## 18.2 Monte Carlo Simulation

Monte Carlo simulation is a sampling scheme. In many important applications within finance and without, Monte Carlo simulation is one of the few viable tools. It is also one of the most important elements of studying econometrics [573]. When the time evolution of a stochastic process is not easy to describe analytically, Monte Carlo simulation may very well be the only strategy that succeeds consistently [400].

Assume $X_1, X_2, \ldots, X_n$ have a joint distribution and $\theta \equiv E[\, g(X_1, X_2, \ldots, X_n)\,]$ for some function $g$ is desired. We generate

$$\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right), \quad 1 \le i \le N$$

independently with the same joint distribution as $(X_1, X_2, \ldots, X_n)$ and set

$$Y_i \equiv g\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right).$$

Now, $Y_1, Y_2, \ldots, Y_N$ are independent and identically distributed random variables, and each $Y_i$ has the same distribution as $Y \equiv g(X_1, X_2, \ldots, X_n)$. Since the average of these $N$ random variables, $\overline{Y}$, satisfies $E[\,\overline{Y}\,] = \theta$, it can be used to estimate $\theta$. In fact, the **strong law of large numbers** says that this procedure converges almost surely [724]. The number of **replications** (or *independent* trials), $N$, is called the **sample size**.

**Example 18.2.1** To evaluate the definite integral $\int_a^b g(x)\,dx$ numerically, consider the random variable $Y \equiv (b-a)\,g(X)$, where $X$ is uniformly distributed over $[\,a, b\,]$. Note that $\text{Prob}[\,X \le x\,] = (x-a)/(b-a)$ for $a \le x \le b$. Since

$$E[\,Y\,] = (b-a)\,E[\,g(X)\,] = (b-a)\int_a^b \frac{g(x)}{b-a}\,dx = \int_a^b g(x)\,dx,$$

any unbiased estimator of $E[\,Y\,]$ can be used to evaluate the integral. □

The Monte Carlo estimate and true value may differ owing to two reasons: sampling variation and the discreteness of the sample paths. The former can be controlled by the number of replications as we shall see shortly, and the latter can be controlled by the number of observations along the sample path [151].

The statistical error of the sample mean $\overline{Y}$ of the random variable $Y$ grows as $1/\sqrt{N}$ because $\text{Var}[\,\overline{Y}\,] = \text{Var}[\,Y\,]/N$. In fact, this convergence rate is asymptotically optimal by the **Berry-Esseen theorem** [429]. As a result, the variance of the estimator $\overline{Y}$ can be reduced by a factor of $1/N$ by doing $N$ times as much work [747]. This property is amazing because the same order of convergence holds independent of the dimension $n$. In contrast, classic numerical integration schemes have an error bound of $O(N^{-c/n})$ for some constant $c > 0$. The required number of evaluations thus grows exponentially in $n$ to achieve a given level of accuracy. This is the familiar curse of dimensionality. The Monte Carlo method, for example, is more efficient than alternative procedures for securities depending on more than one asset, the multivariate derivatives [553].

The statistical efficiency of Monte Carlo simulation can be measured by the variance of its output. If this variance can be lowered without changing the expected value, fewer replications are needed. Methods that improve efficiency in this manner are called **variance reduction techniques**. Such techniques, covered in §18.2.3, become practical when the added costs are outweighed by the reduction in sampling.

### 18.2.1 Monte Carlo option pricing

For the pricing of European options on a dividend-paying stock, we may proceed as follows. From (14.17), stock prices $S_1, S_2, S_3, \ldots$ at times $\Delta t, 2\Delta t, 3\Delta t, \ldots$ can be generated via

$$S_{i+1} = S_i e^{(\mu - \sigma^2/2)\,\Delta t + \sigma\sqrt{\Delta t}\,\xi}, \quad \xi \sim N(0,1)$$

when $dS/S = \mu\,dt + \sigma\,dW$. Non-dividend-paying stock prices in a risk-neutral economy can be generated by setting $\mu = r$. Figure 18.5 contains a pricing algorithm for arithmetic average-rate calls.

Monte Carlo method for pricing average-rate calls on a non-dividend-paying stock:

```
input:   S, X, n, r, σ, τ, m;
real     P, C, M;
real     ξ();   // ξ() ~ N(0,1).
integer  i, j;
C := 0; // Accumulated terminal option value.
for (i = 1 to m) { // Perform m replications.
        P := S; M := S;
        for (j = 1 to n) {
                P := P × e^((r − σ^2/2)(τ/n) + σ√(τ/n) ξ());
                M := M + P;
        }
        C := C + max(M/(n + 1) − X, 0);
}
return Ce^(−rτ)/m;
```

Figure 18.5: MONTE CARLO METHOD FOR AVERAGE-RATE CALLS. Here, $m$ is the number of replications, and $n$ is the number of periods.

The sample standard deviation of the estimation scheme in Fig. 18.5 is proportional to $1/\sqrt{m}$, where $m$ is the number of replications. To narrow down the confidence interval by a factor of $f$, $f^2$ times as many replications need to be carried out. Although we do not know how small $\Delta t \equiv \tau/n$ should be to yield acceptable approximations, it is not hard to figure out $m$. Since the estimate is composed of a simple average across replications, the central limit theorem says the error of the estimate is distributed as $N(0, s^2/m)$ with $s^2$ denoting the variance of each replication. Hence, confidence interval requirements can be used to derive the desired $m$.

The discreteness of sample paths and the variance in prices do not necessarily make Monte Carlo results inferior to closed-form solutions. The judgment ultimately depends on the security being priced. In reality, for instance, a case may be made that, since prices do not move continuously, discrete-time models are more appropriate.

Monte Carlo simulation is a general methodology. It can be used to value virtually any European-style derivative security [151]. Standard Monte Carlo simulation, however, is inappropriate for American options because of early exercise: It is difficult to determine the early exercise point based on one single path. Intriguingly, Tilley showed that Monte Carlo simulation can be modified to price American options [873]; the estimate is biased, however [110].

▷**Exercise 18.2.1** How to price European barrier options by Monte Carlo simulation?

▷**Exercise 18.2.2** Consider the Monte Carlo method that estimates the price of the American call by taking the maximum discounted intrinsic value per simulated path and then averaging them: $E[\max_{i=0,1,\cdots,n} e^{-ri\Delta t}\max(S_i - X, 0)]$. Show that it is biased high.

▷**Programming assignment 18.2.3** Implement the Monte Carlo method for arithmetic average-rate calls and puts.

### 18.2.2 Ito processes

Consider the stochastic differential equation $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$. While it is often difficult to give an analytic solution to the above equation, simulation of the process on a computer is relatively easy [609]. Recall that Euler's method picks a small number $\Delta t$ and then approximates the Ito process by

$$\widehat{X}(t_{n+1}) = \widehat{X}(t_n) + a(\widehat{X}(t_n))\Delta t + b(\widehat{X}(t_n))\sqrt{\Delta t}\,\xi,$$

where $\xi \sim N(0,1)$. See Fig. 18.6 for the algorithm. This simulation is exact for any $\Delta t$ if both the drift $a$ and diffusion $b$ are constants as in Brownian motion because the sum of independent normal distributions remains normal.

Monte Carlo simulation of Ito process:

```
input:   x_0, T, Δt;
real     X[0..⌈T/Δt⌉];
real     ξ();   // ξ() ~ N(0,1).
integer  i;
X[0] := x_0;    // Initial state.
for (i = 1 to ⌈T/Δt⌉)
         X[i] := X[i-1] + a(X[i-1]) Δt
                 +b(X[i-1])√Δt ξ();
return X[];
```

Figure 18.6: Monte Carlo simulation of the Ito process. The Ito process is $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$. A run of the algorithm generates an approximate sample path for the process.

▷**Exercise 18.2.4** The Monte Carlo method for the Ito process in Fig. 18.6 may not be the most ideal theoretically. Consider the geometric Brownian motion $dX/X = \mu\,dt + \sigma\,dW$. Assume you have access to a perfect random number generator for normal distribution. Find a theoretically better algorithm to generate sample paths for $X$.

▷**Programming assignment 18.2.5** Simulate $dX_t = (0.06 - X_t)\,dt + 0.3\,dW_t$ using $\Delta t \equiv 0.01$. Explain its dynamics.

#### Discrete approximations to Ito processes with Brownian bridge

Aside from the Euler method and the related approximation methods in §14.2.1, Brownian bridge is one more alternative. Let the time interval $[0, T]$ be partitioned at time points $t_0, t_1, t_2, \ldots$, where $t_0 = 0$. Instead of employing

$$W(t_n) = W(t_{n-1}) + \sqrt{t_n - t_{n-1}}\,\xi, \quad \xi \sim N(0,1)$$

to generate the discrete-time Wiener process, the new method uses

$$W(t_n) = \frac{t_{n+1} - t_n}{t_{n+1} - t_{n-1}}\,W(t_{n-1}) + \frac{t_n - t_{n-1}}{t_{n+1} - t_{n-1}}\,W(t_{n+1}) + \sqrt{\frac{(t_{n+1} - t_n)(t_n - t_{n-1})}{t_{n+1} - t_{n-1}}}\,\xi,$$

given a past value $W(t_{n-1})$ and a future value $W(t_{n+1})$. In general, the method determines a sample path $W(i(T/2^m))$, $i = 0, 1, \ldots, 2^m$, over $[0, T]$ as follows. First, set $W(0) = 0$ and $W(T) = \sqrt{T}\,\xi$. Then set the mid-point $W(T/2)$ according to the above equation. From here, we find the mid-points for $(W(0), W(T/2))$ and $(W(T/2), W(T))$, that is, $W(T/4)$ and $W(3T/4)$, respectively. Iterate for $m-2$ more times. This scheme increases the accuracy of quasi-Monte Carlo simulation to be introduced shortly by reducing its effective dimension [6, 146, 703, 908].

▷**Programming assignment 18.2.6** Implement the Brownian bridge approach to generate the sample path of geometric Brownian motion.

### 18.2.3 Variance reduction techniques

The success of variance reduction schemes depends critically on the particular problem of interest. Since it is usually impossible to know beforehand how great a reduction in variance may be realized, if at all, preliminary runs should be made to compare the results of a variance reduction scheme with those from standard Monte Carlo simulation.

#### Antithetic variates

Suppose we are interested in estimating $E[\,g(X_1, X_2, \ldots, X_n)\,]$, where $X_1, X_2, \ldots, X_n$ are independent random variables. Let $Y_1$ and $Y_2$ be random variables with the same distribution as $g(X_1, X_2, \ldots, X_n)$. Then

$$\mathrm{Var}\left[\frac{Y_1 + Y_2}{2}\right] = \frac{\mathrm{Var}[\,Y_1\,]}{2} + \frac{\mathrm{Cov}[\,Y_1, Y_2\,]}{2}.$$

Note that $\mathrm{Var}[\,Y_1\,]/2$ is the variance of the Monte Carlo method with two (independent) replications. The variance $\mathrm{Var}[\,(Y_1 + Y_2)/2\,]$ is smaller than $\mathrm{Var}[\,Y_1\,]/2$ when $Y_1$ and $Y_2$ are negatively correlated instead of being independent.

The **antithetic variates** technique is based on the above observation. First, simulate $X_1, X_2, \ldots, X_n$ via the **inverse transform technique**. That is, $X_i$ is generated by $F_i^{-1}(U_i)$, where $U_i$ is a random number uniformly distributed over $(0, 1)$ and $F_i$ is the distribution function of $X_i$. Set

$$Y_1 \equiv g(F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n)).$$

Since $1 - U$ is also uniform over $(0, 1)$ whenever $U$ is, it follows that

$$Y_2 \equiv g(F_1^{-1}(1 - U_1), \ldots, F_n^{-1}(1 - U_n))$$

has the same distribution as $Y_1$. When $g$ is a monotone function, $Y_1$ and $Y_2$ are indeed negatively correlated, and the antithetic variates estimate,

$$\frac{g(F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n)) + g(F_1^{-1}(1 - U_1), \ldots, F_n^{-1}(1 - U_n))}{2},$$

has a lower variance than the Monte Carlo method with two replications [790]. Computation time is also saved because only $n$ rather than $2n$ random numbers need to be generated, with each number used twice.

Computationally, for each simulated sample path $X$, a second one is obtained by reusing the random numbers on which the first path is based, yielding a second sample path $Y$.

Two estimates are then obtained, one based on $X$ and the other on $Y$. If a total of $N$ independent sample paths are generated, the antithetic variates estimator averages over $2N$ estimates.

**Example 18.2.2** Consider the Ito process $dX = a_t\,dt + b_t\sqrt{dt}\,\xi$. Let $g$ be a function of $n$ samples $X_1, X_2, \ldots, X_n$ on the sample path. We are interested in $E[\,g(X_1, X_2, \ldots, X_n)\,]$. Suppose one simulation run has realizations $\xi_1, \xi_2, \ldots, \xi_n$ for the normally distributed fluctuation term $\xi$, generating samples $x_1, x_2, \ldots, x_n$. The estimate is then $g(\boldsymbol{x})$, where $\boldsymbol{x} \equiv (x_1, x_2 \ldots, x_n)$. Instead of sampling $n$ more numbers from $\xi$ for the second estimate, the antithetic variates method computes $g(\boldsymbol{x}')$ from the sample path $\boldsymbol{x}' \equiv (x_1', x_2' \ldots, x_n')$ generated by $-\xi_1, -\xi_2, \ldots, -\xi_n$ and outputs $(g(\boldsymbol{x}) + g(\boldsymbol{x}'))/2$. Figure 18.7 implements the antithetic variates method for average-rate options. □

▷**Exercise 18.2.7** Justify the procedure in Example 18.2.2.

▷**Programming assignment 18.2.8** Implement the antithetic variates method for arithmetic average-rate calls and puts. Compare it with the Monte Carlo method in Programming assignment 18.2.3.

### Conditioning

Let $X$ be a random variable whose expectation is to be estimated. There is another random variable $Z$ such that the conditional expectation $E[\,X \mid Z = z\,]$ can be efficiently and precisely computed. We have $E[\,X\,] = E[\,E[\,X \mid Z\,]\,]$ by the law of iterated conditional expectations. Hence the random variable $E[\,X \mid Z\,]$ is also an unbiased estimator of $\mu$. As $\mathrm{Var}[\,E[\,X \mid Z\,]\,] \leq \mathrm{Var}[\,X\,]$, $E[\,X \mid Z\,]$ indeed has a smaller variance than observing $X$ directly [790]. The computing procedure is to first obtain a random observation $z$ on $Z$, then calculate $E[\,X \mid Z = z\,]$ as our estimate. There is no need to resort to simulation in computing $E[\,X \mid Z = z\,]$. The procedure can be repeated a few times to reduce the variance.

▷**Programming assignment 18.2.9** Apply conditioning to price European options when the stock price volatility is stochastic. The stock price and its volatility may be correlated.

### Control variates

The idea of **control variates** is to use the analytic solution of a similar yet simpler problem to improve the solution. Suppose we want to estimate $E[\,X\,]$ and there exists a random variable $Y$ with a known mean $\mu \equiv E[\,Y\,]$. Then $W \equiv X + \beta(Y - \mu)$ can serve as a "controlled" estimator of $E[\,X\,]$ for any constant $\beta$ which scales the deviation $Y - \mu$ to arrive at an adjustment for $X$. However $\beta$ is chosen, $W$ remains an unbiased estimator of $E[\,X\,]$. As

$$\mathrm{Var}[\,W\,] = \mathrm{Var}[\,X\,] + \beta^2\,\mathrm{Var}[\,Y\,] + 2\beta\,\mathrm{Cov}[\,X, Y\,], \tag{18.5}$$

$W$ is less variable than $X$ if and only if

$$\beta^2\,\mathrm{Var}[\,Y\,] + 2\beta\,\mathrm{Cov}[\,X, Y\,] < 0. \tag{18.6}$$

The success of the scheme clearly depends on both $\beta$ and the choice of $Y$. For example, arithmetic average-rate options can be priced by choosing $Y$ to be the otherwise identical geometric average-rate option's price and $\beta = -1$ [571]. This approach is much more effective than antithetic variates (see Fig. 18.8) [110].

| Antithetic variates | | Control variates | |
|---|---|---|---|
| 1.11039 | 1.11452 | 1.11815 | 1.11864 |
| 1.10952 | 1.10892 | 1.11788 | 1.11853 |
| 1.10476 | 1.10574 | 1.11856 | 1.11789 |
| 1.13225 | 1.10509 | 1.11852 | 1.11868 |
| (0.009032505) | | (0.000331789) | |

Figure 18.7: VARIANCE REDUCTION TECHNIQUES FOR AVERAGE-RATE PUTS. An arithmetic average-rate put is priced using antithetic variates and control variates ($\beta = -1$). The parameters used for each data are $S = 50$, $\sigma = 0.2$, $r = 0.05$, $\tau = 1/3$, $X = 50$, $n = 50$, and $m = 10000$. Sample standard deviations of the computed values are in parentheses.

Equation (18.5) is minimized when $\beta$ equals $\beta^* \equiv -\text{Cov}[X, Y]/\text{Var}[Y]$, which was called beta earlier in Exercise 6.4.1. For this specific $\beta$,

$$\text{Var}[W] = \text{Var}[X] - \frac{\text{Cov}[X, Y]^2}{\text{Var}[Y]} = \left(1 - \rho_{X,Y}^2\right) \text{Var}[X],$$

where $\rho_{X,Y}$ is the correlation between $X$ and $Y$. The stronger $X$ and $Y$ are correlated, the greater the reduction in variance. For example, if this correlation is nearly perfect ($\pm 1$), we could control $X$ almost exactly, eliminating practically all of its variance. Typically, neither $\text{Var}[Y]$ nor $\text{Cov}[X, Y]$ is known, unfortunately. Therefore, we cannot obtain the maximum reduction in variance. One approach in practice is to guess at these values and hope that the resulting $W$ does indeed have a smaller variance than $X$. A second possibility is to use the simulated data to estimate these quantities.

Observe that $-\beta^*$ has the same sign as the correlation between $X$ and $Y$. Hence, if $X$ and $Y$ are positively correlated, hence $\beta^* < 0$, then $X$ is adjusted downward whenever $Y > \mu$ and upward otherwise. The opposite is true when $X$ and $Y$ are negatively correlated, in which case $\beta^* > 0$.

▷**Exercise 18.2.10** Pick $\beta = \pm 1$. The success of the scheme now depends solely on the choice of $Y$. Derive the conditions under which the variance is reduced.

▷**Exercise 18.2.11** Why is it a mistake to use *independent* random numbers in generating $X$ and $Y$?

▷**Programming assignment 18.2.12** Implement the control variates method for arithmetic average-rate calls and puts.

### Other schemes

We briefly mention two more schemes before closing this section. In **stratified sampling**, the support of the random variable being simulated is partitioned into a finite number of disjoint regions and standard Monte Carlo simulation is performed in each region. When there is less variance within regions than across the regions, the sampling variance of the estimate will be reduced. **Importance sampling** samples more frequently in regions of the support where there is more variation.

▷**Exercise 18.2.13** Suppose you are searching in set $A$ for any element from set $B \subseteq A$. The Monte Carlo approach selects $N$ elements randomly from $A$ and checks if any one belongs to $B$. An alternative partitions the set $A$ into $m$ disjoint subsets $A_1, A_2, \ldots, A_m$ of equal size, picks $N/m$ elements from each subset randomly, and checks if there is a hit. Prove that the second approach's probability of failure can never exceed the Monte Carlo's.