Monte Carlo Simulation and Related Methods

In this lecture. . .

- the relationship between option values and expectations
- how to do Monte Carlo simulations to calculate derivative prices and to see the results of speculating with derivatives
- simulations in many dimensions using Cholesky factorization
- numerical integration in high dimensions to calculate the price of options on baskets

Relationship between derivative values and simulations

Simulations are at the very heart of finance. With simulations you can explore the unknown future. Simulations can also be used to price options.

- The fair value of an option is the present value of the expected payoff at expiry under a *risk-neutral* random walk for the underlying.

$$dS = r\,S\,dt + \sigma S\,dX$$

The risk-neutral random walk for S is

- $$dS = rS\,dt + \sigma S\,dX.$$

We can therefore write

- $$\text{option value } = e^{-r(T-t)}E\left[\text{payoff}(S)\right]$$

provided that the expectation is with respect to the risk-neutral random walk, not the *real* one.

The algorithm:

1. Simulate the risk-neutral random walk starting at today's value of the asset S_0 over the required time horizon. This gives one realization of the underlying price path.

2. For this realization calculate the option payoff.

3. Perform many more such realizations over the time horizon.

4. Calculate the average payoff over all realizations.

5. Take the present value of this average, this is the option value.

δS $\quad$ dS

Price paths are simulated using a discrete version of the stochastic differential equation for S. An obvious choice is to use

$$\delta S = rS\,\delta t + \sigma S\sqrt{\delta t}\,\phi,$$

where ϕ is from a standardized Normal distribution.

- This way of simulating the time series is called the **Euler method**. This method has an error of $O(\delta t)$.

For the lognormal random walk we are lucky that we can find a simple, and exact, timestepping algorithm. We can write the risk-neutral stochastic differential equation for S in the form

$$d(\log S) = \left(r - \tfrac{1}{2}\sigma^2\right) dt + \sigma \, dX.$$

This can be integrated exactly to give

$$S(t) = S(0) \exp\left(\left(r - \tfrac{1}{2}\sigma^2\right) t + \sigma \int_0^t dX\right).$$

Or, over a timestep δt,

- $$S(t+\delta t) = S(t) + \delta S = S(t) \exp\left(\left(r - \tfrac{1}{2}\sigma^2\right) \delta t + \sigma\sqrt{\delta t}\, \phi\right).$$

Note that δt need not be small, since the expression is exact. Because this expression is exact and simple it is the best timestepping algorithm to use.

Because it is exact, if we have a payoff that only depends on the final asset value, i.e. is European and path independent, then we can simulate the final asset price in one giant leap, using a timestep of T.

Advantages of Monte Carlo simulation

- The mathematics that you need to perform a Monte Carlo simulation can be very basic.
- Correlations can be easily modeled.
- There is plenty of software available, at the least there are spreadsheet functions that will suffice for most of the time.
- To get a better accuracy, just run more simulations.

- The effort in getting *some* answer is very low.
- The models can often be changed without much work.
- Complex path dependency can often be easily incorporated.
- People accept the technique, and will believe your answers.

Using random numbers

If the size of the timestep is δt then we may introduce errors of $O(\delta t)$ by virtue of the discrete approximation to continuous events.

E.g. if we have a barrier option we may miss the possibility of the barrier being triggered between steps.

- Because we are only simulating a finite number of an infinite number of possible paths, the error due to using N realizations of the asset price paths is $O(N^{-1/2})$.

ε $\quad$ $\text{Time} \propto d\dfrac{N}{\delta t} = O\left(d\,\varepsilon^{-3}\right)$

$\delta t = \varepsilon$

$\dfrac{1}{N^{1/2}} = \varepsilon$

$N = \varepsilon^{-2}$

$d = \dim$

The error in the price is therefore

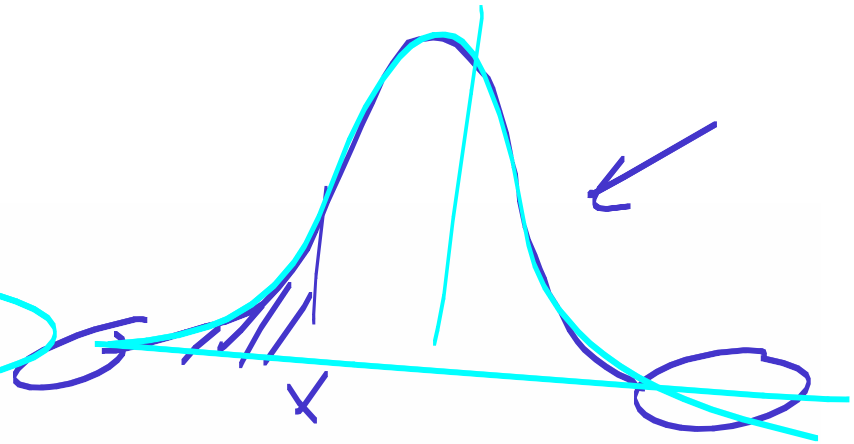
$$O\left(\max\left(\delta t, \frac{1}{\sqrt{N}}\right)\right).$$

The total number of calculations required in the estimation of a derivative price is then $O(N/\delta t)$.

To minimize the error, while keeping the total computing time fixed such that $O(N/\delta t) = K$, we must choose

$$N = O(K^{2/3}) \quad \text{and} \quad \delta t = O(K^{-1/3}).$$

Generating Normal variables



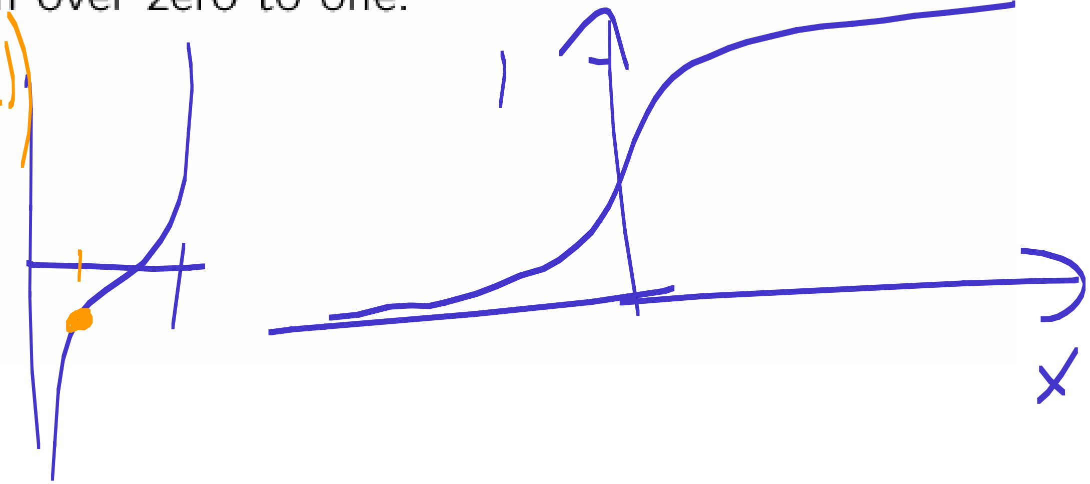
- A useful distribution that is easy to implement on a spreadsheet, and is fast, is the following *approximation* to the Normal distribution:

$$\boxed{\left(\sum_{i=1}^{12}\psi_i\right)-6,}$$

±6

where the ψ_i are independent random variables, drawn from a uniform distribution over zero to one.

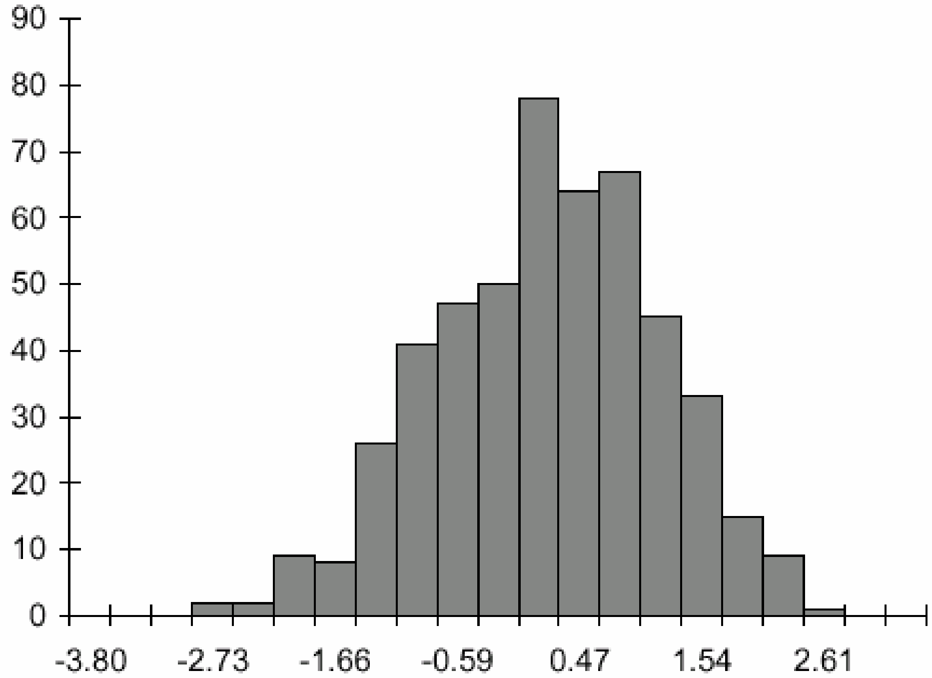
NORMSINV(RAND())



- To generate *genuinely* Normally-distributed random numbers use the **Box–Muller method**.

This method takes uniformly-distributed variables and turns them into Normal. The Box–Muller method takes two uniform random numbers x_1 and x_2 between zero and one and combines them to give two numbers y_1 and y_2 that are both Normally distributed:

$$y_1 = \sqrt{-2 \ln x_1} \cos(2\pi x_2) \quad \text{and} \quad y_2 = \sqrt{-2 \ln x_1} \sin(2\pi x_2).$$



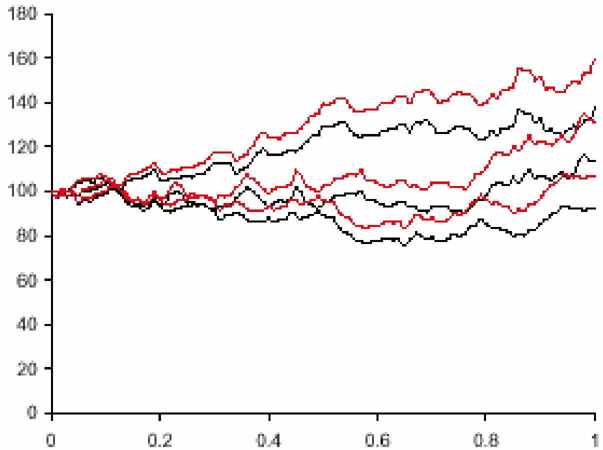
The approximation to the Normal distribution using Box–Muller.

$dS = rS\,dt +$

$dS = \mu S\,dt$

Real versus risk neutral, speculation versus hedging

In the next figure are shown several realizations of a risk-neutral asset price random walk. Also shown in the figure are the corresponding *real* random walks using the same random numbers but here with the real, and here higher, drift instead of the interest rate drift.



We can use simulations to estimate the payoff distribution from holding an *unhedged* option position.

In this situation we are interested in the whole distribution of payoffs (and their present values) and not just the average or expected value.

This is because in holding an unhedged position we cannot guarantee the return that we (theoretically) get from a hedged position.

- It is therefore valid and important to have the real drift as one of the parameters; it would be incorrect to estimate the probability density function for the return from an unhedged position using the risk-neutral drift.

Interest rate products

The relationship between expected payoffs and option values when the interest rate is stochastic is more complicated:

1. Simulate the random walk for the risk-adjusted spot interest rate r over the required time horizon. This gives one realization of the spot rate path.

2. For this realization calculate *two* quantities, the payoff and the *average* interest rate realized up until the payoff is received.

3. Perform many more such realizations.

4. For each realization of the r random walk calculate the present value of the payoff for this realization discounting at the average rate for this realization.

5. Calculate the average present value of the payoffs over all realizations, this is the option value.

In other words,

- $$\text{option value} = E\left[e^{-\int_t^T r(\tau)d\tau}\text{payoff}(r)\right].$$

Why is this different from the deterministic interest rate case? Why discount at the average interest rate?

We discount all cashflows at the average rate because this is the interest rate received by a money market account, and in the risk-neutral world all assets have the same risk-free growth rate.

Recall that cash in the bank grows according to

$$\frac{dM}{dt} = r(t)M.$$

The solution of which is

$$M(t) = M(T)e^{-\int_t^T r(\tau)d\tau}.$$

This contains the same discount factor as in the option value.

The choice of discretization of spot rate models is usually limited to the Euler method

$$\delta r = (u(r,t) - \lambda(r,t)w(r,t))dt + w(r,t)\sqrt{\delta t}\,\phi.$$

Calculating the greeks

$\times 2$

The simplest way to calculate the delta using Monte Carlo simulation is to estimate the option's value twice. The delta of the option is

$$\Delta = \lim_{h \to 0} \frac{V(S+h,t) - V(S-h,t)}{2h}.$$

This is an estimate of the first derivative, with an error of $O(h^2)$.

$O\left(\frac{1}{N^{1/2}}\right)$

However, the error in the measurement of the two option values at $S + h$ and $S - h$ can be much larger than this for the Monte Carlo simulation.

These Monte Carlo errors are then magnified when divided by h, resulting in an error of $O(1/hN^{1/2})$.

- To overcome this problem, estimate the value of the option at $S + h$ and $S - h$ *using the same values for the random numbers*. In this way the errors in the Monte Carlo simulation may cancel each other out.

Higher dimensions: Cholesky factorization

Monte Carlo simulation is a natural method for the pricing of European-style contracts that depend on many underlying assets.

Supposing that we have an option paying off some function of S_1, $S_2, \ldots, S_d$ then we could, in theory, write down a partial differential equation in $d+1$ variables. Such a problem would be horrendously time consuming to compute.

The simulation methods discussed above can easily be extended to cover such a problem. All we need to do is to simulate

$$S_i(t+\delta t) = S_i(t) \exp\left(\left(r - \tfrac{1}{2}\sigma_i^2\right)\delta t + \sigma_i \sqrt{\delta t}\,\phi_i\right).$$

$$dS_1 = r S_1\, dt + \sigma_1 S_1\, dX_1$$
$$dS_2 = r S_2\, dt + \sigma_2 S_2\, dX_2$$

The catch is that the ϕ_i are correlated,

$$E[\phi_i \phi_j] = \rho_{ij}.$$

How can we generate *correlated* random variables? This is where **Cholesky factorization** comes in.

Let us suppose that we can generate d *uncorrelated* Normally distributed variables ϵ_1, ϵ_2,. . . , ϵ_d. We can use these variables to get correlated variables with the transformation

$$\phi = \mathbf{M}\epsilon \tag{1}$$

where ϕ and ϵ are the column vectors with ϕ_i and ϵ_i in the ith rows.

The matrix $\mathbf{M}$ is special and must satisfy

$$\mathbf{M}\mathbf{M}^T = \boldsymbol{\Sigma}$$

with $\boldsymbol{\Sigma}$ being the correlation matrix.

It is easy to show that this transformation will work.

From (1) we have

$$\phi\phi^T = \mathbf{M}\epsilon\epsilon^T\mathbf{M}^T.$$

Taking expectations of each entry in this matrix equation gives

$$E\left[\phi\phi^T\right] = \mathbf{M}E\left[\epsilon\epsilon^T\right]\mathbf{M}^T = \mathbf{M}\mathbf{M}^T = \mathbf{\Sigma}.$$

This decomposition of the correlation matrix is not unique. It results in a matrix $\mathbf{M}$ that is lower triangular.

Speeding up convergence

Monte Carlo simulation is inefficient, compared with finite-difference methods, in dimensions less than about three. It is natural, therefore, to ask how can one speed up the convergence. There are several methods in common use:

- Antithetic variables
- Control variates

Antithetic variables

In this technique one calculates two estimates for an option value using the one set of random numbers. We do this by

- using our Normal random numbers to generate one realization of the asset price path, an option payoff and its present value

- taking the same set of random numbers but changing their signs, thus replacing ϕ with $-\phi$ and simulating a realization, and calculating the option payoff and its present value.

Our estimate for the option value is the average of these two.

Perform this operation many times to get an accurate estimate for the option value

Control variate technique

Suppose we have two similar derivatives, the former is the one we want to value by simulations and the second has a similar (but 'nicer') structure such that we have an explicit formula for its value.

- Use the *one* set of realizations to value *both* options.

Call the values estimated by the Monte Carlo simulation V_1' and V_2'. If the accurate value of the second option is V_2 then a better estimate than V_1' for the value of the first option is

$$V_1' - V_2' + V_2.$$

The argument behind this method is that the error in V_1' will be the same as the error in V_2', and the latter is known.