeks we often resort to some form of *finite difference* approximation. This introduces a truncation error into the computation. In a Monte Carlo framework, in addition to this truncation error we also have some stochastic error associated with the non-perfect simulation of values.

Note the following (one-variable) difference quotient relations, for a suitably differentiable function f. They can be derived using Taylor's formula.

Definition 3.2. (Finite-difference approximation formulae)

(1) The *forward-difference* approximation of $f'(x)$ is given by

$$\frac{f(x+h)-f(x)}{h}.$$

(2) The *backward-difference* approximation of $f'(x)$ is given by

$$\frac{f(x)-f(x-h)}{h}.$$

(3) The *central-difference* approximation of $f'(x)$ is given by

$$\frac{f(x+h)-f(x-h)}{2h}.$$

The truncation error associated with the forward- and backward-difference approximations is $O(h)$; while that of the central-difference approximation is $O(h^2)$.

Example 3.4. (Estimating delta for a European call) The *delta* of a derivative is the sensitivity of its value to changes in the value of the underlying asset[6]. In the case of the call option described in §3.4, it is

$$\Delta(S_0, r, \sigma, T, K) := \frac{\partial}{\partial S_0} c(S_0, r, \sigma, T, K).$$

[6]Assuming that the derivative is based upon a single asset. Multi-asset derivatives will have a delta for each underlying asset.

To estimate delta, consider two stock price processes, $S^{(1)}$ and $S^{(2)}$, whose risk-neutral price processes are given by the SDEs

$$\begin{aligned} d\tilde{S}_t^{(1)} =& r\tilde{S}_t^{(1)}\,dt + \sigma\tilde{S}_t^{(1)}\,d\tilde{W}_t^{(1)}, \qquad \tilde{S}_0^{(1)} := S_0^{(1)} = S_0 + \Delta S_0, \\ d\tilde{S}_t^{(2)} =& r\tilde{S}_t^{(2)}\,dt + \sigma\tilde{S}_t^{(2)}\,d\tilde{W}_t^{(2)}, \qquad \tilde{S}_0^{(2)} := S_0^{(2)} = S_0, \end{aligned}$$

where $\tilde{W}^{(1)}$ and $\tilde{W}^{(2)}$ are independent Brownian motions. Then

$$c\left(S_0^{(1)}, r, \sigma, T, K\right) = c(S_0 + \Delta S_0, r, \sigma, T, K),$$

and

$$c\left(S_0^{(2)}, r, \sigma, T, K\right) = c(S_0, r, \sigma, T, K),$$

are premiums for options on $S^{(1)}$ and $S^{(2)}$, respectively, with exactly the same pricing parameters as in §3.4. By application of the *forward difference approximation* of first order partial derivatives, as well as the Monte Carlo pricing approach developed in §3.4, we get

$$\begin{aligned} \Delta(S_0, r, \sigma, T, K) &\approx \frac{c(S_0 + \Delta S_0, r, \sigma, T, K) - c(S_0, r, \sigma, T, K)}{\Delta S_0} \\ &= \frac{c\left(S_0^{(1)}, r, \sigma, T, K\right) - c\left(S_0^{(2)}, r, \sigma, T, K\right)}{\Delta S_0} \\ &\approx \frac{\hat{c}_n\left(S_0^{(1)}, r, \sigma, T, K\right) - \hat{c}_n\left(S_0^{(2)}, r, \sigma, T, K\right)}{\Delta S_0} \\ &= \frac{\frac{1}{n}\sum_{i=1}^n f_1\left(Z_i^{(1)}\right) - \frac{1}{n}\sum_{i=1}^n f_2\left(Z_i^{(2)}\right)}{\Delta S_0} \\ &=: \widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K), \end{aligned}$$

where, for $i = 1, 2$,

$$f_i(z) := \exp(-rT)\left(S_0^{(i)} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\,z\right) - K\right)^+$$

and

$$\boldsymbol{Z}_1 = \left[Z_1^{(1)}, Z_1^{(2)}\right]^T, \ldots, \boldsymbol{Z}_n = \left[Z_n^{(1)}, Z_n^{(2)}\right]^T$$

are sampled randomly from $\mathcal{N}_2(\mathbf{0}, I)$. Clearly,

$$\begin{aligned}
&\mathbb{E}\left[\lim_{\Delta S_0 \to 0} \widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K)\right] \\
&\quad = \lim_{\Delta S_0 \to 0} \mathbb{E}\left[\widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K)\right] \\
&\quad = \lim_{\Delta S_0 \to 0} \frac{\mathbb{E}\left[\hat{c}_n\left(S_0^{(1)}, r, \sigma, K, T\right)\right] - \mathbb{E}\left[\hat{c}_n\left(S_0^{(2)}, r, \sigma, K, T\right)\right]}{\Delta S_0} \\
&\quad = \lim_{\Delta S_0 \to 0} \frac{c\left(S_0^{(1)}, r, \sigma, K, T\right) - c\left(S_0^{(2)}, r, \sigma, K, T\right)}{\Delta S_0} \\
&\quad = \lim_{\Delta S_0 \to 0} \frac{c\left(S_0 + \Delta S_0, r, \sigma, K, T\right) - c\left(S_0, r, \sigma, K, T\right)}{\Delta S_0} \\
&\quad = \Delta(S_0, r, \sigma, T, K),
\end{aligned}$$

thus showing that $\widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K)$ is an unbiased estimator of $\Delta(S_0, r, \sigma, T, K)$.

Now, since $Z_i^{(1)}$ and $Z_j^{(2)}$ are independent $(1 \le i, j \le n)$,

$$\mathbb{C}\mathrm{ov}\left[f_1\left(Z_i^{(1)}\right), f_2\left(Z_j^{(2)}\right)\right] = 0,$$

and so

$$\begin{aligned}
&\mathbb{V}\mathrm{ar}\left[\widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K)\right] \\
&\quad = \mathbb{V}\mathrm{ar}\left[\frac{\hat{c}_n\left(S_0^{(1)}, r, \sigma, K, T\right) - \hat{c}_n\left(S_0^{(2)}, r, \sigma, K, T\right)}{\Delta S_0}\right] \\
&\quad = \frac{1}{(\Delta S_0)^2}\left(\frac{\mathbb{V}\mathrm{ar}\left[f_1(Z^{(1)})\right]}{n} + \frac{\mathbb{V}\mathrm{ar}\left[f_2(Z^{(2)})\right]}{n}\right) \\
&\quad = O\left(\frac{1}{(\Delta S_0)^2}\right) O\left(\frac{1}{n}\right),
\end{aligned}$$

where $\left[Z^{(1)}, Z^{(2)}\right]^T \sim \mathcal{N}_2(\mathbf{0}, I)$. Consequently, while the variance of the estimate of delta decreases linearly, as $n \to \infty$, it also increases quadratically, as $\Delta S_0 \to 0$. In other words, decreasing ΔS_0 does not lead to improved accuracy in estimating the option delta, unless it is accompanied by an increased sample size. In particular (and contrary to intuition), for fixed n, the variance of $\widehat{\Delta}_n^{\mathrm{fd1}}(S_0, r, \sigma, T, K)$ diverges, as $\Delta S_0 \to 0$; making it a poor estimate of delta.

One way of improving things is by employing the same samples to estimate both $c(S_0 + \Delta S_0, r, \sigma, T, K)$ and $c(S_0, r, \sigma, T, K)$; resulting in the estimator

$$\begin{aligned}\widehat{\Delta}_n^{\text{fd2}}(S_0, r, \sigma, T, K) :=& \frac{\frac{1}{n}\sum_{i=1}^n f_1(Z_i) - \frac{1}{n}\sum_{i=1}^n f_2(Z_i)}{\Delta S_0} \\ =& \frac{1}{n}\sum_{i=1}^n \frac{f_1(Z_i) - f_2(Z_i)}{\Delta S_0},\end{aligned}$$

where $Z_1, \ldots, Z_n$ are sampled randomly from $\mathscr{N}(0,1)$. This is known as the method of *common random numbers*. As before,

$$\mathbb{E}\left[\widehat{\Delta}_n^{\text{fd2}}(S_0, r, \sigma, T, K)\right] = \Delta(S_0, r, \sigma, T, K).$$

But this time $f_1(Z_i)$ and $f_2(Z_i)$ are no longer independent ($1 \leq i \leq n$) — in fact, since f_1 and f_2 are monotone increasing functions, it can be shown that

$$\mathbb{C}\text{ov}[f_1(Z_i), f_2(Z_i)] \geq 0,$$

for each $1 \leq i \leq n$. Thus,

$$\begin{aligned}&\mathbb{V}\text{ar}\left[\widehat{\Delta}_n^{\text{fd2}}(S_0, r, \sigma, T, K)\right] \\ &\quad= \mathbb{V}\text{ar}\left[\frac{f_1(Z) - f_2(Z)}{\Delta S_0}\right]\frac{1}{n} \\ &\quad= \frac{1}{(\Delta S_0)^2}\left(\frac{\mathbb{V}\text{ar}[f_1(Z)]}{n} - 2\frac{\mathbb{C}\text{ov}[f_1(Z), f_2(Z)]}{n} + \frac{\mathbb{V}\text{ar}[f_2(Z)]}{n}\right) \\ &\quad\leq \mathbb{V}\text{ar}\left[\widehat{\Delta}_n^{\text{fd1}}(S_0, r, \sigma, T, K)\right],\end{aligned}$$

where $Z \sim \mathscr{N}(0,1)$.

The intuition underlying this estimator is that using independent samples to estimate $c(S_0 + \Delta S_0, r, \sigma, T, K)$ and $c(S_0, r, \sigma, T, K)$ results in independent errors in these estimates. However, estimating the option values with the same samples results in similar errors on the option premiums (i.e., both option prices are either overestimated or underestimated by similar amounts). When computing delta, correlated errors tend to cancel each other out, since the one option price is subtracted from the other.

We can easily apply the method of common random numbers to get estimates for the other Greeks:

- (*gamma*) $\Gamma(S_0, r, \sigma, T, K) := \frac{\partial^2}{\partial S_0^2} c(S_0, r, \sigma, T, K)$;
- (*rho*) $\mathrm{P}(S_0, r, \sigma, T, K) := \frac{\partial}{\partial r} c(S_0, r, \sigma, T, K)$;
- (*vega*) $\mathrm{V}(S_0, r, \sigma, T, K) := \frac{\partial}{\partial \sigma} c(S_0, r, \sigma, T, K)$; and
- (*theta*) $\Theta(S_0, r, \sigma, T, K) := \frac{\partial}{\partial T} c(S_0, r, \sigma, T, K)$.

For estimating gamma, applying the forward difference approximation twice, gives

$$\begin{aligned}
&\Gamma(S_0, r, \sigma, T, K) \\
&\approx \frac{\Delta(S_0 + \Delta S_0, r, \sigma, T, K) - \Delta(S_0, r, \sigma, T, K)}{\Delta S_0} \\
&\approx \frac{c(S_0 + 2\Delta S_0, r, \sigma, T, K) - 2c(S_0 + \Delta S_0, r, \sigma, T, K) + c(S_0, r, \sigma, T, K)}{(\Delta S_0)^2} \\
&\approx \frac{\hat{c}_n(S_0 + 2\Delta S_0, r, \sigma, T, K) - 2\hat{c}_n(S_0 + \Delta S_0, r, \sigma, T, K) + \hat{c}_n(S_0, r, \sigma, T, K)}{(\Delta S_0)^2}.
\end{aligned}$$

Together with the use of common random variables, this produces the estimator $\widehat{\Gamma}_n^{\mathrm{fd+fd}}(S_0, r, \sigma, T, K)$.

We can improve somewhat on the estimators based on forward difference approximations by employing the *central differences* method. In the case of delta,

$$\Delta(S_0, r, \sigma, T, K) \approx \frac{c(S_0 + \Delta S_0, r, \sigma, T, K) - c(S_0 - \Delta S_0, r, \sigma, T, K)}{2\Delta S_0}.$$

Using the above, in conjunction with the method of common random variables, yields the estimator $\widehat{\Delta}_n^{\mathrm{cd}}(S_0, r, \sigma, T, K)$. This can be adapted easily to estimate the other Greeks.

If we apply the central finite difference approximation twice, we get

$$\begin{aligned}
&\Gamma(S_0, r, \sigma, T, K) \\
&\approx \frac{\Delta(S_0 + \Delta S_0, r, \sigma, T, K) - \Delta(S_0 - \Delta S_0, r, \sigma, T, K)}{2\Delta S_0} \\
&\approx \frac{c(S_0 + 2\Delta S_0, r, \sigma, T, K) - 2c(S_0, r, \sigma, T, K) + c(S_0 - 2\Delta S_0, r, \sigma, T, K)}{4(\Delta S_0)^2} \\
&\approx \frac{\hat{c}_n(S_0 + 2\Delta S_0, r, \sigma, T, K) - 2\hat{c}_n(S_0, r, \sigma, T, K) + \hat{c}_n(S_0 - 2\Delta S_0, r, \sigma, T, K)}{4(\Delta S_0)^2}.
\end{aligned}$$

Again, combining this with the method of common random variables produces the estimator $\widehat{\Gamma}_n^{\mathrm{cd+cd}}(S_0, r, \sigma, T, K)$ for gamma.

3.6.2. Pathwise Derivative Estimates. In the last section we saw that one could produce estimates for the Greeks using finite difference methods. This relied on introducing a small, but finite, deviation of the variable of interest. Now, consider the following approach which allows that deviation to tend toward zero. Computing the delta of a call option:

$$\begin{aligned}
\Delta(S_0, r, \sigma, T, K) &= \lim_{\Delta S_0 \to 0} \frac{c\left(S_0 + \Delta S_0, r, \sigma, K, T\right) - c\left(S_0, r, \sigma, K, T\right)}{\Delta S_0} \\
&\approx \lim_{\Delta S_0 \to 0} \frac{\hat{c}_n\left(S_0 + \Delta S_0, r, \sigma, K, T\right) - \hat{c}_n\left(S_0, r, \sigma, K, T\right)}{\Delta S_0} \\
&= \lim_{\Delta S_0 \to 0} \frac{\frac{1}{n}\sum_{i=1}^{n} f(S_0 + \Delta S_0, Z_i) - \frac{1}{n}\sum_{i=1}^{n} f(S_0, Z_i)}{\Delta S_0} \\
&= \frac{1}{n}\sum_{i=1}^{n} \lim_{\Delta S_0 \to 0} \frac{f(S_0 + \Delta S_0, Z_i) - f(S_0, Z_i)}{\Delta S_0} \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial s_0} f(s_0, Z_i)\Big|_{s_0 = S_0}, \qquad (3.4)
\end{aligned}$$

where

$$f(s_0, z) := \exp(-rT)\left(s_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\, z\right) - K\right)^+, \qquad (3.5)$$

and the (common) random numbers $Z_1, \ldots, Z_n$ are sampled from $\mathscr{N}(0, 1)$.

Expression (3.4) is an expectation of the pathwise derivative of the function f (with respect to S_0). To provide a proof that the expected value of this estimator is equal to the actual value of the delta requires that the following interchange of expectation and derivative is justified

$$\frac{\partial}{\partial x}\mathbb{E}\left[f(\cdot)\right] = \mathbb{E}\left[\frac{\partial}{\partial x} f(\cdot)\right].$$

In a lot of useful cases this is justified. There are, however, instances where it is not. In particular the interchange is not applicable when the pay-off function is discontinuous (i.e., not Lipschitz continuous) — this applies to digital and barrier options and is also a problem when computing the gamma of a standard put or call option (See Glasserman [**5**, §7.2.2] for details).

Example 3.5. (Pathwise estimate of the Black-Scholes delta) For convenience we may express the discounted pay-off function (3.5) as

$$f(s_0, z) := \exp(-rT)(\tilde{S}_T(s_0, z) - K)^+,$$

where

$$\tilde{S}_T(s_0, z) = s_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\, z\right) \qquad (3.6)$$

is the risk-neutral terminal stock price. Then, applying the chain rule, we have

$$\frac{\partial f}{\partial s_0} = \frac{\partial f}{\partial \tilde{S}_T}\frac{\partial \tilde{S}_T}{\partial s_0}.$$

The first factor on the right hand side is given by

$$\frac{\partial f}{\partial \tilde{S}_T} = \frac{\partial}{\partial \tilde{S}_T} \exp(-rT) \max(0, \tilde{S}_T(s_0, z) - K) = \exp(-rT) \mathbb{I}_{\{\tilde{S}_T(s_0,z)>K\}},$$

while the second factor is given by

$$\frac{\partial \tilde{S}_T}{\partial s_0} = \frac{\tilde{S}_T(s_0, z)}{s_0}.$$

Putting everything together, the Black-Scholes delta may be computed using the following estimator

$$\begin{aligned}
\Delta(S_0, r, \sigma, T, K) &= \frac{\partial}{\partial S_0} \mathbb{E}\left[f(S_0, Z)\right] \\
&= \mathbb{E}\left[\frac{\partial}{\partial S_0} f(S_0, Z)\right] \\
&\approx \exp(-rT) \frac{1}{n} \sum_{i=1}^{n} \frac{\tilde{S}_T(S_0, Z_i)}{S_0} \mathbb{I}_{\{\tilde{S}_T(S_0,Z_i)>K\}},
\end{aligned}$$

where $Z_1, \ldots, Z_n$ are sampled from $\mathcal{N}(0, 1)$. In other words it is the discounted mean of $\frac{\tilde{S}_T}{S_0}$ for those paths that end up in the money. It is important to note that this estimator reuses the terminal stock prices required for pricing the option—leading to a substantial computational improvement over the corresponding finite difference approach.

Exercise 3.2. Derive the expression for a pathwise delta estimate for an arithmetic average rate option.

3.6.3. The Likelihood Ratio Method. One of the problems with the pathwise method is that in order to justify the interchange of the derivative and expectation, one needs continuity of the payoff function. We now explore a method that relies instead on the continuity of the density function.

Suppose we may express our sensitivity in the following manner

$$\frac{\partial}{\partial x} \mathbb{E}[f(\cdot)] = \frac{\partial}{\partial x} \left[\int_A f(s) w_x(s)\, ds\right],$$

where the expectation has been written in such a way that the function f does not depend directly on the parameter x, while the density function, w_x, now does. Note that w_x is usually the density of risk neutral stock price paths associated with the payoff function f in terms of those paths. For example, in the case of a European option it is the density of terminal stock prices, $w_x(s) = \frac{\partial}{\partial s}(\mathbb{P}[\tilde{S}_T(x) \leq s])$. The subscript x emphasizes that it is written in terms of a parameter (which, for example, would be S_0 for the delta or σ for the vega). The function $w_x(s)$ is sometimes known as the likelihood function as it may also be interpreted as the likelihood of the parameter x given an observed outcome s.

The advantage of this formulation is that, in most instances of interest, density functions are continuous and this means that the interchange of partial derivative and expectation is justified. Hence, we write

$$\begin{aligned}\frac{\partial}{\partial x}\left[\int_A f(s)w_x(s)\,ds\right] &= \int_A f(s)\frac{\partial}{\partial x}w_x(s)\,ds \\ &= \int_A f(s)\frac{w_x'(s)}{w_x(s)}w_x(s)\,ds \\ &= \mathbb{E}\left[f(s)\frac{w_x'(s)}{w_x(s)}\right].\end{aligned}$$

The quantity w_x'/w_x, which may also be written as $\partial \log w_x/\partial x$, is called the score function and is also used in the computation of maximum likelihood estimates.

Example 3.6. (Likelihood-Ratio estimate of the Black-Scholes delta) We now express the discounted payoff as

$$f(s) := \exp(-rT)(s-K)^+,$$

where the variable s represents the risk-neutral terminal stock price. We must find the from of the likelihood function $w_{S_0}(s)$. We use the fact that in this case $w_{S_0} = \mathbb{P}[\tilde{S}_T = s]$. Given $Z \sim \mathscr{N}(0,1)$, the cumulative distribution function corresponding to w_{S_0} may be written as

$$\begin{aligned}\mathbb{P}[\tilde{S}_T(S_0,Z) \le s] &= \mathbb{P}\left[S_0 \exp\left(\left(r-\frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\,Z\right) \le s\right] \\ &= \mathbb{P}\left[Z \le -\frac{\log(S_0/s) + (r-\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right] \\ &= \Phi\big(z(s)\big),\end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution function for the standard normal density and

$$z(s) = -\frac{\log(S_0/s) + (r-\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

Thus, we may now compute w_{S_0} as

$$w_{S_0}(s) = \frac{\partial\Phi}{\partial z}\frac{\partial z}{\partial s} = \frac{\phi\big(z(s)\big)}{s\sigma\sqrt{T}},$$

where $\phi(\cdot)$ is the standard normal density function. Furthermore

$$\begin{aligned}w_{S_0}'(s) &= \frac{1}{s\sigma\sqrt{T}}\frac{\partial\phi}{\partial z}\frac{\partial z}{\partial S_0} \\ &= \frac{1}{s\sigma\sqrt{T}}\big[-z(s)\phi\big(z(s)\big)\big]\left[\frac{-1}{S_0\sigma\sqrt{T}}\right],\end{aligned}$$

and, as a result, the score function is

$$\frac{w'_{S_0}(s)}{w_{S_0}(s)} = \frac{z(s)}{S_0\sigma\sqrt{T}}.$$

The Black-Scholes delta may then be written as

$$\Delta(S_0, r, \sigma, T, K) = \mathbb{E}\left[f(s)\frac{w'_{S_0}(s)}{w_{S_0}(s)}\right] = \mathbb{E}\left[\frac{f(s)z(s)}{S_0\sigma\sqrt{T}}\right],$$

where $s \sim w_{S_0}$.

To simplify computation, it is possible to generate random variables that are w_{S_0} distributed by using $Z \sim \mathcal{N}(0,1)$ and (3.6), which means that $s = \tilde{S}_T(S_0, Z)$ and $z(s) = Z$, and our estimator is given by

$$\begin{aligned}\Delta(S_0, r, \sigma, T, K) &= \mathbb{E}\left[\frac{f\big(\tilde{S}_T(S_0, Z)\big)Z}{S_0\sigma\sqrt{T}}\right] \\ &\approx \exp(-rT)\frac{1}{n}\sum_{i=1}^{n}\frac{Z_i\big(\tilde{S}_T(S_0, Z_i) - K\big)^+}{S_0\sigma\sqrt{T}},\end{aligned}$$

where $Z_1, \ldots, Z_n$ are sampled from $\mathcal{N}(0,1)$. Again, it is important to note that the estimator uses the same terminal stock prices used in the computation of the option price.

An important aspect of the final form of the estimator is that it directly uses the functional form of the discounted payoff multiplied by the score function. It is, thus, very easy to compute delta estimates for any European payoff function since the likelihood is unaffected by the form of the payoff.

One of the disadvantages of this estimator is that it generally has a higher variance than the finite difference estimate (with common random numbers) and the pathwise derivative estimate. Figure 3.5 shows estimates of the delta for the call option of §3.4 for each of the methods previously described.

Exercise 3.3. Derive the likelihood function for the vega of any European payoff function based on GBM. Once you have the expression in terms of s and $z(s)$ find the simplification in terms of the substitution $z(s) = Z$.

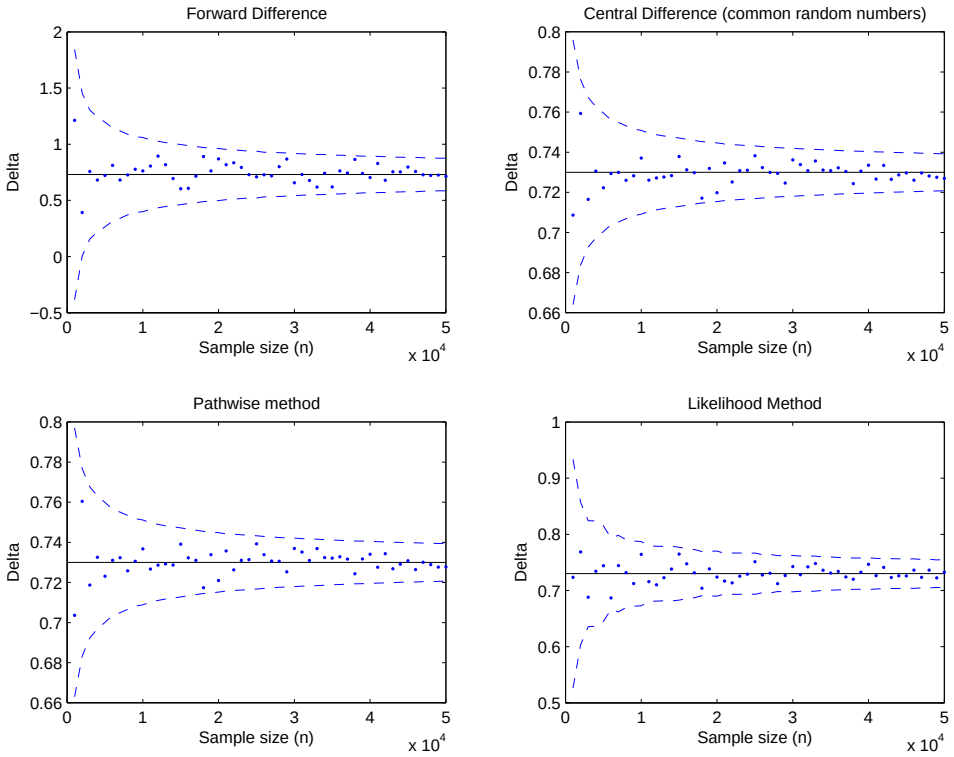


Figure 3.5. Estimates of the delta of a European call, with estimated three standard deviation error bounds, plotted against sample size. The theoretically correct *Black-Scholes* delta (0.73) is also included. The pricing parameters were a current stock price of \$100.00, a risk-free rate of 8.5%, a stock price volatility of 30%, an option maturity of two years, and a strike price of \$100.00. The finite difference methods were performed using $\Delta S_0 = 5$.

Chapter 4

Variance Reduction

> *Johnny*[1] *saw at once its great scope even though in the first hour of our discussion he evinced a certain scepticism. But when I became more persuasive, quoting statistical estimates of how many computations were needed to obtain rough results with this or that probability, he agreed, eventually becoming quite inventive in finding marvellous technical tricks to facilitate or speed up these techniques. The one thing about Monte Carlo is that it never gives an exact answer; rather its conclusions indicate that the answer is so and so, within such and such an error, with such and such probability — that is, with probability differing from one by such and such a small amount.*
>
> — Stanislaw Ulam, *Adventures of a Mathematician.*

4.1. Introduction

As we have seen before, the standard deviation of the crude Monte Carlo estimate of (3.1) is

$$\mathbb{SD}\text{ev}\left[\hat{I}_{A,n}(fw)\right] = \frac{\mathbb{SD}\text{ev}[f(\boldsymbol{X})]}{\sqrt{n}},$$

where $\boldsymbol{X} \sim w$. This means that reducing $\mathbb{SD}\text{ev}\left[\hat{I}_{A,n}(fw)\right]$ by a factor of ten requires a hundred-fold increase in the number of simulation trials. An alternative approach is to concentrate on reducing the size of $\mathbb{SD}\text{ev}[f(\boldsymbol{X})]$. Techniques that do this are known as *variance reduction techniques*.

[1] "Good-time Johnny" aka John von Neumann.

4.2. Control Variates

The motivating principle behind *control variates* is to improve the Monte Carlo estimate of an integral by leveraging off known information. Let us return to the integration problem (3.1). Suppose we can evaluate

$$I_A(gw) := \int_A g(\boldsymbol{x})w(\boldsymbol{x})\,d\boldsymbol{x},$$

exactly[2]. We can use this to construct an estimator for $I_A(fw)$, with a better variance than that of $\hat{I}_{A,n}(fw)$. By evaluating

$$\hat{I}_{A,n}(gw) - I_A(gw)$$

we get an idea of the error in the crude Monte Carlo estimate $\hat{I}_{A,n}(fw)$. We can then adjust $\hat{I}_{A,n}(fw)$ by an amount proportional to $\hat{I}_{A,n}(gw) - I_A(gw)$, to produce the estimator

$$\hat{I}^{\star}_{A,n}(fw) := \hat{I}_{A,n}(fw) + \alpha\left(\hat{I}_{A,n}(gw) - I_A(gw)\right), \tag{4.1}$$

where the parameter α must be chosen judiciously. Notice that

$$\begin{aligned}\hat{I}^{\star}_{A,n}(fw) &= -\alpha I_A(gw) + \left(\hat{I}_{A,n}(fw) + \alpha\hat{I}_{A,n}(gw)\right)\\ &= -\alpha I_A(gw) + \hat{I}_{A,n}((f+\alpha g)w).\end{aligned}$$

Consequently,

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\hat{I}^{\star}_{A,n}(fw)\right] &= \mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}((f+\alpha g)w)\right]\\ &= \frac{\mathbb{V}\mathrm{ar}[(f+\alpha g)(\boldsymbol{X})]}{n}\\ &= \frac{1}{n}\{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})] + 2\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}), \alpha g(\boldsymbol{X})] + \mathbb{V}\mathrm{ar}[\alpha g(\boldsymbol{X})]\}\\ &= \frac{1}{n}\{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})] + 2\alpha\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}), g(\boldsymbol{X})] + \alpha^2\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]\}.\end{aligned}$$

In order to minimize this, set $\frac{d}{d\alpha}\mathbb{V}\mathrm{ar}\left[\hat{I}^{\star}_{A,n}(fw)\right] = 0$. Then,

$$\frac{2}{n}\{\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}), g(\boldsymbol{X})] + \alpha\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]\} = 0,$$

from which it follows that the optimal α is given by

$$\begin{aligned}\alpha^{\mathrm{opt}} &= -\frac{\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}), g(\boldsymbol{X})]}{\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]}\\ &= -\frac{\mathbb{SD}\mathrm{ev}[f(\boldsymbol{X})]}{\mathbb{SD}\mathrm{ev}[g(\boldsymbol{X})]}\mathbb{C}\mathrm{orr}[f(\boldsymbol{X}), g(\boldsymbol{X})].\end{aligned} \tag{4.2}$$

[2]This seems an onerous requirement, but is often easily fulfilled in practice.

For this value of α, we have

$$\begin{aligned}
&\mathbb{V}\mathrm{ar}\left[\hat{I}^{\star}_{A,n}(fw)\right] \\
&\quad= \frac{1}{n}\left\{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})] - 2\frac{(\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}),g(\boldsymbol{X})])^2}{\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]} + \frac{(\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}),g(\boldsymbol{X})])^2}{\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]}\right\} \\
&\quad= \frac{1}{n}\left\{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})] - \frac{(\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}),g(\boldsymbol{X})])^2}{\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]}\right\} \\
&\quad\leq \frac{\mathbb{V}\mathrm{ar}[f(\boldsymbol{X})]}{n} = \mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right].
\end{aligned}$$

We can now see that this approach is good for two reasons:

(1) it can never produce an increase in estimator variance (relative to crude Monte Carlo); and

(2) its variance is strictly less than that of the crude Monte Carlo estimator, whenever $\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}),g(\boldsymbol{X})] \neq 0$.

Of course, the statistics required to determine the optimal value for α from (4.2) aren't known. Nevertheless, they can be estimated, using a subsample of the simulation data. Specifically, let $n_0 \ll n$. Then we can use the first n_0 random variates $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n_0}$, drawn from w, to estimate $\mathbb{C}\mathrm{ov}[f(\boldsymbol{X}),g(\boldsymbol{X})]$ and $\mathbb{V}\mathrm{ar}[g(\boldsymbol{X})]$ as

$$\widehat{\mathbb{V}\mathrm{ar}}_{n_0}[g(\boldsymbol{X})] := \frac{1}{n_0-1}\sum_{i=1}^{n_0}\Big(g(\boldsymbol{X}_i) - \hat{I}_{A,n_0}(gw)\Big)^2 \tag{4.3}$$

$$\widehat{\mathbb{C}\mathrm{ov}}_{n_0}[f(\boldsymbol{X}),g(\boldsymbol{X})] := \frac{1}{n_0-1}\sum_{i=1}^{n_0}\Big(f(\boldsymbol{X}_i) - \hat{I}_{A,n_0}(fw)\Big)\Big(g(\boldsymbol{X}_i) - \hat{I}_{A,n_0}(gw)\Big). \tag{4.4}$$

Using (4.2), the optimal α can be estimated as

$$\hat{\alpha}^{\mathrm{opt}}_{n_0} := -\frac{\widehat{\mathbb{C}\mathrm{ov}}_{n_0}[f(\boldsymbol{X}),g(\boldsymbol{X})]}{\widehat{\mathbb{V}\mathrm{ar}}_{n_0}[g(\boldsymbol{X})]}.$$

Now, the last $n - n_0$ variates can be used to evaluate $\hat{I}^{\star}_{A,(n-n_0)}(fw)$.

Example 4.1. (Control variate for a European basket option) We have three non-dividend paying stocks $S^{(a)}$, $S^{(b)}$ and $S^{(c)}$, whose prices follow geometric Brownian motions. The stock price volatilities are $\sigma_a = 40\%$, $\sigma_b = 25\%$ and $\sigma_c = 30\%$, respectively. The stock price returns are correlated, with the correlation structure specified by

$$\Sigma = \begin{pmatrix} \rho_{aa} & \rho_{ab} & \rho_{ac} \\ \rho_{ab} & \rho_{bb} & \rho_{bc} \\ \rho_{ac} & \rho_{bc} & \rho_{cc} \end{pmatrix} = \begin{pmatrix} 1 & 0.80 & 0.90 \\ 0.80 & 1 & 0.75 \\ 0.90 & 0.75 & 1 \end{pmatrix}.$$

We will determine the price of a European basket option, whose payoff at expiry date $T = 18$ months is

$$\max\Big\{S_T^{(a)} + S_T^{(b)} + S_T^{(c)} - 200, 0\Big\},$$

where $S_0^{(a)} = \text{R}25.00$, $S_0^{(b)} = \text{R}55.00$ and $S_0^{(c)} = \text{R}120.00$. We take the interest rate to be 10% (NACC).

The control variate that we use is the price of a vanilla at-the-money European call on $S^{(c)}$ (with the same maturity as before). We can do this as we have the exact Black-Scholes formula for an at-the-money European call:

$$c_0^{(c)} = S_0^{(c)}(\Phi(d_1) - e^{-rT}\Phi(d_2)),$$

where we write,

$$d_1 = \frac{(r + \frac{1}{2}\sigma_c^2)\sqrt{T}}{\sigma_c}, \qquad d_2 = d_1 - \sigma_c\sqrt{T}.$$

Having generated n $\mathscr{N}_3(\mathbf{0}, \Sigma)$ samples, $\{\boldsymbol{Z_i}\}$, we generate stock prices for our basket,

$$\boldsymbol{S}_{T,i} = \begin{bmatrix} S_{T,i}^{(a)} \\ S_{T,i}^{(b)} \\ S_{T,i}^{(c)} \end{bmatrix} := \begin{bmatrix} S_0^{(a)} \exp\Big(\big(r - \frac{1}{2}\sigma_a^2\big)\,T + \sigma_a\sqrt{T}\,Z_i^{(1)}\Big) \\ S_0^{(b)} \exp\Big(\big(r - \frac{1}{2}\sigma_b^2\big)\,T + \sigma_b\sqrt{T}\,Z_i^{(2)}\Big) \\ S_0^{(c)} \exp\Big(\big(r - \frac{1}{2}\sigma_c^2\big)\,T + \sigma_c\sqrt{T}\,Z_i^{(3)}\Big) \end{bmatrix}$$

Using 10% of the paths (i.e., $n_0 = 0.1n$ in each case), we compute the vectors of payoffs for the Basket and (single stock) Vanilla option from which we can then estimate the optimal α as

$$\alpha = -\frac{\mathbb{C}\text{ov}[\text{Basket}, \text{Vanilla}]}{\mathbb{V}\text{ar}[\text{Vanilla}]} \approx -1.53.$$

Then, we use (4.1) with the remaining samples (i.e., $n - n_0$) to get our control variate price estimates for the Basket option — you can check that the Black-Scholes price for the single stock c is R25.95.

The resulting plot of crude and control variate prices is pictured in Figure 4.1; you will notice that the oscillation of prices is less prevalent in the control variate case when compared to the crude Monte Carlo estimates.

4.2.1. Pricing Arithmetic Average-Rate Options. Consider a *geometric average rate European call option*, with *discrete sampling*, on a stock described by (2.1). Suppose that averaging begins at time $t^* \geq 0$, and that

$$h := \frac{T - t^*}{N},$$

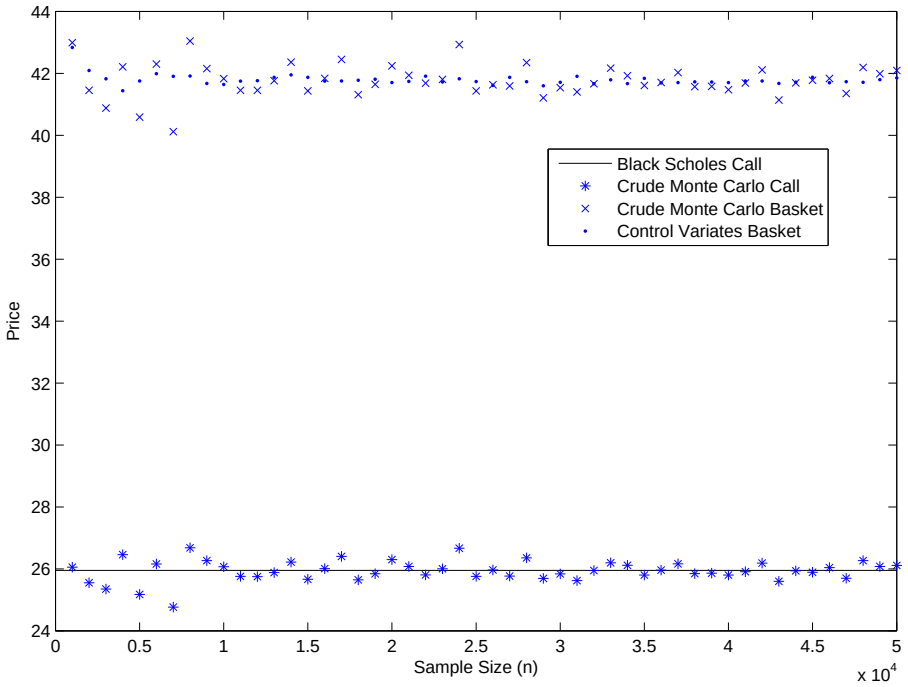


Figure 4.1. Crude and control variate European basket option prices against sample size (for samples of $1000, 2000, \ldots, 50\,000$ random vectors).

is the interval between sample points, where there are $N+1$ such points, including the option maturity T. If the strike is K the payoff for such an instrument is

$$\left(\left(\prod_{i=0}^{N} S_{t^*+ih} \right)^{\frac{1}{N+1}} - K \right)^+$$

while it may be shown that the price of this instrument is given by[3]

$$c^{\text{geom}}(S_0, r, \sigma, t^*, T, N, K)$$
$$= \exp(-rT) \left[S_0 \exp\left(\underline{\mu} + \frac{1}{2}\underline{\sigma}^2 \right) \Phi\left(\frac{\log \frac{S_0}{K} + \underline{\mu}}{\underline{\sigma}} + \underline{\sigma} \right) - K\Phi\left(\frac{\log \frac{S_0}{K} + \underline{\mu}}{\underline{\sigma}} \right) \right].$$

[3] You will derive this sometime during the *Swaps and Exotic Options* course.

where Φ is the *standard normal cumulative distribution function*, and

$$\underline{\mu} := \left(r - \frac{1}{2}\sigma^2\right) t^* + \left(r - \frac{1}{2}\sigma^2\right) \frac{T - t^*}{2}$$
$$\underline{\sigma}^2 := \sigma^2 t^* + \sigma^2 \frac{T - t^*}{6} \left(\frac{T - t^*}{T - t^* + h} + 1\right).$$

Unfortunately, there is no closed-form solution for $c^{\text{arith}}(S_0, r, \sigma, t^*, T, N, K)$, the value of the corresponding *arithmetic average-rate European call* — this is an example of an Asian option that we looked at in Example 3.3. Here the payoff function is given by

$$\left(\frac{1}{N+1} \sum_{j=0}^{N} S_j(\boldsymbol{Z}) - K\right)^+$$

Nevertheless, we can use the price of the geometric average-rate call as a control variate to estimate $c^{\text{arith}}(S_0, r, \sigma, t^*, T, N, K)$. Specifically, generate

$$\boldsymbol{Z}_1 = \left[Z_1^{(0)}, \ldots, Z_1^{(N)}\right]^T, \ldots, \boldsymbol{Z}_n = \left[Z_n^{(0)}, \ldots, Z_n^{(N)}\right]^T,$$

from $\mathcal{N}_{N+1}(\mathbf{0}, I)$. Now, set

$$\hat{c}_n^{\text{arith}}(S_0, r, \sigma, t^*, T, N, K) := \frac{1}{n} \sum_{i=1}^{n} f^{\text{arith}}(\boldsymbol{Z}_i),$$

and

$$\hat{c}_n^{\text{geom}}(S_0, r, \sigma, t^*, T, N, K) := \frac{1}{n} \sum_{i=1}^{n} f^{\text{geom}}(\boldsymbol{Z}_i),$$

where, for $\boldsymbol{Z} = \left[Z^{(0)}, \ldots, Z^{(N)}\right]^T$,

$$f^{\text{arith}}(\boldsymbol{Z}) := \exp(-rT) \left(\frac{1}{N+1} \sum_{j=0}^{N} S_j(\boldsymbol{Z}) - K\right)^+$$

and

$$f^{\text{geom}}(\boldsymbol{Z}) := \exp(-rT) \left(\left(\prod_{j=0}^{N} S_j(\boldsymbol{Z})\right)^{\frac{1}{N+1}} - K\right)^+$$

are the payoff functions of the arithmetic and geometric average-rate calls, respectively, and

$$S_j(\boldsymbol{Z}) := S_0 \exp\left\{\left(r - \frac{1}{2}\sigma^2\right)(t^* + jh) + \sigma\left(\sqrt{t^*} Z^{(0)} + \sqrt{h} \sum_{k=1}^{j} Z^{(k)}\right)\right\},$$

for every $0 \leq j \leq N$. We then have that the control variate estimator for $c^{\text{arith}}(S_0, r, \sigma, t^*, T, N, K)$ is given by

$$\hat{c}_n^{\text{arith}\star}(S_0, \ldots, K) = \hat{c}_n^{\text{arith}} + \alpha\left(\hat{c}_n^{\text{geom}} - c^{\text{geom}}\right),$$

where α is chosen appropriately.

Example 4.2. (Arithmetic average rate European call premium) Figure 4.2 illustrates the behaviour of the above control variate estimator of the arithmetic average rate European call premium, for a stock whose current price is \$100.00, and whose volatility is 45%. A risk-free rate of 12% is assumed, and averaging occurs six times, starting immediately (i.e., $t^* = 0$). The option maturity is one year, and its strike is \$50.00.

The geometric average rate option price (using the analytic pricing formula) is \$48.04. Using a $n_0 = 0.1n$ $\mathcal{N}_5(\mathbf{0}, I)$ samples[4], the optimal value for α was about -1.05. The control variate estimate was computed using the remaining $n - n_0$ samples in each case. The control variate estimate of the arithmetic average rate option price, using $n = 50\,000$ is \$49.91.

4.3. Antithetic Variates

4.3.1. Introduction. The slogan for *antithetic variates* is "get more from your sample". Essentially, the idea is that, under suitable conditions, it is possible to use every element of a sequence of random variates at least twice. This effectively converts a sample of n random variates into a sample of size at least $2n$; hopefully, with an associated increase in the accuracy of the Monte Carlo integral estimate.

4.3.2. The One-Dimensional Case. Suppose that the density w in (3.1) is symmetric (about its mean, obviously). Then, if $\boldsymbol{X} \sim w$, it will also be the case that $2\mathbb{E}[\boldsymbol{X}] - \boldsymbol{X} \sim w$. To check this in the one-dimensional case note that, by the symmetry of w,

$$\begin{aligned}
\mathbb{P}[2\mathbb{E}[X] - X < a] &= \mathbb{P}[X > 2\mathbb{E}[X] - a] \\
&= \int_{2\mathbb{E}[X]-a}^{\infty} w(x)\,dx \\
&= \int_{-\infty}^{a} w(2\mathbb{E}[X] - x)\,dx \\
&= \int_{-\infty}^{a} w(x)\,dx = \mathbb{P}[X < a],
\end{aligned}$$

for any $a \in \mathbb{R}$. So, if we sample $X_1, \ldots, X_n$ from w, we get the "antithetic sample" $2\mathbb{E}[X] - X_1, \ldots, 2\mathbb{E}[X] - X_n$ "for free". Note, however, that the

[4] Since $t^* = 0$, we require one less increment for each stock price path.

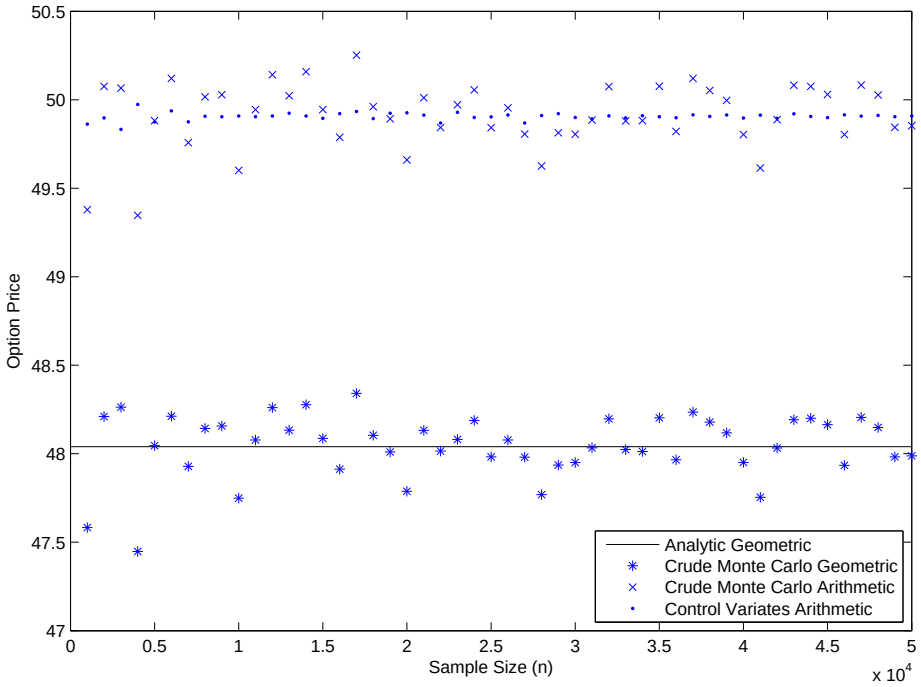


Figure 4.2. Analytic, crude Monte Carlo, and control variate option prices against sample size (for samples of $1000, 2000, \ldots, 50\,000$ random vectors) for a geometric and an arithmetic average rate option. Both options have the same specifications: averaging occurs six times (starting immediately); the maturity date is in one year; and the strike price is \$50.00. The underlying instrument for these contracts is a stock, whose price is \$100.00, and whose volatility is 45%. An interest rate of 12% is assumed.

original sample and the new sample are not independent! So, if we set

$$\hat{I}^{-}_{A,n}(fw) := \frac{1}{n}\sum_{i=1}^{n} f(2\mathbb{E}[X] - X_i),$$

it is clear that $\hat{I}^{-}_{A,n}(fw)$ is an unbiased estimate of $I_A(fw)$, with

$$\mathbb{V}\mathrm{ar}\left[\hat{I}^{-}_{A,n}(fw)\right] = \mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right].$$

Now, put

$$\begin{aligned}\hat{I}^{\pm}_{A,n}(fw) :=& \frac{1}{2}\left(\hat{I}_{A,n}(fw) + \hat{I}^{-}_{A,n}(fw)\right)\\ =& \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i) + f(2\mathbb{E}[X] - X_i)}{2},\end{aligned} \tag{4.5}$$

Then we have the following:

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}\left[\hat{I}^{\pm}_{A,n}(fw)\right] \\
&\quad = \mathbb{V}\mathrm{ar}\left[\frac{1}{2}\left(\hat{I}_{A,n}(fw) + \hat{I}^{-}_{A,n}(fw)\right)\right] \\
&\quad = \frac{1}{4}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] + \frac{1}{2}\mathbb{C}\mathrm{ov}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] + \frac{1}{4}\mathbb{V}\mathrm{ar}\left[\hat{I}^{-}_{A,n}(fw)\right] \\
&\quad = \frac{1}{2}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] + \frac{1}{2}\mathbb{C}\mathrm{ov}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] \qquad (4.6) \\
&\quad = \frac{1}{2}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] \\
&\qquad + \frac{1}{2}\mathbb{SD}\mathrm{ev}\left[\hat{I}_{A,n}(fw)\right]\mathbb{SD}\mathrm{ev}\left[\hat{I}^{-}_{A,n}(fw)\right]\mathbb{C}\mathrm{orr}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] \\
&\quad = \frac{1}{2}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] + \frac{1}{2}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right]\mathbb{C}\mathrm{orr}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] \qquad (4.7) \\
&\quad \leq \mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right].
\end{aligned}
$$

It is also clear from (4.5) and (4.7) that we have

$$
\begin{aligned}
\mathbb{V}\mathrm{ar}\left[\hat{I}^{\pm}_{A,n}(fw)\right] &= \frac{1}{n}\left\{\mathbb{V}\mathrm{ar}\left[\frac{f(X) + f(2\mathbb{E}[X] - X)}{2}\right]\right\} \\
&= \frac{1}{2n}\left\{\mathbb{V}\mathrm{ar}\left[f(X)\right] + \mathbb{V}\mathrm{ar}\left[f(X)\right]\mathbb{C}\mathrm{orr}\left[f(X), f(2\mathbb{E}[X] - X)\right]\right\},
\end{aligned}
$$

which allows the computation of sample estimates for the variance of the estimate $\hat{I}^{\pm}_{A,n}(fw)$.

Based on the above analysis, we may observe the following:

(1) At worst (when $\hat{I}_{A,n}(fw)$ and $\hat{I}^{-}_{A,n}(fw)$ are perfectly correlated), the variance of $\hat{I}^{\pm}_{A,n}(fw)$ equals the variance of $\hat{I}_{A,n}(fw)$;

(2) In general (when $\hat{I}_{A,n}(fw)$ and $\hat{I}^{-}_{A,n}(fw)$ are not perfectly correlated), the variance of $\hat{I}^{\pm}_{A,n}(fw)$ is strictly less than that of $\hat{I}_{A,n}(fw)$; and

(3) At best (when $\hat{I}_{A,n}(fw)$ and $\hat{I}^{-}_{A,n}(fw)$ are perfectly negatively correlated), the variance of $\hat{I}^{\pm}_{A,n}(fw)$ is zero.

The last case above is interesting (and surprising). It occurs precisely when f satisfies

$$f(2\mathbb{E}[X] - x) = 2f(\mathbb{E}[X]) - f(x),$$

and A is symmetric about $\mathbb{E}[X]$. In that case, writing $\hat{A} = A \cap (-\infty, \mathbb{E}[X]]$, and also $\hat{A}' = A \cap (\mathbb{E}[X], \infty)$, we have

$$\begin{aligned}
I_A(fw) &= \int_A f(x)w(x)\,dx \\
&= \int_{\hat{A}} f(x)w(x)\,dx + \int_{\hat{A}'} f(x)w(x)\,dx \\
&= \int_{\hat{A}} f(x)w(x)\,dx + \int_{\hat{A}} f(2\mathbb{E}[X]-x)w(2\mathbb{E}[X]-x)\,dx \\
&= \int_{\hat{A}} f(x)w(x)\,dx + \int_{\hat{A}} (2f(\mathbb{E}[X]) - f(x))w(x)\,dx \\
&= 2f(\mathbb{E}[X]) \int_{\hat{A}} w(x)\,dx \\
&= f(\mathbb{E}[X]) \int_A w(x)\,dx \\
&= f(\mathbb{E}[X]),
\end{aligned}$$

while

$$\begin{aligned}
\hat{I}^{\pm}_{A,n}(fw) &= \frac{1}{2}\left(\hat{I}_{A,n}(fw) + \hat{I}^{-}_{A,n}(fw)\right) \\
&= \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i) + \frac{1}{n}\sum_{i=1}^{n} f(2\mathbb{E}[X]-x_i)\right) \\
&= \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i) + \frac{1}{n}\sum_{i=1}^{n} (2f(\mathbb{E}[X]) - f(x_i))\right) \\
&= \frac{1}{2n}\sum_{i=1}^{n} 2f(\mathbb{E}[X]) \\
&= f(\mathbb{E}[X]),
\end{aligned}$$

where $x_1, \ldots, x_n$ are drawn randomly from w.

Unfortunately, our appraisal of the antithetic variate technique so far has been too optimistic — while it is true that the variance of $\hat{I}^{\pm}_{A,n}(fw)$ can never exceed that of $\hat{I}_{A,n}(fw)$, this is not enough. The problem is that the computation of $\hat{I}^{\pm}_{A,n}(fw)$ requires $2n$ evaluations of $f(x)$, whereas computing $\hat{I}_{A,n}(fw)$ entails only n evaluations of the same function. In the case where f is the payoff function of a complicated derivative, evaluating $f(x)$ is much more time-consuming than generating a random variate; and we may infer that calculating $\hat{I}^{\pm}_{A,n}(fw)$ consumes roughly twice as much time as does calculating $\hat{I}_{A,n}(fw)$. Consequently, for the antithetic variates technique to

be viable, we require

$$\begin{aligned}\mathbb{V}\text{ar}\left[\hat{I}^{\pm}_{A,n}(fw)\right] &\leq \mathbb{V}\text{ar}\left[\hat{I}_{A,2n}(fw)\right] \\ &= \frac{\mathbb{V}\text{ar}[f(X)]}{2n} \\ &= \frac{1}{2}\mathbb{V}\text{ar}\left[\hat{I}_{A,n}(fw)\right],\end{aligned}$$

So, we need $\mathbb{C}\text{ov}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] \leq 0$, by (4.6).

In order to show that this is true for most payoff functions of interest, we need the following results:

Exercise 4.1. Suppose the function $g : [0,1] \to \mathbb{R}$ is continuous and monotonic, show that

$$(g(u) - g(v))(g(1-v) - g(1-u)) \geq 0 \tag{4.8}$$

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

Exercise 4.2. Let U be a $\mathscr{U}[0,1)$ random variable. Suppose the density w is continuous and symmetric about its mean with corresponding distribution function F. Show, for random variable $X \sim w$ defined by $X = F^{-1}(U)$, that

$$F^{-1}(1-U) = 2\mathbb{E}[X] - X. \tag{4.9}$$

Theorem 4.1. (Chebyshev's monotone covariance inequality) *Suppose $f : \mathbb{R} \to \mathbb{R}$ is continuous and monotone and that the density w is continuous and symmetric about its mean. Then, for $X \sim w$,*

$$\mathbb{C}\text{ov}[f(X), f(2\mathbb{E}[X] - X)] \leq 0.$$

Proof. Firstly, since w is continuous, it has a continuous distribution function F and inverse distribution function $F^{-1} : [0,1] \to \mathbb{R}$, which is continuous and monotonic. Thus, the function $g = f \circ F^{-1}$ is continuous and monotonic. Using (4.8) for independent $U, V \sim \mathscr{U}[0,1]$ we have

$$0 \leq g(U)g(1-V) - g(V)g(1-V) - g(U)g(1-U) + g(V)g(1-U),$$

which implies that

$$\begin{aligned}0 &\leq \mathbb{E}[g(U)g(1-V) - g(V)g(1-V) - g(U)g(1-U) + g(V)g(1-U)] \\ &= 2\mathbb{E}[g(U)]\mathbb{E}[g(1-U)] - 2\mathbb{E}[g(U)g(1-U)],\end{aligned}$$

where the last step follows as a result of independence of U and V. Thus,

$$\mathbb{E}[g(U)g(1-U)] - \mathbb{E}[g(U)]\mathbb{E}[g(1-U)] = \mathbb{C}\text{ov}[g(U), g(1-U)] \leq 0.$$

Finally, using (4.9), we have

$$\mathbb{C}\text{ov}[f(X), f(2\mathbb{E}[X] - X)] \leq 0,$$

for $X \sim w$. □

Thus if f satisfies the conditions of the above theorem, then

$$\begin{aligned}\mathbb{C}\mathrm{ov}\left[\hat{I}_{A,n}(fw), \hat{I}^{-}_{A,n}(fw)\right] &= \mathbb{C}\mathrm{ov}\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i), \frac{1}{n}\sum_{i=1}^{n} f(2\mathbb{E}[X] - x_i)\right] \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{C}\mathrm{ov}[f(x_i), f(2\mathbb{E}[X] - x_j)] \\ &= \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{C}\mathrm{ov}[f(x_i), f(2\mathbb{E}[X] - x_i)] \leq 0.\end{aligned}$$

Example 4.3. (Pricing a Call Using an Antithetic Sample) Let us revisit the crude Monte Carlo estimate for the premium of the European call option. Recall from Example 3.2 that its price can be estimated as

$$\hat{c}_n(S_0, r, \sigma, T, K) := \hat{I}_{\mathbb{R},n}(fw) = \frac{1}{n}\sum_{i=1}^{n} f(Z_i),$$

where w is the standard normal density function, $Z_1, \ldots, Z_n$ are independently generated from w, and f is given by

$$f(z) := \exp(-rT)\left(S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}z\right) - K\right)^{+}.$$

Since f clearly satisfies the conditions of Theorem 4.1, we may conclude that the use of antithetic variates for the Monte Carlo pricing of European call options leads to a reduction of variance. Using a sample of half the size of the above estimate ($\frac{n}{2}$) with its corresponding antithetic sample we may compute an estimate of the call option using the estimator (4.5) as

$$\hat{c}^{\pm}_{n/2}(S_0, r, \sigma, T, K) := \hat{I}^{\pm}_{\mathbb{R},n/2}(fw) = \frac{1}{n/2}\sum_{i=1}^{n/2} \frac{f(Z_i) + f(-Z_i)}{2}.$$

This estimator consumes roughly the same amount of computational time as the crude estimator but with reduced variance as evidenced in Figure 4.3, which shows the performance of the two estimators on an at-the-money European call option with two year maturity for a stock with current price \$100.00 and volatility 20%. The risk-free interest rate used is 10%.

Similarly, the use of antithetic variates also improves the variance of the Monte Carlo estimate of a European put premium. The same applies for any other option whose payoff f is monotone (eg. Asian options). Barrier options form a notable class of options whose payoff functions are not monotone (and, hence, for which the method of antithetic variates may not be appropriate).

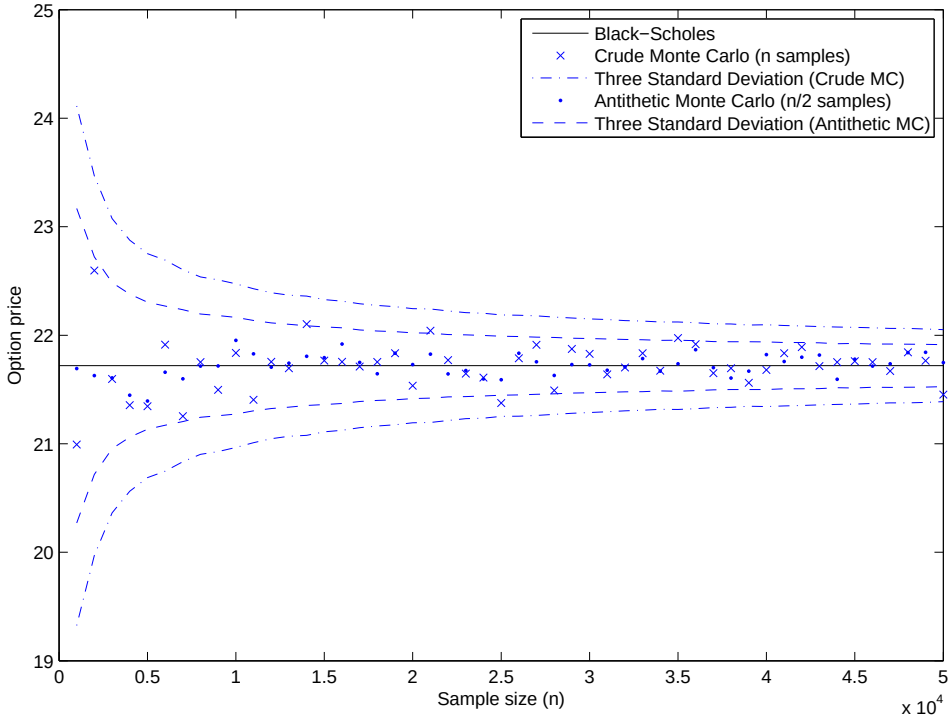


Figure 4.3. Crude and antithetic Monte Carlo estimates of the price of a European call, with estimated three standard deviation error bounds, plotted against sample size. The theoretically correct *Black-Scholes* price ($21.72) is also included. The pricing parameters were a current stock price of $100.00, a risk-free rate of 10%, a stock price volatility of 20%, an option maturity of two years, and a strike price of $100.00.

4.4. Importance Sampling

4.4.1. Introduction. One of the major defects of the crude Monte Carlo estimator for (3.1) is that relatively many of the samples generated from w may lie in regions where f is very small in absolute value, or even zero. These are wasted samples, and contribute little or nothing to the accuracy of the estimate of $I_A(fw)$. Conversely, it is also possible that generating from w may produce relatively few samples in regions where f is large in absolute value. In either case, the estimator will exhibit a retarded rate of convergence.

The problem is that w and f may not be well "matched" with each other (in the sense that w is large, where $|f|$ is small, and conversely). A common response to this is the use of *importance sampling*. The idea is deceptively simple. Suppose that v is another density function, with

$\text{supp}(fw) \subseteq \text{supp}(v)$. Then,

$$\begin{aligned} I_A(fw) &:= \int_A f(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \int_A f(\boldsymbol{x}) \frac{w(\boldsymbol{x})}{v(\boldsymbol{x})} v(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \mathbb{E}\left[f(\boldsymbol{X}) \frac{w(\boldsymbol{X})}{v(\boldsymbol{X})} \right], \end{aligned}$$

where $\boldsymbol{X} \sim v$ is a random variable. We may now estimate $I_A(fw)$ with

$$\hat{I}^v_{A,n}(fw) := \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{X}_i) \frac{w(\boldsymbol{X}_i)}{v(\boldsymbol{X}_i)},$$

where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are generated independently from v. By the strong law of large numbers, $\hat{I}^v_{A,n}(fw) \to I_A(fw)$. Furthermore, from our analysis of the variance of the crude Monte Carlo estimator (here $\boldsymbol{X} \sim v$),

$$\begin{aligned} \mathbb{V}\text{ar}\left[\hat{I}^v_{A,n}(fw) \right] &= \mathbb{V}\text{ar}\left[\hat{I}_{A,n}\left(f \frac{w}{v} v \right) \right] \\ &= \frac{\mathbb{V}\text{ar}\left[f(\boldsymbol{X}) \frac{w(\boldsymbol{X})}{v(\boldsymbol{X})} \right]}{n} \\ &= \frac{\mathbb{E}\left[\left(f(\boldsymbol{X}) \frac{w(\boldsymbol{X})}{v(\boldsymbol{X})} \right)^2 \right] - I_A^2(fw)}{n}. \end{aligned} \tag{4.10}$$

When used in this way, the density v is called an *importance sampler*. Of course, not every importance sampler will produce an improved estimator — a bad choice will make things worse! The question of how to select a good importance sampler is a difficult one.

4.4.2. When to Use Importance Sampling. How can one tell whether the density w makes $\hat{I}_{A,n}(fw)$ a poor estimator of $I_A(fw)$? The obvious answer is to examine

$$\widehat{\mathbb{V}\text{ar}}_n\left[\hat{I}_{A,n}(fw) \right] = \frac{\widehat{\mathbb{V}\text{ar}}_n[f(\boldsymbol{X})]}{n} = \frac{\hat{I}_{A,n}(f^2 w) - \hat{I}^2_{A,n}(fw)}{n},$$

where $\boldsymbol{X} \sim w$. If this number is "relatively" large, for given n, we may conclude that variates sampled from w are not optimally distributed for evaluating the given integral. Of course, if sampling from w produces poor estimates of $I_A(fw)$, then $\widehat{\mathbb{V}\text{ar}}_n\left[\hat{I}_{A,n}(fw) \right]$ may also be a poor estimate of

$$\mathbb{V}\text{ar}\left[\hat{I}_{A,n}(fw) \right] = \frac{\mathbb{V}\text{ar}[f(\boldsymbol{X})]}{n} = \frac{I_A(f^2 w) - I_A^2(fw)}{n}.$$

This is a serious problem, because it may lead us to believe falsely that $\hat{I}_{A,n}(fw)$ is a good estimate of $I_A(fw)$, when, in fact, it is not.

An alternative approach is that we may consider the *coefficient of variation* of $\hat{I}_{A,n}(fw)$. This is given by

$$\mathbb{CV}\left[\hat{I}_{A,n}(fw)\right] := \frac{\mathbb{SD}\text{ev}\left[\hat{I}_{A,n}(fw)\right]}{\mathbb{E}\left[\hat{I}_{A,n}(fw)\right]} = \frac{1}{\sqrt{n}}\frac{\mathbb{SD}\text{ev}[f(\boldsymbol{X})]}{I_A(fw)}.$$

It can be estimated with the sample coefficient of variation of $\hat{I}_{A,n}(fw)$,

$$\widehat{\mathbb{CV}}_n\left[\hat{I}_{A,n}(fw)\right] := \frac{1}{\sqrt{n}}\frac{\widehat{\mathbb{SD}\text{ev}}_n\left[f(\boldsymbol{X})\right]}{\hat{I}_{A,n}(fw)} = \frac{1}{\sqrt{n}}\frac{\sqrt{\hat{I}_{A,n}(f^2w) - \hat{I}^2_{A,n}(fw)}}{\hat{I}_{A,n}(fw)}.$$

The square of this amount has an interesting interpretation, whenever $f \geq 0$, since[5]

$$\begin{aligned}\left(\widehat{\mathbb{CV}}_n\left[\hat{I}_{A,n}(fw)\right]\right)^2 &= \frac{1}{n}\frac{\hat{I}_{A,n}(f^2w) - \hat{I}^2_{A,n}(fw)}{\hat{I}^2_{A,n}(fw)} \\ &= \frac{1}{n}\left(\frac{\hat{I}_{A,n}(f^2w)}{\hat{I}^2_{A,n}(fw)} - 1\right) \\ &= \frac{1}{n^2\hat{I}^2_{A,n}(fw)}\sum_{i=1}^{n} f(\boldsymbol{x}_i)^2 - \frac{1}{n} \\ &= \sum_{i=1}^{n} w_i^2 - \frac{1}{n},\end{aligned}$$

where, for every $1 \leq i \leq n$

$$\omega_i := \frac{f(\boldsymbol{x}_i)}{n\hat{I}_{A,n}(fw)}, \tag{4.11}$$

is called the i-th *normalized weight* (because $0 \leq \omega_i \leq 1$ and $\sum_{i=1}^{n}\omega_i = 1$). We interpret the value of ω_i as a measure of the significance of the point $\boldsymbol{x}_i$, in terms of its impact on the estimate $\hat{I}_{A,n}(fw)$.

Now, since $\left(\widehat{\mathbb{CV}}_n\left[\hat{I}_{A,n}(fw)\right]\right)^2 \geq 0$ and $\omega_i^2 \leq \omega_i$ for all i, we must have

$$\frac{1}{n} \leq \sum_{i=1}^{n}\omega_i^2 \leq \sum_{i=1}^{n}\omega_i = 1. \tag{4.12}$$

We now analyze the upper and lower bounds of $\sum_{i=1}^{n}\omega_i^2$ in (4.12) separately.

Exercise 4.3. Prove by induction that

$$(n-1)\sum_{i=1}^{n} a_i^2 - 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} a_i a_j = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(a_i - a_j)^2, \tag{4.13}$$

[5]This is generally the case for derivative pricing problems.

for $n \geq 1$, and $a_1, \ldots, a_n \in \mathbb{R}$.

Theorem 4.2. $\sum_{i=1}^n \omega_i^2 = \frac{1}{n}$ *iff* $\omega_i = \frac{1}{n}$, *for every* $1 \leq i \leq n$.

Proof. Firstly, we note that

$$\sum_{i=1}^n \omega_i^2 = \frac{\sum_{i=1}^n f^2(\boldsymbol{x}_i)}{\left(\sum_{i=1}^n f(\boldsymbol{x}_i)\right)^2}. \tag{4.14}$$

So, $\sum_{i=1}^n \omega_i^2 = \frac{1}{n}$ is equivalent to

$$\begin{aligned} n \sum_{i=1}^n f^2(\boldsymbol{x}_i) &= \left(\sum_{i=1}^n f(\boldsymbol{x}_i)\right)^2 \\ &= \sum_{i=1}^n f^2(\boldsymbol{x}_i) + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n f(\boldsymbol{x}_i) f(\boldsymbol{x}_j). \end{aligned}$$

Rearranging, this gives

$$\begin{aligned} 0 &= (n-1) \sum_{i=1}^n f^2(\boldsymbol{x}_i) - 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n f(\boldsymbol{x}_i) f(\boldsymbol{x}_j) \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2, \end{aligned}$$

where the second equality follows by (4.13). This holds iff $f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)$, for every $1 \leq i < j \leq n$. By (4.11), this holds iff $\omega_i = \frac{1}{n}$ for every $1 \leq i \leq n$. □

Theorem 4.3. $\sum_{i=1}^n \omega_i^2 = 1$ *iff* $\omega_i = 1$, *for precisely one* $1 \leq i \leq n$, *with all the other* ω*'s identically zero.*

Proof. Using (4.14), $\sum_{i=1}^n \omega_i^2 = 1$, is equivalent to

$$\begin{aligned} \sum_{i=1}^n f^2(\boldsymbol{x}_i) &= \left(\sum_{i=1}^n f(\boldsymbol{x}_i)\right)^2 \\ &= \sum_{i=1}^n f^2(\boldsymbol{x}_i) + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n f(\boldsymbol{x}_i) f(\boldsymbol{x}_j). \end{aligned}$$

Thus,

$$0 = \sum_{i=1}^{n-1} \sum_{j=i+1}^n f(\boldsymbol{x}_i) f(\boldsymbol{x}_j).$$

Since $f \geq 0$, this holds iff $f(\boldsymbol{x}_i) f(\boldsymbol{x}_j) = 0$, for every $1 \leq i < j \leq n$; which, in turn, happens iff $f(\boldsymbol{x}_i) \neq 0$, for at most one $1 \leq i \leq n$. By (4.11), this holds iff $\omega_i = 1$ and $\omega_j = 0$ for $j \neq i$. □

We can now see that

(1) $\sum_{i=1}^n \omega_i^2 = \frac{1}{n}$ precisely when every sampled variate is equally significant in the crude Monte Carlo estimate of $I_A(fw)$; and

(2) $\sum_{i=1}^n \omega_i^2 = 1$ precisely when the crude Monte Carlo estimate of $I_A(fw)$ is determined by only one of the sampled variates, with the rest contributing nothing at all.

Clearly, the first case is much more desirable than the second. So, one way of evaluating the sampling density w is by computing $\sum_{i=1}^n \omega_i^2$, and checking whether it is close to $\frac{1}{n}$ (which is good), or close to 1 (which is bad).

4.4.3. The Optimal Importance Sampler. Our next result does not tell us how to construct a good importance sampler; rather, it is merely an existence theorem[6].

Theorem 4.4. (Optimal Importance Sampler) *Suppose that we have*

$$0 < I_A(|fw|) < \infty.$$

Then there exists an optimal importance sampler, minimizing $\mathbb{V}\mathrm{ar}\left[\hat{I}^v_{A,n}\right]$, *and given by*

$$v_{opt}(\boldsymbol{x}) := \frac{|f(\boldsymbol{x})w(\boldsymbol{x})|}{I_A(|fw|)}.$$

Furthermore, the variance of the corresponding (optimal) importance sampling estimator is

$$\mathbb{V}\mathrm{ar}\left[\hat{I}^{v_{opt}}_{A,n}\right] = \frac{I_A^2(|fw|) - I_A^2(fw)}{n}.$$

Proof. Let the density v be an arbitrary importance sampler, and set

$$q(\boldsymbol{x}) := \frac{|f(\boldsymbol{x})w(\boldsymbol{x})|}{I_A(|fw|)},$$

[6]The extent to which you are impressed by a *mere* existence theorem determines whether you are a pure or applied mathematician.

Then, for $\boldsymbol{X} \sim v$,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{q(\boldsymbol{X}) - v(\boldsymbol{X})}{v(\boldsymbol{X})}\right)^2\right] \\
&\quad = \mathbb{E}\left[\left(\frac{q(\boldsymbol{X})}{v(\boldsymbol{X})} - 1\right)^2\right] \\
&\quad = \mathbb{E}\left[\left(\frac{q(\boldsymbol{X})}{v(\boldsymbol{X})}\right)^2\right] - 2\mathbb{E}\left[\frac{q(\boldsymbol{X})}{v(\boldsymbol{X})}\right] + 1 \\
&\quad = \frac{1}{I_A^2(|fw|)}\mathbb{E}\left[\left(\frac{|f(\boldsymbol{X})w(\boldsymbol{X})|}{v(\boldsymbol{X})}\right)^2\right] - \frac{2}{I_A(|fw|)}\mathbb{E}\left[\frac{|f(\boldsymbol{X})w(\boldsymbol{X})|}{v(\boldsymbol{X})}\right] + 1 \\
&\quad = \frac{1}{I_A^2(|fw|)}\mathbb{E}\left[\left(\frac{|f(\boldsymbol{X})w(\boldsymbol{X})|}{v(\boldsymbol{X})}\right)^2\right] - 1,
\end{aligned}$$

So,

$$\mathbb{E}\left[\left(\frac{|f(\boldsymbol{X})w(\boldsymbol{X})|}{v(\boldsymbol{X})}\right)^2\right] = I_A^2(|fw|)\mathbb{E}\left[\left(\frac{q(\boldsymbol{X}) - v(\boldsymbol{X})}{v(\boldsymbol{X})}\right)^2\right] + I_A^2(|fw|).$$

Consequently, by (4.10),

$$\begin{aligned}
&\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}^{v}(fw)\right] \\
&\quad = \frac{1}{n}\left\{I_A^2(|fw|)\mathbb{E}\left[\left(\frac{q(\boldsymbol{X}) - v(\boldsymbol{X})}{v(\boldsymbol{X})}\right)^2\right] + I_A^2(|fw|) - I_A^2(fw)\right\}.
\end{aligned}$$

This is minimized when $v = q$; in other words, $v_{\mathrm{opt}} = q$. In that case,

$$\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}^{v_{\mathrm{opt}}}(fw)\right] = \frac{I_A^2(|fw|) - I_A^2(fw)}{n},$$

is the optimal (minimal) variance for an importance sampling estimator. □

It is clear from this result that $\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}^{v_{\mathrm{opt}}}(fw)\right] = 0$ precisely when fw is always positive. Then,

$$v_{\mathrm{opt}} = \frac{fw}{I_A(fw)},$$

is simply the result of normalizing fw to produce a density. It is obvious, in this case, that sampling from v_{opt} will produce variates which are optimally distributed for calculating $I_A(fw)$.

There are two obstacles hindering the application of the above theorem to the problem of selecting an importance sampler:

(1) Even if we know which density is the optimal importance sampler, we might not have an algorithm for generating from it; and

(2) Part of the job in determining v_{opt} involves calculating $I_A(|fw|)$ — essentially the integral we want to evaluate in the first place!

Nevertheless, it does give us qualitative information about what a good importance sampler looks like[7].

4.4.4. Pricing Out-of-the-Money Puts. Consider the problem of pricing a European put option, with strike K, on the stock in §3.4. Again, let the option maturity be $T > 0$, and suppose that the risk-free rate is r. By the same argument as in §3.4, the price of this instrument is

$$p(S_0, r, \sigma, T, K) = \mathbb{E}[f(Z)] = \int_{-\infty}^{\infty} f(z)w(z)\,dz,$$

where $Z \sim \mathscr{N}(0,1)$, w is the corresponding density function, and

$$f(z) := \exp(-rT)\left(K - S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\,z\right)\right)^+$$

The crude Monte Carlo estimate of the put premium is

$$\hat{p}_n(S_0, r, \sigma, T, K) := \frac{1}{n}\sum_{i=1}^{n} f(z_i),$$

where $z_1, \ldots, z_n$ are randomly drawn from $\mathscr{N}(0,1)$. But,

$$\begin{aligned}
\mathbb{P}[f(Z) > 0] &= \mathbb{P}\left[S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}Z\right) < K\right] \\
&= \mathbb{P}\left[Z < -\frac{\log(S_0/K) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right] \\
&\approx \mathbb{P}[Z < -1.7829] \\
&\approx 0.0373,
\end{aligned}$$

when $S_0 = 100$, $K = 50$, $r = 0.1$, $\sigma = 0.4$, and $T = 1$. With these (reasonable) parameter values for a deep out-of-the-money put, we see that only about 4% of the samples generated by the crude Monte Carlo estimator have any impact on the final price estimate.

We might consider the density

$$v(x) := \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x - \mu)^2\right),$$

corresponding to the $\mathscr{N}(\mu, 1)$ distribution, as a reasonable importance sampler for the out-of-the-money put pricing problem. We obtain

$$\hat{p}_n^v(S_0, r, \sigma, T, K) := \frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{w(x_i)}{v(x_i)},$$

[7] Showing that existence theorems have their rôle too.

where $x_1, \ldots, x_n$ are sampled from v.

Now, for $X \sim v$, we get

$$\begin{aligned}\mathbb{P}\left[f(X)\frac{w(X)}{v(X)} > 0\right] &= \mathbb{P}[f(X) > 0] \\ &= \mathbb{P}\left[X < -\frac{\log(S_0/K) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}\right] \\ &= \mathbb{P}\left[Z < -\frac{\log(S_0/K) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} - \mu\right].\end{aligned}$$

With the same pricing parameters as before, and setting $\mu = -2$, we get

$$\begin{aligned}\mathbb{P}\left[f(X)\frac{w(X)}{v(X)}\right] &\approx \mathbb{P}[Z < 0.2171] \\ &\approx 0.5859.\end{aligned}$$

In other words, about 59% of the variables generated from v contribute to the option value.

Example 4.4. (Pricing an Out-of-the-Money Put) Figure 4.4 illustrates the behaviour of the of the $\mathcal{N}(\mu, 1)$ importance sampler (with $\mu = -2$) for a European put option, for a stock whose current price is \$100.00, and whose volatility is 40%. A risk-free rate of 10% is assumed. The option maturity is one year, and its strike is \$50.00. The Black-Scholes option price is \$0.24.

4.5. Stratified Sampling

In Chapter 1, our analysis of the performance of random number generators included the use of goodness of fit tests. One approach (in the chi-square test) was to partition the support of the distribution and test the extent to which a sample fitted the distribution by comparing the number of samples in each partition with the expected number in each partition. This suggests a method for improving the distributional characteristics of samples would be to generate the expected number of samples in each partition by design. Each partition is called a *stratum* and the method is known as *stratified sampling*.

For simplicity, we only consider the one-dimensional case. We start by constructing d sets (strata) $A_1, \ldots, A_d \subseteq \mathbb{R}$ such that

- $\text{supp}(w) = A = \bigcup_{i=1}^{d} A_i$; and
- $\forall 1 \leq i \neq j \leq d. A_i \cap A_j = \emptyset$,

where w is the probability density from which we wish to generate random variates.

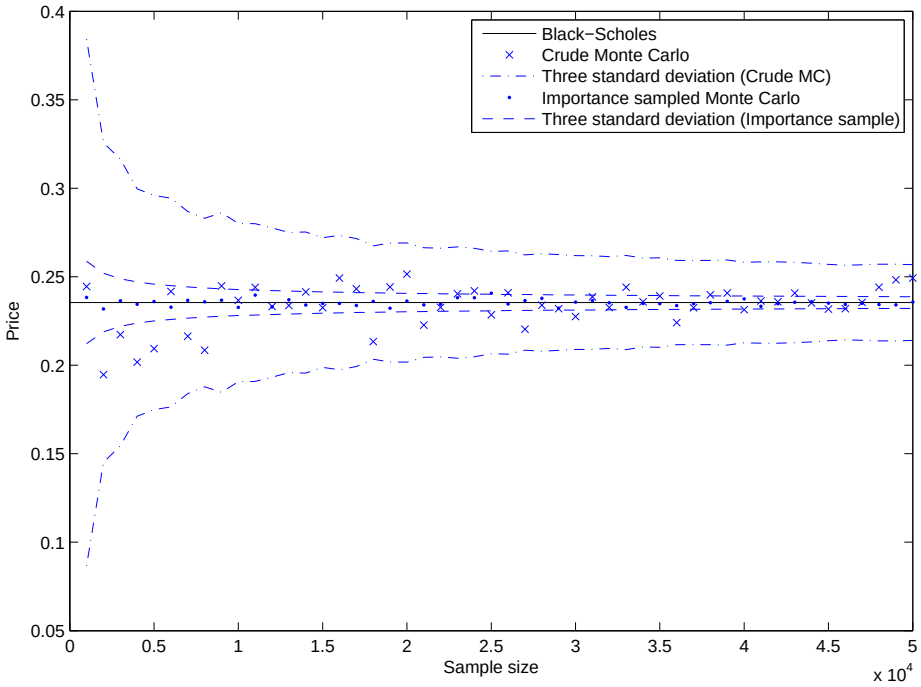


Figure 4.4. Analytic, crude Monte Carlo, and importance sampled option prices against sample size (for samples of $1000, 2000, \ldots, 50\,000$ random variates) for a European put option.

Suppose that we generate n_i samples from each of the i strata, x_{ij} with $1 \leq i \leq d$, $1 \leq j \leq n_i$ and $n = \sum_{i=1}^{d} n_i$. We define the stratified sample estimator as

$$\hat{I}^{\equiv}_{A,n}(fw) = \sum_{i=1}^{d} p_i \hat{I}_{A_i,n_i}(fw) = \sum_{i=1}^{d} p_i \frac{1}{n_i} \sum_{j=1}^{n_i} f(x_{ij}),$$

where $p_i = \mathbb{P}^w[A_i]$. Then, from

$$\mathbb{E}\left[\hat{I}^{\equiv}_{A,n}(fw)\right] = \sum_{i=1}^{d} p_i \mathbb{E}\left[\hat{I}_{A_i,n_i}(fw)\right] = \sum_{i=1}^{d} p_i I_{A_i,n_i}(fw) = I_{A,n}(fw)$$

it follows that $\hat{I}^{\equiv}_{A,n}(fw)$ is an unbiased estimator of $I_{A,n}(fw)$.

Now, for convenience, define the conditional means and variances

$$\mu_i := \mathbb{E}\left[f(X)|X \in A_i\right] = \mathbb{E}[f(x_{ij})]$$

and

$$\sigma_i^2 := \mathbb{V}\mathrm{ar}\left[f(X)|X \in A_i\right] = \mathbb{V}\mathrm{ar}[f(x_{ij})],$$

for all $1 \leq i \leq d$. A proportionally stratified sample is one in which we ensure that $\frac{n_i}{n} = p_i$. In other words we choose $n_i = p_i n$. Note that care should be taken while selecting strata to ensure that the n_i are integers. The variance of a proportionally stratified sample is given by

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\hat{I}^{\equiv}_{A,n}(fw)\right] &= \sum_{i=1}^{d} p_i^2 \mathbb{V}\mathrm{ar}\left[\hat{I}_{A_i,n_i}(fw)\right] \\ &= \sum_{i=1}^{d} p_i^2 \frac{\sigma_i^2}{n_i} \\ &= \frac{1}{n}\sum_{i=1}^{d} p_i \sigma_i^2.\end{aligned}$$

Of course, this last expression can be estimated in the usual manner by using sample estimates for σ_i^2. In particular

$$\widehat{\mathbb{V}\mathrm{ar}}_n\left[\hat{I}^{\equiv}_{A,n}(fw)\right] = \frac{1}{n}\sum_{i=1}^{d} p_i \widehat{\mathbb{V}\mathrm{ar}}_{n_i}[f(x_{ij})].$$

To compare this with the variance of the Monte Carlo estimate without stratification note that

$$\mathbb{E}[f(X)^2] = \sum_{i=1}^{d} p_i \mathbb{E}[f(X)^2 | X \in A_i] = \sum_{i=1}^{d} p_i(\sigma_i^2 + \mu_i^2),$$

where the last equality follows because $\sigma_i^2 = \mathbb{E}[f(X)^2 | X \in A_i] - \mu_i^2$. Now, since $\mathbb{E}[f(X)] = \sum_{i=1}^{d} p_i \mu_i$, we have

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\hat{I}_{A,n}(fw)\right] &= \frac{1}{n}\left(\mathbb{E}[f(X)^2] - \mathbb{E}[f(X)]^2\right) \\ &= \frac{1}{n}\left(\sum_{i=1}^{d} p_i\sigma_i^2 + \sum_{i=1}^{d} p_i\mu_i^2 - \left(\sum_{i=1}^{d} p_i\mu_i\right)^2\right) \\ &\geq \frac{1}{n}\sum_{i=1}^{d} p_i\sigma_i^2 = \mathbb{V}\mathrm{ar}\left[\hat{I}^{\equiv}_{A,n}(fw)\right],\end{aligned}$$

where the last step follows by Jensen's inequality[8]. So, for a proportionally stratified sample, the variance of the Monte Carlo integral is never greater than when using a crude sample.

[8] For a convex function $\varphi : \mathbb{R} \to \mathbb{R}$, numbers $x_1, x_2, \ldots, x_n$ and corresponding positive weights a_i (summing to one), the finite form of Jensen's inequality states that $\varphi(\sum a_i x_i) \leq \sum a_i \varphi(x_i)$.

4.5.1. Stratified Sampling in Higher Dimensions. In considering the method of stratified sampling, we have thus far only looked at the one dimensional case. But, as we know, the real reason for considering Monte Carlo methods (rather than quadrature for example) is so that we can consider the pricing of options in many dimensions.

It is possible to extend the technique described above directly to s dimensions. However, the curse of dimensionality prevents this from being useful. To see this, consider d sets (substrata) $A_{i1}, \ldots, A_{id} \subseteq \mathbb{R}$ for each dimension $1 \leq i \leq s$ such that

- $\text{supp}(w_i) = \bigcup_{j=1}^{d} A_{ij}$; and
- $\forall 1 \leq j \neq k \leq d. A_{ij} \cap A_{ik} = \emptyset$,

where w_i is the marginal probability density for dimension i. There will now be d^s combinations of these sets required to stratify the joint density. Thus given a set of substrata, the number of strata required for the joint density becomes prohibitively unmanageable for even a small number of dimensions (say 4 or more).

This is, however, a way to use one dimensional stratification to improve path dependent options — we explore it now.

4.5.2. End Point Stratification. In the pricing of path-dependent options, the most important feature of the paths generated is usually the terminal value. Much of the variability of an option payoff can be eliminated by stratifying on the basis of terminal stock price value for each path.

This provides us with our first opportunity to make use of Brownian bridges. The idea is rather simple, stratify the terminal stock price into d strata, then generate intermediate stock prices using Brownian bridged paths between S_0 and the terminal prices S_T in each stratum. Since this is effectively a stratification in only one dimension, the analysis in Section 4.5 can be used directly to compute the variance reduction achieved.

Example 4.5. (End Point Stratification using Brownian Bridges)

Consider an additive (vanilla) cliquet call option (also called a ratchet option) with $N = 4$ equi-spaced start dates, which has a discounted payoff function given by

$$\sum_{i=1}^{N} e^{-rih}(S_{ih} - S_{(i-1)h})^+,$$

where $h = T/N$, $T = 1$ is the terminal time and $r = 10\%$ is the risk free rate. This is a compound option which is essentially a sum of forward starting call options struck at the money. The analytical price for this option is given by

$$c = NS_0(\Phi(d_1) - e^{-rh}\Phi(d_2)),$$

where

$$d_1 = \frac{(r + \frac{1}{2}\sigma^2)h}{\sigma\sqrt{h}}, \qquad d_2 = d_1 - \sigma\sqrt{h},$$

and $S_0 = 100$ and $\sigma = 30\%$. We stratify the terminal price of the stock price paths into $d = 50$ strata, each with equal probability, and use Brownian bridges to generate the path values at prior times. The strata for the normal distribution can easily be generated by applying the inverse transform method to strata consisting of equal size segments of the uniform distribution on $[0, 1]$. Figure 4.5 shows the convergence of this end-point-stratified Monte Carlo estimate as a function of sample size in comparison to the crude Monte Carlo estimate and the analytical solution.

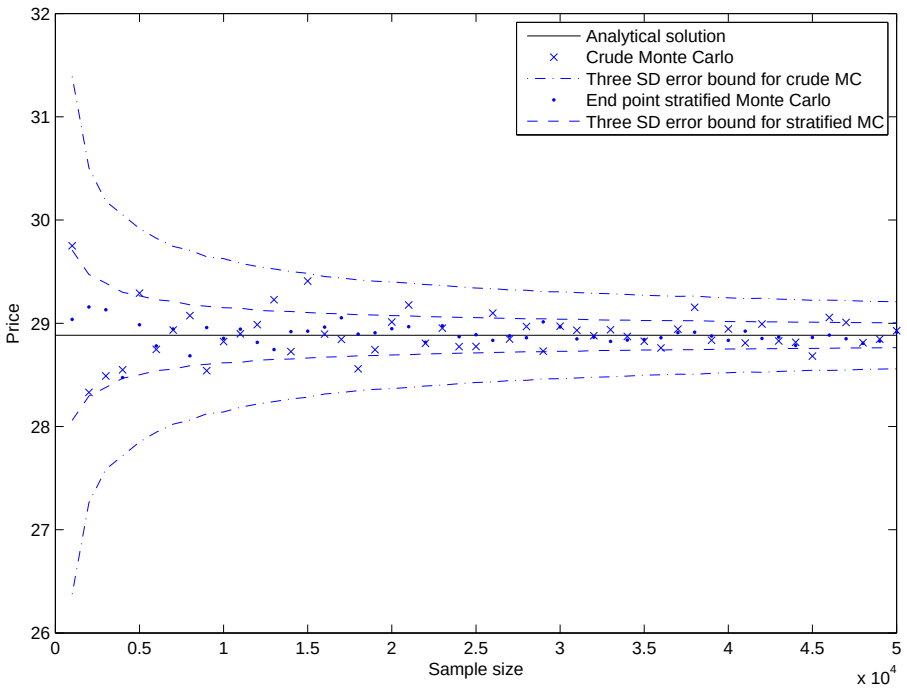


Figure 4.5. Comparison of crude and end-point-stratified Monte Carlo estimates of an additive cliquet option as a function of sample size. For comparison, the analytical price of the option is also provided.

Chapter 5

Quasi-Monte Carlo Integration

5.1. Introduction

In this Chapter, we are concerned with integrals of the form

$$I_{[0,1)^s}(f) := \int_{[0,1)^s} f(\boldsymbol{x})\, d\boldsymbol{x}. \tag{5.1}$$

If the actual integral we want to evaluate is not of this form, it is usually possible to transform the domain of integration into a s-dimensional unit cube, via a suitable sequence of transformations[1]. So, (5.1) offers a reasonably general formulation of the integration problem.

Example 5.1. Many of the integrals we are interested in are of the form $I_{\mathbb{R}^s}(fw)$, where w is a s-dimensional normal density function. In the one-dimensional case, an integral of this form can be transformed as follows,

$$\begin{aligned} I_{\mathbb{R}}(fw) &= \int_{-\infty}^{\infty} f(x)w(x)\, dx \\ &= \int_0^1 f(\tan(2\xi-1)\pi)w(\tan(2\xi-1)\pi)2\pi\left(1+\tan^2(2\xi-1)\pi\right)\, d\xi, \end{aligned}$$

by making the substitution $\xi := \frac{1}{2\pi}\arctan x + \frac{1}{2}$.

From (5.1) it is clear that

$$I_{[0,1)^s}(f) = \mathbb{E}[f(\boldsymbol{U})],$$

[1]It is possible to transform many integration domains into rectangles; and transforming rectangles into unit cubes is easy. For finance applications, the inverse transform method is used in conjunction with the relevant distribution function.

where $\boldsymbol{U} \sim \mathscr{U}[0,1)^s$. Consequently, we could estimate $I_{[0,1)^s}(f)$ using the crude Monte Carlo estimator

$$\hat{I}_{[0,1)^s,n}(f) := \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{u}_i),$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ are randomly generated from $\mathscr{U}[0,1)^s$. The problem with this is that uniform random numbers tend to "cluster" (this is the source of the slow convergence rate for the crude Monte Carlo scheme).

The idea behind *quasi-Monte Carlo methods* is that the points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ are not sampled randomly. Instead, they are generated by a deterministic rule, designed to cover $\mathscr{U}[0,1)^s$ in an *optimally regular* (i.e., better than random) way. The inspiration is the following remarkable result [**17**].

Theorem 5.1. (Weyl's criterion) *Suppose that f is Riemann-integrable on $[0,1)$, and that z_0 is an irrational number. Then*

$$\lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} f(\{(j-1)z_0\}) = I_{[0,1)}(f), \tag{5.2}$$

Here $\{x\} := x - \lfloor x \rfloor$ denotes the fractional part of x, which is computed by subtracting the floor of x from x.

Proof. (Optional) The idea of the proof is to first prove the result for a very simple class of functions, namely complex exponentials, and then to extend the result by suitable approximation arguments. For each integer k set $g_k(x) = \exp(2\pi i k x)$. If $k = 0$ then (5.2) holds trivially. If $k \neq 0$ then the right hand side of (5.2) equals zero. Noting that $\exp(2\pi i l) = 1$ for any $l \in \mathbb{Z}$ we compute and find, using the usual formula for a geometric series,

$$\frac{1}{n}\sum_{j=0}^{n-1} \exp(2\pi i k\{jz_0\}) = \frac{1}{n}\sum_{j=0}^{n-1} \exp(2\pi i k j z_0) = \frac{1}{n}\left(\frac{1-\exp(2\pi i k n z_0)}{1-\exp(2\pi i k z_0)}\right)$$

The denominator (of the bracketed expression) here is always bounded away from zero by the irrationality of z_0, while the numerator is trivially bounded above by two. In total then, the whole expression tends to zero as $n \to \infty$. We can now invoke the Stone-Weierstrass approximation theorem to show that (5.2) holds for continuous one-periodic functions too — we have shown that the result holds for the algebra of complex exponentials, and this is a sufficiently rich algebra for the Stone-Weierstrass theorem to apply. Next, let g be the indicator function of an interval, $[a, b] \subset [0, 1)$. Given $\varepsilon > 0$ we can find continuous periodic functions $M(x)$ and $m(x)$ such that we have $m \leq g \leq M$, and also $||M - m||_1 < \varepsilon$. Using the fact that (5.2) holds for both m and M we can now show that

$$\int_0^1 m\,dx \leq \liminf_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} g\left(\{jz_0\}\right) \leq \limsup_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} g(\{jz_0\}) \leq \int_0^1 M\,dx,$$

and, since $\varepsilon > 0$ was arbitrary we conclude that (5.2) holds for indicator functions of intervals too (and thus for step functions as well). Finally, if f is merely Riemann integrable then we know that, given $\varepsilon > 0$, we can find step functions $U(x)$ and $L(x)$ such that $L \leq f \leq U$ with $\int_0^1 (U - L)\,dx < \varepsilon$. Arguing now as in the case of the indicator functions (using (5.2) for U and L, and our inequality, $L \leq f \leq U$) shows that we have our result, (5.2), holding for f too. □

Corollary 5.1. *Let z_0 be an irrational number. Then,*

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{[a,b)}(\{(i-1)z_0\}) = b - a,$$

for any interval $[a, b) \subset [0, 1)$.

One of the details of the proof of the theorem above is that the approximation error satisfies

$$\left| \frac{1}{n}\sum_{i=1}^{n} f(\{(i-1)z_0\}) - I_{[0,1)}(f) \right| = O\left(\frac{1}{n}\right).$$

This is better than any Monte Carlo method, where (as we have seen) the error declines like $\frac{1}{\sqrt{n}}$. The improvement in convergence rate is the primary reason why quasi-Monte Carlo integration is advocated.

A quasi-Monte Carlo integration technique is very much like crude Monte Carlo integration based upon uniformly distributed samples; except, the samples are not chosen randomly. Quasi-Monte Carlo methods are defined by their rules for covering the s-dimensional unit cube. We differentiate between two main types of method; based upon whether the rules for generating sample points are *open* or *closed*:

Open rules: These have the pleasant feature that, having generated n sample points, the accuracy of an approximation can be improved simply by adding more points to the existing set. An open rule will add new points by, in a sense, trying to "fill the remaining gaps" optimally. Examples of quasi-random sequences generated by open rules are *Sobol' sequences*, *van der Corput sequences*, and *Halton sequences*.

Closed rules: These, on the other hand, are characterized by the feature that, in order to generate n sample points, n must be known in advance. In a sense, a closed rule is designed to produce an optimal sequence of size n, for any given n. The problem is that if we want to improve the accuracy of an estimate, a closed rule doesn't allow us to append new points to an existing sequence — an optimal sequence of size n is not necessarily a subset of an optimal sequence of size $n+1$! The requirement of having to regenerate a sequence from scratch, in order to increase the sample size, is a serious drawback. Examples of closed quasi-random sequences are *Hammersley sequences*, and *(t,m,k)-nets and (t,k)-sequences*.

5.2. Error Analysis

Definition 5.1. (Equidistributed sequences) We will call a sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \in [0,1)^s$ *equidistributed* in $[0,1)^s$ if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x}_i) = \prod_{j=1}^{s} (b_j - a_j),$$

for any cube $[\boldsymbol{a},\boldsymbol{b}) := \prod_{j=1}^{s} [a_j, b_j) \subset [0,1)^s$.

Theorem 5.2. (Multidimensional Weyl criterion) *Let f be Riemann-integrable. Then,*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) = I_{[0,1)^s}(f),$$

for any equidistributed sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \in [0,1)^s$.

Example 5.2. (Multiples of an irrational) By Corollary 5.1,

$$0, \{z_0\}, \{2z_0\}, \{3z_0\}, \ldots,$$

is equidistributed in $[0,1)$, for z_0 irrational.

While, in general, it is not possible to store irrational numbers on a computer, in certain cases we can compute approximations to these sequences. Consider the sequence that makes use of $\sqrt{3}$:[2] $0, \left\{\sqrt{3}\right\}, \left\{2\sqrt{3}\right\}, \left\{3\sqrt{3}\right\}, \ldots$

[2]Recall that the square root of any non-square number is irrational.

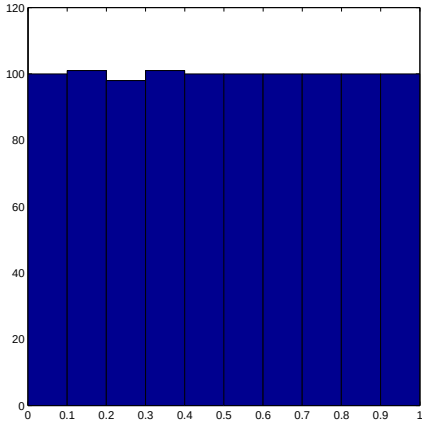


Figure 5.1. Histogram of the first 1000 points of the equidistributed sequence $x_i(\sqrt{3})$.

The ith term in this sequence can be written as

$$\begin{aligned} x_i(\sqrt{3}) &= \left\{(i-1)\sqrt{3}\right\} \\ &= \left\{\sqrt{3(i-1)^2}\right\} \\ &= \sqrt{3(i-1)^2} - \left\lfloor \sqrt{3(i-1)^2} \right\rfloor, \end{aligned} \tag{5.3}$$

where $\lfloor \cdot \rfloor$ is the floor function. So, as long as we implement (5.3) and limit ourselves to the first few terms, machine precision should not be a major issue.

Figure 5.1 shows a histogram of the first 1000 numbers generated using this approach.

Example 5.3. (Linearly independent irrationals) If $z_1, \ldots, z_s$ are irrational numbers, such that $1, z_1, \ldots, z_s$ are linearly independent over $\mathbb{Q}$, then the sequence

$$(0, \ldots, 0)^T, (\{z_1\}, \ldots, \{z_s\})^T, (\{2z_1\}, \ldots, \{2z_s\})^T, (\{3z_1\}, \ldots, \{3z_s\})^T, \ldots,$$

is equidistributed over $[0, 1)^s$.

As in the one dimensional case, it is possible to compute approximations to these sequences in certain cases. Figure 5.2 shows a scatter plot of the sequence $\left\{ \left(x_i(\sqrt{3}), x_i(\sqrt{7})\right) \middle| \, 1 \leq i \leq 1000 \right\}$.

The following example demonstrates what can happen if the linear independence condition in Example 5.3 is violated.

Example 5.4. (Linear independence is important) Suppose that $z_1 \in \left[0, \frac{1}{2}\right)$ is irrational. Then $z_2 := z_1 + 1$ is irrational as well. Now, $1, z_1, z_2$ are

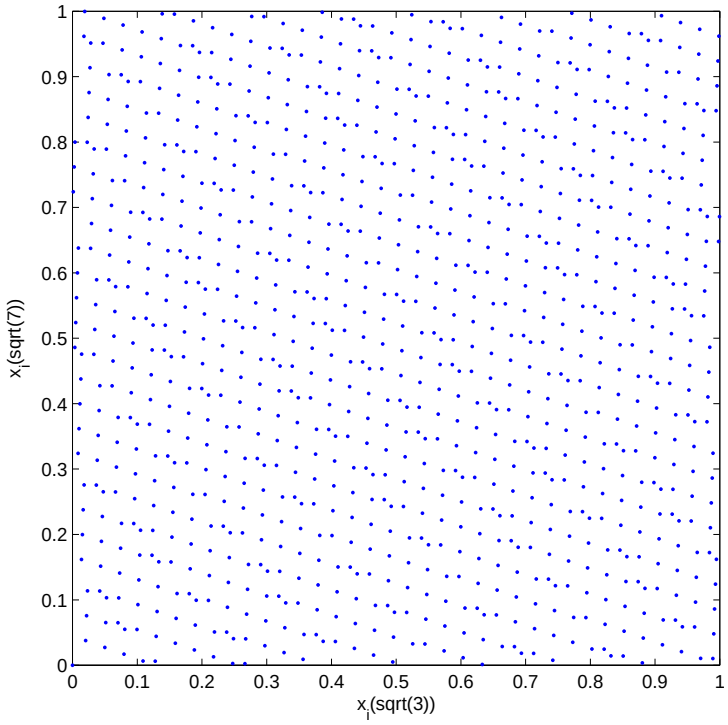


Figure 5.2. Scatter plot of the first 1000 points of the two dimensional equidistributed sequence $\big(x_i(\sqrt{3}), x_i(\sqrt{7})\big)$.

not linearly independent over $\mathbb{Q}$, since

$$1 + z_1 - z_2 = 0.$$

It is also easy to verify that the sequence

$$(0,0)^T, (\{z_1\},\{z_2\})^T, (\{2z_1\},\{2z_2\})^T, (\{3z_1\},\{3z_2\})^T, \ldots$$

is not equidistributed over $[0,1)^2$. Since $\{iz_1\} = \{iz_2\}$, for every $i > 0$, we have

$$\{iz_1\} \in \left[0, \frac{1}{2}\right) \Leftrightarrow \{iz_2\} \notin \left[\frac{1}{2}, 1\right)$$

Consequently,

$$\begin{aligned}\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{[0,\frac{1}{2})\times[\frac{1}{2},1)}(\{iz_1\},\{iz_2\}) &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{[0,\frac{1}{2})}(\{iz_1\})\mathbb{I}_{[\frac{1}{2},1)}(\{iz_2\}) \\ &= 0 \neq \frac{1}{4} = \left(\frac{1}{2} - 0\right)\left(1 - \frac{1}{2}\right).\end{aligned}$$

So, the linear independence condition is necessary for equidistributivity.

By Theorem 5.2, it is clear that equidistributed sequences can be used to approximate the values of integrals such as (5.1).

Sequences such as those in Example 5.3 are not practically useful, however, due to computer precision issues. Fortunately, there are also many computable rules for generating equidistributed sequences. The question is: how does one choose between competing sequences, for the purposes of evaluating the integral in (5.1)?

Equidistributed sequences do not behave like random sequences[3], and, hence, error assessment based upon probabilistic reasoning is not appropriate. Instead, the suitability of such a sequence for estimating (5.1) is determined by asymptotic error analysis — i.e., by examining the convergence rate of

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) - I_{[0,1)^s}(f) \right|,$$

as $n \to \infty$. To this end, the following definitions are useful.

Definition 5.2. (Discrepancy) We define the *discrepancy* of a sequence $\{\boldsymbol{x}_j\} \subset [0,1)^s$ by

$$D_n := \sup_{(s_1,\ldots,s_s) \leq (t_1,\ldots,t_s) \in [0,1)^s} \left| \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\prod_{j=1}^{s}[s_j,t_j)}(\boldsymbol{x}_i) - \prod_{j=1}^{s}(t_j - s_j) \right|.$$

This measures the worst absolute error of a quasi-random rule, when approximating the $\mathscr{U}[0,1)^s$ distribution function. Clearly, $\lim_{n\to\infty} D_n = 0$ is equivalent to the sequence being equidistributed.

Definition 5.3. (Star discrepancy) The *star discrepancy* of a sequence $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \in [0,1)^s$ is defined by

$$D_n^* := \sup_{(t_1,\ldots,t_s) \in [0,1)^s} \left| \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\prod_{j=1}^{s}[0,t_j)}(\boldsymbol{x}_i) - \prod_{j=1}^{s} t_j \right|.$$

Exercise 5.1. (An explicit discrepancy computation) Consider the sequence $x_1, \ldots, x_n \in [0,1)$, defined by

$$x_i := \frac{(i-1)^2}{n^2},$$

for each $1 \leq i \leq n$.

(1) Explain whether this sequence is open or closed.

[3]Since they possess the integration properties of random sequences, but not the independence or randomness qualities, equidistributed sequences are often referred to as *quasi-random sequences*.

(2) Prove that

$$\lim_{n\to\infty} D_n^* = \frac{1}{4},$$

(**Hint:** First demonstrate that

$$D_n^* = \max\left\{ \frac{i}{n} - \frac{(i-1)^2}{n^2} \,\middle|\, 1 \le i \le n \right\}$$

for each $n \ge 1$.)

Exercise 5.2. (Inequalities between discrepancy and star discrepancy) Prove that the inequality

$$D_n^* \le D_n \le 2^s D_n^*, \tag{5.4}$$

always holds.

By (5.4), we have

$$\lim_{n\to\infty} D_n^* = 0 \Rightarrow \lim_{n\to\infty} D_n = 0.$$

Hence, convergence of the star discrepancy of a sequence indicates that it is equidistributed.

For the remainder of this section, let us assume that $f \in C^s([0,1)^s)$. This is a technical condition, required by the next definition.

Definition 5.4. (Vitali variation) The *Vitali variation* of a function f is defined by

$$V^{(s)}(f) := \int_{[0,1)^s} \left| \frac{\partial^s}{\partial x_1 \dots \partial x_s} f(x_1, \dots, x_s) \right| dx_s \dots dx_1.$$

The Vitali variation is a generalization, to multiple dimensions, of the concept of the *variation* of a function. In particular, when $s = 1$ it reduces to the standard definition of variation given by

$$V^{(1)}(f) = \sup \sum_{i=1}^{n} |f(x_i) - f(x_{i-1})|,$$

where the supremum is taken over all possible partitions $0 = x_0 < x_1 < \ldots < x_n < 1$ of $[0,1)$. Note that the differentiability requirement is a convenience and is not required for the most general definition of the Vitali variation.

Now, let $1 \le i_1 \le \ldots \le i_l \le s$, where $1 \le l \le s$. Define the function $f_{i_1,\dots,i_l} \colon [0,1)^s \to \mathbb{R}$ by

$$f_{i_1,\dots,i_l}(x_1, \dots, x_s) := f(x_1', \dots, x_s'),$$

where

$$x_i' := \begin{cases} x_i, & \text{if } i \in \{i_1, \dots, i_l\}, \\ 0, & \text{otherwise.} \end{cases}$$

for every $1 \leq i \leq s$. It should be clear that $f_{i_1,\ldots,i_l}$ is just an l-dimensional cross-section of f.

Definition 5.5. (HARDY-KRAUSE VARIATION) The expression

$$V(f) := \sum_{l=1}^{s} \sum_{1 \leq i_1 \leq \ldots \leq i_l \leq s} V^{(l)}(f_{i_1,\ldots,i_l}),$$

is called the *Hardy-Krause variation* of f.

It is clear from the above definition that when $s = 1$, then we have $V(f) = V^{(1)}(f)$. We now have the following famous inequality.

Theorem 5.3. (KOKSMA-HLAWKA INEQUALITY) *For any sequence* $\{\boldsymbol{x}_j\}$ *in* $[0,1)^s$,

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) - I_{[0,1)^s}(f) \right| \leq V(f) D_n^*,$$

Proof. See [**11**, page 151], for example. □

This theorem gives us a technique for analyzing the convergence rates of integral approximations based on pseudo-random sequences, and for comparing such sequences with each other for suitability with respect to specific integration problems. That technique is simply an examination of the asymptotic behaviour of the star discrepancy of the sequence(s).

5.3. Open Rules

5.3.1. Introduction. As we mentioned, the defining characteristic of *open rules* for generating quasi-random sequences is that, for such a rule, the sequence of length n is always a subsequence of the sequence of length $n+1$. This means that enlarging a given open sequence is merely a matter of adding new points to the existing set. In this section, we will examine *van der Corput sequences*, *Halton sequences* and *Sobol' sequences*.

5.3.2. Van der Corput Sequences. Let $r > 1$ be an integer. Any $m \in \mathbb{N}$ can be expressed uniquely in base (radix) r as

$$m = a_0 + a_1 r + a_2 r^2 + \cdots + a_l r^l,$$

where $0 \leq a_i \leq r-1$, $a_l \neq 0$, and $r^l \leq m < r^{l+1}$. Now, the *base r radical inverse function* $\varphi_r \colon \mathbb{N} \to [0,1)$ is defined by

$$\varphi_r(m) := a_0 r^{-1} + a_1 r^{-2} + a_2 r^{-3} + \cdots + a_l r^{-l-1},$$

(In other words, if the base r representation of m is $a_l a_{l-1} \ldots a_0$, then the base r representation of $\varphi_r(m)$ is $0.a_0 a_1 \ldots a_l$.) The sequence

$$\varphi_r(0), \varphi_r(1), \varphi_r(2), \varphi_r(3), \ldots,$$

is called a *van der Corput sequence*. Table 1 lists the first ten elements of the van der Corput sequence with $r = 2$. It can be shown, for van der Corput

m	m(base 2)	$\varphi_2(m)$(base 2)	$\varphi_2(m)$
0	0	0.0000	0.0000
1	1	0.1000	0.5000
2	10	0.0100	0.2500
3	11	0.1100	0.7500
4	100	0.0010	0.1250
5	101	0.1010	0.6250
6	110	0.0110	0.3750
7	111	0.1110	0.8750
8	1000	0.0001	0.0625
9	1001	0.1001	0.5625
⋮	⋮	⋮	⋮

Table 1. The first ten van der Corput numbers with base $r = 2$.

sequences generated using base r, that

$$\lim_{n\to\infty} D_n = \lim_{n\to\infty} D_n^* = \begin{cases} \frac{\log n}{n}\frac{r^2}{4(r+1)\log r} & \text{for even } r, \\ \frac{\log n}{n}\frac{r-1}{4\log r} & \text{for odd } r, \end{cases} \tag{5.5}$$

which implies that

$$D_n^* = O\left(\frac{\log n}{n}\right).$$

This, in turn, implies that these sequences are equidistributed. It can also be shown that the above rate of convergence is the best that can be achieved for equidistributed sequences with $s = 1$. To minimize D_n^* the optimal choice for the base is $r = 3$ (this is easily verified using the above result).

Example 5.5. (European cash-or-nothing option using a Van der Corput sequence)

Consider a cash-or-nothing call option which has the following payoff function

$$X\mathbb{I}_{\{S_T>K\}},$$

where $X = 10$ is the amount of cash and $K = 110$ is the strike price. The initial stock price $S_0 = 100$ and its volatility is $\sigma = 40\%$, while the risk free rate is $r = 10\%$.

We will use a Van der Corput sequence of base $b = 3$ to provide an estimate for the quasi-Monte Carlo integral $I_{(0,1)^1}(f)$ where

$$f(x_i) = e^{-rT}X\mathbb{I}_{\{S_T(x_i)>K\}},$$

and $S_T(x_i)$ is the risk-neutral terminal stock price given $x_i \in (0,1)$. Because the payoff is so simple, the Hardy-Krause variation is given by $V(f) = e^{-rT}X$. In fact, we have chosen this option precisely because this will allow us to demonstrate definite error bounds for the quasi-Monte Carlo integral through the Koksma-Hlawka inequality.

We use the sequence $\varphi_3(i)$, $0 < i \leq 50000$ to price the option for the sample sizes $1000, 2000, \ldots, 50000$, with the error bounds taken by multiplying (5.5) with the expression for the Hardy-Krause variation above. Figure 5.3 shows the result, where, for comparison, we also show a crude Monte Carlo sample and three standard deviation error bounds.

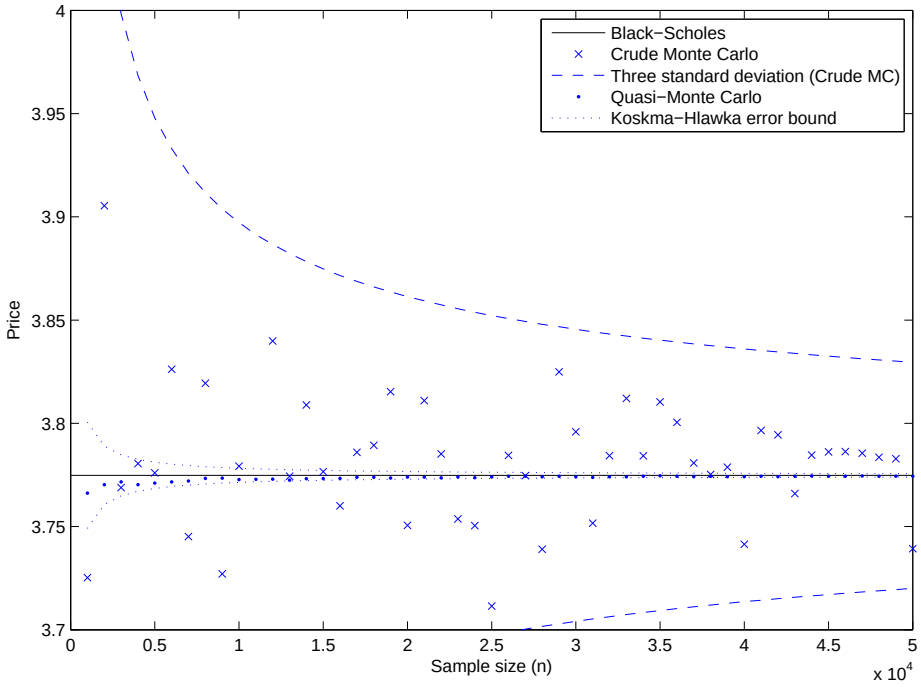


Figure 5.3. Comparison of Crude Monte Carlo, Van der Corput prices for a European cash-or-nothing option versus sample size (for samples of size 1000, 2000, ..., 50 000).

5.3.3. Halton Sequences. *Halton sequences* are multi-dimensional generalizations of van der Corput sequences. Suppose the bases $r_1, \ldots, r_s$ are pairwise relatively prime (in order to avoid correlations between coordinates). Then the Halton sequence in $[0,1)^s$, corresponding to the above bases, is

$$(\varphi_{r_1}(0), \ldots, \varphi_{r_s}(0))^T, (\varphi_{r_1}(1), \ldots, \varphi_{r_s}(1))^T, (\varphi_{r_1}(2), \ldots, \varphi_{r_s}(2))^T, \ldots$$

It is common to take $r_1, \ldots, r_s$ to be the first s primes. It can be shown that

$$D_n^* \leq \left(\prod_{j=1}^{s} \frac{r_j - 1}{2 \log r_j} \right) \frac{(\log n)^s}{n} + \frac{(\log n)^{s-1}}{n},$$

which implies that

$$D_n^* = O\left(\frac{(\log n)^s}{n} \right).$$

It is conjectured that this is the best asymptotic error rate for a sequence in $[0,1)^s$. *Note, however, the dimensional effect implicit in the asymptotic constant and in the rate of convergence.*

If we take the asymptotic error rate for Monte Carlo integration to be $\frac{1}{\sqrt{n}}$, and we assume that the corresponding error rate for integration based upon a Halton sequence is $\frac{(\log n)^s}{n}$, then from

$$\begin{aligned} \lim_{n \to \infty} \left(\frac{\frac{(\log n)^s}{n}}{\frac{1}{\sqrt{n}}} \right) &= \lim_{n \to \infty} \frac{(\log n)^s}{\sqrt{n}} \\ &= 0, \end{aligned}$$

we can conclude that a quasi-Monte Carlo integral estimate, based on a Halton sequence, displays better asymptotic convergence than a Monte Carlo estimate — *irrespective of dimension*! However, the asymptotic constants for Monte Carlo techniques do not exhibit a dimensional effect.

Figure 5.4 plots the first 1000 points of the two-dimensional Halton sequence with $r_1 = 2$ and $r_2 = 3$.

5.3.4. Sobol' Sequences. This section contains a (brief) presentation of *"Algorithm 659"* for generating Sobol' sequences. This algorithm was first published in [**3**]. For a full description of how the Sobol' sequence is in fact a permutation of the van der Corput sequence in base two see [**5**].

To generate a Sobol' sequence, we must first produce a sequence of *direction numbers* $m_1, m_2, \ldots \in \mathbb{N}$. These are odd integers satisfying $0 < m_i < 2^i$, for every $i \in \mathbb{N}$. The direction numbers are, in turn, generated from a *primitive polynomial over* $\mathbb{Z}_2$.

Definition 5.6. (Polynomial) A *polynomial of degree* $d \in \mathbb{Z}^+$ *over* $\mathbb{Z}_2$ is an expression of the form

$$p(x) = x^d + a_1 x^{d-1} + \cdots + a_{d-1} x + a_d,$$

with coefficients in the field $\mathbb{Z}_2$ (i.e., $a_1, \ldots, a_d \in \{0, 1\}$). We denote the set of polynomials in a variable x over $\mathbb{Z}_2$ by $\mathbb{Z}_2[x]$. It inherits all the operations

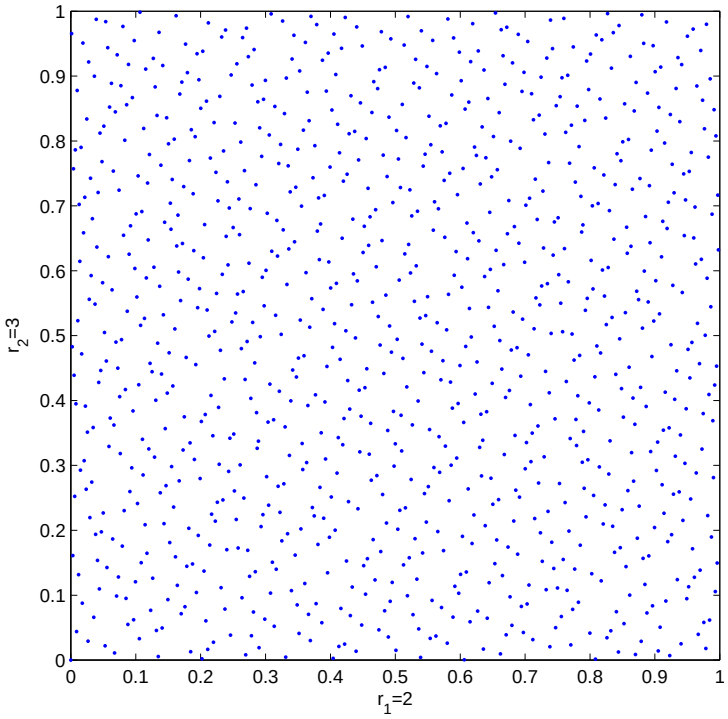


Figure 5.4. The first 1000 two-dimensional Halton points with bases $r_1 = 2$ and $r_2 = 3$.

of $\mathbb{Z}_2$, except inverses of non-zero elements, in a natural way. Thus $\mathbb{Z}_2[x]$ has the structure of a ring — it is an example of a *polynomial ring.*

Definition 5.7. (Order of a polynomial) The *order* of a polynomial $p(x) \in \mathbb{Z}_2[x]$, with $p(0) \neq 0$, is the smallest integer n such that $p(x)|x^n + 1$.

Definition 5.8. (Primitive polynomial) A polynomial $p(x) \in \mathbb{Z}_2[x]$ of degree d is called *primitive* if

(1) it is irreducible (i.e. it cannot be factored); and

(2) it is of order $2^d - 1$.

The first condition implies that $p(x)$ must be of the form

$$p(x) = x^d + a_1 x^{d-1} + \cdots + a_{d-1} x + 1. \tag{5.6}$$

It is customary to represent primitive polynomials in a compact binary format where the binary digits (bits) represent the coefficients of the polynomial. For example, the polynomial $x^3 + x^2 + 1$ is represented by the binary number $1101_2 = 13$. Since many authors have provided lists of primitive

polynomials, it is not necessary to calculate them from scratch. For convenience, Table 2 provides a list of all the primitive polynomials of degree 8 or less [**5**].

Degree	Primitive Polynomials
0	1
1	3 $(x+1)$
2	7 (x^2+x+1)
3	11 (x^3+x+1), 13 (x^3+x^2+1)
4	19, 25
5	37, 59, 47, 61, 55, 41
6	67, 97, 91, 109, 103, 115
7	131, 193, 137, 145, 143, 241, 157, 185, 167, 229, 171, 213, 191, 253, 203, 211, 239, 247
8	285, 369, 299, 425, 301, 361, 333, 357, 351, 501, 355, 397, 391, 451, 463, 487

Table 2. Primitive polynomials of degree 8 or less. The binary representation of each number in the right column gives the coefficients of the polynomial.

Given the primitive polynomial (5.6), the following algorithm generates k direction numbers $m_1, \ldots, m_k$.

Definition 5.9. (Generating direction numbers)

(1) `Select odd integer values for` $m_1, \ldots, m_d$, `subject to the constraint` $0 < m_i < 2^i$;

(2) `For` $i = d+1$ `to` k `set`

$$m_i = 2a_1 m_{i-1} \oplus 2^2 a_2 m_{i-2} \oplus \cdots \oplus 2^{d-1} a_{d-1} m_{i-d+1} \oplus 2^d m_{i-d} \oplus m_{i-d};$$

Definition 5.10. (Bitwise exclusive-or operation) The operation $\oplus$ denotes the *bitwise exclusive-or operation* (so, for example $5 \oplus 7 = 101_2 \oplus 111_2 = 010_2 = 2$).

Definition 5.11. (Generating Sobol' sequences) Having computed a sequence of direction numbers $m_1, \ldots, m_k$, the Sobol' sequence $x_1, \ldots, x_n$ is determined by the recurrence

$$x_1 = 0$$
$$x_{i+1} = x_i \oplus \frac{m_{c(i-1)}}{2^{c(i-1)}}, \tag{5.7}$$

for $1 \le i \le n-1$. Here $c(i)$ is the position of the least significant zero in the binary expansion of i (e.g. $c(5) = c(101_2) = 2$, $c(7) = c(111_2) = 4$ and $c(10) = c(1010_2) = 1$).

From (5.7) it is apparent that the value of k places a bound on n. Explicitly, we must have $c(i-1) \leq k$, for every $1 \leq i \leq n-1$. Now, the smallest integer $n_{\text{crit}} \in \mathbb{N}$ for which $c(n_{\text{crit}}) > k$ is

$$n_{\text{crit}} = \sum_{j=1}^{k} 2^{j-1} = 2^k - 1,$$

Thus, we require that $i - 1 < n_{\text{crit}}$, which means that the requirement on n is given by $n - 2 < n_{\text{crit}} \Rightarrow n < 2^k + 1$.

Up to this point we have only considered generating a Sobol' sequence in one dimension. It should be noted that when generating multi-dimensional sequences, there are further restrictions on the initial values of the direction numbers $m_1, \ldots, m_d$ of Definition 5.9. Further information on these intricacies can be found in Glasserman [**5**].

Example 5.6. (Sobol' sequences) The primitive polynomial $p_1(x) = x^3 + x^2 + 1$ (binary representation 13) was used to generate direction numbers $m_1^{(1)}, \ldots, m_{10}^{(1)}$, with $m_1^{(1)} = 1$, $m_2^{(1)} = 3$ and $m_3^{(1)} = 5$. The final vector of m's is given by $\begin{bmatrix} 1 & 3 & 5 & 3 & 29 & 23 & 53 & 159 & 401 & 703 \end{bmatrix}$. These were then used to produce $x_1^{(1)}, \ldots, x_{1000}^{(1)}$, using (5.7). The direction numbers $m_1^{(2)}, \ldots, m_{10}^{(2)}$ were then generated from the primitive polynomial $p_2(x) = x^4 + x + 1$ (binary representation 19), with $m_1^{(2)} = 1$, $m_2^{(2)} = 1$, $m_3^{(2)} = 3$ and $m_4^{(2)} = 13$. The final vector of m's is given by $\begin{bmatrix} 1 & 1 & 3 & 13 & 25 & 9 & 91 & 21 & 449 & 577 \end{bmatrix}$. Using (5.7), these direction numbers were used to generate $x_1^{(2)}, \ldots, x_{1000}^{(2)}$. Figure 5.5 displays the first 1000 points

$$\boldsymbol{x}_1 = \left(x_1^{(1)}, x_1^{(2)}\right)^T, \ldots, \boldsymbol{x}_{1000} = \left(x_{1000}^{(1)}, x_{1000}^{(2)}\right)^T$$

of the two-dimensional Sobol' sequence.

Exercise 5.3. (Proving polynomials primitive) Demonstrate that the polynomials $p_1(x), p_2(x) \in \mathbb{Z}_2[x]$, from Example 5.6, are primitive.

The star discrepancy of an s-dimensional Sobol' sequence is

$$D_n^* = O\left(\frac{(\log n)^s}{n}\right),$$

as is the case for Halton sequences.

Example 5.7. (European basket option using Sobol' sequences) We return to the problem posed in Example 4.1. There we priced a European basket option using crude Monte Carlo and later improved our estimate using a control variate. We aim to repeat this exercise using Sobol'

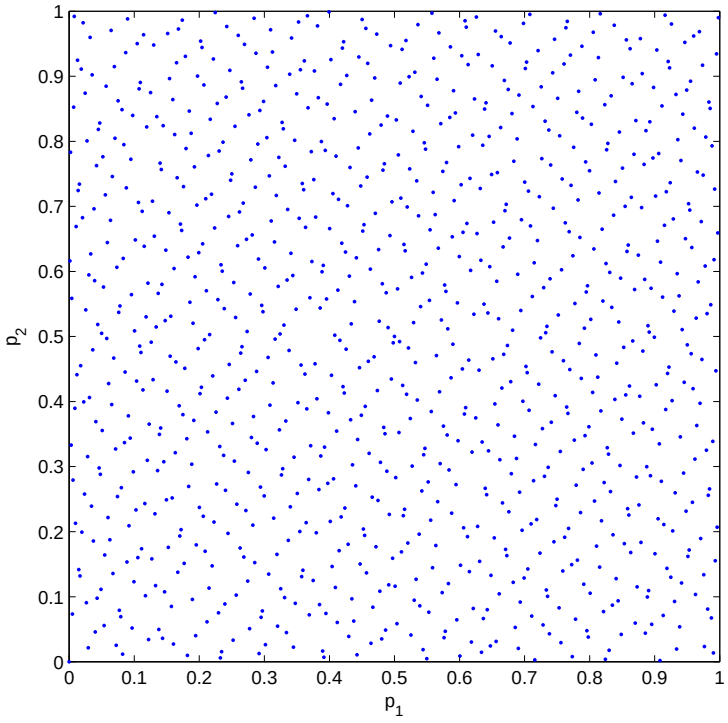


Figure 5.5. The first 1000 points of a two-dimensional Sobol' sequence. The first coordinates were generated from the primitive polynomial $p_1(x) = x^3 + x^2 + 1$, with initial direction numbers $m_1^{(1)} = 1$, $m_2^{(1)} = 3$ and $m_3^{(1)} = 5$. The second coordinates were generated from the primitive polynomial $p_2(x) = x^4 + x + 1$, with initial direction numbers $m_1^{(2)} = 1$, $m_2^{(2)} = 1$, $m_3^{(2)} = 3$ and $m_4^{(2)} = 13$.

sequences. We have three non-dividend paying stocks $S^{(A)}$, $S^{(B)}$ and $S^{(C)}$, whose prices follow geometric Brownian motions. The stock price volatilities are $\sigma_A = 40\%$, $\sigma_B = 25\%$ and $\sigma_C = 30\%$, respectively. The stock price returns are correlated, with correlation coefficients given by $\rho_{AB} = 0.80$, $\rho_{AC} = 0.90$, and $\rho_{BC} = 0.75$. We will determine the price of a European basket option, whose payoff at expiry date $T = 18$ months is

$$\max\left\{S_T^{(A)} + S_T^{(B)} + S_T^{(C)} - 200, 0\right\},$$

where $S_0^{(A)} = \text{R}25.00$, $S_0^{(B)} = \text{R}55.00$ and $S_0^{(C)} = \text{R}120.00$. We take the interest rate to be 10% (NACC).

We use the following primitive polynomials and associated direction numbers in what follows: $p_1(x) = x^2 + x + 1$ (7), with $m_1^{(1)} = 1$ and $m_2^{(1)} = 3$ as the first two direction numbers. $p_2(x) = x^3 + x + 1$ (11), with the initial direction numbers $m_1^{(2)} = 1$, $m_2^{(2)} = 1$ and $m_3^{(2)} = 5$, and

finally $p_3(x) = x^3 + x^2 + 1$ (13), with the first three direction numbers $m_1^{(3)} = 1$, $m_2^{(3)} = 3$ and $m_3^{(3)} = 7$. We employ our Sobol' sequences to price the Basket option for the sample sizes $1000, 2000, \ldots, 50\,000$. We use the primitive polynomial p_1 and associated direction numbers for the first random vector, the polynomial p_2 and associated direction numbers for the second random vector, and the polynomial p_3 and associated direction numbers for the third. We transform $\mathscr{U}(0,1)$ numbers into $\mathscr{N}(0,1)$ numbers using a numerical approximation to Φ^{-1} (the 'inverse transform method'). The resulting quasi-$\mathscr{N}_3(\mathbf{0}, I)$ distributed random vectors are then, finally, transformed into quasi-$\mathscr{N}_3(\mathbf{0}, \Sigma)$ random vectors (Σ, recall the notation of Example 4.1, reflects the correlation of the stocks in our basket) and used to price our basket option. A plot of the prices obtained using the Sobol' sequences, with a comparison to the crude and control variate Monte Carlo results that we obtained in the previous chapter, Example 4.1, is pictured in Figure 5.6.

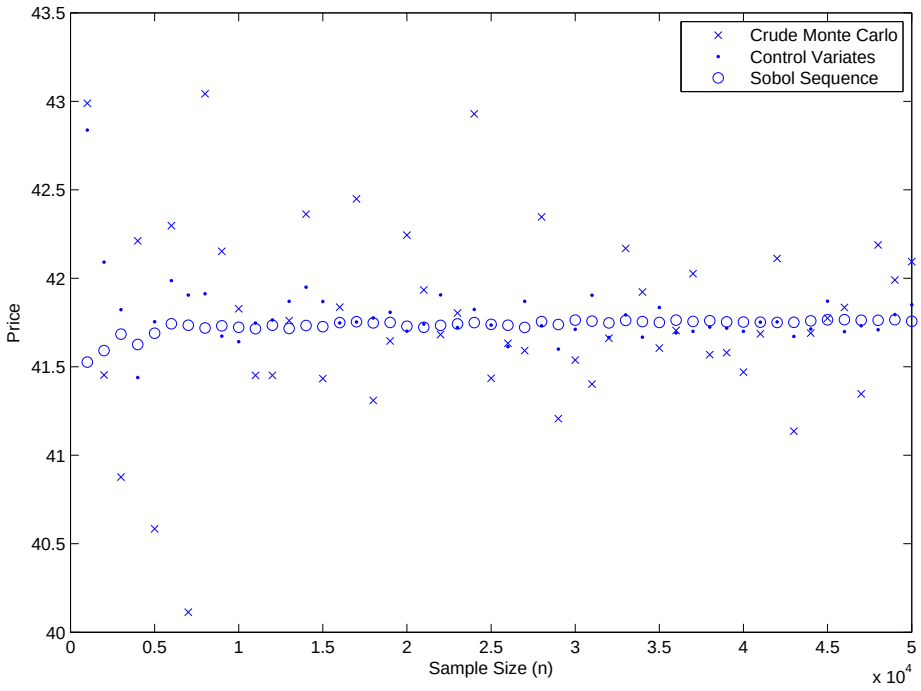


Figure 5.6. Comparison of Crude Monte Carlo, control variate and Sobol' prices for a European basket option versus sample size (for samples of size 1000, 2000, ..., 50 000).

You will note the quite rapid convergence of the price, with very little oscillation, even compared to the control variate price. The downside here

is that the generation of Sobol' sequences can be, computationally, quite expensive.

5.4. Closed Rules

5.4.1. Introduction. *Closed rules* generate quasi-random sequences which cannot be extended. The number of points to be generated by such a rule is a required parameter — if a larger set is required, it must be generated from scratch. In this section, we will examine *Hammersley sets*.

5.4.2. Hammersley Sets. Each Halton sequence leads to a family of *Hammersley sets*, indexed by sample size n. For the Halton sequence in $[0,1)^{s-1}$ generated from the bases $r_1, \ldots, r_{s-1}$, the Hammersley set in $[0,1)^s$ with n elements is given by

$$H_n := \left\{ \left(\frac{i}{n}, \varphi_{r_1}(i), \ldots, \varphi_{r_{s-1}}(i) \right) \middle| \, 0 \le i \le n-1 \right\},$$

Note that $H_n \not\subseteq H_{n+1}$ — i.e., Hammersley sequences are closed! Figure 5.7 displays the two-dimensional Hammersley set H_{1000}, with $r = 3$.

It can be shown for H_n that

$$D_n^* = \left(\prod_{j=1}^{s-1} \frac{r_j - 1}{2 \log r_j} \right) \frac{(\log n)^{s-1}}{n} + \frac{(\log n)^{s-2}}{n},$$

implying that

$$D_n^* = O\left(\frac{(\log n)^{s-1}}{n} \right).$$

It is conjectured that this is the best asymptotic error rate for a sequence of finite sets. (Note that it is also better than the rate for infinite Halton sequences.)

Of course, to increase the accuracy of an integral approximation requires function evaluations at every point in a larger Hammersley set. This set must be generated from scratch. By contrast, for integral estimates based on a Halton or Sobol sequence, the results of previous computations can be used.

5.5. Using Bridging to Improve Quasi-Monte Carlo Estimates

It is often the case that, as a function of the number of dimensions, there is a degradation of uniformity (increase in discrepancy) of the higher dimension quasi-Monte Carlo sub-sequences. For example, this is directly evident for the Halton sequence which is composed of van der Corput sequences of increasing radix r which must be relatively prime (recall that it is standard to use the first s primes, where s is the dimension required). But, the

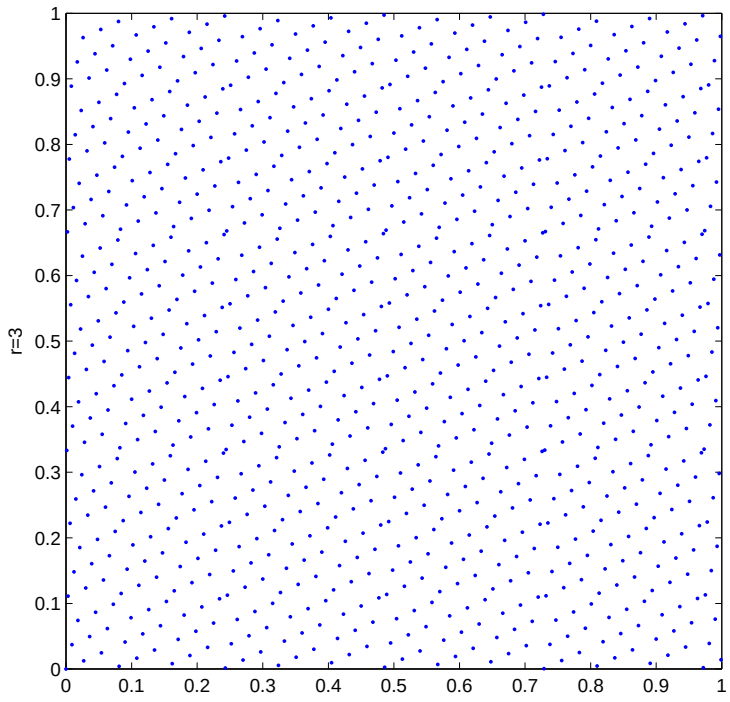


Figure 5.7. The two-dimensional Hammersley set H_{1000}, with $r = 3$.

discrepancy of the van der Corput sequence is a monotonically increasing function of r. This issue also applies to Sobol' and Hammersley sequences.

Thus, it has been observed that when applying these quasi-Monte Carlo sequences to path dependent options, improvement in convergence can be obtained by very carefully considering how to apply the more accurate subsequences (the lower dimensions) to the most important elements used for generating the stock paths.

As we saw in the section on end-point stratification, conditioning on terminal value can lead to decreases in variability of the pay-off. The same is true when applying quasi-Monte Carlo techniques. We now consider using the Hammersley sequence in conjunction with Brownian bridges to compute the cliquet option.

Example 5.8. (Hammersley Sets and Brownian Bridges) Consider the cliquet option of Example 4.5. We generate stock price paths using the standard (increments) formula for GBM and inverse normal transformed values of the Hammersley set[4] $H = \{(i/(n+1), \varphi_2(i), \varphi_3(i), \varphi_5(i)) | 1 \leq i \leq n\}$.

[4]The first element of the set (0) has been excluded.

For comparison, we generate Brownian bridge stock price paths using inverse normal transformed values of the permuted Hammersley set $H' = \{(\varphi_3(i), \varphi_2(i), \varphi_5(i), i/(n+1)) | 1 \leq i \leq n\}$, where the last sequence in the set is used to generate the endpoints of the Brownian motions. Notice how, the dimension with best discrepancy has been assigned to the terminal value, while the next best dimension has been assigned to the mid point between initial and final times, and the higher discrepancy subsequences have been assigned to the other intermediate updates.

Figure 5.8 shows a comparison of the quasi-Monte Carlo estimates for the two sets of stock prices as a function of sample size, and demonstrates the increased accuracy that results from the use of bridging.

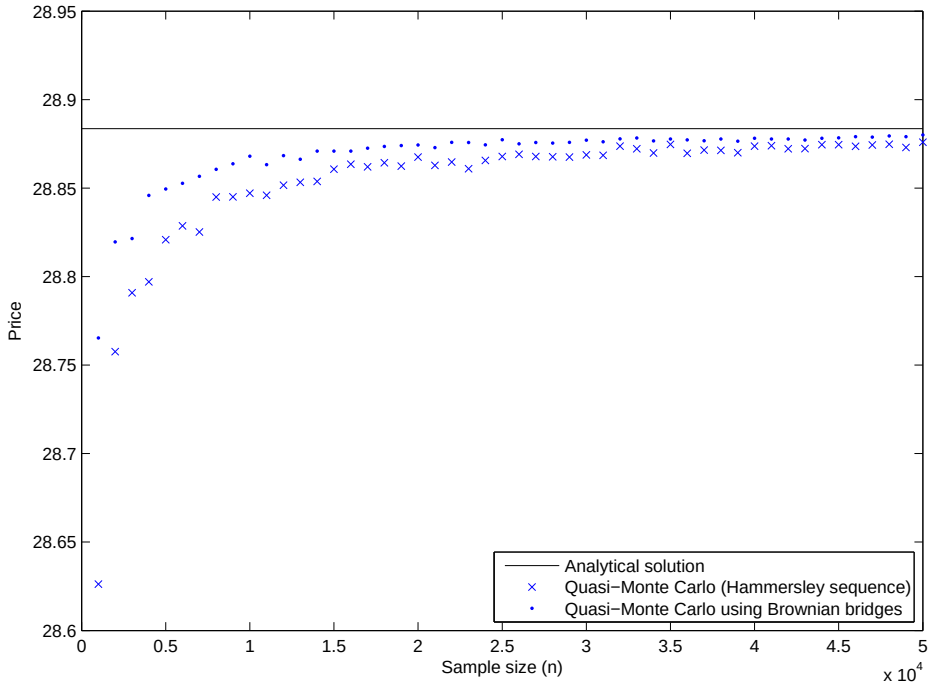


Figure 5.8. Comparison of Hammersley quasi-Monte Carlo estimates with and without bridging as a function of sample size. For comparison, the analytical price of the option is also provided.

Bibliography

[1] Bachelier, L. *Theory of Speculation*. PhD thesis, Faculty of Sciences of the Academy of Paris, 1900.

[2] Box, G., and Muller, M. A note on the generation of random normal deviates. *Annals of Mathematical Statistics 29* (1958), 610–611.

[3] Bratley, P., and Bennet, L. Algorithm 659: Implementing Solbol's quasirandom sequence generator. *ACM Transactions on Mathematical Software 14* (1988), 88–100.

[4] Durret, R. *Probability: Theory and Examples*, second ed. Duxbury Press, 1996.

[5] Glasserman, P. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

[6] Golub, G., and Van Loan, C. *Matrix Computations*, third ed. Johns Hopkins University Press, 1996.

[7] Higham, D. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review 20* (1979), 381–408.

[8] Johnson, R., and Wichern, D. *Applied Multivariate Statistical Analysis*, fourth ed. Prentace-Hall, 1998.

[9] Kloeden, P., and Platen, E. *Numerical Solution of Stochastic Differential Equations*. No. 23 in Applications of Mathematics. Springer-Verlag, 1992.

[10] Knuth, D. *The Art of Computer Programming*, third ed., vol. 2 — Seminumerical Algorithms. Addison-Wesley, 1998.

[11] Kuipers, L., and Niederreiter. *Uniform Distribution of Sequences*. Wiley Interscience, 1974.

[12] Levene, H., and Wolfowitz, J. The convergence of matrix runs up and down. *Annals of Mathematical Statistics 15* (1944), 58–69.

[13] Mood, A., Graybill, F., and Boes, D. *Introduction to the Theory of Statistics*, third ed. McGraw-Hill, 1974.

[14] Moro, B. The full Monte. *Risk Magazine 8*, 2 (February 1995), 57–58.

[15] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. *Numerical Recipes in C*, second ed. Cambridge University Press, 1992.

[16] Samuelson, P. Rational theory of warrent pricing. *Industrial Management Review 6* (1965), 13–39.

[17] Weyl, H. Uber die gleichverteilung von zahlen mod. eins. *Math. Ann. 77* (1916), 313–352.

[18] Wolfowitz, J. Asymptotic distribution of runs up and runs down. *Annals of Mathematical Statistics 15* (1944), 163–172.