Fin 514: Financial Engineering II

Lecture 15: Monte Carlo Methods

Dr. Martin Widdicks

UIUC

Spring, 2018

Outline

- We now look at a numerical scheme that uses the probabilistic solution -Monte Carlo techniques.
- The main idea behind the Monte Carlo technique is that you simulate paths that could be taken by the underlying asset (under the risk-neutral probability) and then use these to estimate an expected option price at expiry, which can be discounted back to today at the appropriate rate.
- Monte Carlo techniques are very useful for path dependent options (such as Lookbacks and Asians) and for options on more than one underlying asset.
- Sadly, the convergence of Monte Carlo methods is slow and it is hard to determine the error terms.

Recap on Binomial tree/FD methods

- In the binomial, the convergence to the correct option value is $1/N$ where N is the number of time steps in the tree. For CN finite difference it's $1/jmax^2$ and $1/imax^2$ where $imax$ and $jmax$ are the number of time and S steps respectively.
- Those methods are particularly suited for valuing American options (or call features) as the process is backward induction and so you use known future option values to determine a strategy now.
- There are less useful for path dependent options such as Asian and lookback options, as this typically requires including another dimension into the problem (See Lecture 5).
- The computational effort increases **exponentially** as you add underlying assets, thus to price an option with d underlying assets (or sources of uncertainty such as stochastic volatility or stochastic interest rates) then an n step tree requires approximately n^d calculations.

Preview of Monte Carlo methods

- The convergence to the correct option value will be at a rate of $1/N^{1/2}$ where N is the number of sample paths.
- As it is a forward induction techniques, it is particularly suitable for valuing path dependent options such as lookback and Asian options.
- It is very unsuitable for valuing American style options (including those with call features) for exactly the same reasons, although we shall see methods for overcoming this.
- The computational effort increases linearly as you add underlying assets, thus to price an option with d underlying assets (or sources of uncertainty such as stochastic volatility or stochastic interest rates) then an N sample path Monte Carlo method requires approximately Nd calculations.
- Products with multiple underlying assets are common: Worst performing securities Note

Law of Large Numbers

- If we have a sequence of independent, identically distributed random variables S_n then we have that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} S_n = E[S_1]$$

  which is the law of large numbers.

- In other words the expectation is exactly like taking a long run average (exactly as we'd expect), so to evaluate the expectation to any desired accuracy we can simply take more and more draws of S_n.
- With the Monte Carlo technique what we are trying to do is to evaluate the value of $E[f(S_n)]$ which is the expectation of a function of a random variable S_n on independent draws (or paths) n.

Application to options

- If we consider S_t as the value of the stock at time t, then the option value at expiry, $t = T$, is $V(S_T)$ and from our probabilistic approach we know that, in the most general case,

$$V(S_t) = E_t^Q[e^{-\int_t^T r(s)ds} V(S_T)]$$

or, more usually,

$$V(S_t) = e^{-r(T-t)} E_t^Q[V(S_T)]$$

if r is constant. Where Q is the risk-neutral measure and E_t denotes taking the expectation at time t.

- Thus if we can estimate the expectation on the right hand side then we can simply discount this value at the risk-free rate to obtain the option price today. In fact, with Monte-Carlo methods it is also fairly straightforward to factor in stochastic interest rates as well.

Simple example with GBM

- If we assume that we are in the risk-neutral world and the underlying asset follows geometric Brownian motion, then, under the real world measure

$$dS_t = (\mu - \delta)S_t dt + \sigma S_t dX$$

and under the risk-neutral measure

$$dS_t = (r - \delta)S_t dt + \sigma S_t d\tilde{X}$$

where X and $\tilde{X}$ are both Brownian motions under the respective measures.

- The above stochastic differential equation is solved to yield:

$$S_T = S_t \exp\left[(r - \delta - \frac{1}{2}\sigma^2)(T - t) + \sigma(X(T) - X(t))\right]$$

or

$$S_T = S_t \exp\left[(r - \delta - \frac{1}{2}\sigma^2)(T - t) + \sigma\phi\sqrt{T - t}\right]$$

Simple example with GBM

- ϕ here is a variable drawn at random from a Normal distribution with a mean of 0 and a variance of 1, $N(0,1)$.
- To estimate the expected option value at time T, $V(S_T)$ then we take N random draws from the $N(0,1)$ distribution which enables us to calculate S_T and then calculate $V(S_T)$. To get an approximation of the expected payoff we average the $V(S_T)$ values.
- Thus if the nth draw from the normal distribution gives $V(S_T^n))$ then by the law of large numbers:

$$\frac{1}{N}\sum_{n=1}^{N} V(S_T^n) \to E_t^Q[V(S_T)] \text{ as } N \to \infty$$

- Now if we define the error from the nth sample path as ϵ_n so

$$\epsilon_n = V(S_T^n) - E_t^Q[V(S_T)]$$

CLT and error

- The Central Limit Theorem tells us that for large N if the individual errors ϵ_n have variance $v = var(\epsilon_n)$ (which is the same for all n) then the error when approximating the expectation:

$$\frac{1}{N}\sum_{n=1}^{N} V(S_T^n) - E_t^Q[V(S_T)]$$

  is approximately normally distributed with mean zero and variance v/N (standard deviation $\sqrt{v/N}$).
- Note that this error bound is difficult to estimate precisely as it is probabilistic, in that we only know the distribution of the errors rather than their actual values.
- Also the standard deviation of the error only declines with the square root of the number of paths N.
- For each individual path the error will be random, depending upon the draw of ϕ.

Quiz

Given what we know about the error in Monte Carlo can we use extrapolation techniques?

- Yes, we know the convergence is $1/N^{1/2}$
- No, we don't know the convergence.
- No, errors will not be monotonic.

Generating (Pseudo-)Random Number, ϕ

- Many statistical packages have (normal) random number generators
- Otherwise, generate Normal random numbers by:
  - Generating random numbers that are uniformly distributed on [0,1]. Transforming them to obtain normally distributed random numbers
- Most straightforward way:
  - Let F^{-1} denote inverse of Normal distribution function.
  - If x is uniformly distributed on $[0, 1]$, then $y = F^{-1}(x)$ is a standard variable from the normal distribution.
- If you have function F^{-1} then this is easy. For example, in Excel:
  - Has function RAND() which generates a random number uniformly distributed on $[0, 1]$.
  - Has function F^{-1}, called NORMSINV.
  - Thus, function call NORMSINV(RAND()) returns a realization of a standard normal random variable.

The Monte-Carlo method for European options

- That gives the basics of the Monte Carlo method, it is very simple to implement for all different types of options.
- For a European call options the payoff at maturity $V(S_T)$ is given by

$$V(S_T) = \max(S_T - K, 0)$$

and so, to value the option one simulates N possible values or paths for S_T by making N independent draws from $N(0,1)$ then to use these possible values, call them ϕ_n we have for $1 \leq n \leq N$,

$$\begin{aligned} S_T^n &= S_t \exp\left[(r - \frac{1}{2}\sigma^2)(T-t) + \sigma\phi_n\sqrt{T-t}\right] \\ V(S_T^n) &= \max(S_T^n - K, 0) \\ V(S_t) &= e^{-r(T-t)}\frac{1}{N}\sum_{n=1}^{N} V(S_T^n) \end{aligned}$$

The Monte-Carlo method for European options

- Note, here that the choice of GBM makes using the Monte Carlo method very straightforward as we can solve the SDE directly and so have very large time steps in the Monte Carlo technique - indeed for a European option, we need only **one** time step.
- In general, if we can solve the SDE directly (or determine an analytic transition density function) then this makes our Monte Carlo method far easier.
- For more advanced stock price (or interest rate) models, this will not always be the case. For example, we saw the CEV model in Lecture 8 and that does not have a solution to the SDE and so is challenging to simulate using Monte Carlo methods.

Valuing Asian options

- An Asian option is an option whose payoff is a function of the average price of the underlying asset over the lifetime of the option. The stock price is observed at M points in time (every day, every trading day, every week etc.) and the average is calculated by using either geometric or arithmetic averaging.
- Consider an Asian option whose payoff is

$$V(T) = \max(A - K, 0)$$

where

$$A = \frac{1}{M} \sum_{i=1}^{M} S(t_i)$$

and $S(t_i)$ are the share prices at the M sampling times $t_1, \ldots, t_M$.

- We need to adjust our Monte-Carlo method slightly to deal with this.

Valuing Asian options

- From the solution to the SDE we know that

$$S_{t_i} = S_t \exp\left[(r - \frac{1}{2}\sigma^2)(t_i - t) + \sigma(X(t_i) - X(t))\right]$$

$$S_{t_{i+1}} = S_{t_i} \exp\left[(r - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma(X(t_{i+1}) - X(t_i))\right]$$

  and so the procedure is as follows:

- Simulate each of the M Brownian increments dX_i by drawing ϕ_i from the Normal distribution and use

$$S^n_{t_i} = S^n_{t_{i-1}} \exp\left[(r - \frac{1}{2}\sigma^2)(t_i - t_{i-1}) + \sigma\sqrt{t_i - t_{i-1}}\phi_i\right]$$

  to estimate the underlying asset values at each time.

- Calculate the value of A^n

$$A^n = \frac{1}{M}\sum_{i=1}^{M} S^n(t_i)$$

  and the payoff $\max(A^n - K, 0)$

Valuing Asian options

- Then repeat this procedure N times to get the option value:

$$V(S_t) = e^{-r(T-t)} \frac{1}{N} \sum_{n=1}^{N} \max(A^n - K, 0)$$

- Note that here, we need to break up the path of S_t into M time steps and then use the N created paths to take the average.

Quiz

What is the number of ϕ values required to calculate the Asian option value?

- N
- M
- NM
- N^M
- M^N

More General Problem

- The two examples we talked about above with the BSM setup (geometric Brownian motion) are particularly easy because increments to $\ln(S)$ are normally distributed or alternatively we can write the solution to the SDE out explicitly in terms of the Normal distribution.
- This solution is equivalent to knowing the transition density function - see Lecture 17.
- What do we do for non-Normal stochastic processes?

More general problem

- Previously in Lecture 6 when we introduced the SDE,

$$dS_t = \mu S_t dt + \sigma S_t dX$$

we first introduced a process

$$S_{t+\Delta t} = S_t + \mu S_t dt + \sigma S_t \sqrt{\Delta t}\phi$$

where δt is a small increment of time and ϕ a Normally distributed random variable.

- The second process is more than a way to explain the first process, it is the **Euler scheme** to approximate the SDE. Letting $S_T^{\Delta t}$ denote the value of the Euler process at a future time T using an increment of Δt, $S_T^{\Delta t} \to S_T$ in a mean-square sense.
- Thus, in cases where we cannot solve the SDE directly, we can use the Euler scheme but only for *small* step sizes, dt.

Stochastic volatility

- The Euler approximation is useful for processes that do not have normally distributed returns, i.e. something other than GBM.
- Example: A Stochastic volatility model. Under Q, we have

$$\begin{aligned} dS_t &= rS_t dt + \sigma_t S_t dX_1 \\ d\sigma_t &= (a - b\sigma_t)dt + v\sqrt{\sigma_t}dX_2 \end{aligned}$$

where dX_1 and dX_2 may be correlated. To explicitly build in the correlation we write

$$\begin{aligned} dS_t &= rS_t dt + \sigma_t S_t dX_1 \\ d\sigma_t &= (a - b\sigma_t)dt + v\rho\sqrt{\sigma_t}dX_1 + v\sqrt{\sigma_t}\sqrt{1-\rho^2}dX_2 \end{aligned}$$

where dX_1 and dX_2 are now independent.

- Here a, b, v, r are constants, σ and S are stochastic processes and σ_t is interpreted as volatility at time t.

Stochastic volatility

- Consider $M+1$ equally spaced times $t_0, t_1, \ldots, t_M$, where $\Delta t = t_i - t_{i-1}$.
- Euler approximation is

$$\begin{aligned} S_{t_i} &= S_{t_{i-1}} + rS_{t_{i-1}}\Delta t + \sigma_{t_{i-1}} S_{t_{i-1}} \phi_{1i}\sqrt{\Delta t} \\ \sigma_{t_i} &= \sigma_{t_{i-1}} + (a - b\sigma_{t_{i-1}})\Delta t + v\rho\sqrt{\sigma_{t_{i-1}}}\phi_{1i}\sqrt{\Delta t} + v\sqrt{\sigma_{t_{i-1}}}\sqrt{1-\rho^2}\phi_{2i}\sqrt{\Delta t} \end{aligned}$$

  where the $\phi_{ki}(k = 1, 2)$ are independent standard normal r.v.'s.
- Only change from before is that we now do simulation using **Euler approximation**.

Quiz

Do you think this treatment of stochastic volatility or different underlying stochastic processes is simpler than adapting the binomial model?

- Yes.
- No.
- Depends

Is it as accurate?

Two underlying assets

- If we have two correlated underlying assets such that

$$\begin{aligned} dS_1 &= \mu_1 S_1 dt + \sigma_1 S_1 dX_1 \\ dS_2 &= \mu_2 S_2 dt + \sigma_2 S_2 dX_2 \end{aligned}$$

where, as above we can write the correlated Brownian motion in terms of two independent ones:

$$X_2 = \rho X_1 + \sqrt{1-\rho^2} X_3$$

or changing the notation slightly

$$\begin{aligned} dS_1 &= \mu_1 S_1 dt + \sigma_1 S_1 dX_1 \\ dS_2 &= \mu_2 S_2 dt + \sigma_2 \rho S_2 dX_1 + \sigma_2 \sqrt{1-\rho^2} S_2 dX_2 \end{aligned}$$

then it is very straightforward to value an option whose payoff depends upon both of the asset prices at expiry.

Two underlying assets

- Consider an option where $V(T) = \max(S_1 - K_1, S_2 - K_2, 0)$. So
$$V(t) = e^{-r(T-t)} E_t^Q [V(T)]$$
- Under risk-neutrality we have
$$\begin{aligned} dS_1 &= rS_1 dt + \sigma_1 S_1 dX_1 \\ dS_2 &= rS_2 dt + \sigma_2 \rho S_2 dX_1 + \sigma_2 \sqrt{1-\rho^2} S_2 dX_2 \end{aligned}$$

  thus from solving the SDEs
$$\begin{aligned} S_{1T}^n &= S_{1t}^n \exp\left[(r - \frac{1}{2}\sigma_1^2)(T-t) + \sigma_1 \sqrt{T-t}\phi_{1n}\right] \\ S_{2T}^n &= S_{2t}^n \exp\left[(r - \frac{1}{2}\sigma_2^2)(T-t) + \sigma_2 \rho \sqrt{T-t}\phi_{1n} + \sigma_2 \sqrt{1-\rho^2}\sqrt{T-t}\phi_{2n}\right] \end{aligned}$$

- So to value the equations simulate $2N$ draws of normally distributed random numbers and then determine the possible values of $V(T)$ and then discount the average.

Two underlying assets

- Thus the option value can be estimated as follows:

$$V(t) = e^{-r(T-t)} E_t^Q[V(T)] = \frac{1}{N} \sum_{n=1} N \max(S_{1T}^n - K_1, S_{2T}^n - K_2, 0)$$

- One question, however, is how to get the general form of the set of SDEs when we have more than one underlying asset.
- The way in which we typically see correlation matrices (in two underlying assets) are in terms of the covariance matrix of continuously compounded returns, i.e. $\ln(S_i(T)/S_i(t))$ and this matrix is typically of the form:

$$\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Two underlying assets

- We can write our two SDEs as follows:

$$\begin{pmatrix} dS_{1t} \\ dS_{2t} \end{pmatrix} = \begin{pmatrix} rS_{1t} \\ rS_{2t} \end{pmatrix} dt + \begin{pmatrix} \sigma_1 S_{1t} & 0 \\ \sigma_2 \rho S_{2t} & \sigma_2 \sqrt{1-\rho^2} S_{2t} \end{pmatrix} \begin{pmatrix} dX_{1t} \\ dX_{2t} \end{pmatrix}$$

- It so happens that the matrix, we'll call it M, here

$$\begin{pmatrix} \sigma_1 & 0 \\ \sigma_2 \rho & \sigma_2 \sqrt{1-\rho^2} \end{pmatrix}$$

is the 'square root'of the covariance matrix, Σ,

$$\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

in that $\Sigma = MM^T$. To check note that

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

More underlying assets

- In general, given any size of covariance matrix for any number of underlying assets it will be possible to generate correlated normally distributed random variables.
- For a covariance matrix for two underlyings the correlated random variables, y_1, y_2 say were obtained from

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = M \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}$$

  where M is the square root of the covariance matrix and ϕ are the independent Normally distributed random variables.
- In general to create d-correlated Normally distributed variables $y_1, \ldots, y_d$ from d uncorrelated variables $\phi_1, \ldots, \phi_d$ and $MM^T = \Sigma =$ covariance matrix, use

$$\begin{pmatrix} y_1 \\ \vdots \\ y_d \end{pmatrix} = M \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_d \end{pmatrix}$$

Finding square roots of the covariance matrix: Cholesky factorization

- We wish to write the process followed by the two assets in the following form

$$\begin{aligned} dS_1/S_1 &= rdt + m_{11}dX_1 + m_{12}dX_2 \\ dS_2/S_2 &= rdt + m_{21}dX_1 + m_{22}dX_2 \end{aligned}$$

where dX_1 and dX_2 are two independent Brownian motions.

- To determine m_{ij} we need to find a square root of the covariance matrix. As Σ is symmetric and generally strictly positive definite then there will be a solution to

$$MM^T = \Sigma$$

in fact, there will be many solutions.

Finding square roots of the covariance matrix: Cholesky factorization

- Let's write out covariance matrix as

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \dots & \sigma_{NN} \end{pmatrix}$$

- One solution is to restrict M to be a lower triangular matrix, where it can be found by Cholesky factorization. The matrix will look like

$$M = \begin{pmatrix} m_{11} & 0 & \dots & 0 \\ m_{21} & m_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \dots & m_{NN} \end{pmatrix}$$

Finding square roots of the covariance matrix: Cholesky factorization

- The algorithm is that, $m_{11} = \sqrt{\sigma_{11}}$ and then

$$\begin{aligned} m_{ii} &= \sqrt{\sigma_{ii} - \sum_{k=1}^{i-1} m_{ik}^2} \\ m_{ij} &= \frac{\sigma_{ij} - \sum_{k=1}^{j-1} m_{ik} m_{jk}}{m_{jj}} \end{aligned}$$

- For our simple 2×2 matrix we obtain

$$\begin{aligned} m_{11} &= \sigma_1 \\ m_{21} &= \rho\sigma_2 \\ m_{22} &= \sqrt{1-\rho^2}\sigma_2 \end{aligned}$$

Cholesky factorization: return to our example

- Thus

$$\begin{aligned} dS_1/S_1 &= rdt + \sigma_1 dX_1 \\ dS_2/S_2 &= rdt + \rho\sigma_2 dX_1 + \sqrt{1-\rho^2}\sigma_2 dX_2 \end{aligned}$$

  as above.

So far...

- Simple to program and to understand, they are ideal for a first approximation to a derivative value.
- Convergence is slow, and determined probabilistically which makes extrapolation impossible.
- It is naturally designed for forward looking problems such as path dependent derivatives such as lookback options and Asian options.
- It is also good for derivatives where there are multiple sources of uncertainty as the computational effort only increases linearly.

Antithetic variables

- Antithetic variables or antithetic sampling is a simple adjustment to generating $\phi_n (1 \leq n \leq N)$.
- Instead of making N independent draws, you draw the sample in pairs, if the ith Normally distributed variable is ϕ_i, then choose ϕ_{i+1} to be $-\phi_i$, then draw again for ϕ_{i+2}.
- $-\phi_i$ is also Normally distributed and most importantly the mean of the two draws is zero, and so this ensures that the mean of the sample paths will be correct and the distributions of draws will be symmetric.
- Thus if $N = 500,000$, you only need to make $250,000$ random draws for ϕ and use the negative of the draw to complete the required $500,000$ values.
- This should improve the convergence as the distribution of paths is better matched to the model, i.e. the mean of ϕ is zero

Control variate technique

- Control variate technique: This is explained through an example:
- We want to compute $E^Q[V(T)]$
- And we can write

$$V(T) = V(T) - V_1(T) + V_1(T)$$

where $E^Q[V_1(T)]$ is known analytically and the error in estimating $E^Q[V(T) - V_1(T)]$ by simulation is less than error in estimating $E^Q[V(T)]$.

- Then, a better estimate of $E^Q[V(T)]$ is the sum of the known value of $E^Q[V_1(T)]$ plus the estimate of $E^Q[V(T) - V_1(T)]$.

Control variate technique

- Consider a basket option with payoff of

$$V(T) = \max[(1/2)S_1(T) + (1/2)S_2(T) - K, 0]$$

(assuming both S_1 and S_2 follow GBM).

- A natural choice of control variate is

$$V_1(T) = \max[(S_1(T)S_2(T))^{1/2} - K, 0]$$

- Why is this a good choice? $E^Q[V_1(T)]$ is known as products and powers of lognormal r.v.'s are lognormal, so this is a similar calculation to Black-Scholes. Remember the midterm?
- The difference $V(T) - V_1(T)$ is relatively small, and thus a Monte Carlo estimate of $E^Q[V(T) - V_1(T)]$ will have a relatively small error

Control variate technique

- In particular, we know that under Q

$$\begin{aligned} dS_1/S_1 &= rdt + \sigma_1 dX_1 \\ dS_2/S_2 &= rdt + \rho\sigma_2 dX_1 + \sqrt{1-\rho^2}\sigma_2 dX_2 \end{aligned}$$

and so, if $Z(t) = (S_1(t)S_2(t))^{1/2}$ then

$$dZ = \frac{1}{2}\left(2r - \frac{1}{4}\left(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2\right)\right) Zdt + \frac{1}{2}\sigma_1 Z dX_1 + \frac{1}{2}\sigma_2 Z(\rho dX_1 + \sqrt{1-\rho^2})dX_2.$$

- From this there is an analytic formula for the call on the geometric average.
- We use the Monte Carlo to value the difference between an option with a payoff equal to the arithmetic average and the geometric average.
- This difference is then added onto the known analytic solution for the geometric average payoff.

Quiz

If you were attempting to value an autocallable bond, what would be a good V_1 to use?

- European call option.
- The standard bond.
- A callable bond.

Control variate for other numerical methods

- The control variate technique is not just useful for Monte Carlo techniques but it can also be useful for binomial lattices.
- The obvious example is for American option when you can calculate the European and American option values from the tree (V_E and V_A) - or even value the difference between them - and then use the Black-Scholes equation to get the true European value (V_{BS}).
- Then a new estimate of the American option value is $V_{BS} + E[V_A - V_E]$.
- Essentially the lattice has estimated the early exercise premium (the difference between European and American) and then this premium is added to the known value.

Importance sampling

- The idea behind importance sampling is that if you know that the payoff function is zero outside of an interval $[a, b]$ then any draw which makes S_T lie outside of $[a, b]$ is wasted.
- Ideally, you would only to sample from distributions that cause S_T to lie in $[a, b]$ and then multiply the result by the actual probability of S_T being in this region.
- Recall that we have a function that turns a uniform random variable $[0, 1]$ into a realization of S_T and so we can invert this map to find an interval $[x_1, x_2]$ that is mapped onto $[a, b]$, thus the probability of S_T being in $[a, b]$ is $x_2 - x_1$.

Importance sampling

- Thus to compute the expectation at time T
  - Draw variables from $[0, 1]$.
  - Multiply by $x_2 - x_1$ and then add to x_1 so that they are all in $[x_1, x_2]$.
  - Convert the x value into ϕ.
  - Determine value of S_T.
  - Determine the option value V_T from S_T.
  - Repeat over N draws of a random number.
  - Average the option values to obtain an expected payoff.
  - Multiply this expectation by $(x_2 - x_1)$ and discount back to 0.

Importance sampling

- For example, $S_0 = 100, K = 100, r = 0.05, \sigma = 0.2, T = 1$. Consider a call option and so for $[0, 100]$ the payoff contribution is zero. Thus we only need to sample in $[100, \infty)$. $S_T = 100$ corresponds to a ϕ value of:

$$\begin{aligned} \phi &= \frac{\ln(100/100) - (r - \frac{1}{2}\sigma^2)}{\sigma} \\ &= \frac{-(0.04 - 0.02)}{0.2} \\ &= -0.1 \end{aligned}$$

  but $P(\phi < -0.1) = 0.461$. So $[0.461, 1]$ is the range of x values required.

- So, every draw of x is multiplied by 0.539 and then added to 0.461 to obtain a ϕ value and this only gives S_T values greater than 100.
- Once you have the final expected payoff, multiply by 0.539 and discount back to $t = 0$.

Quiz

What will this be the most effective for pricing?

- In the money options
- At the money options
- Out of the money options?

Overview

- We have introduced Monte Carlo methods which are very closely related to the probabilistic solution, in that you use simulation to determine an expected value for the option in the future that can be discounted at the risk-free rate (according to the fundamental theorem) to obtain the value today.
- To simulate the paths we typically use the solution to the SDE or the Euler approximation, along with a decent generator of Normally distributed random variables.
- It is easy to apply to path dependent options and to options on more than one underlying, for these you need to know the covariance matrix and be able to 'square root' it by means of Cholesky factorization to be able to perform the simulations.
- We have looked through a variety of extensions to the standard Monte Carlo in an effort to reduce the variance of the error or to improve the convergence.
- Most of the improvements are simple to apply -such as antithetic variables, control variate and importance sampling.

Aside: Recall: Generating (Pseudo-)Random Numbers

- Many statistical packages have (normal) random number generators
- Otherwise, generate Normal random numbers by:
  - Generating random numbers that are uniformly distributed on [0,1]. Transforming them to obtain normally distributed random numbers
- Most straightforward way:
  - Let F^{-1} denote inverse of Normal distribution function.
  - If x is uniformly distributed on $[0, 1]$, then $y = F^{-1}(x)$ is a standard variable from the normal distribution.
- If you have function F^{-1} then this is easy. For example, in Excel:
  - Has function RAND() which generates a random number uniformly distributed on $[0, 1]$.
  - Has function F^{-1}, called NORSINV.
  - Thus, function call NORSINV(RAND()) returns a realization of a standard normal random variable.

Aside: Generating (Pseudo-)Random Numbers

- Disadvantage: F^{-1} is not known in closed form, so this approach is not always fast.
- The other issue is that these are only pseudo random numbers in that the computer program typically has an algorithm for calculating the 'random numbers' , as John von Neumann said in 1951 "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin".
- There are many alternatives (The "Recipe" books of Press, Teuklovsky, Vetterling and Flannery are a good source, and have a good discussion of the problem), such as the Mid-Square method, Congruential generation or the popular Mersenne twister...

The Box-Muller method

- The simplest technique for generating decent' Normally distributed random numbers (According to Wilmott) is the Box-Muller method which takes any uniformly distributed variables (from any source you prefer) and turns them into Normally distributed ones.
- Given two uniformly distributed random numbers x_1, x_2, two Normally distributed random numbers, y_1 and y_2 are given by:

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

Aside: Quiz

Using the Box-Muller method overcomes which problem when generating Normally distributed random numbers?

- $[0, 1]$ Random numbers aren't random.
- F^{-1} is slow to calculate.
- Both?

Aside: Moment matching

- One fairly simple strategy that works in a similar way to using antithetic variables is called moment matching.
- The typical way it is done is to ensure that the variance of the sample paths match the variance of the required distribution (antithetic variables will automatically ensured that the mean (and skewness) match)
- Our Brownian motion modeling requires the variance of ϕ to be 1, so we would like the variance of our random ϕ's to share this property.
- To do this we first sample our N ϕ values (requiring $N/2$ random numbers). Then calculate their variance, V say, now replace all of the ϕ values with $\phi \times V^{-1/2}$ and the variance of the new random draws is 1, as required.

Aside: Low discrepancy sequences

- One of the most useful practical techniques for improving the accuracy of Monte Carlo methods is by using Low discrepancy sequences (also known as Quasi Monte Carlo methods).
- The theory behind Monte Carlo techniques is that as you take more and more sample paths then they will eventually cover the entire distribution of S_T in the correct manner.
- Another way to think of this is that our random numbers drawn from $[0, 1]$ will eventually cover this interval in a uniform manner.
- Unfortunately, we cannot actually draw an infinite number of paths and for any size of N it may well be the case that our S_T values all cluster around particular values while missing out other regions of S space entirely.
- This problem becomes more pronounced as we increase the number of dimensions, d.

Aside: Low discrepancy sequences

- To overcome this problem we throw away the idea of using 'random'numbers at all and instead choose a deterministic sequence of numbers that does a very good job of covering the $[0, 1]$ interval.
- Note that as we have already discussed, most random number generators are deterministic to some extent and so this approach isn't as odd as you would imagine.
- The most interesting thing about a low discrepancy series is that it can improve the convergence of the Monte Carlo method from $1/N^{1/2}$ to $1/N$, making the Monte Carlo method fully competitive with binomial lattices.
- We will deal with a simple sequence here but there is an extensive literature on low discrepancy series, see Jäckel, Monte Carlo Methods in Finance for an excellent summary.

Aside: The Halton Sequence

- Sobol' sequences are the most common of the low discrepancy sequences, but for ease of explanation we will consider the Halton sequence.
- The Halton sequence is a sequence of numbers $h(i; b)$ for $i = 1, 2, \ldots$ where b is the base and all of the numbers in the sequence are in $[0, 1]$. You can choose the base, let us select base 2.
- The Halton sequence is the reflection of the positive integers in the decimal point and is best observed through an example:

Integers base 10	Integers base 2	Halton seqeunce base 2	Halton number base 10
1	1	1 x ½	0.5
2	10	0 x ½ + 1 x ¼	0.25
3	11	1 x ½ + 1 x ¼	0.75
4	100	0 x ½ + 0 x ¼ + 1 x 1/8	0.125

Aside: The Halton Sequence

- So $10 \rightarrow 0.01$ in base 2 terms.
- In general the ith integer, i, can be expressed as:

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

where b is the base and $0 \leq a_j < b$. The Halton numbers are given by:

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

- What is nice about the Halton sequence is that it fills the range $[0, 1]$ gradually.
- When extending to multiple dimensions you should choose d prime bases .

Aside: The Halton Sequence

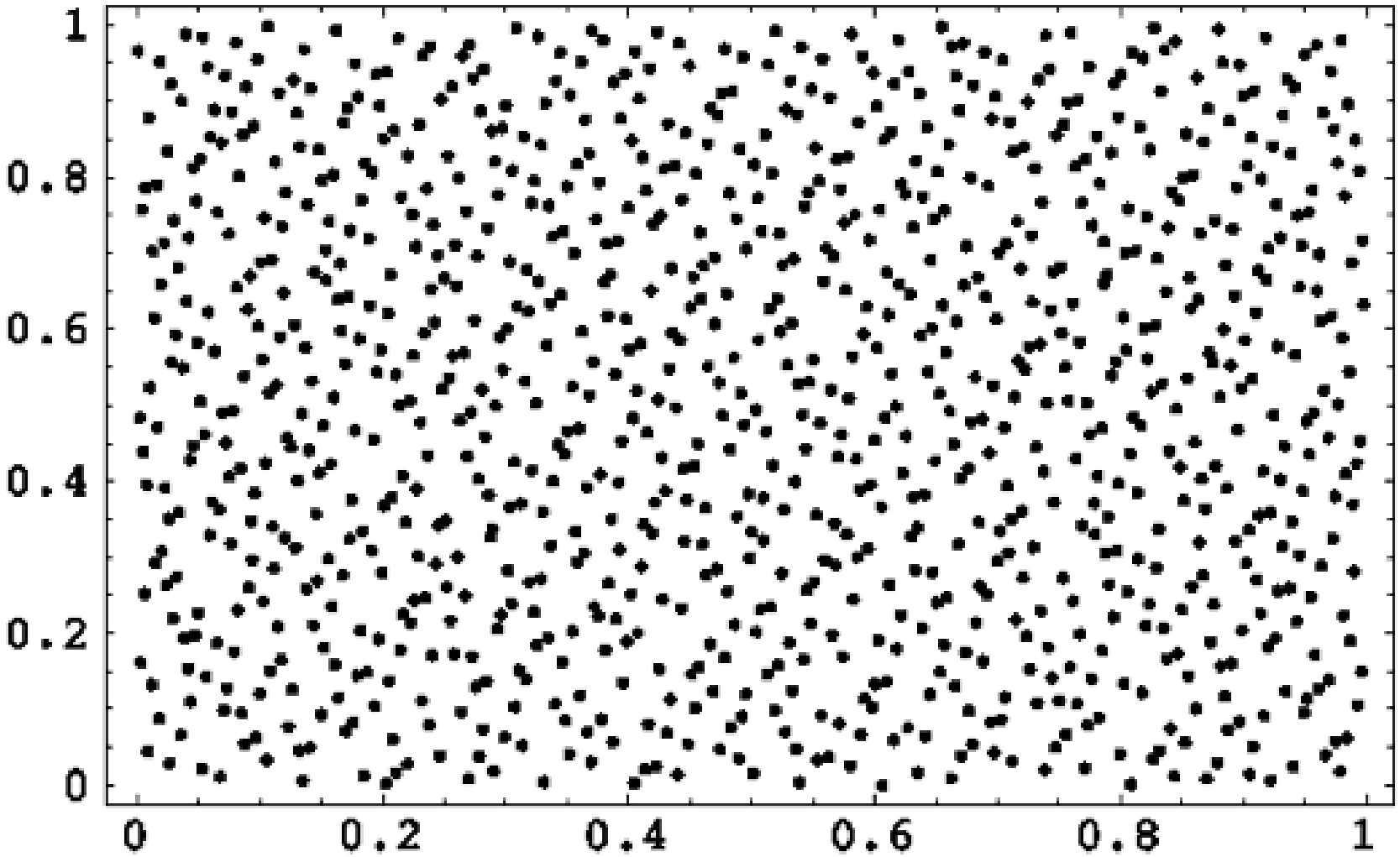
- Thus to estimate a d-dimensional integral

$$\int_0^1 \dots \int_0^1 f(x_1, \dots, x_d) dx_1, \dots, dx_d \approx \frac{1}{N} \sum_{i=1}^{N} f(h(i, b_1), \dots, h(i, b_n))$$

  where b_j are distinct prime numbers.

- The error from the Halton sequence is $O((lnN)^d N^{-1})$ which is better than standard Monte-Carlo.
- This can be improved to $O(1/N)$ by using Sobol' sequences instead but these are more complex algorithms, again Jäckel gives a good summary or alternatively use the algorithms in Press et al.

Aside: 2 Dimensional Halton sequence



Advanced: 2-dimensional pseudo random sequence