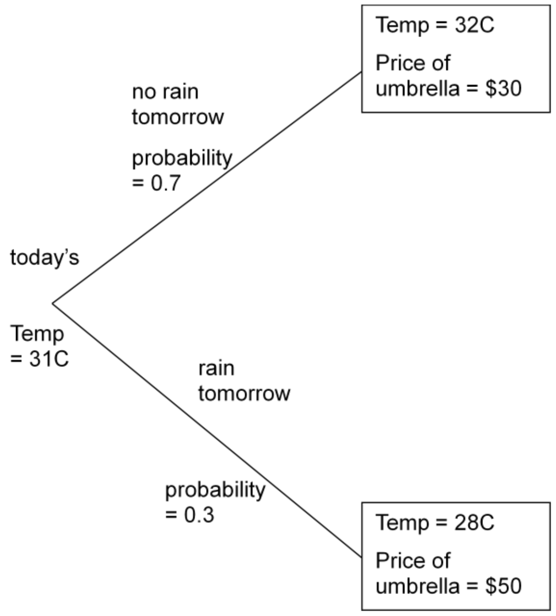
7. Derivative Pricing Models

7.1. Introduction

Before we start to talk about the mathematical models in pricing, we would like to introduce some of the key concepts. One method that would appear repeatedly in derivative pricing problems is known as **pricing by replication** (also known as **pricing by arbitrage**). We can illustrate this idea by a simple example. Let's suppose there is a new kind of drink, "Fresh", selling for $50 a glass, and people just *love* it. Only the seller of the drink knows that the drink could be mixed by adding 50% orange juice, 30% apple juice, and 20% mango juice, and the total costs of the raw ingredients equal to $8. If the market for orange, apple and mango juice is already saturated (i.e. not easy to make money), the "Fresh" drink is a perfect way to make a *huge* amount of money initially. The trick would work until people discover the ingredients and the proportion. When other sellers are prepared to accept a lower profit and sell this drink at a lower price, the price would be pushed down, until the profit margin disappears.

The above pricing method is often applied to derivative pricing, but we may be more used to a typical method known as **pricing by expectation**. Again we can demonstrate this with an example.



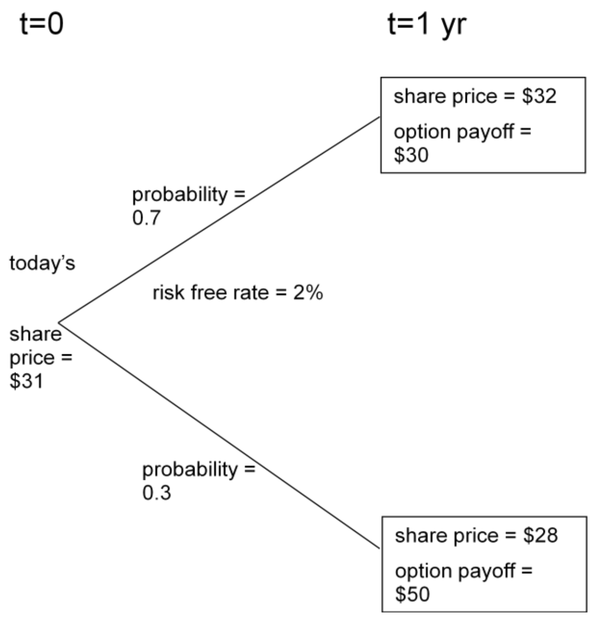
The current temperature is 31C and the probability of raining tomorrow is given. Assume that the price of an umbrella P is a function of temperature, say:

P = 190 – 5 x temperature

We want to calculate the expected price of an umbrella tomorrow. From expectation pricing, this can be obtained by summing the probability and the price, as follows:

$$\begin{aligned} E(P) &= \sum probability \times price \\ &= 0.7 \times 30 + 0.3 \times 50 \\ &= \$36 \end{aligned}$$

In fact, the expected temperature tomorrow can also be calculated from the given information, although it is not required in this problem. The above method is typically what we use when the probability distribution of the future events has been estimated. However, in finance, we can also make use of pricing of arbitrage, which does not require the forecast of the expected value. In the following example, the figures are the same as in the temperature / umbrella example above.



Assume that the payoff of a derivative contract is a function of share price, say:

P = 190 – 5 x share price

We want to calculate the expected price of the contract tomorrow. From expectation pricing:

$$\begin{aligned} E(P) &= \sum probability \times payoff \\ &= 0.7 \times 30 + 0.3 \times 50 \\ &= \$36 \end{aligned}$$

This is an amount in 1 year, so we just need to calculate its present value. Given that the interest rate is 2%, the expected price today = exp(–0.02) x 36 = $35.287.

However, in the context of derivative trading, $35.287 is a wrong price. We can come up with a strategy that can replicate the payoff of the derivative. In this case, we hold a portfolio of Δ shares and $$B$ bond, so that the portfolio's worth today is $$(\Delta S_0 + B)$ (S_0 is the initial share price today, which is $31 in this example). After 1 year, the portfolio is worth $\Delta S + B\exp(rt)$, where S is the share price and B has accumulated interest and become $B\exp(rt)$. Since we have not imposed any constraints in choosing Δ and B, *we set up the portfolio so that its value would always be the same as the derivative contract*, i.e.

$$\Delta \times 32 + B\exp(0.02) = 30$$
$$\Delta \times 28 + B\exp(0.02) = 50$$

Solving the above, we obtain Δ = –5, B = 186.2377, i.e. we would need to hold a short position of 5 shares, and hold bonds with a value of $186.2377 initially. Now since the portfolio is always worth the same as the derivative contract, *by the principle of no arbitrage, they could not have a different price today*. The derivative contract price today is thus –5 x 31 + 186.2377 = $31.2377.

Compare with the earlier example, we notice a strange thing: the subjective probabilities (p = 0.7, 1–p = 0.3) do not enter into the pricing equation; *only the risk free rate is relevant.* In finance, this is known as pricing in the *risk neutral* world. Also, note the difference between the expected share price (=0.7 x 32 + 0.3 x 28 = $30.80) and the fair forward value = ($31 x exp(0.02) = $31.626). We could solve for a value of p^* so that (p^* x 32 + (1–p^*) x 28 = $31.626); this distribution is known as the risk neutral probability distribution, which should be distinguished with real world distribution of the stock prices.

Finally, for the strategy to work, we need to be able to buy/sell the underlying instrument without restrictions (known as a "complete market"). In this example, the strategy relies on a short position in the stock, which may not be feasible in some markets. In the earlier example of umbrella pricing, we could not buy/sell a contract called "Temperature," so the expectation pricing method is appropriate. However, when the underlying market is complete, arbitrage pricing should be used.

7.2. A brief introduction to stochastic processes

7.2.1. Types of Stochastic Processes

We can use any of the four types of stochastic processes to model stock prices

- Discrete time; discrete variable
- Discrete time; continuous variable
- Continuous time; discrete variable
- Continuous time; continuous variable

The continuous time, continuous variable process proves to be the most useful for the purposes of valuing derivatives. However, for implementation purposes, we often need to approximate the continuous process by a discrete time, discrete variable process. This also conforms to real life data, as prices and time only move in discrete steps.

Assume that the current value of the stock is $X(s)$. We would introduce some terms in describing the stock process. Firstly, in a *Markov process*, future movements in a variable depend only on where we are, and not on the history of how we got there. In other words, given the value $X(s)$, the value of $X(t)$, $t>s$, depends only on $X(s)$ but not on any value $X(u)$ where $u<s$. In derivative pricing, a basic assumption is that stock prices follow Markov processes. (However, if this is true, then technical analysis (i.e. the study of charts to predict further stock market movements) would be useless.)

A particular type of Markov process is *Wiener Process*, also known as a Brownian process. We consider a variable $Z(t)$ whose value changes continuously. Next, define $\phi(\mu,v)$ as a normal distribution with mean μ and variance v. The change in a small interval of time Δt is Δz. The variable follows a Wiener process if $\Delta z = \varepsilon\sqrt{\Delta t}$ where ε is a random variable drawn from a standardized normal distribution $\phi(0,1)$.

One property that follows from this definition is that the values of Δz for any two different (non-overlapping) periods of time are independent (because each ε is independently drawn). Other properties include:

- Mean and variance of $[Z(t) - Z(t_0)]$ are 0 and T (= $t - t_0$)
- Probability distribution is:

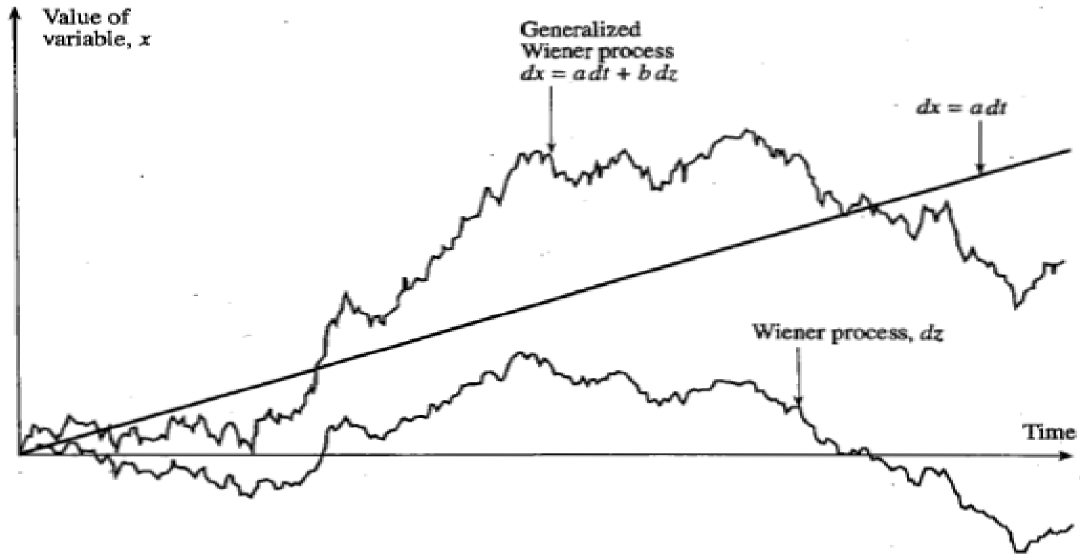
$$
\begin{aligned}
P[Z(t) \leq z | Z(t_0) = z_0] &= P[Z(t) - Z(t_0) \leq z - z_0] \\
&= \frac{1}{\sqrt{2\pi(t-t_0)}} \int_{-\infty}^{z-z_0} \exp\left(-\frac{x^2}{2(t-t_0)}\right) dx \\
&= N\left(\frac{z-z_0}{\sqrt{t-t_0}}\right).
\end{aligned}
$$

where $N()$ denotes a normal distribution.

While a Wiener process has a drift rate of 0 (i.e. average change per unit time) and a variance rate of 1, the drift rate and the variance rate of a *generalized Wiener process* can be set to be equal to any chosen constants. A variable x follows a generalized Wiener process with a drift rate of a and a variance rate of b^2 if we can write: $dx = a\ dt + b\ dz$ where dx is expressed in continuous time. If we approximate this expression in discrete time,

$$\Delta x = a\ \Delta t + b \Delta z = a\ \Delta t + b\ \varepsilon\sqrt{\Delta t}$$

In other words, the mean change in x in time T is aT, variance of change in x in time T is b^2T, and the standard deviation of change in x in time T is $b\sqrt{T}$.



An example is shown in the above diagram. We can see that a generalized Wiener process is just a random process being superimposed on a process with a constant drift (given as $dx=adt$).

Finally, a special case of general Wiener process is an *Itô process*, where the drift rate and the variance rate are functions of time: $dx=a(x,t)\,dt+b(x,t)\,dz$. As Δt tends to zero, the discrete time equivalent is:

$$\Delta x = a(x,t)\Delta t + b(x,t)\varepsilon\sqrt{\Delta t}$$

This is the form that is commonly used for modeling stock prices. We write:

$$dS = \mu S\,dt + \sigma S\,dz$$

where S is the stock price, μ is the expected return and σ is the volatility of the stock. The discrete time equivalent is:

$$\Delta S = \mu S\Delta t + \sigma S\varepsilon\sqrt{\Delta t}$$

7.2.2. Itô's Lemma

If we know the stochastic process followed by a random variable x, Itô's lemma tells us the stochastic process followed by some function $G(x, t)$. Since a derivative contract is a function of the price of the underlying and time, Itô's lemma plays an important part in the analysis of derivative securities.

We won't attempt to provide a rigorous mathematical proof of Itô's Lemma, but would try to demonstrate the idea through a heuristic argument. We start with a Taylor's series expansion of $G(x, t)$:

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + ½\frac{\partial^2 G}{\partial x^2}\Delta x^2 + \frac{\partial^2 G}{\partial x\partial t}\Delta x\,\Delta t + ½\frac{\partial^2 G}{\partial t^2}\Delta t^2 + \ldots$$

An approximation is to ignore higher order terms, i.e. we want to keep only the terms with $O(\Delta t)$. Unlike the case when x is deterministic, even if we ignore the second order terms, the term involving Δx^2 cannot be dropped, because Δx is of order $\sqrt{}\Delta t$.

Suppose

$$dx = a(x,t)dt + b(x,t)dz$$

so that

$$\Delta x = a\,\Delta t + b\,\varepsilon\sqrt{\Delta t}$$

Then ignoring terms of higher order than Δt

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + ½\frac{\partial^2 G}{\partial x^2}b^2\varepsilon^2\Delta t$$

Since $\varepsilon \approx \phi(0,1),\ E(\varepsilon) = 0, E(\varepsilon^2) = 1 + [E(\varepsilon)]^2 = 1$

It follows that $E(\varepsilon^2\Delta t) = \Delta t$, variance of Δt is $O(\Delta t^2)$

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\Delta t$$

Taking limits: $dG = \frac{\partial G}{\partial x}dx + \frac{\partial G}{\partial t}dt + ½\frac{\partial^2 G}{\partial x^2}b^2 dt$

Substituting: $dx = a\,dt + b\,dz$

We obtain: $dG = \left(\frac{\partial G}{\partial x}a + \frac{\partial G}{\partial t} + ½\frac{\partial^2 G}{\partial x^2}b^2\right)dt + \frac{\partial G}{\partial x}b\,dz$

The stock price process is

$$dS = \mu S\,dt + \sigma S\,dz$$

For a function G of S and t

$$dG = \left(\frac{\partial G}{\partial S}\mu S + \frac{\partial G}{\partial t} + ½\frac{\partial^2 G}{\partial S^2}\sigma^2 S^2\right)dt + \frac{\partial G}{\partial S}\sigma S\,dz$$

We can illustrate this relationship with two examples:

1. The forward price of a stock for a contract maturing at time T

$$G = S\,e^{r(T-t)}, \frac{\partial G}{\partial S} = e^{r(T-t)} = \frac{G}{S}, \frac{\partial^2 G}{\partial S^2} = 0, \frac{\partial G}{\partial t} = -rG$$

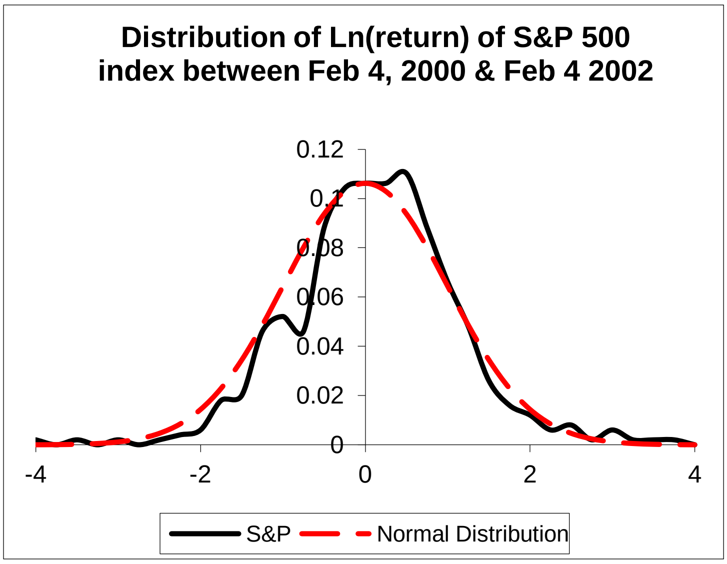
$$dG = (\mu - r)G\,dt + \sigma G\,dz$$

2. $G = \ln S, \frac{\partial G}{\partial S} = \frac{1}{S}, \frac{\partial^2 G}{\partial S^2} = \frac{-1}{S^2}, \frac{\partial G}{\partial t} = 0$

$$dG = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma\,dz$$

7.2.3. Stock price behaviour

The above results have been used in modeling the stock prices. The real world stock process is described as a Geometric Brownian Motion.



$$\frac{dS}{S} = \mu dt + \sigma dz$$

$$d(\ln S) = (\mu - \frac{\sigma^2}{2})dt + \sigma dz$$

$$S_t = S_0 \exp\left[(\mu - \frac{\sigma^2}{2})T + \sigma\varepsilon\sqrt{T}\right]$$

In this formulation, μ is the stock's growth rate, ε is a random variable drawn from a standardized normal distribution (mean = 0, variance = 1), and σ is the annualized volatility of *S*. It follows that:

$$\ln S_T - \ln S_0 \approx \phi\left[\left(\mu - \frac{\sigma^2}{2}\right)T, \sigma^2 T\right]$$

or

$$\ln S_T \approx \phi\left[\ln S_0 + \left(\mu - \frac{\sigma^2}{2}\right)T, \sigma^2 T\right]$$

Since the logarithm of S_T follows a normal distribution, we say that S_T is lognormally distributed. An example from historical data is shown in the diagram above, where the distribution of ln(return) of S&P 500 index is plotted. It is evident that while the historical returns do not conform to a lognormal distribution, the approximation is very reasonable.

It can also be shown that the expected value of S_T and its variance are given by:

$$E(S_T) = S_0\, e^{\mu T}$$

$$\text{var}(S_T) = S_0{}^2 e^{2\mu T}(e^{\sigma^2 T} - 1)$$

If *x* is the continuously compounded return,

$$S_T = S_0\, e^{xT}$$

$$x = \frac{1}{T}\ln\frac{S_T}{S_0}$$

$$x \approx \phi\left(\mu - \frac{\sigma^2}{2}, \frac{\sigma^2}{T}\right)$$

The above formulas can give some non-intuitive results. For example, we need to distinguish between the *short term expected return* of the stock and the *expected continuously compounded return* of the stock. In the short term, the expected percentage change in the stock price is given by μdt, which means that the expected return is μ in a very short period of time. However, the continuously compounded return realized over T years is given as *x* above, which has a mean of $\mu - \sigma^2/2$. . This is because

$$\ln[E(S_T / S_0)] \quad \text{and} \quad E[\ln(S_T / S_0)]$$

are not the same. To illustrate this point, we use a simple numerical example. A sequence of returns of a stock is as follows: 15%, 20%, 30%, –20%, 25%. The arithmetic mean of this series = (15+20+30-20+25)/500 = 14%, and this is the expected return of the stock μ, for each period. However, the expected return of an investor = (1.15 x 1.20 x 1.30 x 0.80 x 1.25)^(1/5) – 1 = 12.4%. This is the expected compound return of the stock.

7.3. The Black-Scholes option pricing framework

7.3.1. The Black-Scholes argument

Historical development

Fischer Black and Myron Scholes managed to publish the famous paper in 1973 (after being rejected by a few journals). Robert Merton published a paper in the same year, giving an alternative derivation which leads to the same formula. Basically, the Black-Scholes formula is almost identical to a formula given by Paul Samuelson in 1965, only that the subjective growth rate μ is now replaced by the risk free rate r. Scholes and Merton earned the Nobel prize in Economics in 1997 (Fischer Black died in 1995).

Set up

We note that the option price and the stock price depend on the same underlying source of uncertainty. Now we construct a portfolio P of short 1 derivative contract (with value *f*) and long Δ shares (with value ΔS). Given that Δ can be arbitrarily chosen, the correct value is to pick Δ such that the portfolio is riskless (i.e. the stochastic term = 0). *Since this portfolio has no risk, the portfolio must earn the risk free rate, otherwise arbitrage exists*.

$$P = -f + \Delta S$$

$$dP = -df + \Delta dS$$

$$df = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \right) dt + \sigma S \frac{\partial f}{\partial S} dz$$

$$\Rightarrow dP = -\left[\left(\Delta - \frac{\partial f}{\partial S} \right) \mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \right] dt + \sigma S \left(\Delta - \frac{\partial f}{\partial S} \right) dz$$

If we choose $\Delta = \dfrac{\partial f}{\partial S}$, the *dz* term is 0, which means the portfolio is riskless.

We note two points here. Firstly, the riskless condition is only correct instantaneously; Δ would change when S, t have changed. Secondly, the growth rate μ does not enter the equation any more as the coefficient to μS has also become zero. It means that the portfolio value does not depend on the subjective growth rate of the stock. On the other hand, since the portfolio is riskless, it earns the risk free rate, so we can write:

$$P_{t+dt} = P_t \exp(rdt), dP = rPdt$$
$$dP = \left[-\frac{\partial f}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \right] dt = rPdt = r\left(\frac{\partial f}{\partial S} S - f \right) dt$$
$$\Rightarrow \frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \quad \text{(Black - Scholes PDE)}$$

The resulting partial differential equation (PDE) must be satisfied by *ANY* derivative under this pricing framework. This equation can be solved by specifying the initial condition and boundary conditions, usually from numerical schemes, e.g. (explicit or implicit) finite difference methods. For example, for European calls,

$$\text{at } t = T,\ f = \max(S - K, 0)$$
$$f(t) = 0 \text{ for } S = 0, \frac{\partial f}{\partial S} = 1 \text{ for } S \to \infty \ \{\text{boundary conditions}\}$$

Risk-Neutral Valuation

An extremely important result is noted in the above derivations. As mentioned before, the variable μ does not appear in the Black-Scholes equation, which means that the equation is independent of all variables affected by risk preference. The solution to the differential equation is therefore the same in a risk-free world as it is in the real world. This leads to the principle of risk-neutral valuation. Later we can see how to make use of the risk free rate in our valuations.

Assumptions in the Black-Scholes formulation

(i) Continuous trading, i.e. prices move in infinitesimal small increments.
(ii) constant riskless interest rate.
(iii) the asset pays no dividend.
(iv) there are no transaction costs and taxes.
(v) the assets are perfectly divisible (c.f. board lots in shares).
(vi) short selling is allowed.
In a later section we will see attempts to relax some of these assumptions.

7.3.2. The Black-Scholes Formulas

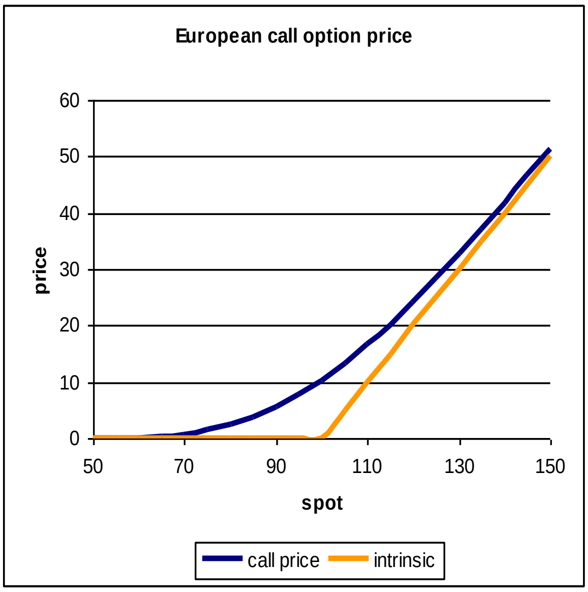
Solutions of the Black-Scholes PDE for European calls and puts lead to the following formulas:

$$c = S_0\, N(d_1) - K\, e^{-rT} N(d_2)$$
$$p = K\, e^{-rT}\, N(-d_2) - S_0\, N(-d_1)$$
$$\text{where} \quad d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

c and p are the European call and put prices and $N()$ is the cumulative normal distribution, with the other variables taking their usual meanings. Two properties are apparent from these formulas. As S_0 becomes very large, c tends to $S_0 - Ke^{-rT}$ and p tends to zero. Conversely, as S_0 becomes very small, c tends to zero and p tends to $Ke^{-rT} - S_0$.

Time value and intrinsic value

With the equations given above, we can graph the option price against the change in the underlying price. The European call option has a strike of 100, maturity 3 months, interest rate = 2%, and stock volatility = 50%.



Recall from section 6.3.1. that we describe the concept of intrinsic value. Option price is made up of two parts:
Option price = time value + intrinsic value.

Intrinsic value is the value if the option is exercised immediately.
= maximum of (S_0 – strike,0) for a call;
= maximum of (strike – S_0, 0) for a put.

Time value is the reward for holding the option. From the diagram, we can see that this value could be very small if the option is deep in-the-money or deep out-of-the-money.

Using this example, when spot price is 110, option price equals 16.6 (from Black-Scholes formula). We can separate this by first calculating the intrinsic value, which is equal to 110 – 100 = 10. Time value is then obtained by: option price – intrinsic value = 16.6 – 10 = 6.6. When spot price is 90 and the option price is 5.4, the intrinsic value is 0, and the time value is equal to the option price = 5.4. As an exercise, try to calculate the time value when spot price = 70 and 130, and compared these values with the cases when spot price equals 90 and 110.

7.3.3. Methods in option pricing

In the above sections, we have provided the solutions to one class of option – European calls and puts. However, the pricing framework is more general than that. For example, the Black-Scholes PDE is supposed to work for every derivative. However, solving the PDE is not the only way to obtain a solution to the problem. Three methods are generally used in practice:

- *Analytical solution*
  Black-Scholes formula is an example of this method, where an explicit formula can be found for particular types of options. This solution can be found by a convolution of payoff and probability density function (see below).

- *Monte Carlo simulation*
  This is a general and powerful technique which is useful in many option pricing problems. Basically we need to generate random paths and obtain the price via averaging the result. More details will be given in section 7.5.
- *Numerical solution of PDE*
  The Black-Scholes PDE can be solved numerically using advanced methods developed in other disciplines. A particular implementation of an explicit finite difference method is known as the Tree method, either binomial or trinomial. Implicit finite difference schemes are also used. These would also be described in section 7.5.

The analytical approach makes use of an important result known as the **Feynman-Kac formulation**. A PDE with a form:

$$\frac{\partial f}{\partial t}+\mu(S,t)\frac{\partial f}{\partial S}+\frac{1}{2}\sigma^2(S,t)\frac{\partial^2 f}{\partial S^2}-rf=0$$

with boundary condition $H(S,t)$ has a solution:

$$f(S,t)=\exp(-rt)\mathrm{E}\big(H(S,t)\big)$$

where the expectation E is taken with respect to a process S defined by:

$$dS=\mu(S,t)dt+\sigma(S,t)dz$$

In the risk neutral world, if S has a constant volatility, we could write:

$$dS=rSdt+\sigma Sdz$$

As an example, the payoff function for a European call at maturity is max(S–K,0). The expected value (i.e. the option price) at maturity is:

$$f=\int_0^\infty H(S)\bullet G(S)dS,H(S)=\begin{cases}S-K,S\geq K\\0,S<K\end{cases}$$

$$G(S)=\frac{1}{S\sigma\sqrt{2\pi t}}\exp\left\{-\frac{1}{2}\left(\frac{\ln(S)-\mu}{\sigma\sqrt{t}}\right)^2\right\}$$

$$\mu=\ln(S_0)+\left(r-\frac{\sigma^2}{2}\right)t$$

$$\therefore f=\int_K^\infty(S-K)\bullet G(S)dS$$

$G(S)$ is the probability density function if S follows a lognormal distribution. Expected value today = f x exp($-rt$), and one could then recover the Black-Scholes formula.

In the Monte Carlo simulation approach, recall that the process for the stock price is given by: $dS = rSdt + \sigma Sdz$. In discrete form, we can approximate the expression by:

$$\Delta S_i=rS_i\Delta t_i+\sigma S_i\varepsilon\sqrt{\Delta t_i}$$
$$S_{i+1}=S_i+\Delta S_i$$

We generate many different paths of $S_0\ldots S_n$ (typically at least 10,000 paths) using a random number generator. For each path, since we know all the prices of S, we can work out the option payoff given the payoff function. The option price is just the present value of the average of all these payoffs.

7.4. Extensions of the Black-Scholes framework

7.4.1. Black's Model

Two of the assumptions in the standard Black-Scholes framework can be relaxed in a straightforward manner.

- If r is a function of time, replace r by $\frac{1}{t}\int_0^t r(u)du$.
- If σ is a function of time, obtain σ from the following: $\sigma^2 = \frac{1}{t}\int_0^t \sigma^2(u)du$.

A significant development is the **Black's Model** (Black (1976)). Instead of assuming the spot price follows a lognormal distribution, a more useful result is obtained when the return of the forward price F is lognormal and has a constant volatility:

$$c = P(0,T)[F_0N(d_1) - KN(d_2)]$$
$$p = P(0,T)[KN(-d_2) - F_0N(-d_1)]$$

$$d_1 = \frac{\ln(F_0/K) + \sigma_F^2 T}{\sigma_F\sqrt{T}} \qquad d_2 = \frac{\ln(F_0/K) - \sigma_F^2 T}{\sigma_F\sqrt{T}}$$

$P(0,T)$ is the price of a zero coupon bond with maturity T. Note the subtle differences between Black-Scholes model and Black's model. We can easily recover the Black-Scholes formula from the Black's formula if interest rate is deterministic, i.e. we can simply write $F = S\exp[(r-D)T]$ where D is the dividend yield. Thus it requires a stronger assumption, that interest rate is somewhat fixed and known. With the Black's model, the underlying variable in the model becomes F, which captures both the dividend yield and stochastic interest rate. This way of modeling is intuitively more appealing, especially when we need to price options with longer time to maturity. However, because of the change in the underlying variable, there is a (small) theoretical difference between the meaning of σ. Now σ represents the volatility of the forward, which aggregates the effect of the movement in the underlying asset price, together with the impact of interest rate (and dividend) changes.

7.4.2. Volatility modeling

A key parameter in the Black-Scholes pricing framework is the volatility of the underlying asset. Formally, it is defined as the standard deviation of the daily log return, expressed in an annualized fashion. Before we discuss the calculation and use of this parameter, we would just mention a trader's "rule of thumb." With the underlying assumption that the stock follows a lognormal distribution, the rule says that if annualized volatility is Y %, then there is a 68% chance that the stock will move +/– Y/16% daily. For example, say stock price is 100 and volatility is 32%. It implies that tomorrow there is a 68% chance that the closing price of the stock would range between 100 +/- (32/16)%, i.e. between 98 and 102. Note that, it says 68% chance that it would stay in the range, which means that there is still a significant chance that it could trade outside of this range. Therefore one should not be surprised if the stock price moves to 104 tomorrow, because this movement is well within statistical prediction. In fact, based on the movements of a few days, we could not say whether the underlying volatility has changed or not. Testing the change in volatility would require standard econometric methods.

The volatility can be calculated from a historical data series:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i-\bar{u})^2}\times\sqrt{252}$$

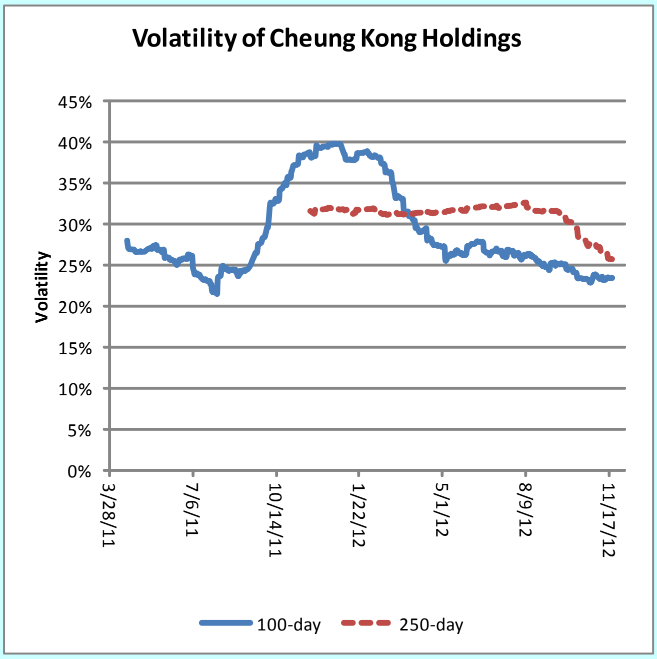
$$u_i = \ln\left(\frac{S_i}{S_{i-1}}\right), S_i \text{ is the stock price at time } i, \bar{u} \text{ is the mean of the } u_i\text{'s}$$

252 is assumed to be the number of trading days per year

We note two points from this calculation. Firstly, S_0 is the first data point and S_n is the last point in the series, with successive S_i representing the stock prices on each trading day. If there are n+1 data points, we can calculate n u_i. Secondly, 252 is used as a scaling factor. Some textbooks use 250 or 260, and there is no absolute correct answer. Furthermore, this scaling factor is independent from the number of data points used (n); it is only used for bringing the daily volatility into an annualized volatility. In fact, it is difficult to know what n should be chosen. In real life calculations, n is commonly set to 100 or 250 of the most recent observations. Other more sophisticated econometric methods could be used, that can make use of some kind of weighting scheme to the more recent data points.

Alternatively, we can plot a moving series of 90 or 180 observations and observe the trend. It is obvious that the volatility does not stay constant and can vary substantially from one period to another.

	Cheung Kong		Volatility	
	closing price	ln(return)	100-day	250-day
10/19/12	113.50	-0.0070	23.47%	28.07%
10/22/12	114.90	0.0123	23.36%	27.35%
10/24/12	119.90	0.0426	23.02%	27.64%
10/25/12	119.40	-0.0042	22.99%	27.56%
10/26/12	117.80	-0.0135	23.06%	27.60%
10/29/12	112.30	-0.0478	23.94%	27.68%
10/30/12	112.20	-0.0009	23.95%	27.68%
10/31/12	114.50	0.0203	23.99%	27.71%
11/01/12	115.00	0.0044	23.90%	27.37%
11/02/12	115.80	0.0069	23.73%	27.24%
11/05/12	115.60	-0.0017	23.44%	27.24%
11/06/12	115.50	-0.0009	23.44%	26.81%
11/07/12	116.10	0.0052	23.44%	26.71%
11/08/12	113.50	-0.0226	23.71%	26.54%
11/09/12	112.90	-0.0053	23.30%	26.44%
11/12/12	114.40	0.0132	23.36%	26.47%
11/13/12	112.50	-0.0167	23.54%	26.50%
11/14/12	113.70	0.0106	23.57%	26.49%
11/15/12	112.60	-0.0097	23.62%	25.86%
11/16/12	112.40	-0.0018	23.52%	25.86%
11/19/12	113.70	0.0115	23.53%	25.77%
11/20/12	113.20	-0.0044	23.56%	25.78%



One way of modeling this behavior is to assume that volatility is a function of time, i.e. we write $\sigma(t)$ instead of a constant σ. From section 7.2.1., the variance of the stock is given by $\sigma(t)^2 t$. As an example, we want to calculate the 3-year volatility of a stock. Suppose the volatility is σ_1 for the first year and σ_2 for the second and third year. Total accumulated variance at the end of three years is $\sigma_1^2 + 2\sigma_2^2$. Thus the 3-year average volatility is given by:

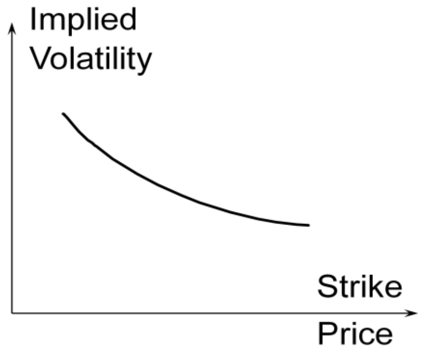
$$3\bar{\sigma}^2 = \sigma_1^2 + 2\sigma_2^2; \quad \bar{\sigma} = \sqrt{\frac{\sigma_1^2 + 2\sigma_2^2}{3}}$$

Using different volatilities with different maturities is called the **term structure of volatility**. In real markets (especially for stock indices), volatility often exhibits another characteristic, known as **volatility skew or smile**. In addition to varying volatilities across time, options at the same maturity but different strikes would be quoted at different volatilities. A typical set of market volatility quotes look like these:

		Strike	
Maturity	**90**	**100**	**110**
3Mth	47%	44%	41%
6Mth	44.5%	42%	38%
1 Year	39%	37%	35%
2 Years	35%	33%	32%
3 Years	32%	31%	30%
4 Years	30%	29.5%	29%
5 Years	29%	28.5%	28%

Theoretically speaking this is counter-intuitive, because the volatility should describe the movement of the underlying variable (e.g. a particular stock's price); it should have nothing to do with the characteristic of the option (i.e. strike). Also we cannot generate this kind of result from using the standard formulas with the historical time series. However, the "volatility" has been used as an alternative way to express the price of an option. Rebonato (1999, p.78) explains the phenomenon like this: smile implied volatility is "a wrong number to put in the wrong formula to obtain the right price." The most common explanations of the smile phenomenon:

- supply and demand of options at different strikes – more people demand downside puts than upside calls.
- the underlying distribution is not lognormal; it could have fat tails or jumps.
- the volatility of the underlying is stochastic – this will be treated in the section below.



Each asset class can have its own characteristic shape. Typical volatility smile for equity options is shown in the diagram here. For foreign exchange, the curve is more symmetrical around the at-the-money strike. Commodity options can show a positive slope, i.e. increasing volatility for a higher strike.

A significant extension to the Black-Scholes framework was first announced in a series of papers published in 1994. These models, known as **local volatility models**, assume volatility is a function of the level of the underlying asset price and time, i.e. $\sigma(S,t)$. If European option prices at all strikes and maturities are available, we would like to find this state-time dependent volatility function which is consistent with the input prices. We will not examine the details of these models, but would just mention some of the most important results. Firstly, we note that we can recover the risk-neutral probability distribution of the asset price. Recall that the call option price C can be written as:

$$C(S,K,T) = e^{-rT}\int_K^\infty (S_T - K)\Psi(S_T)dS_T$$

If we differentiate C w.r.t. K twice, we would get

$$\Psi(S_T) = e^{-rT}\frac{\partial^2 C}{\partial K^2}$$

$\Psi(S_T)$ is the risk-neutral density function of the underlying asset. Thus the (surprising) result is that if option prices at different strikes are known (i.e. $C(S,K,T)$ are given for all K and T), then the probability distribution can be obtained, even without knowing the volatility functions.

A standard formulation of the local volatility model comes from Dupire (1994). If we assume that the asset price follows this process (q is the dividend yield):

$$\frac{dS}{S} = (r-q)dt + \sigma(S,t)dz$$

It can be shown that

$$[\sigma(K,T)]^2 = 2\frac{\frac{\partial C}{\partial T} + qC + K(r-q)\frac{\partial C}{\partial K}}{K^2\frac{\partial^2 C}{\partial K^2}}$$

In practice, the above approach may not work so well because we do not have option prices for all strikes and maturities, even for the most liquid underlying (e.g. stock indices). Extrapolation of option prices would be required. Derman and Kani (1994, 1998) tried to obtain a binomial tree which is implied from the market data, but again it can lead to numerical implementation and stability problems, because of the possibility of violation of the no-arbitrage conditions when option prices need to be interpolated from a limited number of input points.

In the formulation above, it is assumed that the only stochastic variable is the stock price. While the volatility is not constant, its randomness only comes from a dependency on the stock price (as σ is a function of S and t). In **stochastic volatility models**, the volatility V is modeled explicitly as another stochastic process, and the correlation between volatility and underlying price can be specified. For example, we can write:

$$\frac{dS}{S} = (r-q)dt + \sqrt{V}dz_S$$

$$dV = a(V_L - V)dt + \xi V^\alpha dz_V$$

When V and S are uncorrelated, the European option price is the Black-Scholes price integrated over the distribution of the average variance. When V and S are negatively correlated, we obtain a downward sloping volatility skew similar to that observed in the market for equities. When V and S are positively correlated the skew is upward sloping, which is the pattern sometimes observed for commodities. Therefore this model can have enough flexibility to model the market volatility smiles for different underlyings. We can make use of vanilla options of these underlyings to "back out" the parameters of the model (a process known as calibration).

Many versions of stochastic volatility model exist, and one of the earliest and most popular formulation comes from Heston (1993). He observed that volatility often exhibits mean reverting behavior. However, while volatility is driving stock prices, it is not directly observable from the market. Heston's model is written as:

$$\begin{aligned}
\frac{dS}{S} &= (r-q)dt + \sqrt{V}dz_S \\
dV &= a(V_L - V)dt + \xi V^{0.5} dz_V \\
dz_S dz_V &= \rho dt
\end{aligned}$$

V_L is the mean reversion level of the variance, a is the mean reversion speed, ξ is the volatility of variance, and ρ is the correlation between the two Brownian processes. From this set of equations, semi-analytical solutions for call and put prices can be derived.

Why do we need local volatility or stochastic volatility models?

It is evident that the local volatility and stochastic volatility models are more complex than the standard Black-Scholes framework. If our aim is to price vanilla European or American calls and puts, we don't need to use these advanced volatility models. While Black-Scholes does not describe reality, many tests show that its performance is reasonable when it is used to manage the risks of basic options. However, the advanced models would be important in describing the behavior of exotic products which may depend on the accurate (hedgeable) distribution of the underlying asset prices at different strike levels. Examples of such products include various kinds of barrier options and path dependent options.

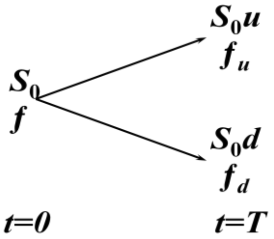
7.5. Numerical methods for option pricing

While the Black-Scholes PDE can be applied to any derivative contract, the solution depends on the particular initial and boundary conditions that correspond to the characteristics of the contract to be priced. Instead of obtaining an analytical solution directly, general numerical methods have been developed in solving the problem. Three particularly useful techniques that can be applied to a wide range of derivative contracts are discussed in the following sections.[1]

[1] Some of the diagrams in this section are taken from Hull, 7th edition, Chapters 11 and 19; or 8th edition, Chapters 12 and 20.

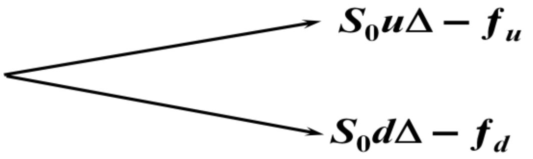
7.5.1. Binomial and trinomial trees

The binomial tree method was proposed by Cox, Ross and Rubinstein in 1979. We will use a simple set up to describe the method below. In the following two-state example, a derivative lasts for time T and its value is dependent on the stock price. There are only two possible states for the stock price after one period.



- S_0 is the initial stock price
- f is the value of the derivative today;
- u and d are the ratios of the upward and downward movements of the stock price
- f_u and f_d are the values of the derivative after the u and d movements

Consider a portfolio that is long Δ shares and short 1 derivative:



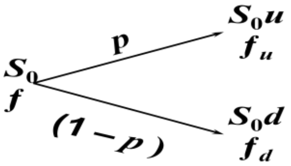
The portfolio's possible values are shown above. When $S_0u\Delta - f_u = S_0d\Delta - f_d$, we say that the portfolio is riskless, because its value is the same no matter how the stock moves (in this two-state world). Solving for Δ, we obtain:

$$\Delta = \frac{f_u - f_d}{S_0u - S_0d}$$

Thus the value of the portfolio at time T is $S_0u\Delta - f_u$, and the value today is thus $(S_0u\Delta - f_u)e^{-rT}$ (where e^{-rT} is the discounting factor to today). Another expression for the portfolio value today is $S_0\Delta - f$ (because the portfolio holds Δ shares and short 1 derivative contract). Equating the two expressions, we obtain $f = S_0\Delta - (S_0u\Delta - f_u)e^{-rT}$. Substituting for Δ we obtain

$$f = [\, pf_u + (1-p)f_d \,]e^{-rT} \quad \text{where} \quad p = \frac{e^{rT} - d}{u - d}$$

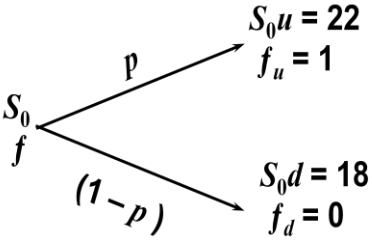
The value of a derivative is then its expected payoff in a risk-neutral world discounted at the risk-free rate:

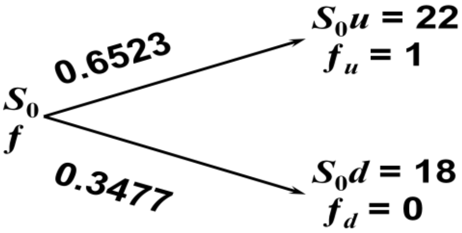


Note that the subjective probability about the upward or downward of the asset price does not appear in the formula for the value of f. In other words, people can have different expectations about the movements of the asset price, but the fair value of the derivative is unaffected by the expectation. In addition, it is natural to interpret p and $1–p$ as some kind of probabilities of up and down movements. If we select u and d such that $u > e^{rT} > d$, it can easily be shown that $0 < p < 1$.

We can also demonstrate that the expected rate of return of the asset price equals the risk free rate: $[\, pS_0u + (1 – p)\, S_0d\,]= S_0e^{rT}$
This is known as using *risk-neutral valuation.* We note that p is a pseudo-probability, and does not represent any realistic probability distribution.

Numerical example





Initial parameters: S_0=20, r= 12%, T=0.25. Since p is the probability that gives a return on the stock equal to the risk-free rate, we can find it from:

$$20e^{0.12\times0.25} = 22p + 18(1 – p\ ), \text{ or } p = 0.6523$$

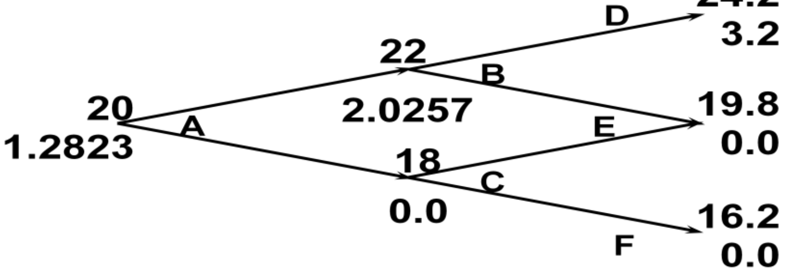
Alternatively, we can use the formula:

$$p = \frac{e^{rT} - d}{u - d} = \frac{e^{0.12\times0.25} - 0.9}{1.1 - 0.9} = 0.6523$$

Using the right diagram, the value of the option can then be calculated accordingly:

$$f = e^{-0.12\times0.25}\,(0.6523\times1 + 0.3477\times0) = 0.633$$

A two-step call option example



The tree configuration can be extended to include more than one time step. The upper number at each node represents the stock price and the lower number is the option value. We start at the last time step (each time step is 3 months = 0.25 year), and the derivative contract to be priced is a European call option with strike K=21; risk free interest rate r=12%.

At nodes D, E, and F, the option value is just the intrinsic value; thus at node D, the value is given by $\max(S - K, 0) = 3.2$.

Moving back one time step, option value at node B is

$$e^{-0.12\times0.25}(0.6523\times3.2 + 0.3477\times0) = 2.0257$$

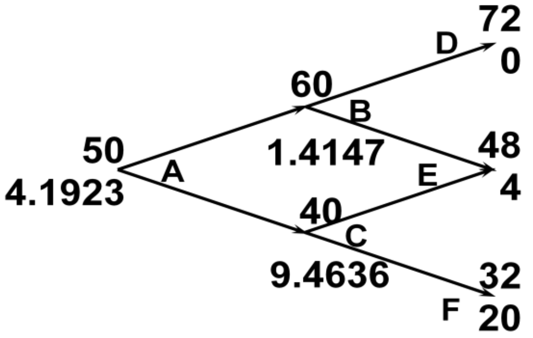
where the probabilities are identical to those obtained in the one-step tree above. After calculating the value at node C in the same way, value at node A is thus:

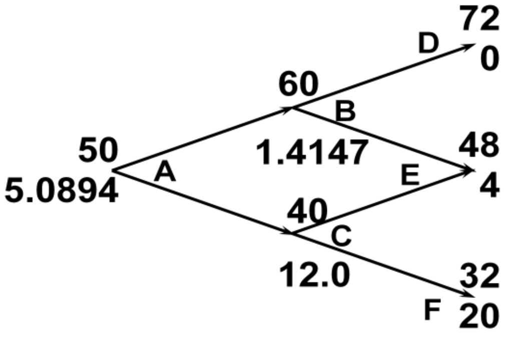
$$e^{-0.12\times0.25}(0.6523\times2.0257 + 0.3477\times0) = 1.2823$$

i.e. option price today = 1.2823.

Put option example

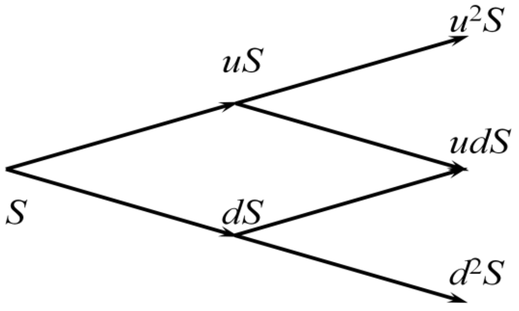
The initial parameters are: $K = 52$, each time step = 1yr, interest rate r = 5%

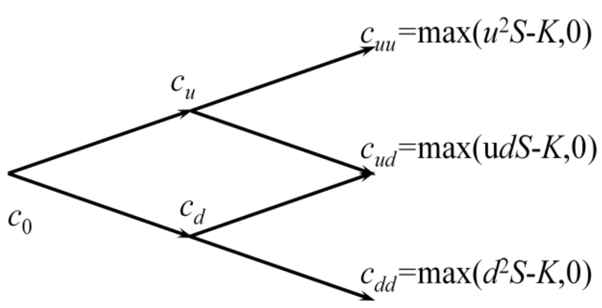




The left diagram represents the European put and the right diagram is an American put. Again we proceed from the last time step (i.e. nodes D, E and F), and the continuation values at nodes B and C can then be obtained accordingly. At node C, if the option cannot be exercised (i.e. a European option), the option value is given as the continuation value, i.e. 9.4636. However, in the case of an American put, the exercise value is given by $\max(K - S, 0) = 12.0$. Thus we can replace the continuation value by the early exercise value in the right diagram. As a result, the final value at node A is higher for the American put.

Multi-period extension




c_{uu}=max(u^2S-K,0)
c_u
c_{ud}=max(udS-K,0)
c_0
c_d
c_{dd}=max(d^2S-K,0)

More formally, the tree can be extended to more than 2 periods. Let Δt be the length of each time step; c_{uu} represents the value of a call option (with strike K) with two consecutive upward moves of the asset price; and $R = \exp(r\Delta t)$. The left diagram shows the asset prices at each node, where u is a upward ratio and d is the downward ratio. In the right diagram, option price at time 0 is given by

$$c_0 = \frac{p^2 c_{uu} + 2p(1-p)c_{ud} + (1-p)^2 c_{dd}}{R^2}$$

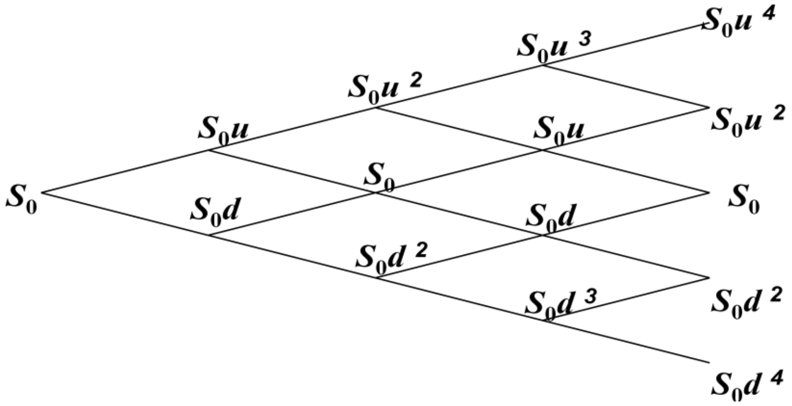
For a tree with n binomial steps, the probability of having j upward moves and $n-j$ downward moves is:

$$probability = C_j^n p^j (1-p)^{n-j}, C_j^n = \frac{n!}{j!(n-j)!}$$

The European call value is thus:

$$c_0 = \frac{\sum_{j=0}^{n} C_j^n p^j (1-p)^{n-j} \max(u^j d^{n-j} S - K, 0)}{R^n}$$

One possible tree configuration is:



- The tree is known as "recombining", meaning that u and d are independent of the time step, i.e. $S_0u_id_{i+1}$ and $S_0d_iu_{i+1}$ will lead to the same node value. Furthermore, in this diagram, $u = 1/d$.
- Number of nodes in an n-step tree is $(n+1)(n+2)/2$
- The corresponding figure for a non-recombining tree is $2^{n+1}-1$.

The above representation can be applied to different types of underlyings. In general, the probability at each branch can be written as:

$$p = \frac{R-d}{u-d}$$

- $R = e^{r\Delta t}$ for a nondividen d paying stock
- $R = e^{(r-q)\Delta t}$ for a stock index wher e q is the dividend yield on the index
- $R = e^{(r-r_f)\Delta t}$ for a currency where r_f is the foreign risk - free rate
- $R = 1$ for a futures contract

As an example, we can calculate the tree parameters for an asset paying a dividend yield of q. Parameters p, u, and d are chosen so that the tree gives correct values for the mean & variance of the stock price changes in a risk-neutral world.

Mean: $SR = pSu + (1-p)Sd$ or $R = pu + (1-p)d$

Variance: $R^2(e^{\sigma^2\Delta t} - 1) = pu^2 + (1-p)d^2 - R^2$

A further condition often imposed is $u = 1/d$. With 3 equations, we can solve for p, u, and d.

$$u = \frac{1}{d} = \frac{R^2e^{\sigma^2\Delta t} + 1 + \sqrt{(R^2e^{\sigma^2\Delta t} + 1)^2 - 4R^2}}{2R}$$

$$p = \frac{R-d}{u-d}$$

$$R = e^{(r-q)\Delta t}$$

When Δt is small, we can approximately u by a simpler expression without sacrificing the order of accuracy (which is of the order of $O(\Delta t)$), so that

$$u = e^{\sigma\sqrt{\Delta t}}, d = e^{-\sigma\sqrt{\Delta t}}$$

Dynamic programming and backward induction

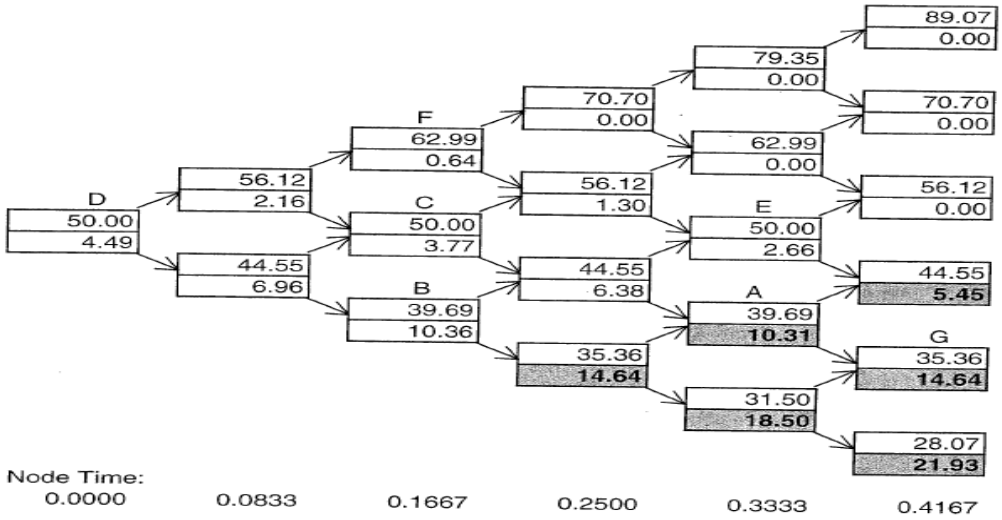
We know the values of the option at the final nodes because they can be calculated simply from the intrinsic values. Then we can work back through the tree using risk-neutral valuation to calculate the value of the option at each node, testing for early exercise when appropriate. Formally, at each tree node, compute $V=\max(V_{cont}, h(S))$, where V_{cont} is the continuation value, being the value obtained from discounting through the risk neutral probability; $h(S)$ is the value obtained through immediate exercise.

Another American put option example

We use the following initial parameters: $S_0 = 50$; $K = 50$; $r = 10\%$; $\sigma = 40\%$; $T = 5$ months = 0.4167; $\Delta t = 1$ month = 0.0833 year. The parameters imply:

$$u = e^{\sigma\sqrt{\Delta t}} = 1.1224, \quad d = e^{-\sigma\sqrt{\Delta t}} = 0.8909, \quad R = e^{r\Delta t} = 1.0084,$$

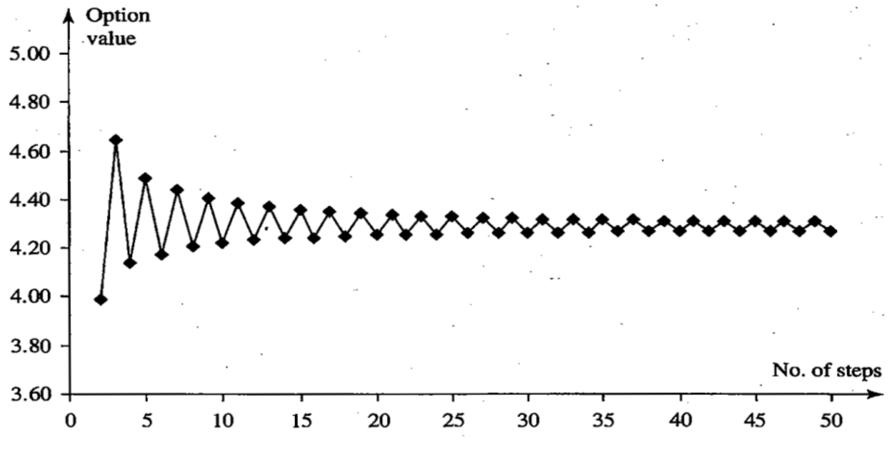
$$p = \frac{R-d}{u-d} = 0.5073, \quad 1 - p = 0.4927.$$



At each node, the upper value is the underlying asset price, the lower value is the option price, and the shaded areas represent the cases where the option is exercised. Option pricing is achieved through three steps:

- Step 1: Calculation of stock price at each node $S_0u^jd^{n-j}$
  e.g. at node A (n=4, j=1), stock price = 50 x 1.1224 x 0.8909^3 = \$39.69
- Step 2: Calculation of the option prices at the final nodes: max ($K - S_T$, 0)
  e.g. at node G, the option price is 50.00 – 35.36 = \$14.64
- Step 3: Backward induction
  Calculate the continuation value and the exercise value at each node; sometimes it should be exercised and sometimes not.
  e.g. at node E, continuation value is (0.5073 x 0 + 0.4927 x 5.45) $e^{-0.10\text{x}0.0833}$ = 2.66, whereas the exercise value is 0 (strike 50, asset price 50); therefore the option value is 2.66. At node A, continuation value is (0.5073 x 5.45 + 0.4927 x 14.64) $e^{-0.10\text{x}0.0833}$ = 9.90, exercise value is max(50 – 39.69, 0) = 10.31; therefore the option value is 10.31 (higher than 9.90). At node B, continuation value is (0.5073 x 6.38 + 0.4927 x 14.64) $e^{-0.10\text{x}0.0833}$ = 10.36, exercise value is max(50 – 39.69, 0) = 10.31; therefore the option value is 10.36. The price of the option is given by the value at the initial node, i.e. \$4.49.

Convergence of option value in a binomial tree



In practice, using more time steps will lead to better accuracy; it is common to use at least 100 steps in a binomial tree.

Binomial tree for stock paying known discrete dividends

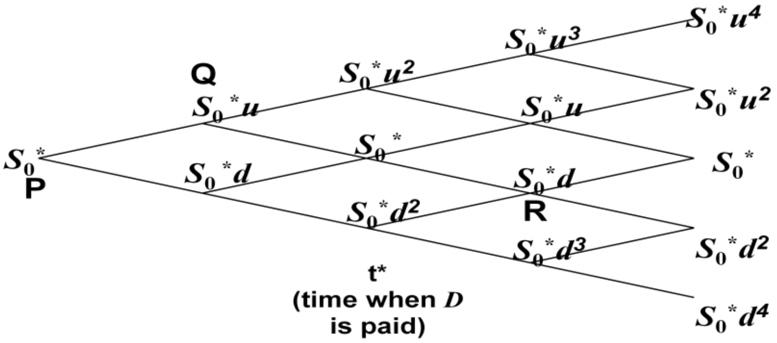
One of the main advantages of using the tree models comes from its flexibility. In the pricing of stock options, a common practical issue is the handling of discrete stock dividends. Dividends are paid in cash at certain dates, instead of being represented as dividend yields. We can modify the tree so that the tree nodes are used to represent the stock price less the present value of the dividends. An "adjusted" tree can then be created by adding the present value of the dividends at each node. This ensures that the unadjusted tree recombines and makes assumptions similar to those when the Black-Scholes model is used.

Suppose there is a known dividend D and the dividend date is at time t^*. The asset price at time t = S_t is split into two components, a risky part S_t* and the known dividend D:

$$S_t^* = \begin{cases} S_t - De^{-r(t^*-t)}, t \le t^* \\ S_t, t > t^* \end{cases}$$

Instead of constructing a tree for S_t, a tree for S_t* is used, where σ* is the volatility of S_t*. A recombining tree structure for S_t* is thus ensured. (Note that it is often assumed that the volatilities of S_t and S_t* are identical, which is theoretically inconsistent; an adjustment is possible but is often ignored.)

We can look at the following example:



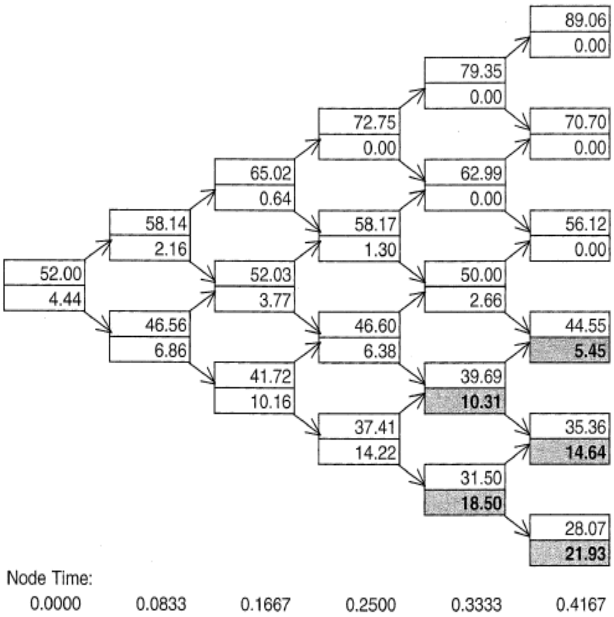
The tree is built based on the values of S*, whereas we need S in order to price an option. From the above relationships, we calculate the asset values S at nodes P, Q, R as S_0*+$De^{-2r\Delta t}$, S_0*u+$De^{-r\Delta t}$, and S_0*d respectively.

We can apply the above in the pricing of an American put option, with the following parameters: S_0 = 52; K = 50; r =10%; σ = 40%; D = 2.06, at 3.5 months; T = 5 months = 0.4167; Δt = 1 month = 0.0833. The parameters imply:

$$u = e^{\sigma\sqrt{\Delta t}} = 1.1224, \quad d = e^{-\sigma\sqrt{\Delta t}} = 0.8909, \quad R = e^{r\Delta t} = 1.0084,$$

$$p = \frac{R-d}{u-d} = 0.5073, \quad 1-p = 0.4927.$$

We construct a tree on S*; assume that the volatility of S* is the same as the volatility of S, which is 40%. Firstly, we note that the present value of the dividend at time 0 is 2.06 $e^{-0.10\text{x}0.0833\text{x}3.5}$ = 2.00, which means that the initial value of S* is 52–2 = 50.



In the tree above, the tree is constructed based on S^*, whereas the upper value in each node represents S. For example, in the upper box at time 0.0833, the stock price is not calculated from 52 x 1.1224 = 58.36, but comes from $50 \text{ x } 1.1224 + 2.06e^{-0.10\text{x}0.0833\text{x}2.5} = 58.14$. Once S is obtained at each node, the option price can again be calculated using backward induction. In this example, option price is found to be 4.44.

Alternative binomial tree geometry

The parameter choices described earlier were suggested by Cox, Ross and Rubinstein (1979), which are popular among practitioners. Other ways of constructing the tree is possible; e.g. Jarrow and Rudd (1983) suggest that, instead of setting $u = 1/d$, we can set each of the two probabilities to 0.5 and

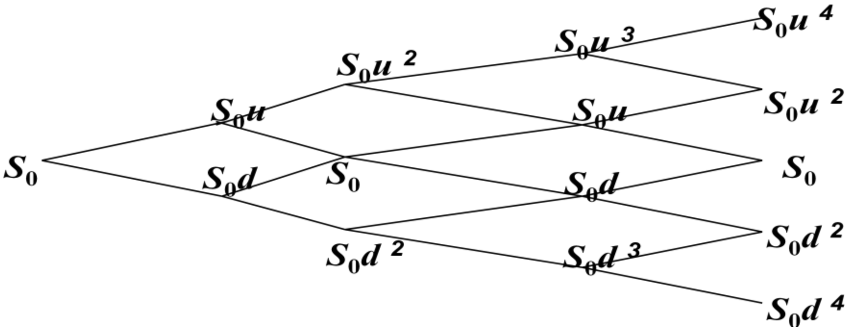
$$u = e^{(r-q-\sigma^2/2)\Delta t+\sigma\sqrt{\Delta t}}$$

$$d = e^{(r-q-\sigma^2/2)\Delta t-\sigma\sqrt{\Delta t}}$$

Extensions of Tree Approach

The basic binomial tree can be modified in a number of ways to take into account more realistic market conditions and specific product characteristics. For example, term structures of interest rates and volatilities are often used, i.e. use $r(t)$ and $\sigma(t)$ in constructing the tree (some comments are made in the section below). If the derivative contract to be priced is a barrier option, we need to develop tricks in order to make the algorithm converge faster, e.g. by placing the nodes right next to the barrier level. Finally, the tree can also be adapted to price some kind of path-dependent options. For example, if we need to price a lookback option, one or more additional values are stored at each tree node, representing the path-dependent state variables. This method is sometimes known as the forward shooting grid algorithm, and is first proposed by Hull and White (1993).

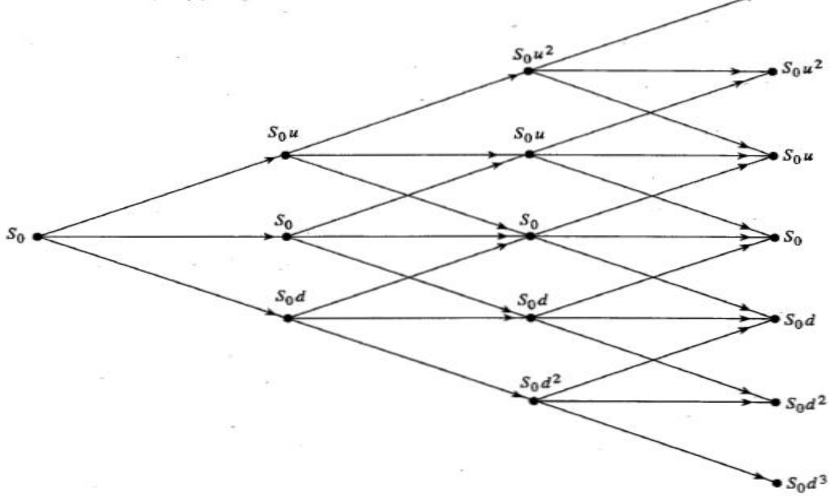
Time Dependent Parameters in a Binomial Tree



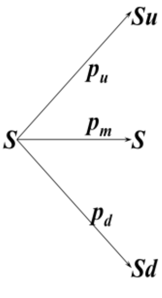
- Making r and/or q a function of time does not affect the geometry of the tree; the only difference is that instead of a constant p, the probabilities on the tree become functions of time
- We can make σ a function of time by making the lengths of the time steps inversely proportional to the variance rate. The tree will recombine as long as $\sigma_i^2\Delta t_i$ is constant

Trinomial tree

Figure 19.12 Trinomial stock price tree.



This method was first proposed in Boyle (1988). Instead of allowing for only two states, a trinomial tree has three branches from each node. It allows for more flexibility due to the extra variable, but it is not as computational efficient as the binomial tree, and there is no guarantee that it can converge faster. We can easily see that the total number of nodes in a n-step tree is $(n+1)^2$.



In this setting, there are six unknowns: the movements u,m,d, and the probabilities to each branch: p_u, p_m, p_d. The problem can be simplified slighting by setting $m=1$, which would reduce the number of unknowns to five. Four equations can be set up:

i) assume $u=1/d$
ii) equating the mean of the stock price distribution (as in the binomial tree)
iii) equating variance of the stock price distribution (as in the binomial tree)
iv) $p_u+p_m+p_d=1$

It means that we are left with four equations and five unknowns, so there is still one free parameter to be determined. With this set of equations, a solution can be:

$$u = e^{\lambda\sigma\sqrt{\Delta t}},\ m = 1,\ d = 1/u$$

$$p_u = \frac{1}{2\lambda^2} + \frac{\sqrt{\Delta t}}{2\lambda\sigma}\left(r - \frac{\sigma^2}{2}\right)$$

$$p_m = 1 - \frac{1}{\lambda^2}$$

$$p_d = \frac{1}{2\lambda^2} - \frac{\sqrt{\Delta t}}{2\lambda\sigma}\left(r - \frac{\sigma^2}{2}\right)$$

λ is a free parameter which supposedly can be arbitrarily chosen. In order to achieve numerical stability, it would be desirable for the probabilities at each branch to be within reasonable limits. Popular choices of λ are $\lambda = \sqrt{\frac{3}{2}}, \sqrt{2} \text{ or } \sqrt{3}$

Note that if we set $\lambda = 1$, the trinomial tree reduces to a binomial tree.

Time dependent parameters in a trinomial tree

In order to have a recombining tree, we only need to make sure that upward and downward movements are identical. In the trinomial tree, the steps depend on

$\lambda\sigma\sqrt{\Delta t}$.

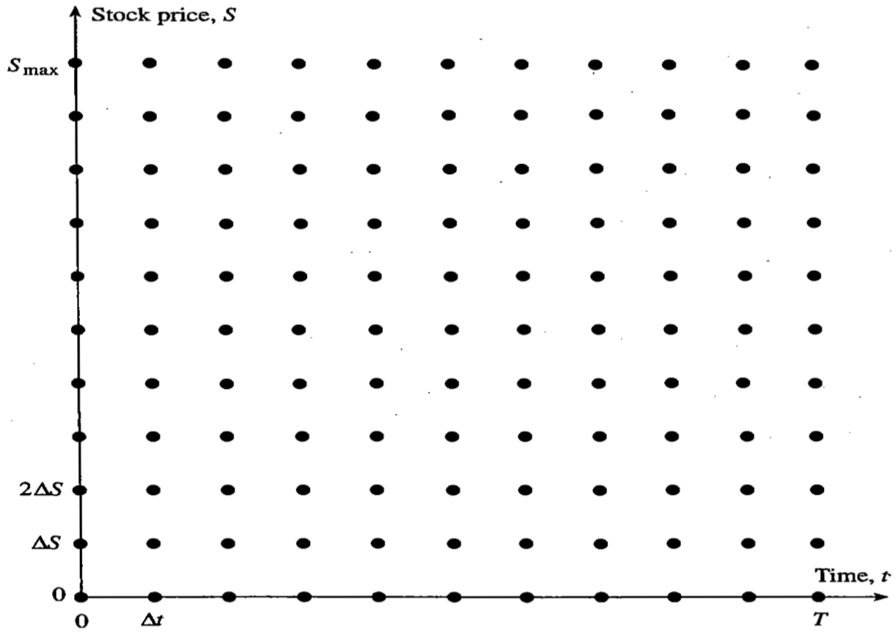
Note that the tree configuration does not require a constant λ; therefore the straightforward way of incorporating the volatility term structure σ(t) is to vary λ, i.e.

$$\lambda(t)\sigma(t)\sqrt{\Delta t} = \text{constant}$$

Unlike the binomial tree, there is no need to vary the time step Δt because the time dependency can be adjusted through $\lambda(t)$. It makes the trinomial scheme more regular and would increase its numerical stability.

7.5.2. Finite difference method

Finite difference methods are well developed and studied in other physical sciences. Basically it aims to represent a differential equation in the form of a difference equation. Applying to the Black-Scholes PDE, we form a grid by considering equally spaced time values and stock price values. Define $f_{i,j}$ as the value of f at time $i\Delta t$ when the stock price is $j\Delta S$. An example of a grid for the finite difference approach is shown below:



Similar to the tree method, each point represents a state of the underlying asset price at a particular time. There are various ways of setting up the relationships between the points. For example, in the implicit finite difference method, we have the following representation:

$$\text{In } \frac{\partial f}{\partial t}+rS\frac{\partial f}{\partial S}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 f}{\partial S^2}=rf \text{ (Black - Scholes PDE)}$$

$$\text{we set } \quad \frac{\partial f}{\partial S}=\frac{f_{i,j+1}-f_{i,j-1}}{2\Delta S}\text{ (centered difference)}$$

$$\frac{\partial^2 f}{\partial S^2}=\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\Delta S^2}$$

$$\text{If we also set } \quad \frac{\partial f}{\partial t}=\frac{f_{i+1,j}-f_{i,j}}{\Delta t}\text{ (forward difference)}$$

we obtain the implicit finite difference method. This involves solving simultaneous equations of the form:

$$a_j f_{i,j-1}+b_j f_{i,j}+c_j f_{i,j+1}=f_{i+1,j}$$

$$a_j=\frac{1}{2}rj\delta t-\frac{1}{2}\sigma^2 j^2\delta t,\ b_j=1+\sigma^2 j^2\delta t+r\delta t, c_j=-\frac{1}{2}rj\delta t-\frac{1}{2}\sigma^2 j^2\delta t$$

We can then apply some boundary conditions; e.g. in valuing a put option where the payoff at time T is $\max(K-S_T,0)$

$$f_{N,j}=\max(K-j\delta S,0), j=0,1,\cdots,M$$
$$f_{i,0}=K, i=0,1,\cdots,N$$
$$f_{i,M}=0, i=0,1,\cdots,N \text{ (an approximation)}$$

We want to solve for all $f_{0,j}$, and the value of the derivative will then be given by one of these values (at the origin, i.e. $f_{0,S0}$). In the above example, we can proceed from the points at time $T-\delta t$; with $i=N-1$:

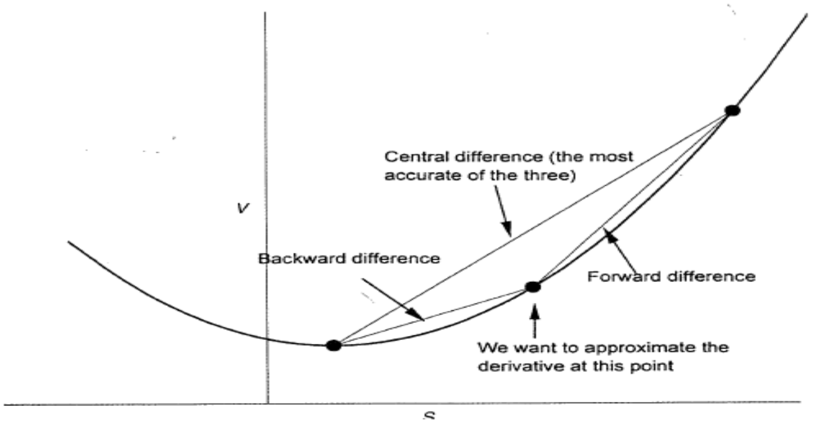
$$a_j f_{N-1,j-1} + b_j f_{N-1,j} + c_j f_{N-1,j+1} = f_{N,j}$$
$$j = 1,2,\cdots,M-1$$

This is a set of simultaneous equations that can be solved for the $M-1$ unknowns:

$$f_{N-1,1}, f_{N-1,2}, \cdots f_{N-1,M-1}$$

The nodes at time $T-2\delta t$ can then be solved in a similar way.

Different representations of the first derivative


Central difference (the most accurate of the three)
V
Backward difference
Forward difference
We want to approximate the derivative at this point
S

Various representations can be used for the first derivative. The above diagram shows the respective forward, backward and centered difference at the ($j\Delta S$, $i\Delta t$) node. The formulas and the order of accuracy are:

$$\text{forward}: \frac{f(x+\Delta x)-f(x)}{\Delta x} = f'(x) + \frac{f''(x)}{2}\Delta x + O(\Delta x^2)$$
$$\text{backward}: \frac{f(x)-f(x-\Delta x)}{\Delta x} = f'(x) - \frac{f''(x)}{2}\Delta x + O(\Delta x^2)$$
$$\text{centered}: \frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x} = f'(x) + \frac{f'''(x)}{6}\Delta x^2 + O(\Delta x^4)$$

It is seen that the centered difference can achieve an order of accuracy of Δx^2, whereas the forward or backward difference can only achieve Δx. However, the centered difference involves two extra calculations in addition to the function valuation $f(x)$ whereas the other two methods require only one extra calculation, hence it is not as efficient as the other methods.

Explicit Finite Difference Method

In one version of the explicit finite difference scheme, we assume that $\partial f / \partial S$ and $\partial^2 f / \partial S^2$ to be the same at the $(i+1, j)$ point as they are at the (i, j) point

$$\frac{\partial f}{\partial S} = \frac{f_{i+1,j+1} - f_{i+1,j-1}}{2\delta S}$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i+1,j+1} + f_{i+1,j-1} - 2f_{i+1,j}}{\delta S^2}$$

The scheme involves solving equations of the form:

$$f_{i,j} = a_j^* f_{i+1,j-1} + b_j^* f_{i+1,j} + c_j^* f_{i+1,j+1}$$

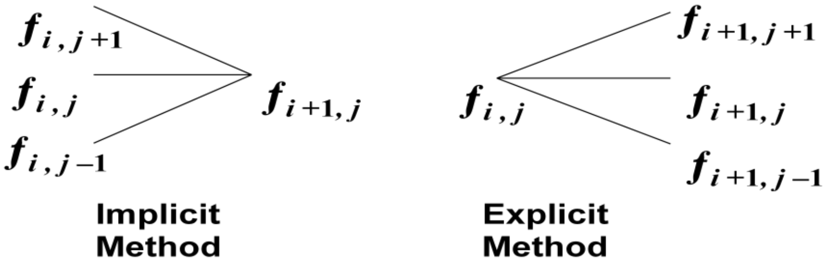
$$a_j^* = \frac{1}{1+r\delta t}\left(-\frac{1}{2}rj\delta t + \frac{1}{2}\sigma^2 j^2 \delta t\right)$$

$$b_j^* = \frac{1}{1+r\delta t}\left(1 - \sigma^2 j^2 \delta t\right)$$

$$c_j^* = \frac{1}{1+r\delta t}\left(\frac{1}{2}rj\delta t + \frac{1}{2}\sigma^2 j^2 \delta t\right)$$

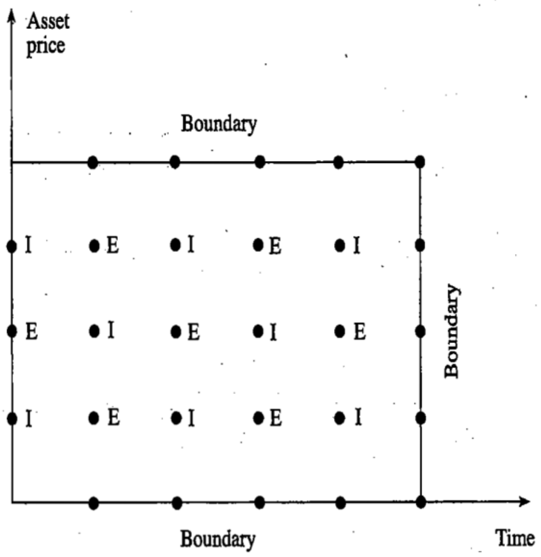
The main advantage of the explicit method over the implicit method is that $f_{i,j}$ can now be obtained directly, instead of requiring the solution of simultaneous equations. However, note that all the coefficients a_j^*, b_j^* and c_j^* should be positive in order for the method to be stable.

A diagrammatic representation of the explicit and implicit methods is shown below:



As can be seen above, the explicit finite difference method is equivalent to the trinomial tree approach. The explicit method is easy to implement, and it is second order accurate in the price direction, but only first order accurate in the time direction. However, it can be inherently unstable, i.e. small errors due to arithmetic inaccuracies or approximations will tend to accumulate and grow.

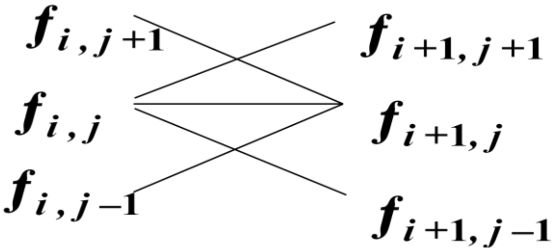
Alternative approaches



Many other kinds of approximations have been proposed as a compromise between efficiency and accuracy. An example is the *hopscotch method*, which alternate between the implicit (I) and explicit (E) methods.

In order to achieve faster convergence than the implicit method (i.e. needing less spatial intervals), the *Crank-Nicolson scheme* takes the “average” of the explicit and implicit schemes. Note that this is still an implicit scheme and the solution requires solving a system of simultaneous equations

$$f_{i,j} = a_j f_{i-1,j-1} + b_j f_{i-1,j} + c_j f_{i-1,j+1}$$
$$f_{i-1,j} = a_j^* f_{i,j-1} + b_j^* f_{i,j} + c_j^* f_{i,j+1}$$
$$f_{i,j} + f_{i-1,j} = a_j f_{i-1,j-1} + b_j f_{i-1,j} + c_j f_{i-1,j+1} + a_j^* f_{i,j-1} + b_j^* f_{i,j} + c_j^* f_{i,j+1}$$



Crank-Nicolson is an example of a class of two-level six-point scheme, involving three option values at each of the two time intervals i and i+1.

PDE with transformed variable

In order to improve the stability of the numerical scheme, it is possible to construct the grid with a transformation. A common method is to use ln S rather than S as the underlying variable. For example, we can define $Z = \ln S$, and the Black-Scholes PDE becomes

$$\frac{\partial f}{\partial t} + (r - \sigma^2 / 2)\frac{\partial f}{\partial Z} + \frac{1}{2}\sigma^2 \frac{\partial^2 f}{\partial Z^2} = rf$$

In the implicit scheme,

$$f_{i+1,j} = \alpha_j f_{i,j-1} + \beta_j f_{i,j} + \gamma_j f_{i,j+1}$$

$$\alpha_j = \frac{\delta t}{2\delta Z}(r - \sigma^2/2) - \frac{\delta t}{2\delta Z^2}\sigma^2$$

$$\beta_j = 1 + \frac{\delta t}{\delta Z^2}\sigma^2 + r\delta t$$

$$\gamma_j = -\frac{\delta t}{2\delta Z}(r - \sigma^2/2) - \frac{\delta t}{2\delta Z^2}\sigma^2$$

Note that the coefficients α_j, β_j and γ_j are independent of state j, i.e. they are constants throughout the grid. However, if we continue to use regularly spaced points in the transformed domain, the asset price will have uniform increment in $\ln S$ rather than S. In fact this transformation is similar to the proportional jumps on the asset price as seen in the binomial/trinomial tree models.

Steps in constructing finite difference schemes

The main steps in the construction of a finite difference scheme involve the following:

- Derivation of the governing model equation. The Black-Scholes PDE is a straightforward example.
- Discretization of the equation, by choosing the state variables.
- Choice of computational domain, i.e. determination of the range of values that the state variables can attain. It usually involves some truncation of the infinite domain.
- Approximation of the auxilliary conditions, including the initial conditions from the terminal payoff structure
- Choice of scheme depends on its stability and efficiency. For example, most implicit schemes are unconditional stable, but may not be as efficient as the explicit schemes.

7.5.3. Monte Carlo method[2]

Theoretical basis of Monte Carlo method

Two mathematical theorems can be used to justify the validity of the Monte Carlo method. In the strong law of large numbers, the arithmetic mean of the realizations of X_i tends to its expectation μ:

$$\mu = E(X_i)$$

$$\frac{1}{n}\sum_{i=1}^{n} X_i \to \mu \text{ as } n \to \infty$$

With the central limit theorem, the normalized error tends to follow the standard normal distribution:

$$\frac{\sum_{i=1}^{N} X_i - N\mu}{\sqrt{N}\sigma} \to \Phi(0,1) \text{ as } N \to \infty$$

[2] Reference: Hull, 7th edition, chapters 19 and 26; 8th edition, chapters 20 and 26.

How do we get random numbers?

The Monte Carlo method is a numerical procedure for estimating the expected value of a random variable, thus the quality of the random number used is very important. In the past, random number generators with short periods can often give rise to erroneous conclusions as the period of the numbers generated are not well understood. In other words, although the test is run many times, the period can be too short which led to repetitive results, which means that increasing the number of runs cannot improve accuracy.

According to Korn et al. (2009), good random number generators should fulfill the following criteria:[3]

- numbers generated should be evenly distributed
- Speed and memory requirements
- Generate independent, identically distributed (i.i.d.) samples
- Results should be reproducible, which is useful for debugging purposes
- Portability – one that is not machine dependent
- Structure of the random points – not all lie in the same hyperplane

Random number generator example

The earliest and easiest type of random number generator is called the Linear Congruential Generator (LCG), which has the form

$$s_{n+1} = (as_n + c) \bmod m, n \in \mathrm{N}$$

$$u_n = \frac{s_n}{m} \text{ where } u_n \text{ lies between } [0,1)$$

Starting with an initial seed for s_0, this generator would produce random numbers with a period of m–1, which is usually converted to limit the output within a range of 0 to 1 (a unit random number). In a version proposed by L'Ecuyer, $m = 2^{31}–249$, $a = 40692$, $c = 0$; this generator can generate numbers with a period of $2^{31}–250$.

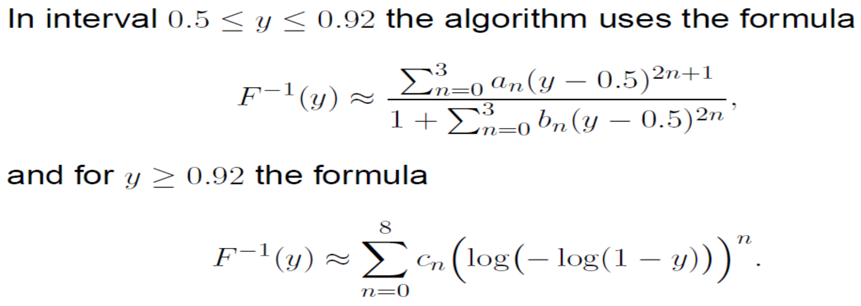
A state-of-the-art generator was proposed by Matsumoto and Nishimura in 1998, known as Mersenne Twister MT19937. The period of this generator is tremendously large, equal to $2^{19937} – 1$, and there is minimal serial correlation up to a dimension of 625. It is a standard generator in packages such as Matlab and S-plus.

Sampling from Normal Distribution

A task that we encounter very often in Monte Carlo simulations (especially in finance applications) is to generate normally distributed random numbers. Formally, we generate a random number u on $[0,1)$, and then obtain the Gaussian sample via $x = \phi^{-1}(u)$. One simple way to obtain a sample from $\phi(0,1)$ is to generate 12 random numbers between 0.0 & 1.0, take the sum, and subtract 6.0. {In Excel, "=NORMSINV(RAND())" gives a random sample from $\phi(0,1)$}.

[3] Ralf Korn, Elke Korn, and Gerald Kroisandt, *Monte Carlo Methods and Models in Finance and Insurance*, Chapman, 2009, pp.6-7.

Several analytical approximations of the inverse normal distribution exist, which can be very accurate. One of these algorithms is the Beasley-Springer Moro formula (accuracy up to 3x10^{-9}):



In interval $0.5 \leq y \leq 0.92$ the algorithm uses the formula

$$F^{-1}(y) \approx \frac{\sum_{n=0}^{3} a_n (y-0.5)^{2n+1}}{1+\sum_{n=0}^{3} b_n (y-0.5)^{2n}},$$

and for $y \geq 0.92$ the formula

$$F^{-1}(y) \approx \sum_{n=0}^{8} c_n \Big(\log(-\log(1-y))\Big)^n.$$

The constants can be found in Paul Glasserman, *Monte Carlo Methods in Financial Engineering* (Springer: 2003), 68.

a_0 = 2.50662823884	b_0 = –8.47351093090
a_1 = -18.61500062529	b_1 = 23.08336743743
a_2 = 41.39119773534	b_2 = -21.06224101826
a_3 = -25.44106049637	b_3 = 3.13082909833
c_0 = 0.3374754822726147	c_5 = 0.0003951896511919
c_1 = 0.9761690190917186	c_6 = 0.0000321767881768
c_2 = 0.1607979714918209	c_7 = 0.0000002888167364
c_3 = 0.0276438810333863	c_8 = 0.0000003960315187
c_4 = 0.0038405729373609	

Another common algorithm is the Box-Muller method. If x_1 and x_2 are uniform deviates on (0,1), it can be shown that y_1 and y_2 are independently distributed according to $\phi(0,1)$.

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

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

However, computing sin() and cos() can be slow; one trick to avoid these is suggested in Press et al (1992). First, we pick v_1 and v_2 as the ordinate and abscissa of a random point inside a unit circle around the origin. This is achieved by rejecting the numbers where

$$R^2 = v_1^2 + v_2^2 >= 1 \text{ or } = 0$$

From this transformation, R^2 is a uniform deviate, and thus it can be used to replace x_1. The angle that (v_1, v_2) defines with respect to the v_1 axis can serve as the random angle $2\pi x_2$, where $\cos(2\pi x_2) = v_1/R$ and $\sin(2\pi x_2) = v_2/R$. Thus we replace the formulas to become

$$y_1 = \sqrt{\frac{-2\ln R^2}{R^2}}v_1,\ y_2 = \sqrt{\frac{-2\ln R^2}{R^2}}v_2$$

Note that there are no trigonometric calls and hence the method is computationally more efficient.

Multivariate random variables

There exist many derivative contracts where the prices depend on several underlying variables : $\theta_i\ (1 \leq i \leq n)$. We can simulate paths for each of them in a risk-neutral world to calculate the values for the derivative, where each of these variables will be correlated with each other.

In the simplest case where we want to generate two correlated normal samples ε_1 and ε_2 (i.e. $n = 2$), we can first obtain independent normal samples x_1 and x_2 from a random number generator. Then we obtain ε_1 and ε_2 as follows:

$$\varepsilon_1 = x_1,\ \varepsilon_2 = \rho x_1 + x_2\sqrt{1-\rho^2}$$

ρ is the correlation coefficient between ε_1 and ε_2. To verify the results

$$\begin{aligned} \operatorname{cov}(\varepsilon_1,\varepsilon_2) &= \operatorname{cov}(x_1, \rho x_1 + x_2\sqrt{1-\rho^2}) \\ &= \rho\operatorname{cov}(x_1,x_1) + \sqrt{1-\rho^2}\operatorname{cov}(x_1,x_2) \\ &= \rho \\ \text{since } \operatorname{cov}(x_1,x_1) &= \operatorname{var}(x_1) = 1,\ \operatorname{cov}(x_1,x_2) = 0 \end{aligned}$$

A general method of obtaining correlated samples is known as the Cholesky decomposition. We can define:

$$\begin{aligned} &\varepsilon = Mx \\ &E[\varepsilon\varepsilon^T] = \Sigma, \text{ which is the covariance matrix} \\ &MM^T = \Sigma, \text{ where } M \text{ is lower triangular} \end{aligned}$$

For example for $n = 3$, we have the following

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} = \begin{pmatrix} \alpha_{11} & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \alpha_{11}x_1 \\ \alpha_{21}x_1 + \alpha_{22}x_2 \\ \alpha_{31}x_1 + \alpha_{32}x_1 + \alpha_{33}x_1 \end{pmatrix}$$

where each coefficient in the matrix can be solved in an efficient manner.

Euler-Maruyama scheme for stochastic differential equations

Assume that a state variable $X(t)$ follows the stochastic differential equation:

$$dX(t) = a(t, X(t))dt + \sigma(t, X(t))dW(t)$$

The SDE can be approximated by the following algorithm:

```
1) Let $\Delta t = T/N$, set $Y_N(0) = X(0) = x_0$
2) For $j = 0$ to $N-1$ do
(a) Simulate a standard normally distributed variable $Z_j$
(b) Set $\Delta W(j\Delta t) = \sqrt{\Delta t}Z_j$ and
$Y_N((j+1)\Delta t) = Y_N(j\Delta t) + a(j\Delta t, Y_N(j\Delta t))\Delta t + \sigma(j\Delta t, Y_N(j\Delta t))\Delta W(j\Delta t)$
```

A simple way to extend the Euler-Maruyama scheme is to include an additional term to account for the error in the discretization. One such method is the Milstein scheme:

```
1) Let $\Delta t = T/N$, set $Y_N(0) = X(0) = x_0$
2) For $j = 0$ to $N-1$ do
(a) Simulate a standard normally distributed variable $Z_j$
(b) Set $\Delta W(j\Delta t) = \sqrt{\Delta t} Z_j$ and
$Y_N((j+1)\Delta t) = Y_N(j\Delta t) + a(j\Delta t, Y_N(j\Delta t))\Delta t + \sigma(j\Delta t, Y_N(j\Delta t))\Delta W(j\Delta t)$
        $+ \frac{1}{2}\sigma(j\Delta t, Y_N(j\Delta t))\sigma'(j\Delta t, Y_N(j\Delta t))(\Delta W(j\Delta t)^2 - \Delta t)$
```

With the additional term, the computation time will increase. However, while the discretization error is reduced, results are mixed in terms of whether the more accurate representation can improve the convergence speed of the Monte Carlo method in general.

Applications of Monte Carlo simulation

Compared with the other numerical methods, Monte Carlo simulation is more suitable in terms of dealing with some specific derivative pricing problems:

- path dependent options
- options dependent on several underlying state variables (no curse of dimensionality)
- options with complex payoffs
- Complex underlying stochastic processes

However, if there is only one state variable, Monte Carlo methods can be time consuming, and cannot easily deal with American-style options (but note the discussion in section 7.5.4.)

Crude Monte Carlo method

Irrespective of the comments above, we start with describing the simplest derivative pricing problem: European call option on a single stock. The required calculation is:

$$e^{-rT} E[\max(S_T - K, 0)]$$

Monte Carlo simulation corresponding to the above formula involves the following steps:

- Simulate one path for the stock price in a risk neutral world
- Calculate the payoff from the stock option
- Repeat these steps many times to get many sample payoffs
- Calculate mean payoff
- Discount the mean payoff at risk free rate

In order to enhance efficiency, the aim is to perform the minimum number of computations within the main calculation routine. In the example above, the discounting factor can be applied to the average payoff instead of being applied per path. For other kinds of options, it may be necessary to generate the stock prices at intermediate time steps. However, the number of time steps required depends on whether there are intermediate cash flows or event dates.

In a risk neutral world the process for a stock price is (assume dividend = 0)

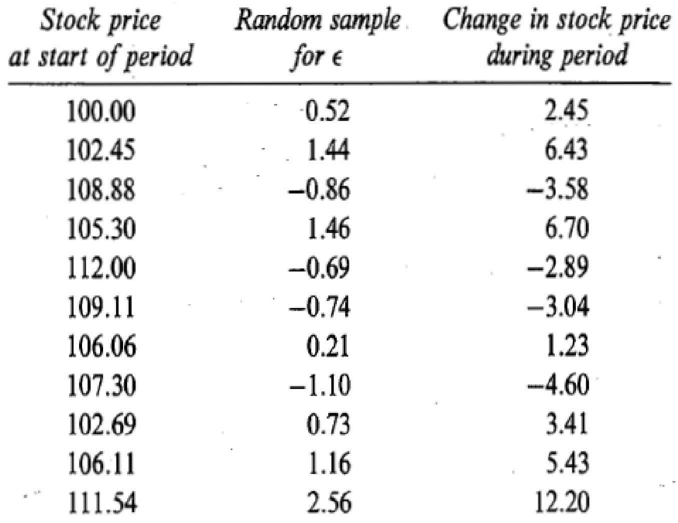
$$dS = rS\,dt + \sigma S\,dz$$

We can simulate a path by choosing time steps of length Δt and using the discrete version of this:

$$\Delta S = rS\,\Delta t + \sigma S\,\varepsilon\sqrt{\Delta t}$$

where ε is a random sample from $\phi(0,1)$.

Example 1: stock price path (from Hull, 7th edition)



Stock price at start of period	*Random sample for ϵ*	*Change in stock price during period*
100.00	0.52	2.45
102.45	1.44	6.43
108.88	−0.86	−3.58
105.30	1.46	6.70
112.00	−0.69	−2.89
109.11	−0.74	−3.04
106.06	0.21	1.23
107.30	−1.10	−4.60
102.69	0.73	3.41
106.11	1.16	5.43
111.54	2.56	12.20

$$r = 15\%,\ \sigma = 30\%,\ \Delta t = 1\,\text{week} = 0.0192$$

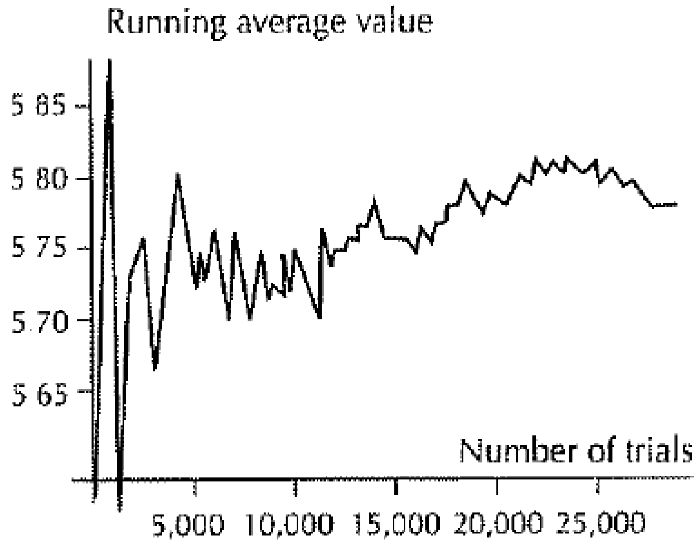
$$\frac{\Delta S}{S} = 0.15\Delta t + 0.30\varepsilon\sqrt{\Delta t}$$

$$\Rightarrow \Delta S = 0.00288S + 0.0416S\varepsilon$$

At the start of the period, initial stock price S=100.00 and the random sample ε is 0.52. In the next period, the generated stock price is calculated from:

ΔS_0 = 0.00288x100 + 0.0416x100x0.52= 2.45, $S_1 = S_0 + \Delta S_0$ = 100.00 + 2.45 = 102.45

Example 2: a test of the Black-Scholes formula (from Luenberger (1998))



The option to be priced has the following characteristics: European call option, S_0=\$62, K=\$60, σ=20%, r=12%, T=5 months = 0.417 years divided into 80 time intervals, Black-Scholes price = \$5.80. In this test, we can see that by increasing the number of trials, the result obtained from the Monte Carlo simulation would slowly converge to the true answer as indicated by the Black-Scholes result.

A more accurate approach

When possible, it is more accurate to use an exact solution of the SDE rather than relying on discretization. In the standard dynamics above, we can use the following:

$$d \ln S = \left(r - \sigma^2/2\right)dt + \sigma\, dz$$

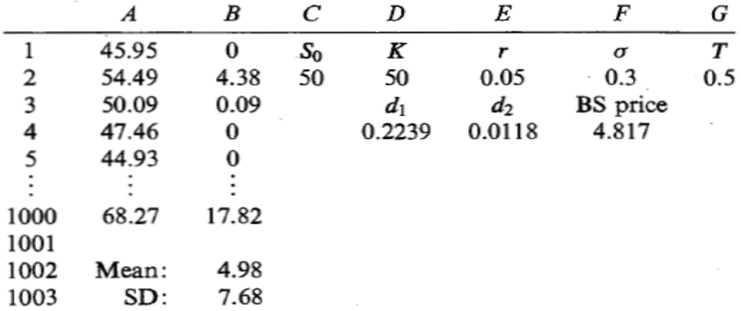
The discrete version of this is

$$\ln S(t + \Delta t) - \ln S(t) = \left(r - \sigma^2/2\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}$$

or

$$S(t + \Delta t) = S(t)\, e^{\left(r - \sigma^2/2\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}}$$

Another test of the Black-Scholes formula (from Hull, 7th edition)



	A	B	C	D	E	F	G
1	45.95	0	S_0	K	r	σ	T
2	54.49	4.38	50	50	0.05	0.3	0.5
3	50.09	0.09		d_1	d_2	BS price	
4	47.46	0		0.2239	0.0118	4.817	
5	44.93	0					
⋮	⋮	⋮					
1000	68.27	17.82					
1001							
1002	Mean:	4.98					
1003	SD:	7.68					

If the volatility and interest rate are not time dependent, the solution to the SDE can directly be written as:

$$S(T) = S(0)\exp\left[\left(r - \frac{\sigma^2}{2}\right)T + \sigma\varepsilon\sqrt{T}\right]$$

Normally, in order to have sufficient accuracy in the discretization process, we would need to have a sufficiently small Δt (typically 1 year should be divided into 100 time steps or more). However, in the above example, we only simulate 1000 times, where each path has only 1 time step (equal to T = 0.5 year). The reason is because the terminal distribution of $S(T)$ can be generated exactly which follows the pre-defined criteria. Column A shows the simulated final stock price $S(T)$, and column B represents the discounted payoff given the simulated stock price at column A. While the number of paths required in order to generate an acceptable answer may be comparable to the crude Monte Carlo method, the number of computations is much reduced because of the savings in the time steps.

Example 3: the Heston model

Refer to section 7.4.2 for a discussion of the Heston (1993) model (assume q=0). The following is based on the Euler-Maruyama scheme:

```
1) Let $\Delta t = T/N$, set $V(0) = V_0, S(0) = S_0$
2) For $j = 0$ to $N-1$ do
(a) Simulate two standard normally distributed variables $Y$ and $Z$
(b) Set $W = \rho Z + \sqrt{1-\rho^2}Y$
(c) Update the volatility
$V(j\Delta t) = V((j-1)\Delta t) + a(V_L - V((j-1)\Delta t))\Delta t + \xi\sqrt{V((j-1)\Delta t)}\sqrt{\Delta t}W$
(d) Update the log-stock price $X(t) = \ln(S(t))$
$X(j\Delta t) = X((j-1)\Delta t) + \left(r - \frac{1}{2}V((j-1)\Delta t)\right)\Delta t + \sqrt{V((j-1)\Delta t)}\sqrt{\Delta t}Z$
```

Note that there is one potential problem – step (c) can potentially produce a negative volatility, which should not be allowed. Special tricks will be required to resolve this issue.

Variance reduction techniques

The naïve Monte Carlo method may only converge slowly to a correct answer. Some techniques have been applied to speed up the convergence. A simple method is the introduction of *antithetic variates.* Variance reduction is achieved by introducing symmetry in the discretization process. If X is a random variable uniformly distributed on [0,1], the crude Monte Carlo estimate is:

$$\overline{f}(X) = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

The antithetic estimator is given by

$$\overline{f}_{anti}(X) = \frac{1}{2}\left(\frac{1}{N}\sum_{i=1}^{N} f(X_i) + \frac{1}{N}\sum_{i=1}^{N} f(1-X_i)\right)$$

Note that as both $f(X_{\text{i}})$ and $f(1-X_{\text{i}})$ are unbiased estimators of $f(X)$, the antithetic estimate is also unbiased. If the variance of $f(X)$ with N samples = σ^2, the variance of the antithetic estimator is:

$$\text{var}(\overline{f}(X) + \overline{f}(1-X)) = 2\sigma^2 + 2\,\text{cov}(\overline{f}(X), \overline{f}(1-X))$$

i.e. there is a reduction in variance if $f(X)$ and $f(1-X_{\text{i}})$ are negatively correlated. Heuristically this is reasonable as errors from the first simulation can be negated by the errors of the simulation using symmetric variates. Furthermore, the random inputs obtained from the collection of the antithetic pair are more evenly distributed than a collection of $2N$ independent samples. However, in many realistic applications, the improvement in convergence is not too significant. Nevertheless as introducing the antithetic variates will not require any increase in computation time, this method should always be implemented as the result will not be worse than the basic Monte Carlo method.

In the *control variate* method, we want to price an option X with Monte Carlo, whereas the price of a similar option Y can be obtained accurately (maybe through an analytical formula). At the same time, the price of Y is calculated using the same Monte Carlo paths as for X. Using the following notation:

$$\begin{aligned} &V_X, V_Y : \text{true values of options X and Y} \\ &\overline{V_X}, \overline{V_Y} : \text{estimated values of the options using Monte Carlo} \end{aligned}$$

The control variate method assumes that

$$V_X - \overline{V_X} \approx V_Y - \overline{V_Y} \Rightarrow \overline{V_X^{CV}} = \overline{V_X} + (V_Y - \overline{V_Y})$$

i.e. we can use the error in the control variate Y to provide a better estimate of V_X. The variance of the estimator can easily be calculated:

$$\begin{aligned} &\overline{V_X^{CV}} = \overline{V_X} - \overline{V_Y} + V_Y \\ &\operatorname{var}(\overline{V_X^{CV}}) = \operatorname{var}(\overline{V_X}) + \operatorname{var}(\overline{V_Y}) - 2\operatorname{cov}(\overline{V_X}, \overline{V_Y}) \end{aligned}$$

Variance reduction is achieved if $\operatorname{var}(\overline{V_Y}) < 2\operatorname{cov}(\overline{V_X}, \overline{V_Y})$. This is true if Y is such chosen that it is closely related to X, so that the covariance between the values of X and Y is high.

Theoretically, better convergence is possible if we optimize the control variate via:

$$\begin{aligned} &\overline{V_X^{\beta}} = \overline{V_X} + \beta(V_Y - \overline{V_Y}) \\ &\operatorname{var}(\overline{V_X^{\beta}}) = \operatorname{var}(\overline{V_X}) + \beta^2 \operatorname{var}(\overline{V_Y}) - 2\beta \operatorname{cov}(\overline{V_X}, \overline{V_Y}) \\ &\operatorname{var}(\overline{V_X^{\beta}}) \text{ is minimzed when } \beta^* = \frac{\operatorname{cov}(\overline{V_X}, \overline{V_Y})}{\operatorname{var}(\overline{V_Y})}, \text{ so that} \\ &\operatorname{var}(\overline{V_X^{\beta}})\Big|_{\beta=\beta^*} = \operatorname{var}(\overline{V_X}) - \frac{\operatorname{cov}(\overline{V_X}, \overline{V_Y})^2}{\operatorname{var}(\overline{V_Y})} \end{aligned}$$

The difficulty is that in general, the covariance term is not available, and it has to be estimated through regression.

Some examples of control variates include:

- Use geometric averaging Asian option as the control variate for arithmetic Asian option
- Use geometric averaging basket option as the control variate for arithmetic basket option
- Use a constant volatility model as the control variate for the stochastic volatility model

Other types of variance reduction methods aim to generate samples with better properties. One such method is known as *stratified sampling*, which samples a small sub-population that mirrors the properties of the total population. Basically the distribution of the random variable X is divided into different parts, and Monte Carlo simulations are applied within each part. However, a disadvantage of this method is that it may not work well if the simulation is based on more than one underlying variable (the curse of dimensionality).

For *importance sampling*, we try to find a distribution of the random variable that assigns a high probability to those values that are important for computing the expectation. Popular methods to obtain importance sampling density include:

- Shifting and/or scaling the density
- Conditional sampling restricted to the important area

However, this method relies on the fact that the characteristics of the distribution are well known before the simulation is performed.

Finally, we introduce the use of *quasi-random sequences*. These sequences are completely deterministic (i.e. not random), but they are *incrementally* evenly distributed.

These are clever ways to insert additional points in the sampling space without clustering. A main advantage of this method is that it is not necessary to decide in advance how many points are needed – the sampling could continue "until" some convergence or termination criterion is met. Examples of quasi-random sequences include Halton, Faure, Sobol, and Niederreiter. They are also known as low-discrepancy sequences.

We can illustrate the idea with a 1-dimensional Halton sequence (also called van der Corput sequence). The j^{th} number H_j in the sequence is obtained from:

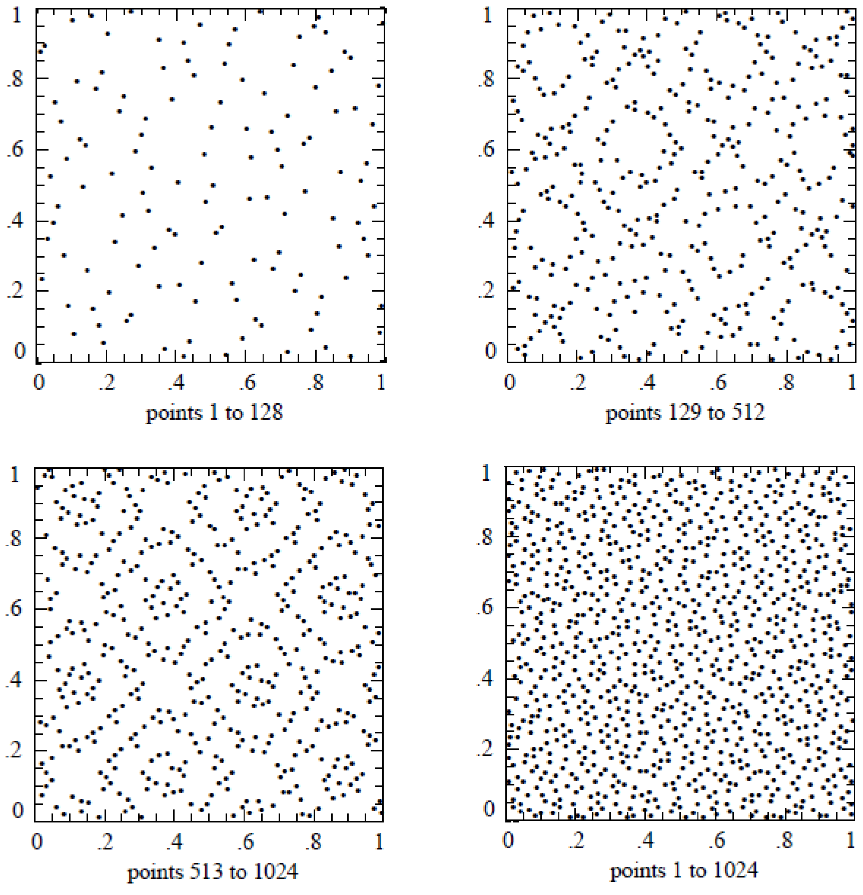
1) write j as a number in base b, where b is a prime number
2) reverse the digits and put a radix (decimal) point in front of the number
3) convert the number back to base 10 and this is H_j

E.g. If we want to find the 17th number in the sequence, j = 17 in base b =5 is 32. Reverse the digits and put a decimal to become 0.23 base 5. Convert back to base 10 via 2x1/5 + 3x1/25 = 0.52.

It is noted that as j increases by 1, the most significant digit in H_j is increased by $1/b$ (1/5 in this example); this results in a kind of "maximally spread-out" order on each grid.

The first 29 numbers in this example are:

0.2, 0.4, 0.6, 0.8, 0.04, 0.24, 0.44, 0.64, 0.84, 0.08, 0.28, 0.48, 0.68, 0.88, 0.12, 0.32, 0.52, 0.72, 0.92, 0.16, 0.36, 0.56, 0.76, 0.96, 0.008, 0.208, 0.408, 0.608, 0.808…



points 1 to 128

points 129 to 512

points 513 to 1024

points 1 to 1024

In another example, the diagrams above show the first 1024 points of a 2-dimensional Sobol sequence.[4] Visually we can see how the algorithm works. As more and more points are inserted, they are not placed in random. Instead, they are inserted in a way so as to sample the domain more or less evenly. This sequence is completely deterministic because the points come from an algorithm and the same results are reproducible.

7.5.4. Valuation of American path-dependent options

There used to be a general belief that simulation methods can only be used to price European style options, apparently due to the backward nature of the early exercise feature. There is no way to know whether exercise is optimal at a particular level of the state variable. Algorithms dealing with this problem started to appear in the mid-1990s. The most popular approach in recent years was proposed by Longstaff and Schwartz (2001), where basis functions and regression techniques are used to approximate the early exercise boundary. Other parameterizations of this boundary are also available. Note that these methods produce a lower bound on the price of the option, as the exercise policy can only be sub-optimal. However, the bound is considered to be close enough to the true answer that it is often accepted as the solution to the problem.

With the increasingly powerful computers and the advance in parallel computing, Monte Carlo methods are becoming the method of choice in many option pricing problems. While the tree methods and the finite difference scheme are suited for American style payoffs due to backward induction, they cannot easily accommodate multiple underlyings or path dependency. This is an intrinsic problem with these two approaches where no solution will be possible. On the other hand, if an efficient Monte Carlo method can be found which can handle American options, this is considered as the favored outcome as it means that products which are path dependent, American, and depend on multiple underlyings can be managed. This is exactly what the Longstaff-Schwartz algorithm is trying to achieve.

Longstaff-Schwartz algorithm

1) Generate a set of N model paths

$$S^j(t_1),\cdots,S^j(t_m),\ j=1,\cdots,N$$

2) Work backwards from $t=t_m$. At final time, the exercise strategy and value are clear. The terminal payoff function of the option for a particular asset price $S^k(t_m)$ is given by

$$h_m(S^k(t_m))$$

3) At one time step before the final time, define continuation value to be the value on same path at final time discounted by one period:

$$e^{-r(t_m-t_{m-1})}h_m(S^k(t_m))$$

4) Regress continuation value against a set of p basis functions $L_i(S)$ to form a relationship between the asset price and the continuation value.

[4] Press et al., *Numerical Recipes in C*, 2nd edition, p.310.

5) Use regressed value to decide exercise strategy

$$V_{reg}=\sum_{i=1}^{p}a_iL_i(S^k(t_{m-1}))$$

6) Define value at second last time according to strategy and value at following time. If the exercise value is lower than the regressed value, it is not optimal to exercise, and the option value at this time will be the continuation value. But if the exercise value is higher than the regressed value, it should be exercised immediately.
7) Proceed recursively in the same manner to early time points; hence we obtain the optimal stopping rule at all time points for every path.
8) We can compute an estimate of the option value by discounting each cash flow back to the issue date and calculate the average

Example (from Longstaff & Schwartz (2001))

Path	*t* = 0	*t* =1	*t* =2	*t* =3	Cash flow at *t* =3
1	1.00	1.09	1.08*	1.34	0.00
2	1.00	1.16	1.26	1.54	0.00
3	1.00	1.22	1.07*	1.03	0.07 = 1.10–1.03
4	1.00	0.93	0.97*	0.92	0.18 = 1.10–0.92
5	1.00	1.11	1.56	1.52	0.00
6	1.00	0.76	0.77*	0.90	0.20 = 1.10–0.90
7	1.00	0.92	0.84*	1.01	0.09 = 1.10–1.01
8	1.00	0.88	1.22	1.34	0.00

Consider a 3-year Bermudan put option which can only be exercised once each year (at t=1,2,3), where the initial asset price is 1.00, the strike price is 1.10, and the risk-free rate is 6%. In the table above, the values in the columns t = 0 to t = 3 are the generated asset prices. A * represents a path which the put is in-the-money at t=2.

Next, we compute values for the 5 paths which are in-the-money at t=2. X is the asset price at t=2, Y is the discounted cash flow at t=3 conditional on no exercise at t=2

Path	X	Exercise value	Y	Regressed value
1	1.08	0.02	0.00x0.94176	0.0369
3	1.07	0.03	0.07x0.94176	0.0461
4	0.97	0.13	0.18x0.94176	0.1176
6	0.77	0.33	0.20x0.94176	0.1520
7	0.84	0.26	0.09x0.94176	0.1565

The discount factor is exp(–0.06)=0.94176. The regressed value is computed based on a least squares fitting procedure. We assume a relationship $Y_{reg}= a+bX+cX^2$ where X is the stock price at t=2, Y_{reg} is the continuation value (with discounting). Using the values in the columns for X and Y in the previous table, we calculate the constants a,b,c based on a least-squares regression, i.e. minimize

$$\sum_{i=1}^{5}\left(Y_i - a - bX_i - cX_i^2\right)^2$$

The result is $Y_{reg} = -1.070 + 2.983X - 1.813X^2$. With this relationship, we can compute the regressed values in the last column in the table; e.g. for path 1 with $X = 1.08$,

$$Y_{reg} = -\ 1.070+2.983\text{x}1.08-1.813\text{x}1.08^2 = 0.0369$$

Comparing the exercise value with the regressed value, we can find the cash flow for each path. For path 1, the exercise value (0.02) is lower than the regressed value (0.0369), thus it is not optimal to exercise at t=2, and the cash flow for this path turns out to be 0. For path 3, the exercise value (0.03) is lower than 0.0461, therefore the cash flow is 0.07 (at time t=3). For paths 4,6,7, the exercise values are higher than the regressed values, thus the cash flows for these paths would be set to the exercise values at t=2.

The cash flows for the 8 paths are thus:

Path	t = 1	t =2	t =3
1		0.00	0.00
2		0.00	0.00
3		0.00	0.07
4		0.13	0.00
5		0.00	0.00
6		0.33	0.00
7		0.26	0.00
8		0.00	0.00

We can use these cash flows to compute the table for t=1. We compute values for the 5 paths which are in-the-money at t=1, i.e. paths 1,4,6,7,8.

Path	X	Exercise value	Y	Regressed value
1	1.09	0.01	$0.00\text{x}0.94176^2$	0.0139
4	0.93	0.17	0.13x0.94176	0.1092
6	0.76	0.34	0.33x0.94176	0.2866
7	0.92	0.18	0.26x0.94176	0.1175
8	0.88	0.22	$0.00\text{x}0.94176^2$	0.1533

As before, the regressed value are computed based on the equation:

$$Y_{reg} = 2.038 - 3.335X + 1.356X^2$$

Thus for paths 4,6,7 and 8, it is optimal to exercise at t=1. The final results are:

	Stopping rule			Option cash flow		
Path	t = 1	t =2	t =3	t = 1	t =2	t =3
1	0	0	0	0.00	0.00	0.00
2	0	0	0	0.00	0.00	0.00
3	0	0	1	0.00	0.00	0.07
4	1	0	0	0.17	0.00	0.00
5	0	0	0	0.00	0.00	0.00
6	1	0	0	0.34	0.00	0.00
7	1	0	0	0.18	0.00	0.00
8	1	0	0	0.22	0.00	0.00

In the stopping rule columns, “1” means that it is optimal to exercise at that date. It is seen that for paths 4,6,7, while is optimal to exercise at t=2 if we compare t=2 and t=3, all these paths should be exercised at t=1 if that option is available.

Finally, the put option value can be computed by discounting each cash flow back to the current time and taking the average over the sample paths:

$$\begin{aligned}&\text{Option price}\\&=\frac{1}{8}(0.07e^{-0.06\times 3}+0.17e^{-0.06\times 1}+0.34e^{-0.06\times 1}+0.18e^{-0.06\times 1}+0.22e^{-0.06\times 1})\\&=0.1144\end{aligned}$$

In practice, more complex functional forms can be used for the regressed value and many more paths are sampled. For example, in the original paper, Longstaff and Schwartz propose to use the Laguerre polynomials as the basis functions:

$$\begin{aligned}&Y=\sum_{j=0}^{\infty} a_j L_j(X)\\&L_n(X)=e^{-X/2}\frac{e^X}{n!}\frac{d^n}{dX^n}(X^n e^{-X})\\&\text{e.g.}\\&L_0(X)=\exp(-X/2), L_1(X)=\exp(-X/2)(1-X),\\&L_2(X)=\exp(-X/2)(1-2X+X^2/2)\end{aligned}$$

However, it is not clear whether more complex basis functions must generate a better early exercise boundary. Remember that the main task for the basis functions is to identify a (smooth) boundary; the accuracy of the regression is of secondary importance.

Early Exercise Boundary Parametrization Approach

In an alternative formulation, we assume that the early exercise boundary can be parameterized in some way. Basically we carry out a first Monte Carlo simulation and work back from the end calculating the optimal parameter values. We then discard the paths from the first Monte Carlo simulation and carry out a new Monte Carlo simulation using the early exercise boundary defined by the parameter values. Note that for the Longstaff/Schwartz algorithm, only in-sample regression is suggested instead of the out-of-sample simulation proposed here.

Example of parametrization approach

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

The initial parameters are the same as in the example in the last section. Denote $S^*(t)$ as the asset price at which the option should be exercised at time t. At time $t = 3$, the option will be exercised if it is in-the-money. At time $t = 2$, we use these sample paths to identify which asset price $S^*(2)$ will lead to an optimal early exercise value.

Path	$S^*(2)<0.77$	$S^*(2)=0.77$	$S^*(2)=0.84$	$S^*(2)=0.97$	$S^*(2)=1.07$	$S^*(2)=1.08$
1	0.00D	0.00D	0.00D	0.00D	0.00D	0.02
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.07D	0.07D	0.07D	0.07D	0.03	0.03
4	0.18D	0.18D	0.18D	0.13	0.13	0.13
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.20D	0.33	0.33	0.33	0.33	0.33
7	0.09D	0.09D	0.26	0.26	0.26	0.26
8	0.00	0.00	0.00	0.00	0.00	0.00
avg	0.0636	0.0813	0.1032	0.0982	0.0938	0.0963

In the table above, $D=\exp(-0.06)$. Suppose we choose $S^*(2)<0.77$, then no path would be exercised early, and the value for each path comes from the discounted value from $t=3$. If $S^*(2)=0.77$, only path 6 is immediately exercised; other paths would have payoffs equal to the value at $t=3$ discounted by one year. If $S^*(2)=1.07$, paths 3,4,6 and 7 are immediately exercised; path 1 would have a payoff equal to the value at $t=3$ discounted by one year, which is 0 in this case. The average payoff for each exercise value can then be computed. With these limited number of paths, the optimal $S^*(2)$ is found to be 0.84 (with the highest average payoff of 0.1032). This process is then repeated at $t=1$ until the optimal exercise boundary $S^*(1)$ is found, which is similar to identifying the stopping rule in the Longstaff/Schwartz algorithm. The option pricing part is also identical to the last procedure in that algorithm.

Reference:

Many excellent textbooks are available, but most of them are fairly technical and mathematically rigorous. Other than the references on specific areas already given in the notes above, a good account suitable for advanced MBA Finance students and introductory MSc Financial Engineering students is Back (2005).

Back, Kerry (2005). *A Course in Derivative Securities: Introduction to Theory and Computation*. Berlin: Springer Finance.