Mod 3 Workshop

In this workshop ..

- Early exercise - detailed review
- Monte Carlo
- Further Finite Difference Methods

The Mathematics of Early Exercise - American Options

American options are contracts that may be exercised early, *prior* to expiry. For example, if the option is a call, we may hand over the exercise price and receive the asset whenever we wish. These options must be contrasted with European options for which exercise is only permitted *at* expiry. Most traded stock and futures options are American style, but most index options are European. The right to exercise at any time at will is clearly valuable, hence the value of an American option cannot be less than an equivalent European option. But as well as giving the holder more rights, they also give them more headaches; when should they exercise? This additional optionality feature clearly make these more interesting from a modelling perspective for mathematicians. Part of the valuation problem is deciding when is the best time to exercise. This is what makes American options much more interesting than their European cousins. The pricing presents us with a computational exercise. However to get a feel for the difference between early exercise problems and plain vanilla contracts, we begin by looking at a *perpetual American option*.

The perpetual American put

There is a very simple example of an American option that we can examine for the insight that it gives us in the general case: the **perpetual American put**. This contract can be exercised for a put payoff at *any* time. There is no expiry. So we can, at any time of *our* choosing, sell the underlying and receive an amount E. That is, the payoff is $\max(E - S, 0)$.

We want to find the value of this option before exercise.

- The solution is independent of time, $V(S)$. It depends only on the level of the underlying.

- The option value can never go below the early-exercise payoff.

In the case under consideration

$$V \geq \max(E - S, 0). \tag{1}$$

Consider what would happen if this 'constraint' were violated. Suppose that the option value were less than $\max(E - S, 0)$, we could buy the option for $\max(E - S, 0)$, immediately exercise it by handing over the asset (worth S) and receive an amount E. We thus make

$$-\text{cost of put} - \text{cost of asset} + \text{strike price} = -V - S + E > 0.$$

This is a riskless profit.

Recalling that the option is perpetual and therefore that the value is independent of t, it must satisfy

$$\tfrac{1}{2}\sigma^2 S^2 \frac{d^2V}{dS^2} + rS\frac{dV}{dS} - rV = 0.$$

This is the ordinary differential equation obtained when the option value is a function of S only. The general solution of this second-order ordinary differential equation is

$$V(S) = AS + BS^{-2r/\sigma^2},$$

where A and B are arbitrary constants. Clearly, for the perpetual American put the coefficient A must be zero; as $S \to \infty$ the value of the option must tend to zero. What about B?

Postulate that while the asset value is 'high' we won't exercise the option. But if it falls too low we immediately exercise the option, receiving $E - S$. Suppose that we decide that $S = S^*$ is the value at which we exercise, i.e. as soon as S reaches this value from above we exercise.

- How do we choose S^*?

When $S = S^*$ the option value must be the same as the exercise payoff:

$$V(S^*) = E - S^*.$$

It cannot be less, that would result in an arbitrage opportunity, and it cannot be more or we wouldn't exercise.

Continuity of the option value with the payoff gives us one equation:

$$V(S^*) = B(S^*)^{-2r/\sigma^2} = E - S^*.$$

But since both B and S^* are unknown, we need one more equation.

Let's look at the value of the option as a function of S^*, eliminating B using the above. We find that for $S > S^*$

$$V(S) = (E - S^*)\left(\frac{S}{S^*}\right)^{-2r/\sigma^2}. \tag{2}$$

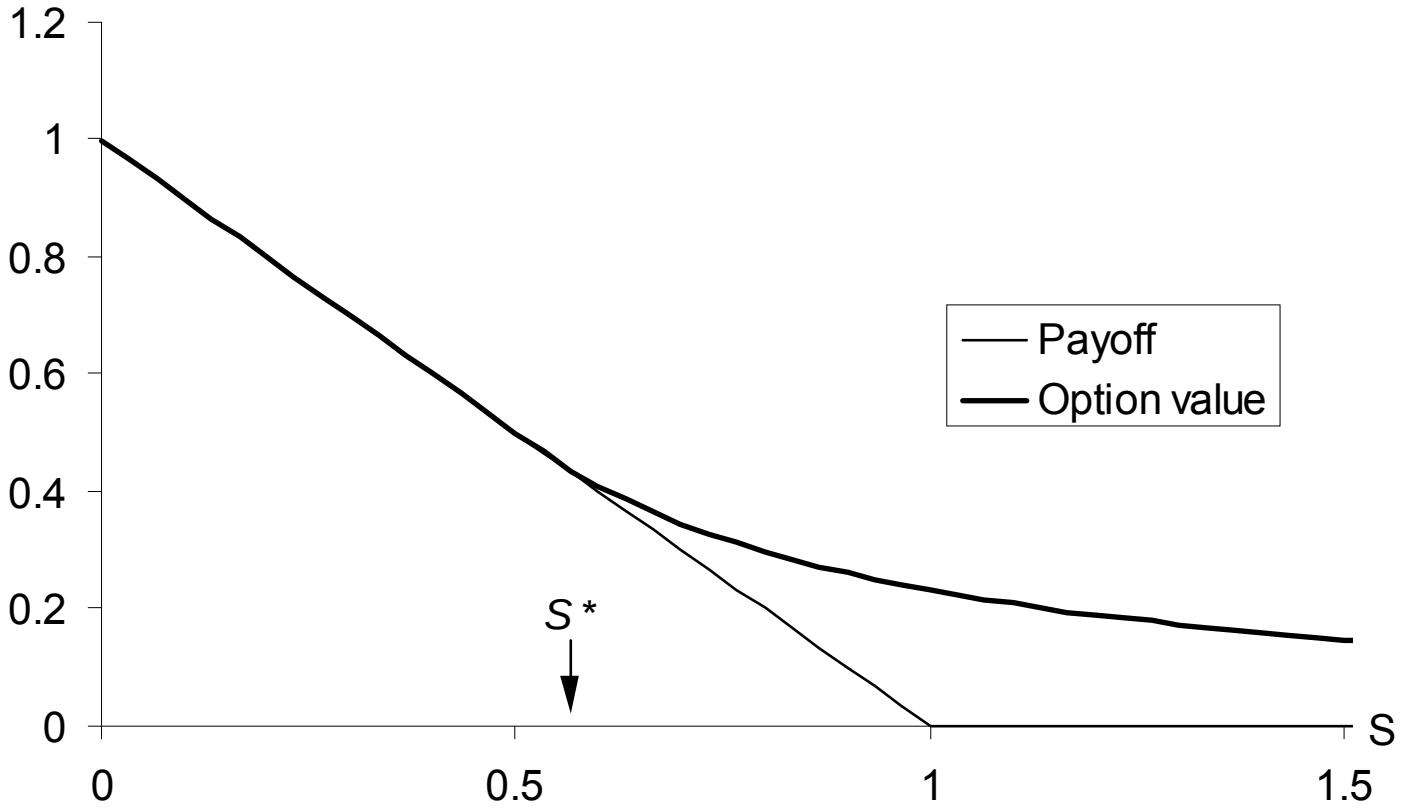
We are going to choose S^* to *maximize the option's value at any time before exercise*. In other words, what choice of S^* makes V given by (2) as large as possible? We find this value by differentiating (2) with respect to S^* and setting the resulting expression equal to zero:

$$\begin{aligned} \frac{\partial}{\partial S^*}(E - S^*)\left(\frac{S}{S^*}\right)^{-2r/\sigma^2} &= \frac{1}{S^*}\left(\frac{S}{S^*}\right)^{-2r/\sigma^2}\left(-S^* + \frac{2r}{\sigma^2}(E - S^*)\right) \\ &= 0. \end{aligned}$$

We find that

$$S^* = \frac{E}{1 + \sigma^2/2r}.$$

This choice maximizes $V(S)$ for *all* $S \geq S^*$. The solution with this choice for S^* is shown below.



The solution for the perpetual American put.

There is something special about this function: the slope of the option value and the slope of the payoff function are the same at $S = S^*$. To see that this follows from the choice of S^* examine the difference between the option value and the payoff function:

$$(E - S^*)\left(\frac{S}{S^*}\right)^{-2r/\sigma^2} - (E - S).$$

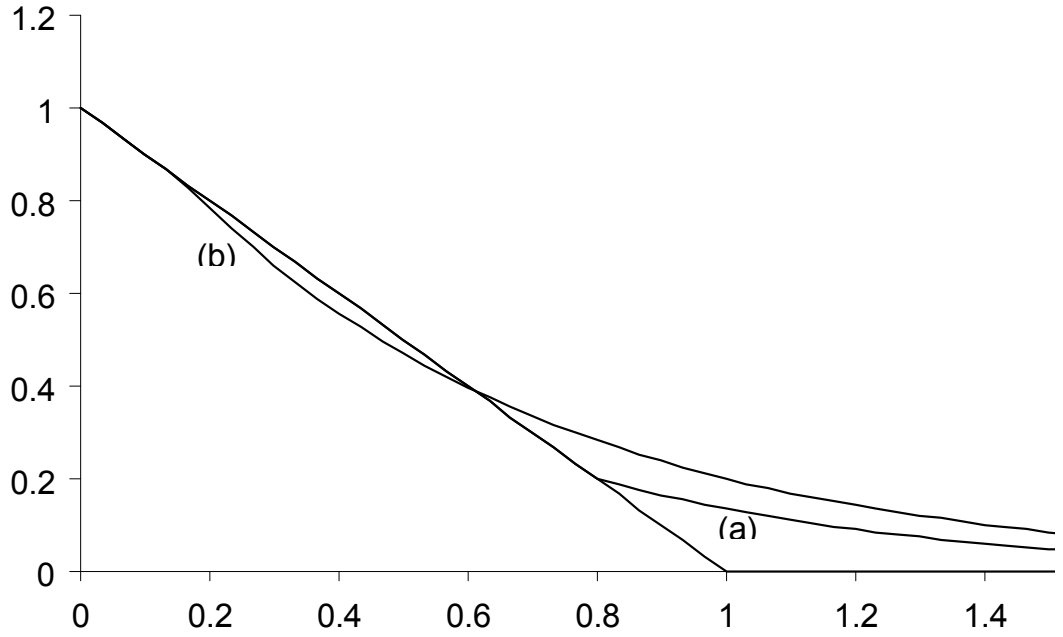
This expression is zero at $S = S^*$.

This demonstrates, in a completely non-rigorous way, that if we want to maximize our option's value by a careful choice of exercise strategy, then this is equivalent to solving the Black–Scholes equation with continuity of option *value* and option *delta*, the slope. This is called the **high-contact** or **smooth-pasting condition**.

The American option value is maximized by
an exercise strategy that makes the
option value and option delta continuous

We exercise the option as soon as the asset price reaches the level at which the option price and the payoff meet. This position, S^*, is called the **optimal exercise point**.

Consider what happens if the delta is not continuous at the exercise point. The two possibilities are shown below.



Option price when exercise is (a) too soon (b) too late.

Case (a) corresponds to exercise that is not optimal because it is premature, the option value is lower than it could be.

In case (b) there is clearly an arbitrage opportunity.

To summarize:

For a put solve

$$\tfrac{1}{2}\sigma^2 S^2 \frac{d^2V}{dS^2} + rS\frac{dV}{dS} - rV = 0.$$

subject to

$$\left.\begin{array}{r} V(S^*) = E - S^* \\ \frac{dV}{dS}(S^*) = -1 \end{array}\right\} \text{ Smooth pasting condition}$$

$$\lim_{S \longrightarrow \infty} V(S) \longrightarrow 0$$

For a **call** solve the same BSE together with

$$\left.\begin{array}{r} V(S^*) = S^* - E \\ \frac{dV}{dS}(S^*) = +1 \end{array}\right\} \text{ Smooth pasting condition}$$

$$V(S = 0) = 0 \Longrightarrow B = 0$$

Monte Carlo Techniques

Earlier we derived the Black-Scholes problem to price a European option $V(S,t)$, where the underlying asset follows GBM

$$dS = \mu S dt + \sigma S dW.$$

The resulting PDE and payoff $P(S)$ at expiry T which is satisfied by $V(S,t)$ is

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV &= 0, \\ V(S,T) &= P(S). \end{aligned}$$

The BSE is a linear parabolic PDE and as such the solution can be expressed as an integral of the form

$$V(S,t) = e^{-r(T-t)} \int_0^\infty \widetilde{p}\left(S,t;S',T\right) V\left(S',T\right) dS'.$$

$\widetilde{p}\left(S,t;S',T\right)$ represents the transition density and is the solution of the backward Kolmogorov problem

$$\begin{aligned}\frac{\partial\widetilde{p}}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2\widetilde{p}}{\partial S^2}+rS\frac{\partial\widetilde{p}}{\partial S} &= 0,\\ \widetilde{p}\left(S,t;S',T\right) &= \delta\left(S'-S\right).\end{aligned}$$

We have discussed that the function $\widetilde{p}\left(S,t;S',T\right)$ can be considered as one of two entities.

Firstly in the PDE framework it can be thought of as a Green's function for the general backward problem

$$\begin{aligned}\frac{\partial U}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2U}{\partial S^2}+rS\frac{\partial U}{\partial S} &= 0,\\ U\left(S,T\right) &= f\left(S\right).\end{aligned}$$

Secondly, and more importantly for this section in probabilistic terms it is the probability density function for the risk-neutral random walk mentioned earlier. This is also called the risk-neutral measure.

We can write the value of the option in the form

$$V(S,t) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}\left[P\left(S\right)\right]$$

which is the present value of the expected payoff wrt the risk-neutral probability density $\mathbb{Q}$ and recall

$$\mathbb{E}^{\mathbb{Q}}\left[P\left(S\right)\right] = \int_0^{\infty} \widetilde{p}\left(S,t;S',T\right) P\left(S'\right) dS'.$$

The precise form of the integral obtained in the BSE work is

$$\int_0^{\infty} e^{-\left(\log(S/S')+\left(r-\frac{1}{2}\sigma^2\right)(T-t)\right)^2/2\sigma^2(T-t)} \text{Payoff}(S')\frac{dS'}{S'}. \qquad \text{(A)}$$

This expression works because the equation is linear - so we just need to specify the payoff condition. It can be applied to any European option on a single lognormal underlying asset.

Equation (A) gives us the risk-neutral valuation. $e^{-r(T-t)}$ present values to today time t.

The integral is the expected value of the payoff with respect to the lognormal transition pdf. The future state is (S', T) and today is (S, t) . So it represents the probability of going from $(S, t) \longrightarrow (S', T)$.

Also note the presence of the risk-free IR r in the pdf. So the expected payoff is as if the underlying evolves according to the *risk-neutral* random walk

$$\frac{dS}{S} = rdt + \sigma dW.$$

The real world drift μ is now replaced by the risk-free return r. The delta hedging has eliminated all the associated risk. This means that if two investors agree on the volatility they will also agree on the price of the derivatives even if they disagree on the drift. This brings us on to the idea of *risk-neutrality* and risk-neutral pricing.

So we can think of the option as discounted expectation of the payoff under the assumption that S follows the risk neutral random walk

$$V(S,t) = e^{-r(T-t)} \int_0^\infty \widetilde{p}\left(S,t;S',T\right) V\left(S',T\right) dS'$$

where $p\left(S,t;S',T\right)$ represents the transition density and gives the probability of going from (S,t) to (S',T) under $\frac{dS}{S} = rdt + \sigma dW$, i.e. the risk-neutral random walk.

So clearly we have a definition for $\widetilde{p}$, i.e. the lognormal density given by

$$\widetilde{p}\left(S,t;S',T\right) = \frac{1}{\sigma S' \sqrt{2\pi(T-t)}} e^{-\left(\log(S/S') + \left(r - \frac{1}{2}\sigma^2\right)(T-t)\right)^2 / 2\sigma^2(T-t)}.$$

Two important points

- $\widetilde{p}\left(S,t;S',T\right)$ is a Green's for the BSE. As the PDE is linear we can write the solution down as the integrand consisting of this function and the final condition.

- The BSE is essentially the backward Kolmogorov equation whose solution is the transition density $\widetilde{p}\,(S,t;S',T)$ with (S',T) fixed and varying (S,t) ; but with the discounting factor.

Introduction

Simulations are at the very heart of finance and used to forecast future scenarios. More importantly they can also be used to price options. Partically all option-pricing problems have to be treated by numerical means. It is highly unlikely that analytical solutions can be obtained to a pricing problem unless it is simple and ideal.
The most useful numerical techniques are the Monte Carlo Scheme and finite-difference methods (both seen earlier).

The former is a simulation based approach hence uses probability as the underlying method.In finance we are concerned with estimating expectations of discounted payoffs where the underlying asset price can be modelled by stochastic differential equations. Whether pricing European options where just the terminal value is required or path dependent contracts requring the entire trajectory, simulations lie at the centre of derivative pricing.

For us the objective of the Monte Carlo method is to

- estimate prices which correspond to expected values of discounted payoffs

$$V(S,t) = \mathbb{E}\left[e^{-\int_t^T r_\tau d\tau} P(S_T) \right]$$

- estimate price sensitivities, i.e. Greeks, for hedging/risk management

$$\frac{\partial V}{\partial x}$$

  where x represents an underlying variable (stock S) or 'parameter' such as volatility or risk free interest, etc.

The view of some academics and practitioners as to the use of numerical methods in industry presents the split as follows:

- 60% Monte Carlo
- 30% finite differences
- 10% binomial trees and transfrom methods

MC vs. FDM

Monte Carlo strengths

- Simple to implement and flexible - the maths needed can be very basic. The effort in getting a price can be very low

- Easy to handle high dimensions - correlations can be easily modelled and it is easy to price options on several underlyings. This can be a 'nightmare' in the finite difference method!

- Computationally quite efficient in high dimensions

- Accuracy easily improved by increasing the number of simulations

- Large availability of software

- Complex path dependency can be easily incorporated

Monte Carlo weaknesses

- The method can be slow for very low dimensions (1-3)

- Accuracy comes at the expense of computational cost due to the large number of simulations required

- The method does not cope well with embedded decisions - early exercise features.

So which method is used when in industry?

- **Equity** - **MC** due to high dimenisonal problem (basket options)
- **Foreign Exchange** - **FDM** due to the low (three) dimensional nature. Domestic interest rate r, one foreign exchange rate r_f.
- **Fixed Income** - **MC** for LIBOR (and HJM) models because of high dimensionality.
- **Credit** - **MC** because high dimensional due to multiple companies

We know from earlier that the SDE $\dfrac{dS_t}{S_t} = rdt + \sigma dW_t$ with constant r and σ has the solution

$$S_T = S_0 \exp\left\{\left(r - \tfrac{1}{2}\sigma^2\right)T + \sigma\phi\sqrt{T}\right\},$$

for some time horizon T; with $\phi \sim N(0,1)$; $W_t \sim N(0,t)$ and can be written $\phi\sqrt{T}$.

It is often more convenient to express in time stepping form

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

In general a closed form solution of an arbitrary SDE is difficult if e.g. $r = r(t)$ and $\sigma = \sigma(S,t)$, i.e. the parameters are no longer constant; or the SDE is complicated.

The need for Monte Carlo requires numerical integration of stochastic differential equations. Previously we considered the Forward **Euler-Maruyama** method. Why did this work?

Consider

$$dX_t = a\left(X_t, t\right) dt + b\left(X_t, t\right) dW_t \tag{1}$$

The simplest scheme for solving (1) is using the E-M method. That is

$$\int_{t_n}^{t_{n+1}} dX_s = \int_{t_n}^{t_{n+1}} a\left(X_s, s\right) ds + \int_{t_n}^{t_{n+1}} b\left(X_s, s\right) dW_s$$

$$X_{n+1} = X_n + \int_{t_n}^{t_{n+1}} a\left(X_s, s\right) ds + \int_{t_n}^{t_{n+1}} b\left(X_s, s\right) dW_s$$

Using the left hand integration rule:

$$\begin{aligned}
\int_{t_n}^{t_{n+1}} a\left(s, X_s\right) ds &\approx a\left(t_n, X_n\right) \int_{t_n}^{t_{n+1}} ds = a\left(t_n, X_n\right) \delta t \\
\int_{t_n}^{t_{n+1}} b\left(s, X_s\right) ds &\approx b\left(t_n, X_n\right) \int_{t_n}^{t_{n+1}} dW_s = b\left(t_n, X_n\right) \Delta W_n
\end{aligned}$$

$$X_{n+1} = X_n + a\left(t_n, X_n\right) \delta t + b\left(t_n, X_n\right) \Delta W_n$$

where $\Delta W_n = \left(W_{n+1} - W_n\right)$.

The Forward **Euler-Maruyama** method for GBM gives

$$\frac{\delta S_t}{S_t} = \frac{S_{t+\delta t} - S_t}{S_t} \sim r\delta t + \sigma\phi\sqrt{\delta t}$$

i.e

$$S_{t+\delta t} \sim S_t \left(1 + r\delta t + \sigma\phi\sqrt{\delta t}\right).$$

Now do a Taylor series expansion of the exact solution, i.e.

$$e^{\left(r-\frac{1}{2}\sigma^2\right)\delta t+\sigma\phi\sqrt{\delta t}} \sim 1 + \left(r - \tfrac{1}{2}\sigma^2\right)\delta t + \sigma\phi\sqrt{\delta t} + \tfrac{1}{2}\sigma^2\phi^2\delta t$$

so we have

$$S_{t+\delta t} \sim S_t \left(1 + r\delta t + \sigma\phi\sqrt{\delta t} + \tfrac{1}{2}\sigma^2\left(\phi^2 - 1\right)\delta t + ..\right)$$

which differs from the Euler method at $O(\delta t)$ by the term $\frac{1}{2}\sigma^2\left(\phi^2 - 1\right)\delta t$. The term

$$\tfrac{1}{2}\left(\phi^2 - 1\right)\delta t,$$

is called the *Milstein correction*.

Milstein Integration

We approximate the solution of the SDE

$$dG_t = A\left(G_t, t\right) dt + B\left(G_t, t\right) dW_t$$

which is compact form for

$$G_{t+\delta t} = G_t + \int_t^{t+\delta t} A\left(G_s, s\right) ds + \int_t^{t+\delta t} B\left(G_s, s\right) dW_s,$$

by

$$G_{t+\delta t} \sim G_t + A\left(G_t, t\right) \delta t + B\left(G_t, t\right) \sqrt{\delta t}\phi + B\left(G_t, t\right) \frac{\partial}{\partial G_t} B\left(G_t, t\right) \cdot \frac{1}{2}\left(\phi^2 - 1\right) \delta t.$$

Note: We use the same value of the random number $\phi \sim N\left(0, 1\right)$ in both of the expressions

$$B\left(G_t, t\right) \sqrt{\delta t}\phi$$

and

$$B\left(G_t, t\right) \frac{\partial}{\partial G_t} B\left(G_t, t\right) \cdot \frac{1}{2}\left(\phi^2 - 1\right) \delta t.$$

The error of the Milstein scheme is $O(\delta t)$ which makes it better than the Euler-Maruyama method which is $O\left(\delta t^{1/2}\right)$. The Milstein makes use of Itô's lemma to increase the accuracy of the approximation by adding the second order term.

Some texts express the scheme in difference form. So a SDE written

$$dY_t = A(Y_t, t)\,dt + B(Y_t, t)\,dW_t$$

can be discretized as

$$Y_{i+1} = Y_i + A\Delta t + B\Delta W_t + \tfrac{1}{2}B\frac{\partial B}{\partial Y_i}\left((\Delta W_t)^2 - \Delta t\right)$$

Applying Milstein to the earlier example of GBM

$$dS_t = rS_t dt + \sigma S_t dW_t$$

where

$$\begin{aligned} A(S_t, t) &= rS_t \\ B(S_t, t) &= \sigma S_t \end{aligned}$$

gives

$$\begin{aligned} S_{t+\delta t} &\sim S_t + rS_t\delta t + \sigma S_t\sqrt{\delta t}\phi + \sigma S_t\frac{\partial}{\partial S_t}\sigma S_t \cdot \tfrac{1}{2}\sigma^2\left(\phi^2-1\right)\delta t \\ &= S_t\left(1 + r\delta t + \sigma\phi\sqrt{\delta t} + \tfrac{1}{2}\sigma^2\left(\phi^2-1\right)\delta t\right) \end{aligned}$$

As another example, the CIR model for the spot rate is

$$dr_t = (\eta - \gamma r_t)\,dt + \sqrt{\alpha r_t}dW_t.$$

So identifying

$$\begin{aligned} A(r_t, t) &= \eta - \gamma r_t \\ B(r_t, t) &= \sqrt{\alpha r_t} \end{aligned}$$

and substituting into the Milstein scheme gives

$$\begin{aligned} r_{t+\delta t} &\sim r_t + (\eta - \gamma r_t)\,\delta t + \sqrt{\alpha r_t \delta t}\phi + \sqrt{\alpha r_t}\frac{\partial}{\partial r_t}\sqrt{\alpha/r_t} \cdot \tfrac{1}{2}\left(\phi^2-1\right)\delta t \\ &= r_t + (\eta - \gamma r_t)\,\delta t + \sqrt{\alpha r_t \delta t}\phi + \tfrac{1}{4}\alpha\left(\phi^2-1\right)\delta t. \end{aligned}$$

Derivation of Milstein

Recall if

$$dG_t = A\left(G_t, t\right) dt + B\left(G_t, t\right) dW_t,$$

where A, B only depend on G, not t directly and $F = F\left(G_t\right)$ then Itô gives

$$dF = \left(A\frac{dF}{dG} + \tfrac{1}{2} B^2 \frac{d^2F}{dG^2} \right) dt + B\frac{dF}{dG} dW_t$$

Now consider a GBM

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

Put

$$\mu_t = \mu\left(S_t\right); \quad \sigma_t = \sigma\left(S_t\right)$$

to give

$$dS_t = \mu_t dt + \sigma_t dW_t;$$

which in integral form is

$$S_{t+\delta t} = S_t + \int_t^{t+\delta t} \mu_s ds + \int_t^{t+\delta t} \sigma_s dW_s, \tag{2}$$

We want to improve the accuracy of discreatization by considering expansions of coefficients $\mu_t,\ \sigma_t$ using Itô. Here we note the coefficients are functions of S and do not depend directly on t. To minimize the amount of working, primed variables $' \equiv \frac{d}{dS}$ are used to denote differentiation w.r.t. S. Then by Itô

$$\begin{aligned} d\mu_t &= \left(\mu_t\mu_t' + \tfrac{1}{2}\sigma_t^2\mu_t''\right)dt + \left(\sigma_t\mu_t'\right)dW_t \\ d\sigma_t &= \left(\mu_t\sigma_t' + \tfrac{1}{2}\sigma_t^2\sigma_t''\right)dt + \left(\sigma_t\sigma_t'\right)dW_t \end{aligned}$$

The integral form of the two SDEs above at time s such that $t < s < t + dt$

$$\begin{aligned} \mu_s &= \mu_t + \int_t^s \left(\mu_u\mu_u' + \tfrac{1}{2}\sigma_u^2\mu_u''\right)du + \int_t^s \left(\sigma_u\mu_u'\right)dW_u \\ \sigma_s &= \sigma_t + \int_t^s \left(\mu_u\sigma_u' + \tfrac{1}{2}\sigma_u^2\sigma_u''\right)du + \int_t^s \left(\sigma_u\sigma_u'\right)dW_u. \end{aligned}$$

Substituting for μ_s and σ_s in (2) gives

$$\begin{aligned} S_{t+\delta t} &= S_t + \\ &\int_t^{t+\delta t} \left(\mu_t + \int_t^s \left(\mu_u \mu_u' + \tfrac{1}{2}\sigma_u^2 \mu_u'' \right) du + \int_t^s \left(\sigma_u \mu_u' \right) dW_u \right) ds + \\ &\int_t^{t+\delta t} \left(\sigma_t + \int_t^s \left(\mu_u \sigma_u' + \tfrac{1}{2}\sigma_u^2 \sigma_u'' \right) du + \int_t^s \left(\sigma_u \sigma_u' \right) dW_u \right) dW_s. \end{aligned}$$

Now look at the orders

$$\begin{aligned} dsdu &= O\left(dt^2\right) && \text{(a)} \\ dsdW_u &= O\left(dt^{3/2}\right) && \text{(b)} \\ dW_u dW_s &= O\left(dt\right) && \text{(c)} \end{aligned}$$

therefore we can ignore double integrals of type (a), (b). This gives

$$\begin{aligned} S_{t+\delta t} &\approx S_t + \int_t^{t+\delta t} \mu_t ds + \int_t^{t+\delta t} \sigma_t dW_s + \int_t^{t+\delta t} \int_t^s \left(\sigma_u \sigma_u' \right) dW_u dW_s \\ &= S_t + \mu_t \delta t + \sigma_t \Delta W_t + \int_t^{t+\delta t} \int_t^s \left(\sigma_u \sigma_u' \right) dW_u dW_s \qquad (3) \end{aligned}$$

Now focus on approximating the double integral

$$\begin{aligned}\int_t^s \left(\sigma_u \sigma_u'\right) dW_u &= \sigma_t \sigma_t' \int_t^{t+\delta t} \left(W_s - W_t\right) dW_s \\ &= \sigma_t \sigma_t' \left(\int_t^{t+\delta t} W_s dW_s - W_t dW_s \right) \qquad (4)\end{aligned}$$

We know from earlier work using the stochastic integral formula that

$$\int_t^{t+\delta t} W_s dW_s = \frac{1}{2} W_{t+\delta t}^2 - \frac{1}{2} W_t^2 - \frac{1}{2}\delta t$$

and

$$\begin{aligned}\left(\int_t^{t+\delta t} W_t dW_s \right) &= W_t \int_t^{t+\delta t} dW_s \\ &= W_t \left(W_{t+\delta t} - W_t\right) = W_t W_{t+\delta t} - W_t^2\end{aligned}$$

Putting these in the integral term of expression (4)

$$\begin{aligned}\int_t^{t+\delta t} W_s dW_s - W_t dW_s &= \frac{1}{2}W_{t+\delta t}^2 - \frac{1}{2}W_t^2 - \frac{1}{2}\delta t - W_t W_{t+\delta t} + W_t^2 \\ &= \frac{1}{2}\left(W_{t+\delta t}^2 + W_t^2 - 2W_t W_{t+\delta t} - \delta t\right) \\ &= \frac{1}{2}\left(W_{t+\delta t} - W_t\right)^2 - \frac{1}{2}\delta t \\ &= \frac{1}{2}\left(\Delta W_t\right)^2 - \frac{1}{2}\delta t\end{aligned}$$

So (4) becomes

$$\sigma_t \sigma_t' \left(\int_t^{t+\delta t} W_s dW_s - W_t dW_s\right) = \sigma_t \sigma_t' \times \frac{1}{2}\left(\phi^2 - 1\right)\delta t$$

and we are able to write the earlier expression (3) as

$$\begin{aligned} S_{t+\delta t} &\approx S_t + \int_t^{t+\delta t} \mu_t ds + \int_t^{t+\delta t} \sigma_t dW_s + \int_t^{t+\delta t} \int_t^s \left(\sigma_u \sigma_u'\right) dW_u dW_s \\ &= S_t + \mu_t \delta t + \sigma_t \Delta W_t + \sigma_t \sigma_t' \times \frac{1}{2}\left(\phi^2 - 1\right)\delta t \\ &= S_t \left(1 + \mu \delta t + \sigma \phi \sqrt{\delta t} + \sigma^2 \times \frac{1}{2}\left(\phi^2 - 1\right)\delta t\right). \end{aligned}$$

To conclude, a SDE for the process Y_t

$$dY_t = A\left(Y_t, t\right) dt + B\left(Y_t, t\right) dW_t$$

can be discretized using Milstein as

$$Y_{i+1} = Y_i + A\delta t + B\phi\sqrt{\delta t} + \tfrac{1}{2} B \frac{\partial B}{\partial Y_i}\left(\phi^2 - 1\right)\delta t,$$

where $\frac{1}{2}\left(\phi^2 - 1\right)\delta t$ is the **Milstein correction term.** The same random number $\phi \sim N\left(0, 1\right)$ is used per time-step.

Let us remind ourselves of some basic statistics terminology:

An *estimator* is a rule for calculating an estimate for a given estimate (e.g. some statistical parameter) based on observed data. As a simple example consider a fixed set of n i.i.d observations $\{x_i\}_{1 \leq i \leq n}$ from a given distribution. Then the sample mean

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

is an **estimator** for the population mean μ.

In general an estimator $\widehat{\theta}$ is an unbiased estimator of θ if

$$\mathbb{E}\left[\widehat{\theta}\right] = \theta.$$

In other words in the average we get to the correct value for our estimate. So in the earlier example Now define the *bias* of an estimator $\widehat{\theta}$ as

$$B\left(\widehat{\theta}\right) \equiv \mathbb{E}\left[\widehat{\theta}\right] - \theta.$$

We can write

$$\begin{aligned}\widehat{\theta} - \theta &= \left(\widehat{\theta} - \mathbb{E}\left[\widehat{\theta}\right]\right) + \left(\mathbb{E}\left[\widehat{\theta}\right] - \theta\right) \\ &= \left(\widehat{\theta} - \mathbb{E}\left[\widehat{\theta}\right]\right) + B\left(\widehat{\theta}\right)\end{aligned}$$

hence an estimator for which $B = 0$ is an unbiased estimator.

So in the example of the mean above, in the average we want to get the correct value for our estimate so that

$$\mathbb{E}\left[\widehat{\mu}\right] = \mu,$$

i.e. the estimate $\widehat{\mu}$ is said to be unbiased if its expected value $\mathbb{E}\left[\widehat{\mu}\right]$ is equal to its theoretical value, μ. The bias which was defined as the difference of the expecetd value and the true value becomes

$$B\left(\widehat{\mu}\right) \equiv \mathbb{E}\left[\widehat{\mu}\right] - \mu.$$

Strong Law of Large Numbers (SLLN): Consider the earlier sequence of i.i.d random numbers with

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Also assume the first moment $\mathbb{E}[X]$ is finite. Then the SLLN asserts that $\overline{X}$ converges **a.s.** to $\mathbb{E}[X]$, i.e.

$$\mathbb{P}\left(\lim_{n\to\infty} \overline{X} = \mathbb{E}[X]\right) = 1.$$

There are two properties of a *good* estimator $\widehat{\theta}$

1. $\widehat{\theta}$ is unbiased. That is $\mathbb{E}\left[\widehat{\theta}\right] = \theta$

2. $\widehat{\theta}$ is consistent. That is $\widehat{\theta} \longrightarrow \theta$ with probability 1.This follows from the Strong Law of Large Numbers.

Monte-Carlo methods are centred on evaluating definite integrals as expectations (or averages). Before studying this in greater detail, we consider the simple problem of estimating expectations of functions of uniformly distributed random numbers.

Motivating Example: Estimate $\theta = \mathbb{E}\left[e^{U^2}\right]$, where $U \sim U(0,1)$.

We note that $\mathbb{E}\left[e^{U^2}\right]$ can be expressed in integral form, i.e.

$$\mathbb{E}\left[e^{U^2}\right] = \int_0^1 e^{x^2} p(x)\, dx$$

where $p(x)$ is the density function of a $U(0,1)$

$$p(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

hence

$$\mathbb{E}\left[e^{U^2}\right] = \int_0^1 e^{x^2} dx.$$

This integral does not have an analytical solution. The theme of this section is to consider solving numerically, using simulations. We use the Monte Carlo simulation procedures:

1. Generate a sequence $U_1, U_2, ..., U_n \sim U(0,1)$ where U_i are i.i.d (independent and identically distributed)

2. Compute $Y_i = e^{U_i^2}$ $(i = 1, ..., n)$

3. Estimate θ by

$$\begin{aligned} \widehat{\theta}_n &\equiv \frac{1}{n}\sum_{i=1}^{n} Y_i \\ &= \frac{1}{n}\sum_{i=1}^{n} e^{U_i^2} \end{aligned}$$

   i.e. use the sample mean of the $e^{U_i^2}$ terms.

Why is this a good procedure? That is, why is $\widehat{\theta}_n$ a good estimator of θ?

Recall there are two properties that need to be a satisfied

1. $\widehat{\theta}_n$ is unbiased. So we need to show $\mathbb{E}\left[\widehat{\theta}_n\right] = \theta$.
   We know

$$\begin{aligned}
\mathbb{E}\left[\widehat{\theta}_n\right] &= \mathbb{E}\left[\frac{\sum_{i=1}^{n} e^{U_i^2}}{n}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[e^{U_i^2}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[e^{U^2}\right] = \frac{1}{n}\times n\mathbb{E}\left[e^{U^2}\right] \\
&= \mathbb{E}\left[e^{U^2}\right] = \theta.
\end{aligned}$$

2. $\widehat{\theta}_n$ is consistent, i.e. $\widehat{\theta}_n \longrightarrow \theta$ with probability 1 as $n \longrightarrow \infty$.

$$\begin{aligned}
\lim_{n\to\infty} \widehat{\theta}_n &= \lim_{n\to\infty} \frac{\sum_{i=1}^{n} e^{U_i^2}}{n} \\
&= \lim_{n\to\infty} \frac{\sum_{i=1}^{n} Y_i}{n}, \text{ where } Y_i \equiv e^{U_i^2} \\
&= \mathbb{E}[Y] \text{ by the Strong Law of large numbers} \\
&= \mathbb{E}\left[e^{U^2}\right] \text{ since } Y \equiv e^{U^2} \\
&= \theta.
\end{aligned}$$

Monte Carlo Integration

When a closed form solution for evaluating an integral is not available, numerical techniques are used. The purpose of Monte Carlo schemes is to use simulation methods to approximate integrals in the form of expectations.

Suppose $f(\cdot)$ is some function such that $f : [0,1] \rightarrow \mathbb{R}$. The basic problem is to evaluate the integral

$$I = \int_0^1 f(x)\, dx$$

Consider e.g. the earlier problem $f(x) = e^{x^2}$, for which an analytical solution cannotbe obtained.
Note that if $U \sim U(0,1)$ then

$$\mathbb{E}[f(U)] = \int_0^1 f(u)\, p(u)\, du$$

where the density $p(u)$ of a uniformly distributed random variable $U(0,1)$ is given earlier. Hence

$$\begin{aligned}\mathbb{E}[f(U)] &= \int_0^1 f(u)\,p(u)\,du \\ &= I.\end{aligned}$$

So the problem of estimating I becomes equivalent to the exercise of estimating $\mathbb{E}[f(U)]$ where $U \sim U(0,1)$.

Very often we will be concerned with an arbitrary domain, other than $[0,1]$. This simply means that the initial part of the problem will involve seeking a transformation that converts $[a,b]$ to the domain $[0,1]$. We consider two fundamental cases.

1. Let $f(\cdot)$ be a function s.t. $f : [a,b] \to \mathbb{R}$ where $-\infty < a < b < \infty$. The problem is to evaluate the integral

$$I = \int_a^b f(x)\,dx.$$

In this case consider the following substitution

$$y = \frac{x - a}{b - a}$$

which gives $dy = dx/(b-a)$. This gives

$$\begin{aligned} I &= (b-a)\int_0^1 f(y \times (b-a) + a)\, dy \\ &= (b-a)\,\mathbb{E}[f(U \times (b-a) + a)] \end{aligned}$$

where $U \sim U(0,1)$. Hence I has been expressed as the product of a constant and expected value of a function of a $U(0,1)$ random number; the latter can be estimated by simulation.

2. Let $g(\cdot)$ be some function s.t. $g : [0,\infty) \to \mathbb{R}$ where $-\infty < a < b < \infty$. The problem is to evaluate the integral

$$I = \int_0^\infty g(x)\, dx,$$

provided $I < \infty$. So this is the area under the curve $g(x)$ between 0 and ∞. In this case use the following substitution

$$y = \frac{1}{1+x}$$

which is equivalent to $x = -1 + \frac{1}{y}$. This gives

$$\begin{aligned} dy &= -dx/(1+x)^2 \\ &= -y^2 dx. \end{aligned}$$

The resulting problem is

$$\begin{aligned} I &= \int_0^1 \frac{g\left(\frac{1}{y}-1\right)}{y^2} dy \\ &= \mathbb{E}\left[\frac{g\left(-1+\frac{1}{U}\right)}{U^2}\right] \end{aligned}$$

where $U \sim U(0,1)$. Hence I has again been expressed as the expected value of a function of a $U(0,1)$ random number; to be estimated by simulation.

Monte Carlo Estimation

Consider a *random vector* $\mathbf{Y} = (Y_1, ..., Y_n)^{\mathsf{T}} \in \mathbb{R}^n$ and a function $h(\cdot)$ s.t.

$$h : \mathbb{R}^n \longrightarrow \mathbb{R}$$

The aim is to estimate $\theta = \mathbb{E}[h(\mathbf{Y})]$. Clearly we want $\mathbb{E}[h(\mathbf{Y})] < \infty$.

$\mathbf{Y}$ could represent the values of a stochastic processes at different points in time. As a particular example suppose Y_i is a stock price at time i, with $h(\cdot)$ defined by

$$h(\mathbf{Y}) = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

So then θ is the expected average value of the stock price.

The MC algorithm for the estimation of θ can be written

$$\begin{aligned} &\textbf{for } i = 1 \textbf{ to } n \\ &\qquad \textbf{simulate } Y_i \\ &\qquad \textbf{set } h_i = h\left(Y_i\right) \\ &\textbf{set } \widehat{\theta}_n = \frac{h_1 + h_2 + ... + h_n}{n} \end{aligned}$$

A note on computation: If n is large, a sensible step would be to keep track of $\sum_i h_i$ within the for loop so as not to store each value of h_i.

So why is $\widehat{\theta}$ a good estimator? Because of the two reasons considered earlier.

1. $\widehat{\theta}$ **is unbiased**. That is

$$\mathbb{E}\left[\widehat{\theta}\right] = \frac{1}{n}\mathbb{E}\left[\sum_i h_i\right] = \frac{1}{n}\mathbb{E}\left[\sum_i h\left(Y_i\right)\right] = \frac{1}{n} n\theta = \theta.$$

2. $\widehat{\theta}$ **is consistent.** That is

$$\widehat{\theta}_n \longrightarrow \theta \text{ with probabilty 1 as } n \longrightarrow \infty.$$

This follows from the Strong Law of Large Numbers (SLLN).

Example: Describe a Monte Carlo algorithm for estimating

$$\theta = \int_0^\infty e^{-x^3} dx.$$

To estimate θ requires a change of variable (and limits of integration) since we are working with $\mathbf{Unif}_{[0,1]}$. Consider the transformation

$$\begin{aligned} x &= \frac{1-y}{y} \\ dx &= -\frac{1}{y^2} dy \\ \int_0^\infty f(x)\, dx &\longrightarrow \int_1^0 F(y)\, dy \end{aligned}$$

So

$$\begin{aligned}\theta &= \int_0^\infty e^{-x^3}dx = \int_1^0 e^{-(-1+1/y)^3}\left(-\frac{1}{y^2}\right)dy \\ &= \int_0^1 \frac{1}{y^2}e^{-(-1+1/y)^3}dy\end{aligned}$$

where the final integral is an expectation, written

$$\mathbb{E}\left[\frac{1}{U^2}e^{-(-1+1/U)^3}\right]$$

of a random variable $U \sim U(0,1)$.

The Monte Carlo algorithm now becomes

1. Simulate $\{U_i\}_{i=1,.....,N} \sim U(0,1)$

2. Calculate $X_i = \frac{1}{U_i^2}e^{-(-1+1/U_i)^3}$ for each $i = 1, ..., N$

3. Put $\widehat{\theta} = \frac{1}{N} \sum_{n=1}^{N} X_i$

As with any numerical scheme, the size of the associated errors is a chief concern. The Central Limit Theorem (CLT) describes the statistical properties of the errors involved in Monte Carlo integration. Assuming we are sampling from a distribution with a finite second moment, the CLT asserts that

$$\lim_{N \longrightarrow \infty} \varepsilon_N (f) \sim \frac{1}{\sqrt{N}} \phi \sigma_f$$

where $\phi \sim N(0, 1)$ is a standard normal and σ_f the standard deviation of f.

Transformation Methods

Most programming languages have random number generators that produce uniformly distributed random numbers. Then applying various transformations,

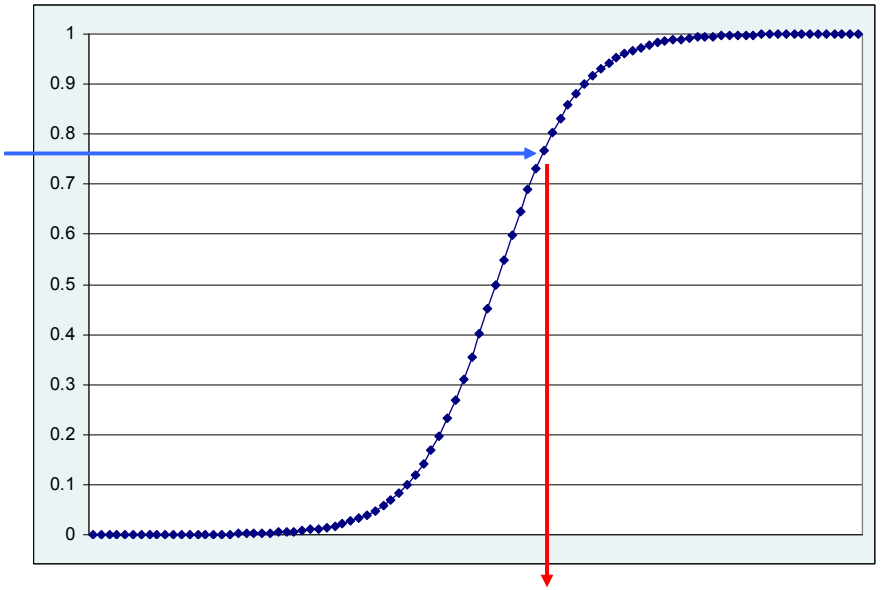
these can be conveniently converted to non-uniformly distributed random numbers. We know that given a non-uniform random variable with density, $p(x)$ its expectation and error in turn are

$$\begin{aligned}\mathbb{E}[f(x)] &= \int_{\mathbb{R}} f(x)\, p(x)\, dx \\ \varepsilon_N(f) &= \int_{\mathbb{R}} f(x)\, p(x)\, dx - \frac{1}{N}\sum_{n=1}^{N} f(x_n)\end{aligned}$$

The CLT gives

$$\begin{aligned}\varepsilon_N(f) &\sim \frac{1}{\sqrt{N}}\phi\sigma_f \\ \sigma_f^2 &= \int_{\mathbb{R}} \left(f(x) - \overline{f}\right)^2 dx\end{aligned}$$

Earlier we discussed how to create random variables following $N(0,1)$ in excel using **NORMSINV(RAND())**. This represents the inverse cumulative density function with a uniformly distributed RV $U(0,1)$ as the function parameter.



Suppose y is a RV which is $U(0,1)$ and is to be converted into another variable x with pdf $p(x)$. If its CDF $F(x)$ is defined as

$$F(x) = \int_{-\infty}^{x} p(s)\, ds$$

then we wish to invert this to obtain

$$x = F^{-1}(y) : y \sim U(0,1)$$

in a computationally efficient manner, assuming of course that F^{-1} exists. Recall that the **NORMSINV()** function is an accurate but slow mode of numerically inverting the integral.

Justification: To turn a $y \sim U(0,1)$ RV into a random variable with density $p(x)$, we know the CDF $F(x)$ is

$$F(x) = \int_{-\infty}^{x} p(s)\, ds.$$

Introduce $x = X(y)$ where X is to be determined. We know

$$\mathbb{E}[f(x)] = \mathbb{E}_{U(0,1)}[f(X(y))]$$

which in integral form

$$\int f(x)\, p(x)\, dx = \int f(X(y))\, dy$$

The second integral becomes (using $x = X(y)$)

$$\int f(x)\, p(x)\, dx = \int f(x)\, \frac{dy}{dx} dx$$

where $\frac{dx}{dy} = X'(y) = X_y$ so $\frac{dy}{dx} = X_y^{-1}$.
Thus

$$p(x) = \frac{dy}{dx} = X_y^{-1}$$

hence $p(x)\,dx = dy$.
This gives

$$F(X(y)) = y$$

i.e. $X(y) = F^{-1}(y)$.
In summary, to transform a $U(0,1)$ RV to x from a density $p(x)$ by the inverse function of the CDF F^{-1}

$$x = F^{-1}(y) : Y \sim U(0,1)$$

we are of course assumming that F^{-1} exists. We will only use this method if computing F^{-1} is indeed practical.
Consider the following simple case of generating x from a unit Cauchy distribution $p(x) = \frac{\pi^{-1}}{1+x^2}$.

Firstly write

$$\begin{aligned} F(x) &= \frac{1}{\pi}\int_{-\infty}^{x} \frac{1}{1+s^2} ds \\ &= \frac{1}{\pi}\left(\arctan x + \frac{\pi}{2}\right) \end{aligned}$$

upon rearranging we have

$$\begin{aligned} F(x) - \frac{1}{2} &= \frac{1}{\pi}\arctan x \\ x &= \tan\left(\pi\left(F(x) - \frac{1}{2}\right)\right) \end{aligned}$$

i.e.

$$\begin{aligned} x &= F^{-1}(y) \\ &= \tan\left(\pi\left(y - \frac{1}{2}\right)\right). \end{aligned}$$

As a second example we wish to generate from a Normal distribution. If $p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, then the CDF $N(x)$ is given by

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$$

and related to the error function

$$\text{erf}\,(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2/2} dt$$

through

$$N\,(x) = \frac{1}{2} + \frac{1}{2}\,\text{erf}\left(x/\sqrt{2}\right).$$

So to sample a standard normal x, using a uniform variable $y \sim U\,(0,1)$, write

$$y = \frac{1}{2} + \frac{1}{2}\,\text{erf}\left(x/\sqrt{2}\right)$$

and rearrange to get

$$x = \sqrt{2}\,\text{erf}^{-1}\,(2y-1)\,.$$

Numerical Algorithms for evaluating $N^{-1}\,(x)$ are available in some languages. In excel this is NORMSINV(RAND ()) which we have already seen.

We need a numerical technique for the conversion. Most programming languages generate uniformly distributed random variables over 0 and 1. How can

these be transformed to standard normals $N(0,1)$?

The Box Müller Method

Recall the CDF for the standardized normal distribution is defined as

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-s^2/2} ds.$$

In this technique the inefficient inversion process of $N(x)$ is not required as the random number $x \sim N(0,1)$ can be directly captured using the following algorithm:

1. generate two independent uniformly distributed random variables $y_1, y_2 \overset{\text{i.i.d}}{\sim} U(0,1)$

2. compute $x_1, x_2 \sim N(0,1)$ by

$$\begin{aligned} x_1 &= \sqrt{-2\log y_1}\cos(2\pi y_2) \\ x_2 &= \sqrt{-2\log y_1}\sin(2\pi y_2). \end{aligned}$$

which are independent standard varaibles.

The procedure underlying this technique is as follows.

Consider the joint distribution of two independent normal variables $(X,Y) \overset{\text{i.i.d}}{\sim} N(0,1)$ i.i.d given by

$$\begin{aligned} F(x,y) &= \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-x'^2/2} dx' \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-y'^2/2} dy' \\ &= \frac{1}{2\pi} \int_{-\infty}^{y} \int_{-\infty}^{x} e^{-\frac{1}{2}\left(x'^2+y'^2\right)} dx' dy' \end{aligned}$$

$$\Phi(x_1,x_2) = \frac{1}{2\pi}\exp\left(-\frac{1}{2}\left(x_1^2+x_2^2\right)\right).$$

Now, define random variables R, θ as

$$R = \sqrt{X^2 + Y^2};\ \theta = \arctan\frac{Y}{X} \text{ with } \theta \in [0, 2\pi]$$

$$X = R\cos\theta;\ Y = R\sin\theta$$

Then express $F(x, y)$ using r, θ and $dxdy = rdrd\theta$

$$F(x, y) = F(r, \theta) = \frac{1}{2\pi}\int_0^\theta \int_0^r e^{-\frac{1}{2}r'^2} r' dr' d\theta'$$

where we know $\int_0^r e^{-\frac{1}{2}r'^2} r' dr' = 1 - e^{-\frac{1}{2}r^2}$. Hence the double integral simplifies to

$$\begin{aligned}
\frac{1}{2\pi}\int_0^\theta e^{-\frac{1}{2}r'^2} d\theta' &= \frac{1}{2\pi}\left(1 - e^{-\frac{1}{2}r^2}\right)\theta'\Big|_0^\theta \\
&= \left(\frac{\theta}{2\pi}\right)\left(1 - e^{-\frac{1}{2}r^2}\right) \\
&= F(r)\,F(\theta).
\end{aligned}$$

Now draw $U_1, U_2 \overset{\text{i.i.d}}{\sim} U(0,1)$. Then we say as follows

$$F(\theta) \leq u_1,\ F(r) \leq u_2.$$

Since

$$U_2 \in [0,1] \sim U(0,1)$$

we can say

$$1 - U_2 \in [0,1] \sim U(0,1).$$

Then

$$F(\theta) \leq u_1;\ F(r) \leq 1 - u_2$$

Therefore we can get $u_1,\ u_2$ as follows. Firstly

$$u_1 = \frac{\theta}{2\pi} \Rightarrow \theta = 2\pi u_1.$$

Secondly

$$\begin{aligned} 1 - u_2 &= 1 - e^{-\frac{1}{2}r^2}, \\ u_2 &= e^{-\frac{1}{2}r^2} \Rightarrow r = \sqrt{-2\log u_2} \end{aligned}$$

Now we can generate X, Y using $U_1,\ U_2$

$$\begin{aligned} X &= R\cos\theta = \sqrt{-2\log U_2}\cos 2\pi U_1 \\ Y &= R\sin\theta = \sqrt{-2\log U_2}\sin 2\pi U_1 \end{aligned}$$

The disadvantage with this method lies in the computation of the trigonometric and transcendental functions sin, cos and log . This leads on to a more efficient scheme which employs an *acceptance-rejection* method.

Polar Marsaglia Method

1. Generate $U_1, U_2 \overset{\text{i.i.d}}{\sim} U(0,1)$

2. Set

$$\left.\begin{array}{l} V_1 = 2U_1 - 1 \\ V_2 = 2U_2 - 1 \end{array}\right\} \quad R = V_1^2 + V_2^2$$

   where $(V_1, V_2) \sim U(-1,1)$.

3. While $R \leq 1$, draw $U \sim U(0,1)$ and returnset

$$X = \sqrt{\frac{-2\log U}{R}} V_1 \; ; \; Y = \sqrt{\frac{-2\log U}{R}} V_2$$

   Else go back to step 1. Can also use U in place of R.

Here

$$\cos(2\pi U_1) \text{ is replaced with } \frac{V_1}{\sqrt{R}}$$

$$\sin(2\pi U_1) \text{ is replaced with } \frac{V_2}{\sqrt{R}}$$

Then $X, Y \sim N(0,1)$. The probability of of S being accepted (i.e. area of unit circle to area of square) is

$$\mathbb{P}(S \leq 1) = \frac{\text{Area of circle}}{\text{Area of square}} = \frac{\pi \times 1^2}{2 \times 2} = \frac{\pi}{4} \approx 0.785$$

which means less than 21.5% of uniform deviates $V_1,\ V_2$ are rejected for which $S > 1$. This is far more efficient than the BM Method.

Generating Correlated Normal Random Variables

Earlier we looked at how to obtain correlated random variables given a pair of uncorrelated ones that follow $N(0,1)$. Recall, if X, Y are random variables then the covariance written σ_{XY} is

$$\begin{aligned} \mathbf{Cov}(X,Y) &= \sigma_{XY} \\ &= \mathbb{E}[X,Y] - \mathbb{E}[X]\,\mathbb{E}[Y] \end{aligned}$$

and the correlation of X and Y written ρ_{XY}

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y},$$

where $-1 \leq \rho_{XY} \leq 1$ (due to Cauchy-Schwartz for random variables).

If X, Y are independent then $\rho_{XY} = 0$, the converse is not generally true. This is a theorem which most students encounter for the first time in linear

algebra. Let's start off with the version for random variables (RVs) X and Y, then the Cauchy-Schwartz inequality is

$$\left[\mathbb{E}\left[XY\right]\right]^2 \leq \mathbb{E}\left[X^2\right]\mathbb{E}\left[Y^2\right].$$

We know that the covariance of X, Y is

$$\sigma_{XY} = \mathbb{E}\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right]$$

If we put

$$\begin{aligned}\mathbb{V}\left[X\right] &= \sigma_X^2 = \mathbb{E}\left[\left(X-\mu_X\right)^2\right] \\ \mathbb{V}\left[Y\right] &= \sigma_Y^2 = \mathbb{E}\left[\left(Y-\mu_Y\right)^2\right].\end{aligned}$$

From Cauchy-Schwartz we have

$$\left(\mathbb{E}\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right]\right)^2 \leq \mathbb{E}\left[\left(X-\mu_X\right)^2\right]\mathbb{E}\left[\left(Y-\mu_Y\right)^2\right]$$

or we can write

$$\sigma_{XY}^2 \leq \sigma_X^2\sigma_Y^2$$

Divide through by $\sigma_X^2 \sigma_Y^2$

$$\frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2} \leq 1$$

and we know that the left hand side above is ρ_{XY}^2, hence

$$\rho_{XY}^2 = \frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2} \leq 1$$

and since ρ_{XY} is a real number, this implies $|\rho_{XY}| \leq 1$ which is the same as

$$-1 \leq \rho_{XY} \leq +1.$$

Suppose now that $\mathbf{Z} = (z_1, z_2, ..., z_n)^{\mathsf{T}}$ is a random vector where each $z_i \sim N(0, 1)$ and are i.i.d.

Then define the *covariance matrix* $\mathbf{\Sigma}$ of $\mathbf{Z}$ as the $n \times n$ matrix that has element

at (i, j) given by

$$\begin{aligned} \Sigma_{ij} &= \mathbf{Cov}\left(Z_i, Z_j\right) \\ &= \sigma_{Z_i Z_j} \end{aligned}$$

So for example, in the case of a 3×3, we have

$$\begin{aligned} \Sigma &= \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{31}\sigma_3\sigma_1 & \rho_{32}\sigma_3\sigma_2 & \sigma_3^2 \end{pmatrix} \\ \Sigma_C &= \begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{pmatrix} \end{aligned}$$

The properties of Σ are

- It is a symmetric matrix $\Sigma^{\mathsf{T}} = \Sigma$
- The leading diagonal elements are non-negative $\Sigma_{ii} \geq 0$

- It is *positive semi-definite,* (PSD) i.e. $\forall \mathbf{y} \in \mathbb{R}^n$, $\mathbf{y}^\mathsf{T}\mathbf{\Sigma}\mathbf{y} \geq 0$.

The new vector $\mathbf{X} = (x_1, x_2, ..., x_n)^\mathsf{T}$ where $\mathbf{X} \sim \mathbf{MN}(0, \mathbf{\Sigma})$, i.e. a multivariate normal distribution. We write

$$\mathbf{X} = L\mathbf{Z}$$

such that

$$\boxed{\mathbf{\Sigma} = LL^\mathsf{T}}$$

Cholesky Decomposition

Here we perform Cholesky factorisation of a symmetric positive-definite matrix, M. Such a matrix can be written as

$$M = lDl^\mathsf{T},$$

where

l is a lower triangular matrix

D is a diagonal matrix with positive elements

So we can write

$$\begin{aligned}\mathbf{\Sigma} &= lDl^{\mathsf{T}} \\ &= \left(l\sqrt{D}\right)\left(\sqrt{D}l^{\mathsf{T}}\right) \\ &= \underbrace{\left(l\sqrt{D}\right)}_{L}\underbrace{\left(l\sqrt{D}\right)^{\mathsf{T}}}_{L^{\mathsf{T}}}\end{aligned}$$

Therefore $L = l\sqrt{D}$ satisfies $\mathbf{\Sigma} = LL^{\mathsf{T}}$. It is called the Cholesky Decomposition of $\mathbf{\Sigma}$.

Consider a 2×2 case, and write

$$\begin{aligned} \mathbf{\Sigma} &= \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_2\sigma_1 & \sigma_2^2 \end{pmatrix} = LL^{\mathsf{T}} \\ &= \begin{pmatrix} a & 0 \\ b & c \end{pmatrix} \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \end{aligned}$$

Equating elements gives

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

Now put

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = L\mathbf{Z} \\ &= \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sigma_2\sqrt{1-\rho^2} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\ &= \begin{pmatrix} \sigma_1 z_1 \\ \rho\sigma_2 z_1 + \sigma_2\sqrt{1-\rho^2} z_2 \end{pmatrix}. \end{aligned}$$

In the case of a standard normal distribution we have $\begin{pmatrix} z_1 \\ \rho z_1 + \sqrt{1-\rho^2} z_2 \end{pmatrix}$.

Relationship between derivative values and simulations

The fair value of an option is the present value of the expected payoff at expiry under the risk-neutral random walk for the underlying. (Phelim Boyle 1977)

Recall an amount of cash $M = M(t)$ in the bank grows according to

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

where $r(t)$ is the variable (risk-free) interest rate. This differential equation has solution

$$M(t) = M(T) \exp\left(-\int_t^T r(\tau)\, d\tau\right)$$

i.e. the present value (time t) of a future cash flow (time T). The exponential term is the discount factor. In the simple case of a fixed rate of interest it becomes $e^{(-r(T-t))}$.

This gives the fair price of an option V to be

$$V = \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T r(\tau) d\tau} \text{Payoff}\,(S) \right]$$

where

S =asset price, r = stochastic domestic interest rate, T = expiry, t = current time, $\mathbb{Q}$= risk neutral density.

Scheme: The Monte Carlo method consists of the following steps

1. Simulate sample paths/realizations for the underlying asset price (e.g. equities or interest rates) over the relevant time horizon, according to the risk-neutral measure. Here we use the discretized SDE

$$S_{i+1} = S_i \left(1 + r\delta t + \sigma\phi\sqrt{\delta t} \right).$$

2. Evaluate the discounted cashflows (using domestic rate of interest) of a derivative on each sample path, as determined by the structure of the security being priced.

3. Average the discounted cashflows over sample paths.

So the option price becomes

$$e^{(-r(T-t))}.\frac{1}{N}\sum_{n=1}^{N}\text{Payoff}\left(S\right)$$

The payoff for a European Call Option with strike E is $C = \max\left(S\left(T\right) - E,\ 0\right)$, where $S\left(T\right)$ is obtained from

$$S_T = S_0 e^{\left(r-0.5\sigma^2\right)T+\sigma W_T}$$

in discrete form, for each value $1 \leq n \leq N$.

Although we are not concerned with the path followed by the process $S(t)$ in getting to $S(T)$ we will nevertheless simulate this as we can price other options which are *path dependent*.

Based upon the N realizations an estimate for the price of an option becomes $\overline{C}(S,\ t)$

$$\overline{C}(S,\ t) = \frac{1}{N}\sum_{n=1}^{N} C^{(n)}(S,\ T)$$

which is equivalent to

$$e^{(-r(T-t))}.\frac{1}{N}\sum_{n=1}^{N} \mathsf{max}\left(S^{(n)}(T) - E,\ 0\right).$$

If we put $S(T) = S_T$, where

$$S_T = S_0 e^{\left(r-0.5\sigma^2\right)T+\sigma W_T}$$

then an algorithm for the estimator $\widehat{C}$ of the BS option price can be written as

$$\begin{array}{c} \textbf{set } sum = 0 \\ \textbf{for } i = 1 \textbf{ to } n \\ \quad \textbf{simulate } S_T \\ \quad \textbf{set } sum = sum + \max\left(S_T - E, 0\right) \\ \textbf{End for} \\ \textbf{set } \widehat{C} = e^{-r(T-t)}\frac{sum}{n} \end{array}$$

Now consider the example of an Asian call option with arithmetic averaging (fixed strike). Recall at expiry T, the payoff is

$$h\left(Y_1, ..., Y_n\right) = \max\left(\frac{1}{m}\sum_{i=1}^{m} S_i - E, 0\right).$$

As earlier we can write the option price as $C = \mathbb{E}^{\mathbb{Q}}\left[e^{-r(T-t)}h\left(\mathbf{Y}\right)\right]$ with the

following algorithm

set $sum = 0$
for $i = 1$ **to** n
simulate S_i
set $sum = sum + \max\left(\frac{1}{m}\sum_{i=1}^{m} S_i - E, 0\right)$
End for
set $\widehat{C} = e^{-r(T-t)}\frac{sum}{n}$

Variance Reduction

As before the idea is to estimate $\mathbb{E}[Y]$ where Y is an output random variable obtained from a simulation. We are motivated by the standard error $\varepsilon = \frac{\sigma}{\sqrt{N}}$, where N is the number of sample paths. The main idea is to reduce σ^2. Earlier the method was to simulate Y such that $\mathbb{E}[X] = \mathbb{E}[Y]$ and $\mathbb{V}[X] \leq \mathbb{V}[Y]$. The idea now is to work with the original output variable Y.

Instead of generating a sequence of Y_i random variables in an i.i.d fashion; correlation will be induced to reduce the variance.

For the simulation of an asset price, samples are drawn from a probability (normal) distribution. If these samples are generated in a fashion, which is not entirely random, but in a manner that reduces the fluctuations (i.e. volatility) of the resulting samples, computational time can be reduced considerably to obtain the desired degree of accuracy.

A similar effect can be obtained by performing suitable transformations on the function, which forms the basis of the simulation, so that dependency upon the fluctuations arising in the samples is reduced. The disadvantage is that correlations are introduced. It then becomes a choice, whether to compromise computational time over the risk of correlations being introduced. Recall that the standard error ε associated with the Monte Carlo method is

$$\varepsilon = \frac{\sigma}{\sqrt{N}}.$$

Increasing the number of sample paths generated,by increasing N, leads to a reduction in ε. In addition we are able to manipulate the variance, i.e. reduce the value of σ. For this reason a whole area of Monte Carlo, namely *variance reduction techniques* has been developed.

Antithetic Variable Technique

This method attempts to reduce the variance by introducing negative correlation between pairs of observations. The estimation of $\mathbb{E}[X]$ is the main problem, for the ouptut variable (from a simulation) X. So suppose

$$X = h(\mathbf{Z})$$

where the components of the m dimensional vector $\mathbf{Z}$ are independent. The function $h(\cdot)$ is

$$h : \mathbb{R}^m \to \mathbb{R}.$$

As a motivating example suppose X represents the payoff of a path dependent option with sampling dates $0 < t_1 < \cdots < t_{m-1} < t_m = T$. Then note that $X = h(Z_1, Z_2, \cdots, Z_m)$, for some function $h(Z_1, Z_2, \cdots, Z_m)$, where the Z_i terms are i.i.d and $N(0,1)$.

When a sequence of $U_i \sim U(0,1)$ are used to generate $Z \sim N(0,1)$, many simulations can be represented as

A very simple technique is by use of *antithetic variates*, which can reduce computational time and be implemented at no additional effort, was introduced to option pricing by Boyle (1977).

The method, which was initially used in the pricing of a European call option on a dividend paying stock, is outlined below. It is based upon the observation that if $\phi^{(n)} \sim N(0,1)$, then $-\phi^{(n)}$ also has a standard Normal distribution.

In this technique, by using the one set of random numbers generated, two estimates for an option are calculated. If ϕ_i is used to obtain $\overline{C}$, then $-\phi_i$ gives $\widehat{S}(t)$ and hence a second approximation for the option price $\widehat{C}$ where

$$\widehat{C} = e^{(-r(T-t))}.\frac{1}{N}\sum_{n=1}^{N} \max\left(\widehat{S}^{(n)}(T) - E,\ 0\right)$$

The estimate for the option C_μ is now the average of the two values, $\overline{C}$ & $\widehat{C}$, so

$$C_\mu = \frac{\overline{C} + \widehat{C}}{2}$$

The technique converges because of the symmetry of the Normal Distribution. Justification for obtaining C_μ is based upon the distribution of the antithetic variates.

The pairs $\left\{\left(\phi^{(n)}, -\phi^{(n)}\right)\right\}$are distributed more regularly than a collection of $2n$ independent samples with the sample mean over the antithetic pairs always equal to the population mean of 0. The data set has a lower variance.

Finite Difference Methods

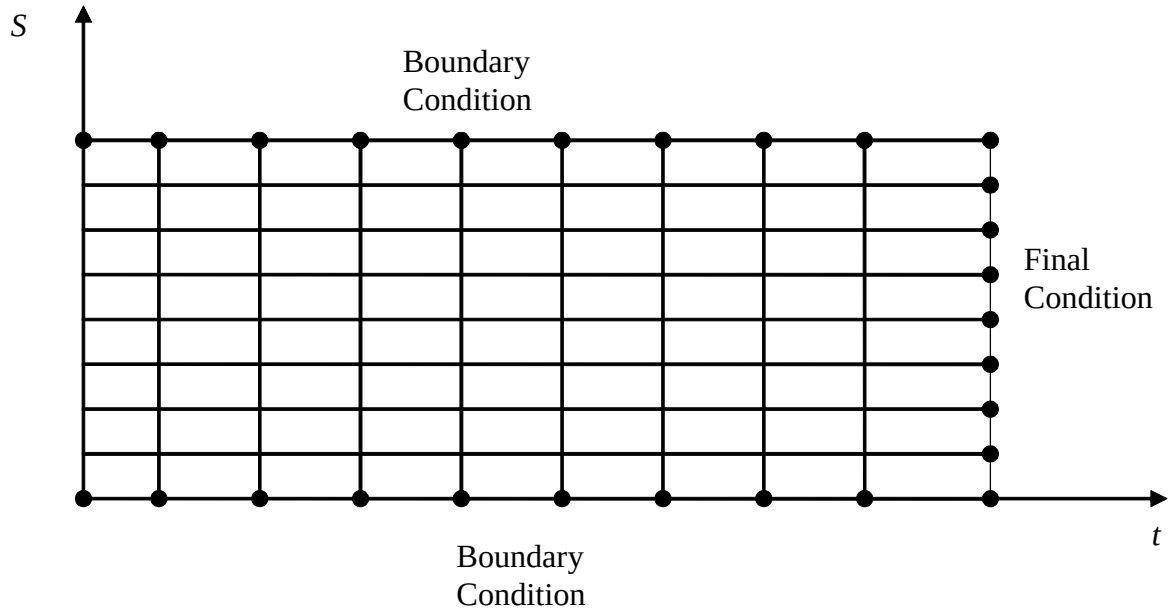
Model Problem

Consider the following Black-Scholes pricing problem for the value of a European Call Option $V = V(S,t)$:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D) S \frac{\partial V}{\partial S} - rV = 0$$

$$\begin{aligned} V(0,t) &= 0 \\ \lim_{S \to \infty} V(S,t) &\to S \\ V(S,T) &= \max(S - E, 0), \\ S &\in [0, \infty), \quad t \in [0,T] \end{aligned}$$

The problem is to be solved over the following region:



This is to be solved by a Finite Difference (FD) Scheme.

By expressing $S = n\delta S$ and $t = m\delta t$, we will obtain a difference equation for the Black-Scholes equation.

$V(S,t) = V(n\delta S,\ m\delta t) = V_n^m$. $\delta S = S^*/N$ where $S^* \gg E$ is a suitably large value of S ; $\delta t = T/M$.

Take N and M steps for S and t respectively, so

$$\begin{aligned} S &= n\delta S \qquad & 0 \leq n \leq N \\ t &= m\delta t \qquad & 0 \leq m \leq M. \end{aligned}$$

Taylor Series Approximations

We begin with the Taylor-series approximations for the 1st and 2nd order derivatives.

If $V = V(S,t)$ and S & t both change by an amount δS and δt in turn, so that $S \longrightarrow S + \delta S$ and $t \longrightarrow t + \delta t$ then the change in V can be obtained using a two-dimensional Taylor expansion

$$V(S + \delta S,\ t + \delta t) =$$

$$V(S,t) + \frac{\partial V}{\partial S}\delta S + \frac{\partial V}{\partial t}\delta t + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\delta S^2$$
$$+\frac{1}{2}\frac{\partial^2 V}{\partial t^2}\delta t^2 + \frac{\partial^2 V}{\partial S \partial t}\delta S \delta t + O\left(\delta S^3,\ \delta t^3\right)$$

Consider

$$V(S,\ t + \delta t) = V(S,t) + \frac{\partial V}{\partial t}\delta t + O\left(\delta t^2\right)$$

and rearranging gives a forward difference

$$\frac{\partial V}{\partial t} = \frac{V\left(S, t+\delta t\right) - V\left(S, t\right)}{\delta t} + O\left(\delta t\right).$$

Let us use a backward time difference

$$\frac{\partial V}{\partial t} = \frac{V\left(S, t\right) - V\left(S, t-\delta t\right)}{\delta t} + O\left(\delta t\right),$$

which becomes, in finite difference form

$$\frac{\partial V}{\partial t}\left(n\delta S,\ m\delta t\right) \sim \frac{V_n^m - V_n^{m-1}}{\delta t}.$$

We now derive approximations for derivative terms involving S. Start by considering V at $S + \delta S$ and $S - \delta S$

$$V\left(S+\delta S,\ t\right) = V\left(S,t\right) + \frac{\partial V}{\partial S}\delta S + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\delta S^2 + O\left(\delta S^3\right) \qquad (1)$$

$$V\left(S-\delta S,\ t\right) = V\left(S,t\right) - \frac{\partial V}{\partial S}\delta S + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\delta S^2 - O\left(\delta S^3\right) \qquad (2)$$

(1) − (2) gives

$$V\left(S+\delta S,\ t\right)-V\left(S-\delta S,\ t\right)=2\frac{\partial V}{\partial S}\delta S+O\left(\delta S^{3}\right)$$

which upon rearranging and using finite difference notation yields

$$\frac{\partial V}{\partial S}\left(n\delta S,\ m\delta t\right)=\frac{V_{n+1}^{m}-V_{n-1}^{m}}{2\delta S}+O\left(\delta S^{2}\right)$$

and hence giving us a scheme for the first derivative $\frac{\partial V}{\partial S}$.

(1) + (2) gives

$$V\left(S+\delta S,t\right)+V\left(S-\delta S,t\right)=2V\left(S,t\right)+\frac{\partial^{2}V}{\partial S^{2}}\delta S^{2}+O\left(\delta S^{4}\right)$$

and hence

$$\frac{\partial^{2}V}{\partial S^{2}}=\frac{V(S+\delta S,\ t)-2V(S,t)+V(S-\delta S,\ t)}{\delta S^{2}}+O\left(\delta S^{2}\right)$$

and hence a finite difference approximation for the second derivative

$$\frac{\partial^2 V}{\partial S^2}\left(n\delta S,\ m\delta t\right) = \frac{V_{n-1}^m - 2V_n^m + V_{n+1}^m}{\delta S^2} + O\left(\delta S^2\right)$$

$$\begin{array}{rcl} \frac{\partial V}{\partial t} & \sim & \frac{V_n^m - V_n^{m-1}}{\delta t}, \qquad \frac{\partial V}{\partial S} \sim \frac{V_{n+1}^m - V_{n-1}^m}{2\delta S}, \\ \frac{\partial^2 V}{\partial S^2} & \sim & \frac{V_{n-1}^m - 2V_n^m + V_{n+1}^m}{\delta S^2} \end{array} \tag{3}$$

Substituting (3) in the BSE gives

$$\begin{array}{rl} & \frac{V_n^m - V_n^{m-1}}{\delta t} + \frac{1}{2}n^2\sigma^2\left(V_{n-1}^m - 2V_n^m + V_{n+1}^m\right) + \\ & \frac{1}{2}\left(r - D\right)n\left(V_{n+1}^m - V_{n-1}^m\right) - rV_n^m \\ = & 0 \end{array}$$

and rearrange to obtain a *backward marching* scheme in time

$$\begin{aligned} V_n^{m-1} &= V_n^m + \delta t \left(\frac{1}{2} n^2 \sigma^2 \left(V_{n-1}^m - 2V_n^m + V_{n+1}^m \right) \right) \\ &\quad + \delta t \left(\frac{1}{2} (r - D)\, n \left(V_{n+1}^m - V_{n-1}^m \right) - r V_n^m \right) \\ &\equiv F\left(V_{n-1}^m,\ V_n^m,\ V_{n+1}^m \right) \end{aligned}$$

Now for the RHS collect coefficients of each variable term V, to get

$$V_n^{m-1} = \alpha_n V_{n-1}^m + \beta_n V_n^m + \gamma_n V_{n+1}^m \tag{4}$$

where

$$\begin{aligned} \alpha_n &= \frac{1}{2} \left(n^2 \sigma^2 - n\,(r - D) \right) \delta t, \qquad (5) \\ \beta_n &= 1 - \left(r + n^2 \sigma^2 \right) \delta t, \\ \gamma_n &= \frac{1}{2} \left(n^2 \sigma^2 + n\,(r - D) \right) \delta t \end{aligned}$$

(4) is a linear difference equation. We will use this to march backwards in time, i.e. given a solution at time step m we can use (4) to approximate a solution at the next time step $(m-1)$ .The difference equation (4) is not valid at the boundaries.

Boundary conditions:

At $S = 0$, i.e. $n = 0$, the BSE becomes

$$\begin{aligned} \frac{\partial V}{\partial t} &= rV \Rightarrow \\ V_0^{m-1} &= (1 - r\delta t)\, V_0^m \end{aligned}$$

This also follows from

$$\alpha_0 = 0 = \gamma_0, \quad \beta_0 = 1 - r\delta t.$$

As S becomes very large, i.e. $S \to \infty$ i.e. S^*, the probability of it becoming lower than the Exercise becomes negligible, therefore $\Delta = \Delta(t)$ only, hence $\Gamma \to 0$.

The problem arises at $n = N$. We cannot use our difference equation at the boundary, as we end up with a term V^m_{N+1}, which is not defined. So we use the gamma condition mentioned above.

We know $\Gamma \sim \dfrac{V^m_{n-1} - 2V^m_n + V^m_{n+1}}{\delta S^2} = 0$, which upon rearranging gives at $n = N$

$$V^m_{N+1} = 2V^m_N - V^m_{N-1}$$

and substituting in the difference equation gives

$$V^{m-1}_N = (\alpha_N - \gamma_N)\, V^m_{N-1} + (\beta_N + 2\gamma_N)\, V^m_N.$$

In summary, the scheme is

$$\left.\begin{array}{c} V_n^{m-1} = \alpha_n V_{n-1}^m + \beta_n V_n^m + \gamma_n V_{n+1}^m \\ M > m \geq 1; \quad 1 \leq n \leq N-1 \end{array}\right\} \text{ D.E}$$

$$\left.\begin{array}{c} V_n^M = \max\left(n\delta S - E, 0\right) \\ 0 \leq n \leq N; \end{array}\right\} \text{ Final Payoff Condition}$$

$$\left.\begin{array}{c} V_0^{m-1} = \beta_0 V_0^m \\ M \geq m \geq 1 \end{array}\right\} \text{BC at } \quad (S = 0)$$

$$\left.\begin{array}{c} V_N^{m-1} = \left(\alpha_N - \gamma_N\right) V_{N-1}^m + \left(\beta_N + 2\gamma_N\right) V_N^m \\ M \geq m \geq 1; \quad S = N\delta S \end{array}\right\} \text{BC at } \quad S^*$$

Fourier Stability (Von Neumann's) Method

A method is called step-wise unstable if for a fixed grid (i.e. δt, δS constant) there exists an initial perturbation which "blows up" as $t \to \infty$, i.e. as we march in time. Here in a backward marching scheme we have $t \to 0$ $(m \to 0)$. The question we wish to answer is "do small errors propagate along the grid and grow exponentially?". We hope not!

Assume an initial disturbance which is proportional to $\exp(in\omega)$. We therefore study the propagation of perturbations created at any given point in time.

If $\widehat{V}_n^m$ is an approximation to the exact solution V_n^m then

$$\widehat{V}_n^m = V_n^m + E_n^m$$

where E_n^m is the associated error. Then E_n^m also satisfies the difference equation (4) to give

$$E_n^{m-1} = \alpha_n E_{n-1}^m + \beta_n E_n^m + \gamma_n E_{n+1}^m.$$

Put

$$E_n^m = \overline{a}^m \exp(in\omega) \tag{6}$$

which is oscillatory of amplitude $\overline{a}$ and frequency ω. Substituting (6) into (4) gives

$$\overline{a}^{m-1} e^{(in\omega)} = \alpha_n \overline{a}^m e^{i(n-1)\omega} + \beta_n \overline{a}^m e^{in\omega} + \gamma_n \overline{a}^m e^{i(n+1)\omega}$$

which becomes

$$\overline{a}^{-1} = \alpha_n e^{-i\omega} + \beta_n + \gamma_n e^{i\omega}.$$

Now stability criteria arises from the balancing of the time dependency and diffusion terms, so that

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = 0$$

From (5) we take the following contributions

$$\alpha_n = \frac{1}{2}n^2\sigma^2\delta t,\ \beta_n = 1 - n^2\sigma^2\delta t,\ \gamma_n = \frac{1}{2}n^2\sigma^2\delta t$$

$$\begin{aligned}\overline{a}^{-1} &= \frac{1}{2}n^2\sigma^2\delta t\left(e^{i\omega}+e^{-i\omega}\right)+1-n^2\sigma^2\delta t \\ &= n^2\sigma^2\delta t\left(\cos\omega-1\right)+1.\end{aligned}$$

we have

$$\overline{a}^{-1} = 1-2n^2\sigma^2\sin^2\frac{\omega}{2}\delta t$$

For stability $\overline{a}^{-1}$ must be bounded, i.e. $\left|\overline{a}^{-1}\right| \leq 1 \Longleftrightarrow |a| \geq 1$ (as it is a backward marching scheme), i.e.

$$\left|1-2n^2\sigma^2\sin^2\frac{\omega}{2}\delta t\right| \leq 1$$

which upon simplifying we find is

$$\delta t \leq \frac{1}{\sigma^2 N^2} \tag{7}$$

so $\delta t \sim O\left(N^{-2}\right)$.

The beauty of the explicit method lies in its simplicity, both in numerical and computational terms.

However, the main disadvantage is associated with the stability criteria, given by (7) .

This condition puts severe constraints on the viability of the method. If it is not satisfied, we will observe exponentially growing oscillations in our numerical solution as we iterate backwards in time.

Given that $\delta t = O\left(\frac{1}{N^2}\right)$, we see that the accuracy can be improved by increasing the number of asset steps N. However doubling N requires the use of four times as many time-steps, to satisfy the stability condition.

Variable Parameters

There are a number of parameters in the BSE, which need not be constant. FDM can easily handle problems involving non-constant parameters. Suppose the volatility, dividend yield and interest rates are functions of asset price and time, such that the pricing equation becomes

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma\left(S,t\right)^2 S^2 \frac{\partial^2 V}{\partial S^2} + \left(r\left(t\right) - D\left(S,t\right)\right) S \frac{\partial V}{\partial S} = r\left(t\right) V.$$

The explicit FD scheme now becomes

$$V_n^{m-1} = \alpha_n^m V_{n-1}^m + \beta_n^m V_n^m + \gamma_n^m V_{n+1}^m$$

where

$$\begin{aligned}
\alpha_n &= \frac{1}{2}\left(\left(\sigma_n^m\right)^2 n^2 - \left(r_n - D_n^m\right) n\right)\delta t, \\
\beta_n &= 1 - \left(r_n + \left(\sigma_n^m\right)^2 n^2\right)\delta t, \\
\gamma_n &= \frac{1}{2}\left(\left(\sigma_n^m\right)^2 n^2 + \left(r_n - D_n^m\right) n\right)\delta t
\end{aligned}$$

Early Exercise Feature - American Options

If we can exercise an option during some time interval, before its expiry date, then the *no-arbitrage* argument tells us that the value of the option V can not be less than the payoff $P(S,t)$ during that time period, so

$$V \geq P(S,t).$$

In the explicit scheme, the early exercise constraint can be implemented in a most trivial manner. Consider the time interval T in which the option may be exercised. As we step backwards in time, the option value is computed. If this price is less than the payoff during T, it is set equal to the payoff. So at each time step, we solve the explicit scheme to obtain the option price $\overline{V}$. Then check the condition $\overline{V} < P(S,t)$? If this is true then the option price $V = P(S,t)$, else $V = U$.

This strategy of checking the early exercise constraint is called the *cutoff* method. Thus the explicit FDM can be expressed in compact form as

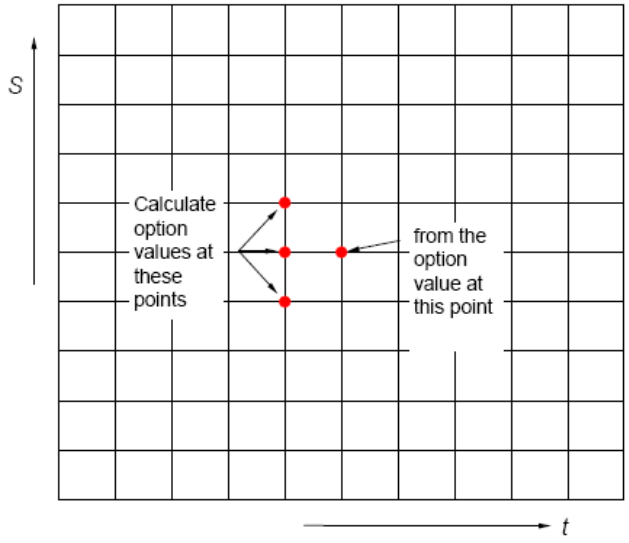
$$\begin{aligned} U_n^{m-1} &= \alpha_n V_{n-1}^m + \beta_n V_n^m + \gamma_n V_{n+1}^m \\ V_n^{m-1} &= \begin{cases} U_n^{m-1} & \text{if } U_n^{m-1} \geq P_n^{m-1} \\ P_n^{m-1} & \text{if } U_n^{m-1} < P_n^{m-1} \end{cases} \end{aligned}$$

where P_n^m is the FDA for the payoff at $(S,\ t)$, i.e. P_n^m =Payoff($n\delta S,\ m\delta t$) as opposed to simply P_n^M at time $t = T$. So we have a time dependent payoff function, defined (at each time step) for the life of the option at the time the contract is written.

Implicit Finite Difference Approximations

We now introduce the Implicit Finite Difference (FD) Scheme.

By using the same notation as in the explicit case we will obtain an implicit difference method for the Black-Scholes equation.



We begin with the Taylor-series approximations for the 1st and 2nd order derivatives, where $V(S,t) = V(n\delta S,\ m\delta t) = V_n^m$:

$$\begin{aligned}
\frac{\partial V}{\partial t} &\sim \frac{V_n^m - V_n^{m-1}}{\delta t} + O(\delta t), \\
\frac{\partial V}{\partial S} &\sim \frac{V_{n+1}^{m-1} - V_{n-1}^{m-1}}{2\delta S} + O\left(\delta S^2\right), \\
\frac{\partial^2 V}{\partial S^2} &\sim \frac{V_{n-1}^{m-1} - 2V_n^{m-1} + V_{n+1}^{m-1}}{\delta S^2} + O\left(\delta S^2\right)
\end{aligned}$$

Substituting in the BSE gives and rearranging to obtain another *backward marching* scheme in time gives the linear system

$$a_n V_{n-1}^{m-1} + b_n V_n^{m-1} + c_n V_{n+1}^{m-1} = V_n^m \tag{8}$$

where

$$\begin{aligned}
a_n &= -\frac{1}{2}\left(\sigma^2 n^2 - n(r-D)\right)\delta t,\ b_n = 1 + \left(\sigma^2 n^2 + r\right)\delta t, \\
c_n &= -\frac{1}{2}\left(\sigma^2 n^2 + n(r-D)\right)\delta t
\end{aligned} \tag{9}$$

This expression is accurate to $O\left(\delta S^2,\ \delta t\right)$.

The chief attraction of this method lies in its instability - it is stable for all values of δt and we call this scheme *unconditionally stable.*

As we know V_n^m before V_n^{m-1}, the system is implicit for the V_n^{m-1} term.

Boundary conditions:

At $S = 0$, i.e. $n = 0$, the BSE becomes

$$(1 + r\delta t)\, V_0^{m-1} = V_0^m$$

The problem again arises at $n = N$ $(S \to \infty)$. We cannot use our difference equation at the boundary, as we end up with a term V_{N+1}^{m-1}, which is not defined. So we use the gamma condition mentioned above ($\Gamma \to 0$).

As before we know

$$\Gamma \sim \frac{V_{n-1}^{m-1} - 2V_n^{m-1} + V_{n+1}^{m-1}}{\delta S^2} = 0.$$

Upon rearranging gives at $n = N$

$$V_{N+1}^{m-1} = 2V_N^{m-1} - V_{N-1}^{m-1}$$

and substituting in the difference equation gives

$$\widehat{a}_N V_{N-1}^{m-1} + \widehat{b}_N V_N^{m-1} = V_N^{m-1},$$

where

$$\widehat{a}_N = N\left(r - D\right)\delta t, \quad \widehat{b}_N = 1 - \left(N\left(r - D\right) - r\right)\delta t,$$

In summary, the scheme is:

$$\left.\begin{array}{c} a_n V_{n-1}^{m-1} + b_n V_n^{m-1} + c_n V_{n+1}^{m-1} = V_n^m \\ M \geq m \geq 1; \quad 1 \leq n \leq N-1 \end{array}\right\} \text{ D.E}$$

$$\left.\begin{array}{c} V_n^M = \max\left(n\delta S - E, 0\right) \\ 0 \leq n \leq N; \end{array}\right\} \text{ Final Payoff Condition}$$

$$\left.\begin{array}{c} \left(1 + r\delta t\right) V_0^{m-1} = V_0^m \\ M \geq m \geq 1 \end{array}\right\} \text{Boundary condition at } \quad (S = 0)$$

$$\left.\begin{array}{c} \widehat{a}_N V_{N-1}^{m-1} + \widehat{b}_N V_N^{m-1} = V_N^m \\ M \geq m \geq 1; \quad S = N\delta S \end{array}\right\} \text{Boundary condition at } \quad S^*$$

We can write the problem as a system of linear equations, called a *linear system*,

$$
\begin{aligned}
a_1 V_0^{m-1} + b_1 V_1^{m-1} + c_1 V_2^{m-1} &= V_1^m \\
a_2 V_1^{m-1} + b_2 V_2^{m-1} + c_2 V_3^{m-1} &= V_2^m \\
&\vdots \\
&\vdots \\
a_{N-1} V_{N-2}^{m-1} + b_{N-1} V_{N-1}^{m-1} + c_{N-1} V_N^{m-1} &= V_{N-1}^m
\end{aligned}
$$

$$
\begin{pmatrix}
b_0 & c_0 & \cdots & \cdots & \cdots & 0 \\
a_1 & b_1 & c_1 & 0 & & \vdots \\
0 & a_2 & b_2 & c_2 & & \vdots \\
\vdots & & \ddots & \ddots & \ddots & 0 \\
\vdots & & & a_{N-1} & b_{N-1} & c_{N-1} \\
0 & \cdots & \cdots & 0 & \widehat{a}_N & \widehat{b}_N
\end{pmatrix}
\begin{pmatrix}
V_0^{m-1} \\ V_1^{m-1} \\ V_2^{m-1} \\ \vdots \\ V_{N-1}^{m-1} \\ V_N^{m-1}
\end{pmatrix}
=
\begin{pmatrix}
V_0^m \\ V_1^m \\ V_2^m \\ \vdots \\ V_{N-1}^m \\ V_N^m
\end{pmatrix}
$$

So we are solving

$$\mathbf{A}\underline{V}^{m-1} = \underline{V}^{m},$$

at each time step for the unknown vector $\underline{V}^{m-1}$. We note that the matrix A is extremely sparse. A matrix consisting of a main diagonal together with one above (*super-diagonal*) and one below (*sub-diagonal*) is called a *tri-diagonal matrix*.

This linear system can now be solved directly or iteratively using e.g. the Gauss-Seidel Method.

Note on Forward Marching

Recall in the Black-Scholes equation the transformation $\tau = T - t$ gives the option value as a function of time to expiry. Write the time variable as

$$t = T - m\delta t,$$

so that $m = 0$ represents $t = T$ and $m = M$ gives $t = 0$.The increasing m from 0 to M gives a forward marching scheme

$$a_n V_{n-1}^{m+1} + b_n V_n^{m+1} + c_n V_{n+1}^{m+1} = V_n^m$$

with a_n, b_n, c_n given by (4) .

Stability

Suppose that the payoff contains small errors. These will propagate as we march backwards in time. As with the explicit scheme write

$$\widehat{V}_n^m = V_n{}^m + E_n{}^m$$

where $E_n{}^m$ is the associated error. Then $E_n{}^m$ also satisfies the difference equation (8) to give

$$a_n E_{n-1}^{m-1} + b_n E_n^{m-1} + c_n E_{n+1}^{m-1} = E_n^m. \tag{10}$$

By general Fourier analysis we consider harmonic perturbations of the form

$$E_n^m = \overline{a}^m \exp(in\omega). \tag{11}$$

So oscillatory of amplitude $\overline{a}$ and frequency ω. Substituting (11) into (10) gives

$$a_n \overline{a}^{m-1} e^{i(n-1)\omega} + b_n \overline{a}^{m-1} e^{in\omega} + c_n \overline{a}^{m-1} e^{i(n+1)\omega} = \overline{a}^m e^{(in\omega)}$$

which on simplification becomes

$$a_n e^{-i\omega} + b_n + c_n e^{i\omega} = \overline{a}.$$

Now stability criteria arises from the balancing of the time dependency and diffusion terms, so that

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = 0$$

From (9) we take the following contributions

$$a_n = -\frac{1}{2}n^2\sigma^2\delta t,\ b_n = 1 + n^2\sigma^2\delta t,\ c_n = -\frac{1}{2}n^2\sigma^2\delta t$$

$$\begin{aligned} \overline{a} &= 1 + n^2\sigma^2\delta t - \frac{1}{2}n^2\sigma^2\delta t \underbrace{\left(e^{i\omega} + e^{-i\omega}\right)}_{=2\cos\omega} \\ &= 1 + n^2\sigma^2\left(1 - \cos\omega\right)\delta t. \end{aligned}$$

Using the trigonometric identity $\cos 2x = 1 - 2\sin^2 x$, we have

$$\overline{a} = 1 + 2n^2\sigma^2 \sin^2 \frac{\omega}{2}\delta t$$

In order that errors do not grow as we step backwards in time, we require $|\overline{a}| \geq 1$,

$$\left|1 + 2n^2\sigma^2 \sin^2 \frac{\omega}{2}\delta t\right| \geq 1$$

which upon simplifying we find is valid for

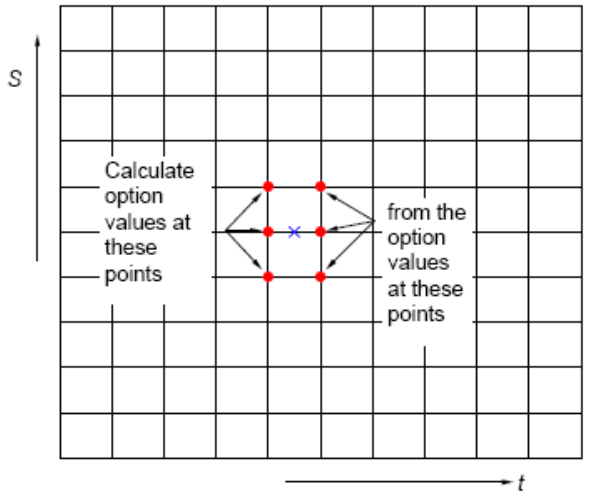
$$\delta t > 0,$$

so stability is guaranteed for all values of δt and the disturbances will die out as we time-step back from expiry.

The Crank-Nicolson Scheme

The fully implicit method has the same order of accuracy (in time and asset price) as the explicit scheme but is unconditionally stable , i.e. $\forall\ \delta t > 0$.

The Crank-Nicolson method whilst being unconditionally stable has the additional advantage that it is also second order accurate in time.



Consider a PDE being satisfied at the midpoint $\left(n\delta S, \left(m-\frac{1}{2}\right)\delta t\right)$ and replace V_S and V_{SS} by means of a FD approximation at the m^{th} and $(m+1)^{\text{th}}$ time steps, i.e.

$$\tfrac{1}{2}\left(m+(m-1)\right) \rightarrow \left(m-\tfrac{1}{2}\right).$$

The method is regarded as an **equally** weighted average of the explicit and implicit schemes with advantage over both individual cases due to accuracy being $O\left(\delta t^2\right)$.

We can then write the FD approximations as

$$\begin{aligned}
\frac{\partial V}{\partial t}\left(n\delta S, \left(m-\tfrac{1}{2}\right)\delta t\right) &\sim \frac{V_n^m - V_n^{m-1}}{\delta t} \\
\frac{\partial V}{\partial S}\left(n\delta S, \left(m-\tfrac{1}{2}\right)\delta t\right) &\sim \tfrac{1}{2}\frac{\partial V}{\partial S}\left(n\delta S, m\delta t\right) + \\
&\quad \tfrac{1}{2}\frac{\partial V}{\partial S}\left(n\delta S, (m-1)\,\delta t\right) \\
\frac{\partial^2 V}{\partial S^2}\left(n\delta S, \left(m-\tfrac{1}{2}\right)\delta t\right) &\sim \tfrac{1}{2}\frac{\partial^2 V}{\partial S^2}\left(n\delta S, m\delta t\right) + \\
&\quad \tfrac{1}{2}\frac{\partial^2 V}{\partial S^2}\left(n\delta S, (m-1)\,\delta t\right)
\end{aligned}$$

and substitute into the BSE as earlier - keeping note of the fact that we are stepping backwards in time. The resulting difference equation is

$$a_n V_{n-1}^{m-1} + b_n V_n^{m-1} + c_n V_{n+1}^{m-1} = A_n V_{n-1}^m + B_n V_n^m + C_n V_{n+1}^m$$

where

$$\begin{aligned}
a_n &= -\frac{1}{4}\left(\sigma^2 n^2 - n\left(r - D\right)\right)\delta t \\
b_n &= 1 + \frac{1}{2}\left(\sigma^2 n^2 + r\right)\delta t \\
c_n &= -\frac{1}{4}\left(\sigma^2 n^2 + n\left(r - D\right)\right)\delta t. \\
A_n &= \frac{1}{4}\left(\sigma^2 n^2 - n\left(r - D\right)\right)\delta t \\
B_n &= 1 - \frac{1}{2}\left(\sigma^2 n^2 + r\right)\delta t \\
C_n &= \frac{1}{4}\left(\sigma^2 n^2 + n\left(r - D\right)\right)\delta t.
\end{aligned}$$

and the matrix inversion problem we solve is

$$\mathbf{A}\underline{V}^{m-1} = \mathbf{B}\underline{V}^m.$$

The $\theta-$Method

While the attraction of the fully implicit scheme lies in its unconditionally stability, it is only first order accurate in time. The Crank–Nicolson scheme is an equally weighted average of both implicit and explicit methods and enjoys second order accuracy in time. Now we construct a generalisation of the Crank-Nicolson to obtain a weighted average of the two schemes with a weighted parameter θ such that $\theta \in [0, 1]$. On a simple heat equation it would be

$$\begin{aligned} \frac{V_n^{m+1} - V_n^m}{\delta t} &= \theta \left(\frac{V_{n+1}^{m+1} - 2V_n^{m+1} + V_{n-1}^{m+1}}{\delta S^2} \right) \\ &\quad + (1-\theta) \left(\frac{V_{n+1}^{m} - 2V_n^{m} + V_{n-1}^{m}}{\delta S^2} \right), \end{aligned}$$

i.e. $\theta \times$Implicit $+ (1-\theta) \times$Explicit.

For the BSE we have

$$a_n V_{n-1}^{m-1} + b_n V_n^{m-1} + c_n V_{n+1}^{m-1} = A_n V_{n-1}^m + B_n V_n^m + C_n V_{n+1}^m$$

where

$$\begin{aligned}
a_n &= -\frac{1}{2}\theta\left(\sigma^2 n^2 - n\left(r - D\right)\right)\delta t \\
b_n &= 1 + \vartheta\left(\sigma^2 n^2 + r\right)\delta t \\
c_n &= -\frac{1}{2}\theta\left(\sigma^2 n^2 + n\left(r - D\right)\right)\delta t. \\
A_n &= \frac{1}{2}\left(1-\theta\right)\left(\sigma^2 n^2 - n\left(r - D\right)\right)\delta t \\
B_n &= 1 - \left(1-\theta\right)\left(\sigma^2 n^2 + r\right)\delta t \\
C_n &= \frac{1}{2}\left(1-\theta\right)\left(\sigma^2 n^2 + n\left(r - D\right)\right)\delta t.
\end{aligned}$$

When $\theta = 0, \frac{1}{2}$ and 1 the θ method becomes the Explicit, Crank Nicolson and Fully Implicit scheme, in turn.

The θ method is a generalisation of the Crank-Nicolson scheme.

For a general value of θ the local truncation error is

$$O\left(\frac{1}{2}\delta t + \frac{1}{12}\delta S^2 - \theta\delta t, \delta S^4, \delta t^2\right).$$

When $\theta = 0$, $1/2$ or 1 we get the results we have seen so far.

But if

$$\theta = \frac{1}{2} - \frac{\delta S^2}{12\delta t}$$

then the local truncation error is improved.

The implementation of the method is no harder than the Crank–Nicolson scheme.

Three time-level methods

Numerical schemes are not restricted to the use of just two time levels. So far we have considered $\left(V^{m+1}, V^m\right)$ or $\left(V^m, V^{m-1}\right)$. We can construct many algorithms using three or more time levels, i.e.

$$V^{m+1} = g\left(V^{m-1}, V^m\right)$$

Again, we would do this if it gave us a better local truncation error or had better convergence properties. We already know that the centred time difference

$$\frac{V_n^{m+1} - V_n^{m-1}}{2\delta t}$$

is unstable for all values of δt. However the scheme

$$\frac{\left(V_n^{m+1} - V_n^{m-1}\right)}{2\delta t} = \frac{1}{\delta S^2}\left(V_{n+1}^m \underbrace{-V_n^{m+1} - V_n^{m-1}}_{=-2V_n^m} + V_{n-1}^m\right)$$

is stable for all δt.

Solving the linear System

We need to solve

$$\begin{pmatrix} b_0 & c_0 & 0 & \cdots & \cdots & 0 \\ a_1 & b_1 & c_1 & & & \vdots \\ 0 & a_2 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & \cdots & \cdots & 0 & \widehat{a}_n & \widehat{b}_n \end{pmatrix} \begin{pmatrix} V_0^{m-1} \\ V_1^{m-1} \\ \vdots \\ \vdots \\ V_{N-1}^{m-1} \\ V_N^{m-1} \end{pmatrix} = \begin{pmatrix} V_0^m \\ V_1^m \\ V_2^m \\ \vdots \\ V_{N-1}^m \\ V_N^m \end{pmatrix} \tag{12}$$

which is written as

$$\mathbf{M}\mathbf{V}^{m-1} = \mathbf{V}^m$$

which in principle is

$$\mathbf{V}^{m-1} = \mathbf{M}^{-1}\mathbf{V}^m$$

which presents us with a highly inefficient way of solving this linear system.

However we can exploit the structure of $\mathbf{M}$ given that it is a tridiagonal matrix.

- LU Decomposition (factorisation) of $\mathbf{M}$. Only advantageous if the matrix is non-time dependent, i.e. one factor model.

- Iterative schemes for solution. Under this heading we have Jacobi/Gauss-Seidel, SOR. Although slower than LU decomposition, they are more efficient in terms of memory and programming.

The LU Decomposition

Consider the matrix inversion problem

$$\mathbf{M}.\mathbf{y} = \mathbf{p}$$

It is often advantageous to think of Gaussian elimination as constructing a lower tridiagonal matrix L and an upper triangular matrix U, so that $\mathbf{LU} = \mathbf{M}$. The problem becomes

$$\begin{aligned} \mathbf{M}.\mathbf{y} &= (\mathbf{LU}).\mathbf{y} \\ &= \mathbf{L}(\mathbf{U}.\mathbf{y}) = \mathbf{p} \end{aligned}$$

We introduce an intermediate vector $\mathbf{z} = \mathbf{U}\mathbf{y}$ so that

$$\mathbf{L}\mathbf{z} = \mathbf{p}, \quad \mathbf{U}\mathbf{y} = \mathbf{z}.$$

$$L = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ l_1 & 1 & 0 & & & \vdots \\ 0 & l_2 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & l_{N-1} & 1 & 0 \\ 0 & \cdots & \cdots & 0 & l_N & 1 \end{pmatrix}, \; U = \begin{pmatrix} d_0 & u_0 & 0 & \cdots & \cdots & 0 \\ 0 & d_1 & u_1 & & & \vdots \\ 0 & 0 & d_2 & u_2 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & 0 & d_{N-1} & u_{N-1} \\ 0 & \cdots & \cdots & 0 & 0 & d_N \end{pmatrix}$$

It is trivial to solve

$$\mathbf{Lz} = \mathbf{p}$$

by forward substitution.

$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ l_1 & 1 & 0 & & & \vdots \\ 0 & l_2 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & l_{N-1} & 1 & 0 \\ 0 & \cdots & \cdots & 0 & l_N & 1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ \vdots \\ z_{N-1} \\ z_N \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p_N \end{pmatrix}$$

Having obtained $\mathbf{z}$ we find $\mathbf{y}$ by solving $\mathbf{Uy} = \mathbf{z}$;

$$\begin{pmatrix} d_0 & u_0 & 0 & \cdots & \cdots & 0 \\ 0 & d_1 & u_1 & & & \vdots \\ 0 & 0 & d_2 & u_2 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & 0 & d_{N-1} & u_{N-1} \\ 0 & \cdots & \cdots & 0 & 0 & d_N \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ \vdots \\ z_{N-1} \\ z_N \end{pmatrix}$$

This gives us the solution, $\mathbf{y}$, of our original problem, $\mathbf{My} = \mathbf{p}$.

This method can be also extended to decompose non-sparse matrices, i.e. $A = LU$, thus opening up a wider class of associated methods.

$$= \begin{pmatrix} l_{11} & 0 & \cdots & \cdots & \cdots & 0 \\ l_{21} & l_{22} & 0 & & & \vdots \\ \vdots & l_{32} & & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & l_{n-1} & l_{n-1,n-1} & 0 \\ l_{n1} & l_{n2} & \cdots & \cdots & \cdots & l_{nn} \end{pmatrix} \times$$

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & \cdots & u_{2n} \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

If $l_{ii} = 1 \ \forall \ 1 \leq i \leq n$ as mentioned in the earlier discussion, this ensures a unique solution and is called *Doolittle's method*. *Crout's method* requires $u_{ii} = 1 \ \forall \ 1 \leq i \leq n$.

Iterative Techniques

The linear system

$$\mathbf{M}\mathbf{v} = \mathbf{w}$$

can be written

$$a_n v_{n-1} + b_n v_n + c_n v_{n+1} = w_n.$$

Rearranging allows us to write as

$$v_n = \frac{1}{b_n}\left(w_n - a_n v_{n-1} - c_n v_{n+1}\right),$$

and forms the basis of an iterative scheme. If the initial guess to the solution is $\widehat{v}_n^{(0)}$, then we have the following iterative scheme

$$\widehat{v}_n^{(k+1)} = \frac{1}{b_n}\left(w_n - a_n \widehat{v}_{n-1}^{(k)} - c_n \widehat{v}_{n+1}^{k}\right),$$

for varying k. Some texts present the above as

$$v_n = \frac{1}{b_n}\left(-a_n v_{n-1} - c_n v_{n+1}\right) + \frac{w_n}{b_n}.$$

So the unknown vector approximation becomes

$$\widehat{\mathbf{v}}^{(k+1)} = \mathbf{T}\widehat{\mathbf{v}}^{(k)} + \mathbf{c}$$

Recalling from linear algebra, a matrix $\mathbf{M}$ with entries $\mathbf{M}_{ij}$ is *strictly diagonally dominant* if

$$|\mathbf{M}_{ii}| > \sum_{j \neq i} \left|\mathbf{M}_{ij}\right|,$$

So in this case

$$|b_n| > |a_n| + |c_n|$$

and hence the scheme will converge to the solution of the original problem,

$$\lim_{k \to \infty} \widehat{v}_n^{(k)} = v_n$$

Suppose the matrix $\mathbf{M}$ can be written in the form

$$\mathbf{M} = L + D + U$$

where D is a matrix with the diagonal elements of $\mathbf{M}$ (zero everywhere else).

L is a strictly lower-triangular part of $\mathbf{M}$.

U is strictly upper part of $\mathbf{M}$

$$\underbrace{\begin{pmatrix} b_0 & c_0 & 0 & \cdots \\ a_1 & b_1 & c_1 & \\ 0 & a_2 & \ddots & \ddots \\ \vdots & & \ddots & \ddots \end{pmatrix}}_{\mathbf{M}} =$$

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots \\ a_1 & 0 & 0 & \\ 0 & a_2 & 0 & \ddots \\ \vdots & & \ddots & \ddots \end{pmatrix}}_{\mathbf{A}} + \underbrace{\begin{pmatrix} b_0 & 0 & 0 & \cdots \\ 0 & b_1 & 0 & \\ 0 & 0 & b_2 & \ddots \\ \vdots & & \ddots & \ddots \end{pmatrix}}_{\mathbf{B}} + \underbrace{\begin{pmatrix} 0 & c_0 & 0 & \cdots \\ 0 & 0 & c_1 & \\ 0 & 0 & \ddots & \ddots \\ \vdots & & \ddots & \ddots \end{pmatrix}}_{\mathbf{C}}$$

The scheme now becomes

$$\widehat{\mathbf{v}}^{(k+1)} = \mathbf{B}^{-1}\mathbf{w} - \mathbf{B}^{-1}\left(\mathbf{A}+\mathbf{C}\right)\widehat{\mathbf{v}}^{(k)}.$$

$\mathbf{B}^{-1}(\mathbf{A}+\mathbf{C})$ is another matrix which premultiplies the iteration vector, call this $\mathbf{IM}$, to write

$$\widehat{\mathbf{v}}^{(k+1)} = \mathbf{B}^{-1}\mathbf{w} - \mathbf{IM}\widehat{\mathbf{v}}^{(k)}$$

Given an $N \times N$ matrix $\mathbf{M}$ the *spectral radius* $\rho(\mathbf{M})$ is defined as

$$\rho(\mathbf{M}) = \max|\lambda_i| \quad i = 1, 2, \ldots\ldots, n$$

where λ is the eigenvalue of $\mathbf{M}$.

Note: If $\lambda = \alpha + i\beta$ then $|\lambda| = \sqrt{\alpha^2 + \beta^2}$

Convergence of the Jacobi method is proved theoretically by showing that the spectral radius of the iteration matrix $\mathbf{IM} < 1$.

The Gauss Seidel Method

We can now refine the Jacobi method very simply by using the most up-to-date $\widehat{v}^{k+1}$ available to us, by writing

$$\widehat{v}_n^{(k+1)} = \frac{1}{b_n}\left(w_n - a_n\widehat{v}_{n-1}^{(k+1)} - c_n\widehat{v}_{n+1}^{k}\right).$$

Note that at iteration step $(k+1)$, we are using the most recent $\widehat{v}_{n-1}^{(k+1)}$, instead of $\widehat{v}_{n-1}^{(k)}$ which is used in Jacobi. Again, iteration convergence is guaranteed if the coefficient matrix is strictly diagonally dominant. Write the above as

$$b_n\widehat{v}_n^{(k+1)} + a_n\widehat{v}_{n-1}^{(k+1)} = w_n - c_n\widehat{v}_{n+1}^{k}$$

which as a matrix problem becomes

$$(\mathbf{A}+\mathbf{B})\,\widehat{\mathbf{v}}^{(k+1)} = \mathbf{w} - \mathbf{C}\widehat{\mathbf{v}}^{(k)}.$$

Write

$$\mathbf{Q} = (\mathbf{A}+\mathbf{B}),\ \mathbf{IM} = \mathbf{Q}^{-1}\mathbf{C}$$

so our iteration problem becomes

$$\widehat{\mathbf{v}}^{(k+1)} = \mathbf{Q}^{-1}\mathbf{w} - \mathbf{IM}\widehat{\mathbf{v}}^{(k)}.$$

As before convergence of this GS method is proved theoretically by showing that the spectral radius of the iteration matrix $\mathbf{IM} < 1$, but we will not be implementing in terms of the iteration matrix.

Successive-Over-Relaxation (SOR) Methods

A large part of numerical analysis is based upon the need to improve the efficiency and speed of existing techniques. Although we have seen from the Gauss-Seidel method that it is superior to Jacobi in this regard, we can nevertheless continue to look for improvements in convergence speed. In this section we will briefly look at a method called *successive-over-relaxation*.

Let $\underline{\widehat{x}} \in \mathbb{R}^n$ be an approximation to the actual solution for $A\underline{x} = \underline{b}$.

The *residual vector* $\underline{r}$ for $\underline{\widehat{x}}$ is

$$\underline{r} = \underline{b} - A\underline{\widehat{x}}$$

In Jacobi/Gauss-Seidel methods a residual vector is associated with each calculation of an approximation component to the solution vector.

Aim: Generate a sequence of approximations that cause the associated residual vectors to converge rapidly to zero.

SOR can be applied to either the Jacobi or the Gauss-Seidel method.

The sequence of approximate solutions $\widehat{v}_n^{(k)}$ such that $\lim_{k\to\infty} \widehat{v}_n^{(k)} \longrightarrow v_n$, can be speeded up. This is the essence of SOR. To see how this works, begin by writing

$$\widehat{v}_n^{(k+1)} = \widehat{v}_n^{(k)} + \left(\widehat{v}_n^{(k+1)} - \widehat{v}_n^{(k)}\right).$$

Of interest is the term $\left(\widehat{v}_n^{(k+1)} - \widehat{v}_n^{(k)}\right)$ which can be regarded as a correction to be added to $\widehat{v}_n^{(k)}$ in order for it to get closer to the exact solution v_n. Introduce ω to control the magnitude of this correction term

$$\widehat{x}_n^{(k+1)} = \widehat{v}_n^{(k)} + \omega\left(\widehat{v}_n^{(k+1)} - \widehat{v}_n^{(k)}\right).$$

So with careful choice of ω we hope that $\widehat{x}_n^{(k+1)}$ will be a better approximation to v_n than $\widehat{v}_n^{(k+1)}$.

SOR correction for **Jacobi**

$$\begin{aligned}\widehat{u}_n^{(k)} &= \frac{1}{b_n}\left(w_n - a_n\widehat{v}_{n-1}^{(k)} - c_n\widehat{v}_{n+1}^{k}\right)\\ \widehat{v}_n^{(k+1)} &= \widehat{v}_n^{(k)} + \omega_{\mathbf{J}}\left(\widehat{u}_n^{(k)} - \widehat{v}_n^{(k)}\right)\end{aligned}$$

SOR correction for **GS**

$$\begin{aligned}\widehat{u}_n^{(k)} &= \frac{1}{b_n}\left(w_n - a_n\widehat{v}_{n-1}^{(k+1)} - c_n\widehat{v}_{n+1}^{k}\right)\\ \widehat{v}_n^{(k+1)} &= \widehat{v}_n^{(k)} + \omega_{\mathbf{GS}}\left(\widehat{u}_n^{(k)} - \widehat{v}_n^{(k)}\right)\end{aligned}$$

For $0 < \omega < 1$, the procedures are called **under-relaxation methods** and used for convergence of systems which do not converge by Jacobi or Gauss-Seidel.

For $\omega > 1$, the procedures are called **over-relaxation methods** which are used to accelerate convergence of systems which are convergent by Jacobi or Gauss-Seidel.