CHAPTER 19

Monte Carlo Valuation

So far we have primarily discussed derivatives for which there is a (relatively simple) valuation formula, or which can be valued binomially. For many common derivatives, however, a different approach is necessary. For example, consider arithmetic Asian options (see Section 14.2). There is no simple valuation formula for such options, and the binomial pricing approach is difficult because the final payoff depends on the specific path the stock price takes through the tree—i.e., the payoff is path-dependent. A pricing method that can be used in such cases is **Monte Carlo valuation.** In Monte Carlo valuation we simulate future stock prices and then use these simulated prices to compute the discounted expected payoff of the option. The idea that an option price is a discounted expected value is familiar from our discussion of the binomial model in Chapter 11 and the Black-Scholes formula in Chapter 18.

Monte Carlo valuation is performed using the risk-neutral distribution, where we assume that assets earn the risk-free rate on average, and we then discount the expected payoff using the risk-free rate. We will see in this chapter that risk-neutral pricing is a cornerstone of Monte Carlo valuation; using the actual distribution instead would create a complicated discounting problem.

Since with Monte Carlo you simulate the possible future values of the security, as a byproduct you generate the *distribution* of payoffs. The distribution can be extremely useful when you want to compare two investment strategies that have different distributions of outcomes. Computing value-at-risk for complicated portfolios is a common use of Monte Carlo.

In this chapter we will see why risk-neutral valuation is important for Monte Carlo, see how to produce normal random numbers, discuss the efficiency of Monte Carlo, introduce the Poisson distribution to help account for nonlognormal patterns in the data, and see how to create correlated random stock prices.

19.1 COMPUTING THE OPTION PRICE AS A DISCOUNTED EXPECTED VALUE

The concept of risk-neutral valuation is familiar from earlier Chapters 15. We saw that option valuation can be performed *as if* all assets earned the risk-free rate of return and investors performed all discounting at this rate. Monte Carlo valuation exploits this insight. We *assume* that assets earn the risk-free rate of return and simulate their returns.

For example, for any given stock price 3 months from now, we can compute the payoff on a call. We perform the simulation many times and average the outcomes. Since we use risk-neutral valuation, we then discount the average payoff at the risk-free rate in order to arrive at the price.

As a practical matter, Monte Carlo valuation depends critically on risk-neutral valuation. In order to see why this is so, we will compute an option price as an expected value with both risk-neutral and true probabilities, using an example we discussed in Chapters 10 and 11.

Valuation with Risk-Neutral Probabilities

We saw in equation (10.6) that we can interpret the one-period binomial option pricing calculation as an expected value, in which the expectation is computed using the risk-neutral probability p^*, and discounting is at the risk-free rate.

In a multiperiod tree, we repeat this process at each node. For a European option, the result obtained by working backward through the tree is equivalent to computing the expected option price in the final period, and discounting at the risk-free rate.

If there are n binomial periods, equation (11.17) gives the probability of reaching any given stock price at expiration. Let n represent the number of binomial steps and i the number of stock price down moves. We can value a European call option by computing the expected option payoff at the final node of the binomial tree and then discounting at the risk-free rate. For example, for a European call,

European call price =

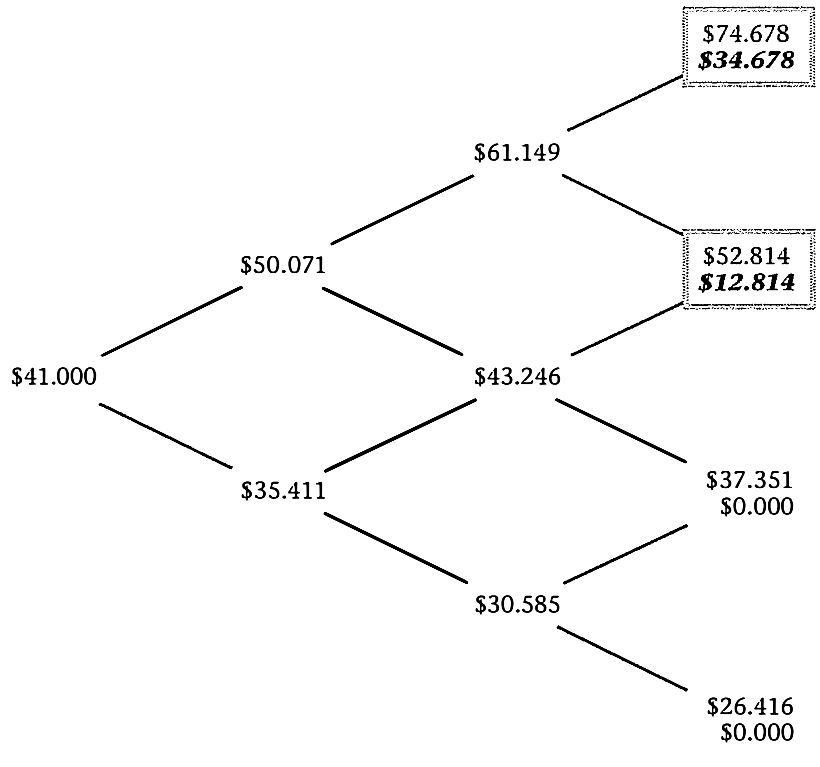
$$e^{-rT}\sum_{i=1}^{n}\max[0, Su^{n-i}d^{i} - K](p^*)^{n-i}(1-p^*)^{i}\frac{n!}{(n-i)!\,i!} \tag{19.1}$$

To illustrate this calculation, Figure 19.1 shows the stock price tree from Figure 10.5, with the addition of the total risk-neutral probabilities of reaching each of the terminal nodes. Figure 19.1 demonstrates that the option can be priced by computing the expected payoff at expiration using the probability of reaching each final node, and then discounting at the risk-free rate. You can verify that the option price in Figure 19.1 is the same as that in Figure 10.5.

We can also use the tree in Figure 19.1 to illustrate Monte Carlo valuation. Imagine a gambling wheel divided into four unequal sections, where each section has a probability corresponding to one of the option payoffs in Figure 19.1: 9.5% (\$34.678), 34% (\$12.814), 40.4% (0), and 16% (0). Each spin of the wheel therefore selects one of the final stock price nodes and option payoffs in Figure 19.1. If we spin the wheel numerous times and then average the resulting option values, we will have an estimate of the expected payoff. Discounting this expected payoff at the risk-free rate provides an estimate of the option value.

It is easy to compute the actual expected payoff for the option in Figure 19.1 without using a gambling wheel. However, the example illustrates how random trials can be used to perform valuation.

FIGURE 19.1



Price and Payoff	Probability	Prob. × Payoff
$74.678 ***$34.678***	p^{*3} = 0.095	$3.306
$52.814 ***$12.814***	$3p^{*2}(1-p^*)$ = 0.340	$4.357
$37.351 $0.000	$3p^*(1-p^*)^2$ = 0.404	$0.000
$26.416 $0.000	$(1-p^*)^3$ = 0.160	$0.000
Total	1.000	$7.663
$e^{-0.08}$ × **Total**		$7.074

Binomial tree (the same as in Figure 10.5) showing stock price paths, along with risk-neutral probabilities of reaching the various terminal prices. Assumes $S = \$41.00$, $K = \$40.00$, $\sigma = 0.30$, $r = 0.08$, $t = 1.00$ years, $\delta = 0.00$, and $h = 0.333$. The risk-neutral probability of going up is $p^* = 0.4568$. At the final node the stock price and terminal option payoff (beneath the price) are given.

Valuation with True Probabilities

The simple procedure we used to discount payoffs for the risk-neutral tree in Figure 19.1 *does not work* when we use actual probabilities. We analyzed the pricing of this option using true probabilities in Chapter 11, in Figure 11.4. We saw there that when using true probabilities to evaluate the option, the discount rate is different at different nodes on the tree. In fact, if we are to compute an option price as an expected value using true probabilities, we need to compute the discount rate *for each path*. There are eight possible paths for the stock price, four of which result in a positive option payoff. All of these paths have a first-period annualized continuously compounded discount rate of 35.7%. The subsequent discount rates depend on the path the stock takes. Table 19.1 verifies that discounting payoffs at path-dependent discount rates gives the correct option price. To take just the first row, the discounted expected option payoff for that row is computed as follows:

$$e^{-(0.357\times\frac{1}{3}+0.323\times\frac{1}{3}+0.269\times\frac{1}{3})} \times (0.5246)^3 \times (\$74.678 - \$40) = \$3.649$$

This calculation uses the fact that the actual probability that the stock price will move up in any period is 52.46%.

As Table 19.1 illustrates, it is necessary to have a different cumulative discount rate along each *path* the stock can take. A call option is a high-beta security when it

TABLE 19.1 Computation of option price using expected value calculation and true probabilities. The stock price tree and parameters are the same as in Figure 11.4. The column entitled "Discount Rates Along Path" reports the node-specific true annualized continuously compounded discount rates from that figure. "Discount Rate for Path" is the compound annualized discount rate for the entire path. "Prob. of Path" is the probability that the particular path will occur, computed using the true probability of an up move (52.46%). The last column is the probability times the payoff, discounted at the continuously compounded rate for the path.

Path	Discount Rates Along Path			Discount Rate for Path	Prob. of Path	Payoff ($)	Discounted ($) (Prob. x Payoff)
uuu	35.7%	32.3%	26.9%	31.64%	0.1444	34.678	3.649
uud	35.7%	32.3%	26.9%	31.64%	0.1308	12.814	1.222
udu	35.7%	32.3%	49.5%	39.18%	0.1308	12.814	1.133
duu	35.7%	49.5%	49.5%	44.91%	0.1308	12.814	1.070
udd	—	—	—	—	—	0	0
dud	—	—	—	—	—	0	0
ddu	—	—	—	—	—	0	0
ddd	—	—	—	—	—	0	0
					Sum		7.074

is out-of-the-money and it has a lower beta (but still higher than the stock) when it is in-the-money. This variation in the discount rate complicates discounting if we are using the true distribution of stock prices.[1]

Risk-neutral valuation neatly sidesteps the hardest problem about using discounted cash flow valuation techniques with an option. While it is easy to compute the expected payoff of an option if the stock is lognormally distributed, it is hard to compute the discount rate. If we value options *as if* the world were risk-neutral, this complication is avoided.

19.2 COMPUTING RANDOM NUMBERS

In this section we discuss how to compute the normally distributed random numbers required for Monte Carlo valuation. We will take for granted that you can compute a uniformly distributed random number between 0 and 1. The uniform distribution is defined on a specified range, over which the probability is 1, and assigns equal probabilities to every interval of equal length on that range. A random variable, u, that is uniformly distributed on the interval (a, b), has the distribution $\mathcal{U}(a, b)$. The uniform probability density, $f(x; a, b)$, is defined as

$$f(x; a, b) = \frac{1}{b-a}; a \leq x \leq b \tag{19.2}$$

and is 0 otherwise. When $a = 0$ and $b = 1$, the uniform distribution is a flat line at a height of 1 over the range 0 to 1.

Drawing uniformly distributed random variables is very common; virtually all programming languages and spreadsheets have a way to do this.[2] The *Rand* built-in function in Excel does this, for example. It turns out that once we have a way to compute uniformly distributed random variables, there are two common ways to compute a normally distributed random variable. Many programs also have functions to compute normal random numbers directly, in which case it is not necessary to use these methods. However, the second method we will discuss can be used to compute random numbers drawn from *any* distribution.

[1]Here is why a single discount rate does not work. Suppose we represent the terminal option price associated with a particular pattern of stock price up and down movements by $C_i(T)$ and the compound discount factor for that path by β_i. Since both the payoff and the discount rates are uncertain, we need to compute $E[C_i(T)/(1+\beta_i)]$. However, if we average the payoff and then separately average the discount factors, we are computing the ratio of the averages, $E[C_i(T)]/E[(1+\beta_i)]$, rather than the average of the ratios. Jensen's inequality tells us that these are not the same calculation.

[2]Since computers are ultimately deterministic devices, it is virtually impossible to compute "true" random numbers. See Judd (1998, pp. 285–287) for a discussion and additional references.

Using Sums of Uniformly Distributed Random Variables

One standard technique to compute normally distributed random variables is to sum 12 uniform (0,1) random variables and subtract 6. Thus, we compute the $\mathcal{N}(0, 1)$ random variable $\tilde{Z}$ as

$$\tilde{Z} = \sum_{i=1}^{12} u_i - 6$$

where the u_i are distributed uniformly on (0,1).

This technique works because the variance of a variable that is uniformly distributed between 0 and 1 is 1/12 and the mean is 1/2. Thus, if you sum 12 uniformly distributed random variables and subtract 6, you get a random variable with a variance of 1 and a mean of 0. The sum of 12 uniform variables is not precisely normal, but it is close. This technique is an application of the central limit theorem.

Using the Inverse Cumulative Normal Distribution

It is also possible to draw a *single* uniformly distributed random number and convert it to a normally distributed random number. Suppose that $u \sim \mathcal{U}(0, 1)$ and $z \sim \mathcal{N}(0, 1)$. As we saw in Chapter 18, the *cumulative distribution function*, denoted $U(w)$ for the uniform and $N(y)$ for the normal, is the probability that $u < w$ or $z < y$, i.e.,

$$U(w) = \text{Prob}(u \leq w)$$
$$N(y) = \text{Prob}(z \leq y)$$

As discussed in Chapter 18, w is the $U(w)$ quantile and y is the $N(y)$ quantile of the two distributions. If we randomly draw a uniform number u, how can we use u to construct a corresponding normal random number, z?

It turns out that the same idea we used to construct normal plots in Section 18.6 permits us to generate a normal random number from a uniform random number. Instead of interpreting a random draw from the uniform distribution as a *number*, we interpret it as a *quantile*. So, for example, if we draw 0.7 from a $\mathcal{U}(0, 1)$ distribution, we interpret this as a draw corresponding to the 70% quantile. We then use the inverse distribution function, $N^{-1}(u)$, to find the value from the normal distribution corresponding to that quantile.[3] This technique works because, for any distribution, quantiles are uniformly

[3]The Excel function *NormSInv* computes the inverse cumulative normal distribution. Unfortunately, there is a serious bug in this function in Office 97 and Office 2000. In both versions of Excel, *NormSInv*(0.9999996) = 5.066, and *NormSInv*(0.9999997) = 5,000,000. Because of this, Excel will on occasion produce a randomly drawn normal value of 5,000,000, which ruins a Monte Carlo valuation. I thank Mark Broadie for pointing out this problem with using Excel to produce random normal numbers.

distributed: If you draw from a distribution, by definition any quantile is equally likely to be drawn.

The algorithm is therefore as follows:

1. Generate a uniformly distributed random number between 0 and 1. Say this is 0.7.
2. Ask: What is the value of z such that $N(z) = 0.7$? The answer to this question is computed using the *inverse cumulative distribution function*. In this case we have $N^{-1}(0.7) = 0.5244$. This value is a single draw of a standard normal random variable (0.5244).
3. Repeat.

This procedure simulates draws from a normal distribution. To simulate a log-normal random variable, simulate a normal random variable and exponentiate the draws.

This procedure of using the inverse cumulative probability distribution is valuable because it works for any distribution for which you can compute the inverse cumulative distribution.

19.3 SIMULATING LOGNORMAL STOCK PRICES

Recall from Chapter 18 that if $Z \sim \mathcal{N}(0, 1)$, a lognormal stock price can be written

$$S_T = S_0 e^{(\alpha-\delta-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z} \tag{19.3}$$

Suppose we wish to draw random stock prices for 2 years from today. From equation (19.3), the stock price is driven by the normally distributed random variable Z. Set $T = 2$, $\alpha = 0.10$, $\delta = 0$, and $\sigma = 0.30$. If we then randomly draw a set of standard normal Z's and substitute the results into equation (19.3), the result is a random set of lognormally distributed S_2's. The continuously compounded mean return will be 20% (10% per year) and the continuously compounded standard deviation of $\ln(S_2)$ will be $0.3 \times \sqrt{2} = 42.43\%$.

Simulating a Sequence of Stock Prices

There is another way to create a random set of prices 2 years from now. We can also generate *annual* random prices and compound these to get a 2-year price. This will give us exactly the same distribution for 2-year prices. Here is how to do it:

- Compute the 1-year price, S_1 as

$$S_1 = S_0 e^{(0.1-\frac{1}{2}0.3^2)\times 1+\sigma\sqrt{1}Z(1)}$$

- Using this S_1 as the starting price, compute S_2:

$$S_2 = S_1 e^{(0.1-\frac{1}{2}0.3^2)\times 1+0.3\sqrt{1}Z(2)}$$

In these expressions, $Z(1)$ and $Z(2)$ are two draws from the standard normal distribution.

If we substititute the expression for S_1 into S_2, we get

$$S_2 = S_0 e^{(0.1-\frac{1}{2}0.3^2)\times 2+0.3\sqrt{1}[Z(1)+Z(2)]} \tag{19.4}$$

The difference between this expression and equation (19.3) is that instead of the term $\sqrt{2}Z$, we have $[Z(1) + Z(2)]$. Note that

$$\mathrm{Var}(\sqrt{2}Z) = 2$$

and

$$\mathrm{Var}[Z(1) + Z(2)] = 2$$

Therefore, equations (19.3) and (19.4) generate S_2's with the same distribution.

If we really want to simulate a random stock price after 2 years, there is no reason to draw two random variables instead of one. But if we want to simulate the *path* of the stock price over 2 years (for example, to price a path-dependent option), then we can do so by splitting up the 2 years into multiple periods.

In general, if we wish to split up a period of length T into intervals of length h, the number of such intervals will be $n = T/h$. We have

$$S_h = S_0 e^{(\alpha-\delta-\frac{1}{2}\sigma^2)h+\sigma\sqrt{h}Z(1)}$$
$$S_{2h} = S_h e^{(\alpha-\delta-\frac{1}{2}\sigma^2)h+\sigma\sqrt{h}Z(2)}$$

and so on, up to

$$S_{nh} = S_{(n-1)h} e^{(\alpha-\delta-\frac{1}{2}\sigma^2)h+\sigma\sqrt{h}Z(n)}$$

These n stock prices can be intepreted as equally spaced points on the stock price path between times 0 and T. Note that if we substitute S_h into the expression for S_{2h}, the expression for S_{2h} into that for S_{3h}, and so on, we get

$$\begin{aligned} S_T &= S_0 e^{(\alpha-\delta-\frac{1}{2}\sigma^2)T+\sigma\sqrt{h}[\sum_{i=1}^{n} Z(i)]} \\ &= S_0 e^{(\alpha-\delta-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}\left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n} Z(i)\right]} \end{aligned} \tag{19.5}$$

Since $\frac{1}{\sqrt{n}}\sum_{i=1}^{n} Z(i) \sim \mathcal{N}(0, 1)$, we get the same distribution at time T with equation (19.5) as if we had drawn a single $\mathcal{N}(0, 1)$ random variable, as in equation (19.3). The important difference is that by splitting up the problem into n draws, we simulate the path taken by S. The simulation of a path is useful in computing the value of path-dependent derivatives, such as Asian and barrier options, the value of which depend on the path by which the price arrives at S_T.

19.4 MONTE CARLO VALUATION

In Monte Carlo valuation, we perform a calculation similar to that in equation (19.1). The option payoff at time T is a function of the stock price, S_T. Represent this payoff as $V(S_T, T)$. The time-0 Monte Carlo price, $V(S_0, 0)$, is then

$$\boxed{V(S_0, 0) = \frac{1}{n} e^{-rT} \sum_{i=1}^{n} V(S_T^i, T)} \tag{19.6}$$

where $S_T^1, \ldots, S_T^n$ are n randomly drawn time-T stock prices. For the case of a call option, for example, $V(S_T^i, T) = \max(0, S_T^i - K)$.

Both equations (19.1) and (19.6) use approximations to the time-T stock price distribution to compute an option price. Equation (19.1) uses the binomial distribution to approximate the lognormal stock price distribution, while equation (19.6) uses simulated lognormal prices to approximate the lognormal stock price distribution.

As an illustration of Monte Carlo techniques, we will first work with a problem for which we already know the answer. Suppose we have a European option that expires in T periods. The underlying asset has volatility σ and the risk-free rate is r. We can use the Black-Scholes option pricing formula to price the option, but we will price the option using *both* Black-Scholes and Monte Carlo so that we can assess the performance of Monte Carlo valuation.

Monte Carlo Valuation of a European Call

We assume that the stock price follows equation (19.3), with $\alpha = r$. We generate random standard normal variables, Z, substitute them into equation (19.3), and generate many random future stock prices. Each Z creates one trial. Suppose we compute N trials. For each trial, i, we compute the value of a call as

$$\max(0, S_T^i - K) = \max\left(0, S_0 e^{(r-\delta-0.5\sigma^2)T+\sigma\sqrt{T}Z_i} - K\right); \qquad i = 1, \ldots, N$$

Average the resulting values:

$$\frac{1}{N}\sum_{i=1}^{N} \max(0, S_T^i - K)$$

This expression gives us an estimate of the expected option payoff at time T. We discount the average payoff back at the risk-free rate in order to get an estimate of the option value:

$$\overline{C} = e^{-rT}\frac{1}{N}\sum_{i=1}^{N} \max(0, S_T^i - K)$$

Example 19.1 Suppose we wish to value a 3-month European call option where the stock price is \$40, the strike price is \$40, the risk-free rate is 8%, the dividend yield is zero, and the volatility is 30%. We draw random 3-month stock prices by using the expression

$$S_{3\text{ months}} = S_0 e^{(0.08-0.3^2/2)\times 0.25+0.3\sqrt{0.25}Z}$$

TABLE 19.2 Results of Monte Carlo valuation of European call with $S = \$40$, $K = \$40$, $\sigma = 30\%$, $r = 8\%$, $t = 91$ days, and $\delta = 0$. The Black-Scholes price is \$2.78. Each trial uses 500 random draws.

Trial	Computed Price (\$)
1	2.98
2	2.75
3	2.63
4	2.75
5	2.91
Average	2.804

For each stock price, we compute

$$\text{Option payoff} = \max(0, S_{3\text{ months}} - \$40)$$

We repeat this procedure many times, average the resulting option payoffs, and discount the average back 3 months at the risk-free rate. With a single estimate using 2500 draws, we get an answer of \$2.804 (see Table 19.2), close to the true value of \$2.78.

In this example we priced a European-style option. We will discuss in Section 19.6 the use of Monte Carlo simulation to value American-style options.

Accuracy of Monte Carlo

There is no need to value a European call using Monte Carlo methods, but doing so allows us to assess the accuracy of Monte Carlo valuation for a given number of simulated stock price paths. The key question is how many simulated stock prices suffice to value an option to a desired degree of accuracy. Monte Carlo valuation is simple but relatively inefficient. There are methods that improve the efficiency of Monte Carlo; we discuss several of these in Section 19.5.

To assess the accuracy of a Monte Carlo estimate, we can run the simulation different times and see how much variability there is in the results. Of course in this case, we also know that the Black-Scholes solution is \$2.78.

Table 19.2 shows the results from running five Monte Carlo valuations, each containing 500 random stock price draws. The result of 2500 simulations is close to the correct answer (\$2.804 is close to \$2.78). However, there is considerable variation among the individual trials of 500 simulations.

To assess accuracy, we need to know the standard deviation of the estimate. Let $C(\tilde{S}_i)$ be the call price generated from the randomly drawn $\tilde{S}_i$. If there are n trials, the Monte Carlo estimate is

$$\overline{C}_n = \frac{1}{n}\sum_{i=1}^{n} C(\tilde{S}_i)$$

Let σ_C denote the standard deviation of one draw and σ_n the standard deviation of n draws. The variance of a mean, given independent and identically distributed $\tilde{S}_i$'s, is

$$\sigma_n^2 = \frac{1}{n}\sigma_C^2$$

or

$$\sigma_n = \frac{1}{\sqrt{n}}\sigma_C$$

Thus, the standard deviation of the Monte Carlo estimate is inversely proportional to the square root of the number of draws.

In the Monte Carlo results reported in Table 19.2, the standard deviation of a draw is about \$4.05. (This value is computed by taking the standard deviation of the 2500 price estimates used to compute the average.) For 500 draws, the standard deviation is

$$\frac{\$4.05}{\sqrt{500}} = \$0.18$$

Given that the correct price is \$2.78, a \$0.18 standard deviation is a substantial percentage of the option price (6.5%). With 2500 observations, the standard deviation is cut to \$0.08, suggesting that the \$2.80 estimate from averaging the five answers was only accidentally close to the correct answer. In order to have a 1% (\$0.028) standard deviation, we would need to have 21,000 trials.

Arithmetic Asian Option

In the previous example of Monte Carlo valuation we valued an option that we already could value with the Black-Scholes formula. In practice, Monte Carlo valuation is useful under these conditions:

- Where the number of random elements in the option valuation problem is too great to permit direct numerical solution.
- Where underlying variables are distributed in such a way that direct solutions are difficult.
- Where options are path-dependent, i.e., the payoff at expiration depends upon the path of the underlying asset price.

For the case of a path-dependent option, the use of Monte Carlo estimation is straightforward. As discussed above, we can simulate the path of the stock as well as its terminal value. For example, consider the valuation of a security that at the end of 3 months makes a payment based on the arithmetic average of the stock price at the end of months 1, 2, and 3. As discussed in Chapter 14, this is an arithmetic average price Asian option: "Asian" because the payoff is based on an average, and "arithmetic average price" because the arithmetic average stock price replaces the actual stock price at expiration.

How will the value of an option on the average compare with an option that settles based on the actual expiration-day stock price? Intuitively, averaging should reduce the likelihood of large gains and losses. Any time the stock ends up high (in which case the call will have a high value at expiration), it will have traversed intermediate stock prices in the process of reaching a high value. The payoff to the Asian option will reflect these lower intermediate prices, and, hence, large payoffs will be much less likely.

We compute the 1-month, 2-month, and 3-month stock prices as follows:

$$S_1 = 40e^{(r-\delta-\sigma^2/2)T/3+\sigma\sqrt{T/3}Z(1)}$$

$$S_2 = S_1 e^{(r-\delta-\sigma^2/2)T/3+\sigma\sqrt{T/3}Z(2)}$$

$$S_3 = S_2 e^{(r-\delta-\sigma^2/2)T/3+\sigma\sqrt{T/3}Z(3)}$$

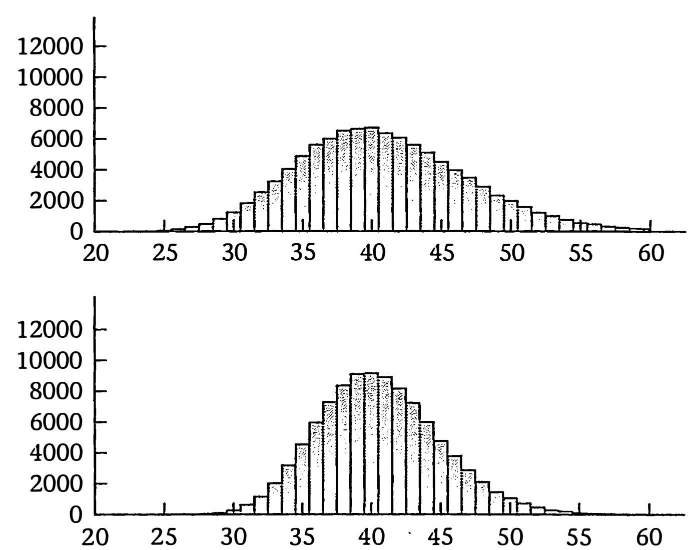
where $Z(1)$, $Z(2)$, and $Z(3)$ are independent draws from a standard normal distribution. We repeat the trial many times and draw many Z_i's. The value of the security is then computed as

$$C_{\text{Asian}} = e^{-rT} E\,(\max[(S_1 + S_2 + S_3)/3 - K, 0]) \tag{19.7}$$

Example 19.2 Let r = 8%, $\sigma = 0.3$, and suppose that the initial stock price is \$40. Figure 19.2 compares histograms for the actual risk-neutral stock price distri-

FIGURE 19.2

Histograms for risk-neutral stock price distribution after 3 months (top panel) and risk-neutral distribution for the average stock price created by averaging the three month-end stock prices during the same period (bottom panel). Assumes $S_0 = \$40$, $r = 8\%$, $\sigma = 30\%$, and $\delta = 0$. These histograms were generated using 100,000 trials.



bution after 3 months and that for the average stock price created by averaging the three month-end prices. As expected, the nonaveraged distribution has significantly higher tail probabilities and a lower probability of being close to the initial stock price of $40.

Table 19.3 lists prices of Asian options computed using 10,000 Monte Carlo trials each.[4] The first row (where a single terminal price is averaged) represents the price of an ordinary call option with 3 months to expiration. The others represent more frequent averaging. The Asian price declines as the averaging frequency increases, with the largest price decline obtained by moving from no averaging (the first row in Table 19.3) to monthly averaging (the second row of Table 19.3).

Note also in Table 19.3 that, in any row, the arithmetic average price is always above the geometric average price. This is Jensen's inequality at work: Geometric

TABLE 19.3 Prices of arithmetic average-price Asian options estimated using Monte Carlo and exact prices of geometric average price options. Assumes option has 3 months to expiration and average is computed using equal intervals over the period. Each price is computed using 10,000 trials, assuming $S = \$40$, $K = \$40$, $\sigma = 30\%$, $r = 8\%$, $T = 0.25$, and $\delta = 0$. In each row, the same random numbers were used to compute both the geometric and arithmetic average price options. σ_n is the standard deviation of the estimated prices, divided by $\sqrt{10{,}000}$.

Number of Averages	Monte Carlo Prices ($) Arithmetic	Monte Carlo Prices ($) Geometric	Exact Geometric Price ($)	σ_n
1	2.79	2.79	2.78	0.0408
3	2.03	1.99	1.94	0.0291
5	1.78	1.74	1.77	0.0259
10	1.70	1.66	1.65	0.0241
20	1.66	1.61	1.59	0.0231
40	1.63	1.58	1.56	0.0226

[4]A trial in this case means the computation of a single option price at expiration. When 40 prices are averaged over 3 months, each trial consists of drawing 40 random numbers; hence, 400,000 random numbers are drawn in order to compute the price.

averaging produces a lower average stock price than arithmetic averaging, and hence a lower option price.

19.5 EFFICIENT MONTE CARLO VALUATION

We have been describing what might be called "naive" Monte Carlo, making no attempt to reduce the variance of the simulated answer for a given number of trials. There are a number of methods to achieve faster Monte Carlo valuations.[5]

Control Variate Method

We have seen that naive Monte Carlo estimation of an arithmetic Asian option requires many simulations. In Table 19.3, even with 10,000 simulations, there is still a standard deviation of several percent in the option price.

In each row of Table 19.3, the same random numbers are used to estimate the option price. As a result, the errors in the estimated arithmetic and geometric prices are correlated: When the estimated price for the geometric option is high, this occurs because we have had high returns in the stock price simulation. This should result in a high arithmetic price as well.

This observation suggests the **control variate method** to increase Monte Carlo accuracy. The idea underlying this method is to estimate the error on each trial by using the price of a related option that does have a pricing formula. The error estimate obtained from this control price can be used to improve the accuracy of the Monte Carlo price on each trial.

Asian options provide an effective illustration of this idea.[6] Because we have a formula for the price of a geometric Asian option (see Section 14.2), we know whether the geometric price from a Monte Carlo valuation is too high or too low. For a given set of random stock prices, the arithmetic and geometric prices will typically be too high or too low in tandem, so we can use information on the error in the geometric price to adjust our estimate of the arithmetic price, for which there is no formula.

To be specific, we use simulation to estimate the arithmetic price, $\overline{A}$, and the geometric price, $\overline{G}$. Let G and A represent the true geometric and arithmetic prices. The error for the Monte Carlo estimate of the geometric price is $(G - \overline{G})$. We want to use this error to improve our estimate of the arithmetic price.

Consider calculating

$$A^* = \overline{A} + \left(G - \overline{G}\right) \tag{19.8}$$

[5]Excellent overviews are Boyle et al. (1997) and Glasserman (2004). See also Judd (1998, ch. 8), which in turn contains other references, and Campbell et al. (1997, ch. 9).

[6]This example follows Kemna and Vorst (1990), who used the control variate method to price arithmetic Asian options.

This is a control variate estimate. Since Monte Carlo provides an unbiased estimate, $E(\overline{G}) = G$. Hence, $E(A^*) = E(\overline{A}) = A$. Moreover, the variance of A^* is

$$\text{Var}(A^*) = \text{Var}(\overline{A}) + \text{Var}(\overline{G}) - 2\text{Cov}(\overline{A}, \overline{G}) \tag{19.9}$$

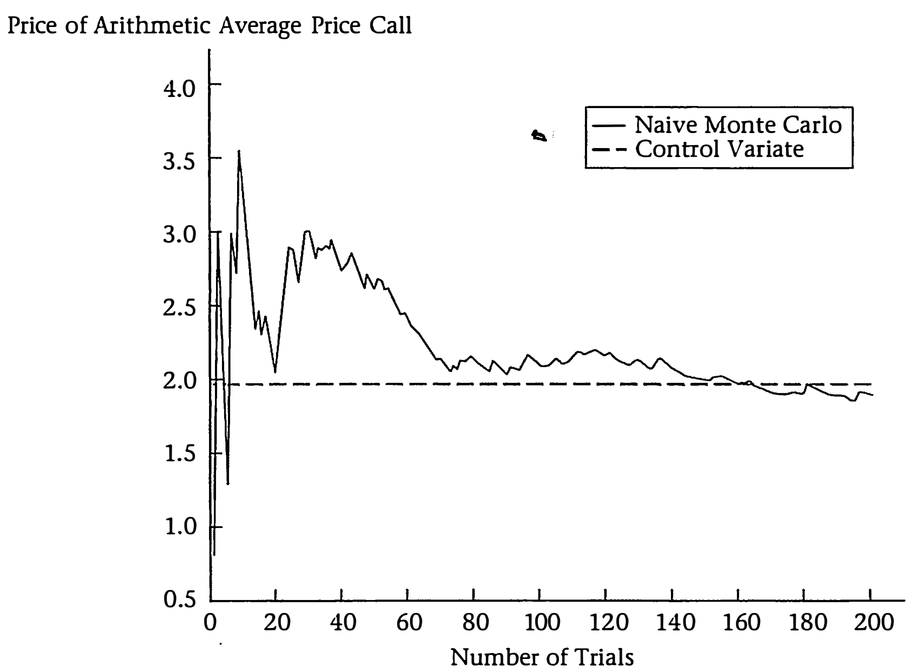
As long as the estimate $\overline{G}$ is highly correlated with the estimate $\overline{A}$, the variance of the estimate A^* can be less than the variance of $\overline{A}$.

In practice, the variance reduction from the control variate method can be dramatic. Figure 19.3 graphs the results from the first 200 simulations in pricing an arithmetic Asian option. The control variate estimate converges in just a few trials to the correct value of about \$1.98. For example, the very first draw in the graphed simulation gave an arithmetic option price of \$0.80 and a geometric price—using the same random prices—of \$0.75. The correct geometric price is \$1.94. Correcting the estimate gives a price of

$$\text{Control variate price} = \$0.80 + (\$1.94 - \$0.75) = \$1.99$$

This example illustrates that if the correlation between the two estimates is high, the control variate method works very well.

FIGURE 19.3



Comparison of "naive" Monte Carlo estimate of arithmetic average option price with control variate method. Graph depicts first 200 simulations for an option with $S = \$40$, $K = \$40$, $\sigma = 0.3$, $r = 0.08$, $T = 0.25$, $\delta = 0$, and the final price computed with three averages.

Boyle et al. (1997) point out that equation (19.8) does not in general provide the minimum variance Monte Carlo estimate, and in some cases can even increase the variance of the estimate. They suggest that instead of estimating equation (19.8), you estimate

$$A^* = \overline{A} + \beta\left(G - \overline{G}\right) \tag{19.10}$$

The variance of this estimate is

$$\text{Var}(A^*) = \text{Var}(\overline{A}) + \beta^2\text{Var}(\overline{G}) - 2\beta\text{Cov}(\overline{A}, \overline{G}) \tag{19.11}$$

The variance $\text{Var}(A^*)$ is minimized by setting $\beta = \text{Cov}(\overline{A}, \overline{G})/\text{Var}(\overline{G})$. One way to obtain β is to perform a small number of Monte Carlo trials, run a regression of equation (19.10) to obtain $\hat{\beta}$, and then use $\hat{\beta}$ for the remaining trials. The optimal value of β will vary depending on the application.

Other Monte Carlo Methods

The control variate example is just one method for improving the efficiency of Monte Carlo valuation. The **antithetic variate method** uses the insight that for every simulated realization, there is an opposite and equally likely realization. For example, if we draw a random normal number of 0.5, we could just as well have drawn −0.5. By using the opposite of each normal draw we can get two simulated outcomes for each random path we draw. This seems as if it would help, since it doubles the number of draws. But drawing a random number is often not the time-consuming part of a Monte Carlo calculation.

There can be an efficiency gain because the two estimates are negatively correlated; adding them reduces the variance of the estimate. In practical terms, this means that if you draw an extreme estimate from one tail of the distribution, you will also draw an extreme estimate from the other tail, balancing the effect of the first draw. Boyle et al. (1997) find modest benefits from using the antithetic variate method.

Another important class of methods controls the region in which random numbers are generated. **Stratified sampling** is an example of this kind of method. Suppose you have 100 uniform random numbers, u_i, $i = 1, \ldots, 100$. With naive Monte Carlo you would compute $z_i = N^{-1}(u_i)$. This calculation treats each random number as representing a random draw from the cumulative distribution. However, because of random variation, 100 uniform random numbers will not be exactly uniformly distributed and therefore the z_i will not be exactly normal. We can improve the distribution of the u_i, and therefore of the z_i, if we treat each number *as a random draw from each percentile of the uniform distribution*. Thus, take the first draw, u_1, and divide it by 100. The resulting $\hat{u}_1$ is now uniformly distribtuted over [0,0.01]. Take the second draw, divide it by 100, and add 0.01. The resulting $\hat{u}_2$ is uniformly distributed over (0.01,0.02). For the ith draw, compute $\hat{u}_i = (i - 1 + u_i)/100$. This value is uniformly distributed over the ith percentile. Proceeding in this way we are guaranteed to generate a random number for each percentile of the normal distribution. You can select a number of intervals different from 100, and you can repeat the simulation multiple times. A generalization of this technique when the payoff depends on more than one random variable is *Latin hypercube sampling,* discussed by Boyle et al. (1997).

There are other techniques for improving the efficiency of Monte Carlo. The approach called *importance sampling* concentrates the generation of random numbers where they have the most value for pricing a particular claim. *Low discrepancy sequences* use carefully selected deterministic points to create more uniform coverage of the distribution. Boyle et al. (1997) provide an excellent summary and comparison of the different methods.

If you are performing a one-time calculation, the simplicity of naive Monte Carlo is appealing. However, if you are performing a Monte Carlo valuation repeatedly, you may achieve large efficiency gains by analyzing the problem and using one or more variance reduction techniques to increase efficiency.

19.6 VALUATION OF AMERICAN OPTIONS

It is generally more difficult to value American-style options than to value European-style options, and this remains true when using Monte Carlo valuation. Standard Monte Carlo entails simulating stock price paths *forward*, then averaging and discounting the maturity payoffs. In American option valuation, the difficulty is knowing when to exercise the option; this requires working *backward* to determine the times at which the option should be exercised. Recently, Broadie and Glasserman (1997) and Longstaff and Schwartz (2001) have demonstrated feasible methods for using Monte Carlo to value American options.

We will discuss pricing a 3-year put option with a strike of $1.10, the example used in Longstaff and Schwartz (2001). In order to analyze early exercise we need to consider times before maturity, so we must simulate stock price *paths*. Figure 19.4, taken from Longstaff and Schwartz (2001), illustrates eight hypothetical simulation paths, with intermediate stock prices generated annually. The in-the-money nodes (those for which

FIGURE 19.4

Assumes $S_0 = 1.0$, $K = 1.1$, and $r = 6\%$. Prices in **bold** are nodes where early exercise might be optimal.

Stock Price Paths

Path	$t = 0$	$t = 1$	$t = 2$	$t = 3$
1	1.00	**1.09**	**1.08**	1.34
2	1.00	1.16	1.26	1.54
3	1.00	1.22	**1.07**	**1.03**
4	1.00	**0.93**	**0.97**	**0.92**
5	1.00	1.11	1.56	1.52
6	1.00	**0.76**	**0.77**	**0.90**
7	1.00	**0.92**	**0.84**	**1.01**
8	1.00	**0.88**	1.22	1.34

Source: Longstaff and Schwartz (2001).

$S < \$1.10$) are candidate nodes for exercise of the option; in Figure 19.4 they are in bold (this ignores exercise at time 0). How do we determine at which of these nodes early exercise is optimal?

The idea underlying any method of American option valuation is to compare the value of immediate exercise to the **continuation value** of the option—i.e., the value of keeping the option alive.[7] The problem is therefore to estimate the continuation value at each point in time.

It is worth noting one potential problem in estimating continuation value, which stems from the use of future stock prices on a given path to decide whether to exercise on that path. Consider path 1 in Figure 19.4. The option is out-of-the-money, and therefore worthless, at $t = 3$. Therefore, on this path we would be better off at $t = 2$ exercising rather than waiting. However, in deciding to exercise by looking ahead on the path, we are using knowledge of the future stock price, which is information we will not have in real life. Valuing the option assuming we know the future price will give us too high a value. The way to mitigate such "lookahead bias" is to base an exercise decision on *average* outcomes from a given point forward. There are at least two ways to characterize the average outcome from a given point. One is to use a regression to characterize the continuation value based on analysis of multiple paths. This is the method proposed by Longstaff and Schwartz (2001). Another is create additional branches from each node, providing multiple outcomes that we can average to characterize continuation value at that node. This is the basis for the Broadie and Glasserman (1997) procedure.

To price the option using regression analysis, we work backward through the columns of Figure 19.4, running a regression at each time to estimate continuation value as a function of the stock price. We work backward because the continuation value at $t = 1$ will depend upon whether exercise is optimal on a given path at $t = 2$. At $t = 2$, there are five paths (1, 3, 4, 6, and 7) where the option is in-the-money and exercise could be optimal. For each of these paths, we know the value of exercising immediately and the value of waiting. Longstaff and Schwartz run a regression of the present value of waiting to exercise (i.e., the continuation value) against the stock price and stock price squared. At time 2, we obtain the following result:

$$\text{Continuation value at time 2} = -1.07 + 2.98 \times S - 1.81 \times S^2$$

where S is the time 2 stock price. Now for each node where exercise could be optimal, we insert the stock price at that node into the regression equation and obtain an estimate of continuation value. By comparing this to intrinsic value, we decide whether to exercise at that node. For example, when $S = 1.08$ in row 1, the esimated value of waiting to exercise is

$$-1.07 + 2.98 \times 1.08 - 1.81 \times 1.166 = 0.037 \tag{19.12}$$

[7] For example, this is the comparison in the binomial valuation in equation (10.10).

Since the immediate exercise value is $1.10 - 1.08 = 0.02$, which is less the 0.037, we wait at that node. Table 19.4 summarizes the results.

We then repeat the analysis at $t = 1$, using the results at $t = 2$. The final decision about where to exercise the option is summarized in Figure 19.5. We can value the option by computing the present value of cash flows based on exercising at the nodes where doing so is optimal. The final American put value is \$0.1144, compared with \$0.0564 for a European value computed using the same simulated paths.

A problem with the regression approach is that it is not obvious how to select an appropriate functional form for the continuation regression. Longstaff and Schwartz (2001) report obtaining similar results for a variety of functional forms, but for each new problem it will be desirable to experiment with different functions.

Broadie and Glasserman (1997) adopt a different approach, pointing out that American option valuations are subject to different kinds of biases. As we discussed above, an estimator will give too high a valuation to the extent it uses information about the future to decide whether to exercise at a given time. Estimators will be biased low to the extent that early exercise is suboptimal (since optimal exercise maximizes the value

TABLE 19.4 Exercise analysis at $t = 2$ for those nodes in Figure 19.4 where $S < \$1.10$ at $t = 2$.

Path	PV(Wait)	S	S^2	Exercise	Continuation	Result
1	0.000	1.08	1.166	0.02	0.037	Wait
3	0.066	1.07	1.145	0.03	0.046	Wait
4	0.170	0.97	0.941	0.13	0.118	Exercise
6	0.188	0.77	0.593	0.33	0.152	Exercise
7	0.085	0.84	0.706	0.26	0.156	Exercise

FIGURE 19.5

Summary of results showing the nodes at which exercise is optimal (in **bold**) for the paths in Figure 19.4.

Stock Price Paths

Path	$t = 0$	$t = 1$	$t = 2$	$t = 3$
1	1.00	1.09	1.08	1.34
2	1.00	1.16	1.26	1.54
3	1.00	1.22	1.07	**1.03**
4	1.00	**0.93**	0.97	0.92
5	1.00	1.11	1.56	1.52
6	1.00	**0.76**	0.77	0.90
7	1.00	**0.92**	0.84	1.01
8	1.00	**0.88**	1.22	1.34

of the option). To address the two errors, Broadie and Glasserman use two estimators, one with high bias and one with low bias. In constructing these estimators, they create sample paths in which there are multiple branches from each node. The resulting set of paths resembles a nonrecombining binomial tree with more than two branches from each node.

The high bias estimator assesses the continuation value by averaging the discounted values on branches emanating from a point and exercising if the value of doing so is greater than the value of continuing. Because the subsequent branches are constructed by simulation, there will be sampling error. To see the effects of such error, suppose exercise is optimal at a node. If the subsequent branches are too high due to sampling error, we will not exercise and assign an even higher value to the node than would be obtained by (optimally) exercising. Now suppose that exercise is not optimal at a node but subsequent branches are too low due to sampling error. We will then exercise and again assign a higher value to the node than we should, given the subsequent branch values.[8]

The low bias estimator is obtained by splitting the branches from each node into two sets. Using the first set, we estimate the value of continuation and decide whether to exercise. If it is optimal to continue, we use the second set of nodes to estimate the continuation value. By using separate sets of nodes to make the exercise decision and to estimate continuation value, this estimator avoids the "high bias" discussed above. But to the extent the exercise decision is suboptimal, the inferred option value will be too low. Both estimators are biased, but both also converge to the true option value as the number of paths increases.

The Broadie and Glasserman approach is computationally involved, but provides a general method for accomodating early exercise in a simulation model.

19.7 THE POISSON DISTRIBUTION

We have seen that the lognormal distribution assigns a low probability to large stock price moves. One approach to generating a more realistic stock price distribution is to permit large stock price moves to occur randomly. Occasional large price moves can generate the fat tails observed in the data in Section 18.6.

The **Poisson distribution** is a discrete probability distribution that counts the number of events—such as large stock price moves—that occur over a period of time. The Poisson distribution is summarized by the parameter λ, where λh is the probability that one event occurs over the short interval h. A Poisson-distributed event is very

[8]Note that the other two kinds of sampling errors do not matter for assessing the value of early exercise. If it is not optimal to exercise and subsequent branches are too high, we will not exercise and therefore not erroneously attribute value to exercising. Similarly, if it is optimal to exercise and subsequent values are too low, we will exercise, giving the correct value to early exercise.

unlikely to occur more than once over a sufficiently short interval. Thus, λ is like an annualized probability of the event occurring over a short interval.[9]

Over a longer period of time, t, the probability that the event occurs exactly m times is given by

$$p(m, \lambda t) = \frac{e^{-\lambda t}(\lambda t)^m}{m!}$$

The cumulative Poisson distribution is then the probability that there are m or fewer events from 0 to t.[10]

$$\mathcal{P}(m, \lambda t) = \text{Prob}(x \leq m; \lambda t) = \sum_{i=0}^{m} \frac{e^{-\lambda t}(\lambda t)^i}{i!}$$

Given an expected number of events, the Poisson distribution tells us the probability that we will see a particular number of the events over a given period of time.[11] The mean of the Poisson distribution is λt.

Example 19.3 Suppose the probability of a market crash is $\lambda = 2\%$ per year. Then the probability of seeing no market crashes in any given year can be computed as $p(0, 0.02 \times 1) = 0.9802$. The probability of seeing no crashes over a 10-year period would be $p(0, 0.02 \times 10) = 0.8187$. The probability of seeing exactly two crashes over a 10-year period would be $p(2, 0.02 \times 10) = 0.0164$.

Figure 19.6 graphs the Poisson distribution for three values of the Poisson parameter, λt. Suppose we are interested in the number of times an event will occur over a

[9]By definition, the number of occurrences of an event is Poisson-distributed if four assumptions are satisfied:

1. The probability that one event will occur in a small interval h is proportional to the length of the interval.
2. The probability that more than one event will occur in a small interval h is substantially smaller than the probability that a single event will occur.
3. The number of events in nonoverlapping time intervals is independent.
4. The expected number of events between time t and time $t + s$ is independent of t.

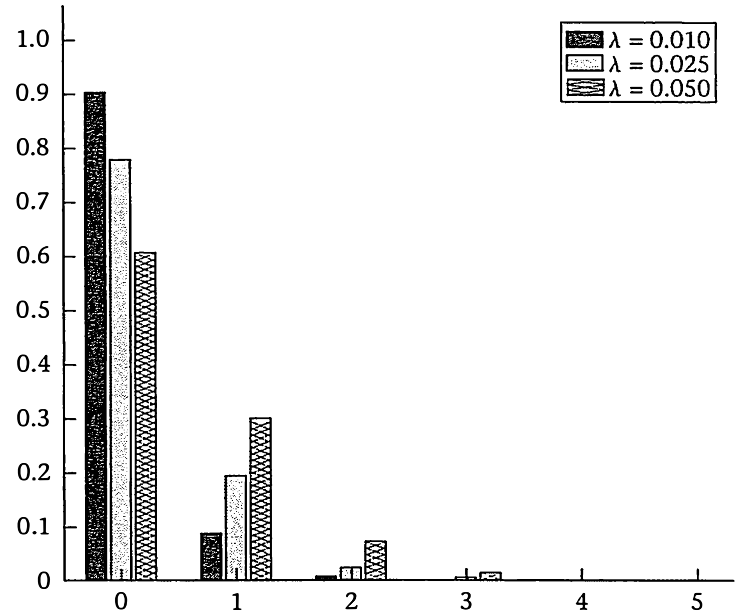
The Poisson distribution can be derived from these four assumptions. See Casella and Berger (2002).

[10]In Excel, you can compute $p(m, \lambda t)$ as Poisson(m, λt, *false*), and the cumulative distribution, $\mathcal{P}(m, \lambda t)$, as Poisson($m$, λt, *true*).

[11]The probability that no event occurs between time 0 and time t is $p(0, \lambda t) = e^{-\lambda t}$. The probability that one or more events occurs between 0 and t is therefore $1 - e^{-\lambda t}$. This expression is also the cumulative distribution of the **exponential distribution,** which models the time until the first event. The density function of the exponential distribution is $f(t, \lambda) = \lambda e^{-\lambda t}$.

FIGURE 19.6

Graph of Poisson distribution for λt of 0.010, 0.025, and 0.050. This graph may be interpreted as the distribution of the number of events observed over a 10-year period, given annual event probabilities of 1%, 2.5%, and 5%. In any of the cases, there is a tiny probability of seeing more than five events over the 10-year period.



10-year period. Figure 19.6 shows us the distribution for $t = 10$ and $\lambda = 0.01$ (1% per year), $\lambda = 0.025$ (2.5% per year), and $\lambda = 0.05$ (5% per year). The likeliest occurrence in all three scenarios is that no events occur. It is also extremely unlikely that four or more events occur.

The Poisson distribution only counts the number of events. If an event occurs, we need to determine the magnitude of the jump as an independent draw from some other density; the lognormal is frequently used. Thus, in those periods when a Poisson event occurs, we would draw a separate random variable to determine the magnitude of the jump.

Using the inverse cumulative distribution function for a Poisson random variable, it is easy to generate a Poisson-distributed random variable. Even without the inverse cumulative distribution function (which Excel does not provide), we can construct the inverse distribution function from the cumulative distribution function.

Table 19.5 calculates the Poisson distribution for a mean of 0.8. Using this table we can easily see how to randomly draw a Poisson event. First we draw a uniform (0,1) random variable. Then we use the values in the table to decide how many events occur. If the uniform random variable is less than 0.4493, for example, we say that no events occur. If the value is between 0.4493 and 0.8088, we say that one event occurs, and so forth.

TABLE 19.5 Values of Poisson distribution and cumulative Poisson distribution with mean $(\lambda t) = 0.8$.

Number of Events	Probability	Cumulative Probability
0	0.4493	0.4493
1	0.3595	0.8088
2	0.1438	0.9526
3	0.0383	0.9909

19.8 SIMULATING JUMPS WITH THE POISSON DISTRIBUTION

As we discussed, stock prices sometimes move more than would be expected from a lognormal distribution. If market volatility is 20% and the expected return is 15%, a one-day 5% drop in the market occurs about once every 2.5 million days. (See Problem 19.8.) A 20% one-day drop (as in October 1987) is virtually impossible if prices are lognormally distributed with a reasonable volatility.

Merton (1976) introduced the use of the Poisson distribution in an option pricing context. The Poisson distribution counts the number of events that occur in a given period of time. If each event is a jump in the price, we can then use the lognormal (or other) distribution to compute the size of the jump. This Poisson-lognormal model assumes that jumps are independent. In addition to independence, we will assume that jumps are idiosyncratic, meaning that jumps can be diversified. In this case, the possibility of a jump does not affect the risk premium of the asset. (This is a common assumption made for tractability, but it is not always appropriate. While some jumps are idiosyncratic, a large market move is by definition systematic.)

Let the lognormally distributed jump magnitude Y be given by

$$Y = e^{\alpha_J - 0.5\sigma_J^2 + \sigma_J W}$$

where W is a standard normal variable. If S is the pre-jump price, YS is the post-jump price. Using the calculations in Chapter 18, e^{α_J} is the expected jump and σ_J is the standard deviation of the log of the jump. The expected percentage jump is

$$E\left(\frac{YS - S}{S}\right) = e^{\alpha_J} - 1 = k \tag{19.13}$$

Simulating the Stock Price with Jumps

To simulate the stock price over a period of time h, we first pick two uniform random variables to determine the number of jumps and the ordinary (nonjump) lognormal return.

If there are m jumps, we must then pick m additional random variables to determine the magnitudes of the jumps. Each jump has a multiplicative effect on the stock price.

Specifically, suppose the stock price is S_t. If a stock cannot jump, its price at time $t + h$ is

$$S_{t+h} = S_t e^{(\alpha-\delta-0.5\sigma^2)h+\sigma\sqrt{h}Z}$$

where α is the expected return.

Now consider an otherwise identical stock that can jump, with price $\hat{S}_t$. The stock price will have two components, one with and one without jumps. The no-jump lognormal component is

$$S_t e^{(\hat{\alpha}-\delta-0.5\sigma^2)h+\sigma\sqrt{h}Z}$$

where the expected stock return, conditional on no jump, is $\hat{\alpha}$. We will see in a moment why we use a different notation for the expected return in this expression. If the stock jumps m times between t and $t + h$, each jump changes the price by a factor of

$$Y_i = e^{\alpha_J-0.5\sigma_J^2+\sigma_J W(i)}$$

Where Z and $W(i)$, $i = 1, \ldots, m$ are standard normal random variables. The cumulative jump is the product of the Y_i's, or

$$\prod_{i=1}^{m} Y_i = e^{m(\alpha_J-0.5\sigma_J^2)+\sigma_J \Sigma_{i=1}^{m} W(i)}$$

Notice that the cumulative jump is lognormal, since it is the product of lognormal random variables. The stock price at time $t + h$, taking account of both the normal lognormal return and jumps, is then

$$\hat{S}_{t+h} = \hat{S}_t e^{(\hat{\alpha}-\delta-0.5\sigma^2)h+\sigma\sqrt{h}Z} \times e^{m(\alpha_J-0.5\sigma_J^2)+\sigma_J \Sigma_{i=1}^{m} W(i)} \tag{19.14}$$

It is possible to simulate $\hat{S}_{t+h}$ using this expression. There are three steps:

1. Select a standard normal Z.
2. Select m from the Poisson distribution.
3. Select m draws, $W(i)$, $i = 1, \ldots, m$, from the standard normal distribution.

By inserting these values into equation (19.14), we generate $\hat{S}_{t+h}$, which is lognormal since it is a product of lognormal expressions.

We have not answered the question: What is $\hat{\alpha}$? There is a subtlety associated with modeling jumps. When a jump occurs, the expected percentage change in the stock price is $e^{\alpha_J} - 1$. If $\alpha_J \neq 0$, jumps will induce average up or down movement in the stock, depending upon whether $\alpha_J > 0$ or $\alpha_J < 0$. Recall, however, that we assumed jumps are idiosyncratic. Therefore, *the unconditional (meaning that we do not know whether jumps will occur) expected return for a stock that does not jump should be the same as the unconditional expected return for an otherwise identical stock that does jump.* When jumps have no systematic risk, the jump does not affect the stock's expected return. However, we have to adjust the nonjump expected return, $\hat{\alpha}$, in order for jumps not to affect the expected return. For example, if the average jump return is −10%, then over time the stock price will drift down on average due to jumps. In equilibrium, the stock must appreciate when not jumping in order to give the owner a fair return unconditionally. If $\alpha_J = -10\%$, we would need to raise the average expected return on the stock in order for it to earn a fair rate of return on average.

We adjust for α_J by subtracting λk from the no-jump expected return, where λ is the Poisson parameter and k is given by equation (19.13). Thus,

$$\hat{\alpha} = \alpha - \lambda k \tag{19.15}$$

With this correction, if the expected jump is positive, we lower the expected return on the stock when it is not jumping, and vice versa for a negative expected jump.

The final expression for the stock price is thus

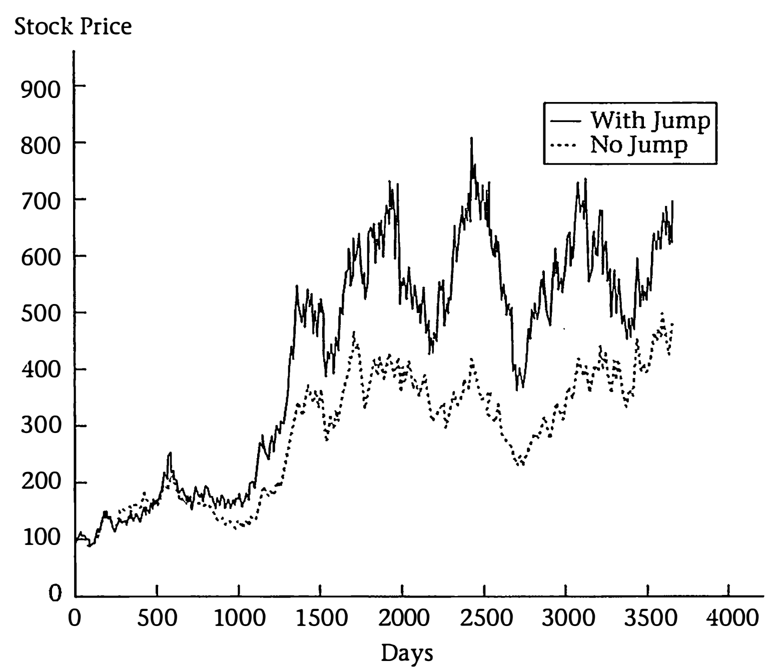
$$\begin{aligned}\hat{S}_{t+h} &= \hat{S}_t e^{(\alpha-\delta-\lambda k-0.5\sigma^2)h+\sigma\sqrt{h}Z} \prod_{i=0}^{m} e^{\alpha_J-0.5\sigma_J^2+\sigma_J W_i} \\ &= \hat{S}_t e^{(\alpha-\delta-\lambda k-0.5\sigma^2)h+\sigma\sqrt{h}Z} e^{m(\alpha_J-0.5\sigma_J^2)+\sigma_J\Sigma_{i=0}^{m}W_i}\end{aligned} \tag{19.16}$$

where α_J and σ_J are the mean and standard deviation of the jump magnitude, Z and W_i are random standard normal variables and m is Poisson-distributed. A similar expression appears in Merton (1976).

Figure 19.7 displays two simulated stock price series, one for which jumps do not occur, and one generated using equation (19.16). In the absence of jumps, the stock

FIGURE 19.7

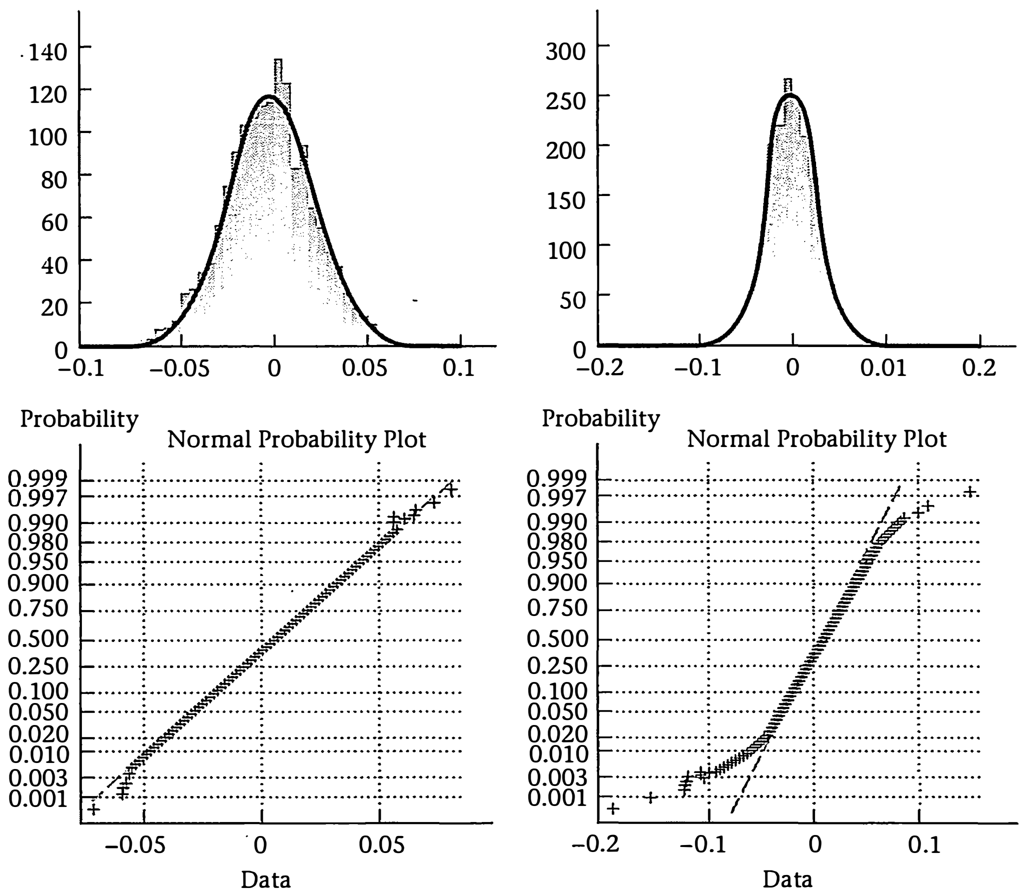
Simulated stock price paths over 10 years (3650 days). One stock cannot jump; the other is the same except that jumps can occur. The simulation assumes that $\alpha = 8\%$, $\delta = 0$, $\sigma = 30\%$, $\lambda = 3$, $\alpha_J = -2\%$, and $\sigma_J = 5\%$.



price is assumed to follow a lognormal process with $\alpha = 8\%$ and $\sigma = 30\%$. For the jump component, we assume $\lambda = 3$ (an average of three jumps per year), $\alpha_J = -2\%$, and $\sigma_J = 5\%$. In the figure, we can detect jumps because the no-jump series is drawn using the same random Z's. Some of the disparity, for example between days 1000 and 1500, is due to the approximate extra 6% return (λk) that is added to the stock when it does not jump.

What happens if we apply the normality tests from Chapter 18 to the stock price series in Figure 19.7? Figure 19.8 displays histograms and normal probability plots for the two series. Without jumps, continuously compounded returns look normal. With jumps, the data look nonnormal and resemble Figures 18.4 and 18.5. The jump results

FIGURE 19.8



Histograms and normal probability plots for the daily returns generated from the two series in Figure 19.7. Graphs on the left are for the no-jump series.

in data that do not look normal. The kurtosis of the continuously compounded returns without jumps is 2.93, very close to the value of 3 expected for a normal distribution. With jumps, kurtosis is 7.40.

Multiple Jumps

When we assume lognormal moves of the stock conditional on a single jump event, we can only get large up *and* down moves by assuming a large standard deviation of the jump move. The reason is that we are drawing from a single lognormal distribution, conditional on the Poisson event. An alternative is to assume there are *two* (or more) Poisson variables, one controlling up jumps and one controlling down jumps. The lognormal moves associated with each can have different means and standard deviations. This obviously provides for a richer and potentially more realistic set of outcomes.

19.9 SIMULATING CORRELATED STOCK PRICES

Suppose that S and Q are both lognormally distributed stock prices such that

$$\ln(S_t) = \ln(S_0) + (\alpha_S - 0.5\sigma_S^2)t + \sigma_S\sqrt{t}W$$
$$\ln(Q_t) = \ln(Q_0) + (\alpha_Q - 0.5\sigma_Q^2)t + \sigma_Q\sqrt{t}Z$$

If S and Q are uncorrelated, then we can simulate both prices by drawing independent W and Z. However, suppose that the correlation between S and Q is ρ. Here is how to simulate these two random variables taking account of their correlation.

Let ϵ_1 and ϵ_2 be independent and distributed as $\mathcal{N}(0, 1)$. Let

$$\begin{aligned} W &= \epsilon_1 \\ Z &= \rho\epsilon_1 + \epsilon_2\sqrt{1-\rho^2} \end{aligned} \tag{19.17}$$

Then $\text{Corr}(Z, W) = \rho$, and Z is distributed $\mathcal{N}(0, 1)$.

To see this, note first that Z and W both have zero mean. Compute the covariance between Z and W and the variance of Z:

$$E(WZ) = E[\epsilon_1(\rho\epsilon_1 + \epsilon_2\sqrt{1-\rho^2})] = \rho E(\epsilon_1^2) = \rho$$
$$E(Z^2) = E[(\rho\epsilon_1 + \epsilon_2\sqrt{1-\rho^2})^2] = \rho^2 + 1 - \rho^2 = 1$$

Thus, W and Z are both $\mathcal{N}(0, 1)$ and have a correlation coefficient of ρ.

Now we will check that the continuously compounded returns of S and Q have correlation ρ. The covariance between $\ln(S_t)$ and $\ln(Q_t)$ is

$$\begin{aligned} E\left[(\ln(S_t) - E[\ln(S_t)])(\ln(Q_t) - E[\ln(Q_t)])\right] &= E\left(\sigma_S W\sqrt{t}\sigma_Q Z\sqrt{t}\right) \\ &= \sigma_S\sigma_Q\rho t \end{aligned}$$

The correlation coefficient is

$$\text{Correlation} = \frac{\sigma_S \sigma_Q \rho t}{\sigma_S \sqrt{t} \sigma_Q \sqrt{t}} = \rho$$

Thus, if W and Z have correlation ρ, so will the continuously compounded returns of S and Q.

Generating n Correlated Lognormal Random Variables

Suppose we have n correlated lognormal variables. The question we address here is how to generalize the previous analysis. The first of the n random variables will have $n-1$ pairwise correlations with the others. The second will have $n-2$ (not counting its correlation with the first, which we have already counted). Continuing in this way, we will have

$$n - 1 + n - 2 + \cdots + 1 = \frac{1}{2}n(n-1)$$

pairwise correlations we have to take into account. We will denote the correlation between variables i and j as $\rho_{i,j}$.

We denote the original uncorrelated random $\mathcal{N}(0, 1)$ variables as $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$. The correlated random variables are $Z(1), Z(2), \ldots, Z(n)$, with

$$E[Z(i)Z(j)] = \rho_{i,j}$$

We can generate the $Z(i)$ as

$$Z(i) = \sum_{j=1}^{i} a_{i,j}\epsilon_j$$

where the $a_{i,j}$ are coefficients selected to make sure the pairwise correlations are correct.

Creating the coefficients $a_{i,j}$ has a recursive solution. That is, we construct $Z(1)$, then $Z(2)$ using the solution to $Z(1)$, and so on. The formula for the $a_{i,j}$ is

$$\begin{aligned} a_{i,j} &= \frac{1}{a_{j,j}}\left[\rho_{i,j} - \sum_{k=1}^{j-1} a_{j,k}a_{i,k}\right] \quad i > j \\ a_{i,i} &= \sqrt{1 - \sum_{k=1}^{i-1} a_{i,k}^2} \end{aligned} \tag{19.18}$$

For the case of two random variables, this reduces to equation (19.17).

The matrix of $a_{i,j}$'s is called the **Cholesky decomposition** of the original correlation matrix. In order for equation (19.18) to give correct coefficients, the set of correlations must be **positive-definite,** which means that the correlations must be such that there is no way to sum random variables and compute a negative variance. This is *not* an arbitrary condition: If this condition is not satisfied, the set of correlations is not valid.[12]

The point—and the reason for mentioning this—is that correlations and covariances cannot be arbitrary. In practice, depending upon how a covariance matrix is estimated, this can be an important concern. The *true* covariances among hundreds of bonds, stocks, currencies, and commodities *must* create a positive-definite covariance matrix. However, *estimated* covariances might not be positive-definite. If there are m assets and $n > m$ observations, the covariance matrix estimated from these data will be positive-definite. However, if different covariances are estimated from different data sets, positive-definiteness is not assured. The results of a simulation based on such covariances may produce nonsensical results.

CHAPTER SUMMARY

Monte Carlo methods entail simulating asset returns in order to obtain a future distribution for prices. This distribution can then be used to price claims on the asset (for example, Asian options) or to assess the risk of the asset (we will focus on such uses in Chapter 25). Performing simulations requires that we draw random numbers from an appropriate distribution (for example, the normal) in order to generate future asset prices. There are adjustments, such as the control variate method, which can dramatically increase the speed with which Monte Carlo estimates converge to the correct price.

It is possible to incorporate jumps in the price by mixing Poisson and lognormal random variables. Simulated correlated random variables can be created using the Cholesky decomposition.

FURTHER READING

The first use of Monte Carlo methods to price options was Boyle (1977) and the technique is now quite widespread. An excellent survey of the use of Monte Carlo in valuing derivatives is Boyle et al. (1997). We will see how Monte Carlo is used in value-at-risk calculations in Chapter 25. Bodie and Crane (1999) use Monte Carlo to analyze

[12]Suppose there are three random variables, a, b, and c, each with a variance of 1. Suppose that a is perfectly correlated with b ($\rho_{a,b} = 1$) and b is perfectly correlated with c ($\rho_{b,c} = 1$). It must then be the case that c is perfectly correlated with a. If $\rho_{a,c} \neq 1$, the matrix of correlations is not positive-definite.

To see this, suppose that $\rho_{a,c} = 0$, then compute $\text{Var}(a - b + c)$. You will find that the variance is -1, which is impossible. To take a different example, suppose $\rho_{a,c} = 0.9$. You will then find that $\text{Var}(a - 2b + c) = -0.2$, which is again impossible.

If a matrix of correlations is not positive-definite, it means that there is some combination of the random variables for which you will compute a negative variance. (For many combinations the variance will still be positive.) The interpretation of a negative variance is that you had an invalid correlation matrix to start with.

retirement investment products. Schwartz and Moon (2000) use Monte Carlo to value a firm by simulating future cash flows.

The papers by Broadie and Glasserman (1997) and Longstaff and Schwartz (2001), which present techniques for using Monte Carlo to value American-style options, have clear discussions of their respective methodologies.

Merton (1976) derived an option pricing formula in the presence of idiosyncratic jumps. Naik and Lee (1990) illustrate option pricing in the presence of systematic jumps. Risk aversion affects the option price in such cases.

PROBLEMS

19.1. Let $u_i \sim \mathcal{U}(0, 1)$. Draw 1000 random u_i and construct a histogram of the results. What are the mean and standard deviation?

19.2. Let $u_i \sim \mathcal{U}(0, 1)$. Compute $\sum_{i=1}^{12} u_i - 6$, 1000 times. (This will use 12,000 random numbers.) Construct a histogram and compare it to a theoretical standard normal density. What are the mean and standard deviation?

19.3. Suppose that $x_1 \sim \mathcal{N}(0, 1)$ and $x_2 \sim \mathcal{N}(0.7, 3)$. Compute 2000 random draws of e^{x_1} and e^{x_2}.

a. What are the means of e^{x_1} and e^{x_2}? Why?

b. Create a graph that displays a frequency distribution in each case

19.4. The Black-Scholes price for a European put option with $S = \$40$, $K = \$40$, $\sigma = 0.30$, $r = 0.08$, $\delta = 0$, and $t = 0.25$ is \$1.99. Use Monte Carlo to compute this price. Compute the standard deviation of your estimates. How many trials do you need to achieve a standard deviation of \$0.01 for your estimates?

19.5. Let $r = 0.08$, $S = \$100$, $\delta = 0$, and $\sigma = 0.30$. Using the risk-neutral distribution, simulate $1/S_1$. What is $E(1/S_1)$? What is the forward price for a contract paying $1/S_1$?

19.6. Suppose $S_0 = 100$, $r = 0.06$, $\sigma_S = 0.4$ and $\delta = 0$. Use Monte Carlo to compute prices for claims that pay the following:

a. S_1^2

b. $\sqrt{S_1}$

c. S_1^{-2}

19.7. Suppose that $\ln(S)$ and $\ln(Q)$ have correlation $\rho = -0.3$ and that $S_0 = \$100$, $Q_0 = \$100$, $r = 0.06$, $\sigma_S = 0.4$ and $\sigma_Q = 0.2$. Neither stock pays dividends. Use Monte Carlo to find the price today of claims that pay

a. $S_1 Q_1$

b. S_1/Q_1

c. $\sqrt{S_1 Q_1}$

d. $1/(S_1 Q_1)$

e. $S_1^2 Q_1$

19.8. Assume that the market index is 100. Show that if the expected return on the market is 15%, the dividend yield is zero, and volatility is 20%, the probability of the index falling below 95 over a 1-day horizon is approximately 0.0000004.

19.9. Suppose that on any given day the annualized continuously compounded stock return has a volatility of either 15%, with a probability of 80%, or 30%, with a probability of 20%. This is a **mixture of normals** model. Simulate the daily stock return and construct a histogram and normal plot. What happens to the normal plot as you vary the probability of the high volatility distribution?

19.10. For stocks 1 and 2, $S_1 = \$40$, $S_2 = \$100$, and the return correlation is 0.45. Let $r = 0.08$, $\sigma_1 = 0.30$, $\sigma_2 = 0.50$, and $\delta_1 = \delta_2 = 0$. Generate 1000 1-month prices for the two stocks. For each stock, compute the mean and standard deviation of the continuously compounded return. Also compute the return correlation.

19.11. Assume $S_0 = \$100$, $r = 0.05$, $\sigma = 0.25$, $\delta = 0$, and $T = 1$. Use Monte Carlo valuation to compute the price of a claim that pays \$1 if $S_T > \$100$, and 0 otherwise. (This is called a *cash-or-nothing call* and will be further discussed in Chapter 22. The actual price of this claim is \$0.5040.)

a. Running 1000 simulations, what is the estimated price of the contract? How close is it to \$0.5040?

b. What is the standard deviation of your Monte Carlo estimate? What is the 95% confidence interval for your estimate?

c. Use a 1-year at-the-money call as a control variate and compute a price using equation (19.8).

d. Again use a 1-year at-the-money call as a control variate, only this time use equation (19.10). What is the standard deviation of your estimate?

For the following three problems, assume that $S_0 = \$100$, $r = 0.08$, $\alpha = 0.20$, $\sigma = 0.30$, and $\delta = 0$. Perform 2000 simulations. Note that most spreadsheets have built-in functions to compute skewness and kurtosis. (In Excel, the functions are *Skew* and *Kurt*.) For the normal distribution, skewness, which measures asymmetry, is zero. Kurtosis, discussed in Chapter 18, equals 3.

19.12. Let $h = 1/52$. Simulate both the continuously compounded actual return and the actual stock price, S_{t+h}. What are the mean, standard deviation, skewness, and kurtosis of both the continuously compounded return on the stock and the stock price? Use the same random normal numbers and repeat for $h = 1$. Do any of your answers change? Why?

19.13. An options trader purchases 1000 1-year at-the-money calls on a nondividend-paying stock with $S_0 = \$100$, $\alpha = 0.20$, and $\sigma = 0.25$. Assume the options are priced according to the Black-Scholes formula and $r = 0.05$.

a. Use Monte Carlo (with 1000 simulations) to estimate the expected return, standard deviation, skewness, and kurtosis of the return on the call when it is held until expiration. Interpret your answers.

b. Repeat for an at-the-money put.

19.14. Repeat the previous problem, only assume that the options trader purchases 1000 1-year at-the-money *straddles*.

19.15. Refer to Table 19.1.

a. Verify the regression coefficients in equation (19.12).

b. Perform the analysis for $t = 1$, verifying that exercise is optimal on paths 4, 6, 7, and 8, and not on path 1.

19.16. Refer to Figure 19.2.

a. Verify that the price of a European put option is \$0.0564.

b. Verify that the price of an American put option is \$0.1144. Be sure to allow for the possibility of exercise at time 0.

19.17. Assume $S_0 = \$50$, $r = 0.05$, $\sigma = 0.50$, and $\delta = 0$. The Black-Scholes price for a 2-year at-the-money put is \$10.906. Suppose that the stock price is lognormal but can also jump, with the number of jumps Poisson-distributed. Assume $\alpha = 0.05$ (the expected return to the stock is equal to the risk-free rate), $\sigma = 0.50$, $\lambda = 2$, $\alpha_J = -0.04$, $\sigma_J = 0.08$.

a. Using 2000 simulations incorporating jumps, simulate the 2-year price and draw a histogram of continuously compounded returns.

b. Using Monte Carlo incorporating jumps, value a 2-year at-the-money put. Is this value significantly different from the Black-Scholes value?

APPENDIX 19.A: FORMULAS FOR GEOMETRIC AVERAGE OPTIONS

Appendix available online at www.aw-bc.com/mcdonald.