William Carpenter
github.com/wc
linkedin.com/in/wc

Equity Derivative Models
Methods for Derivative Pricing

March 15, 2023

Objectives

This document first describes three option pricing models and their implementation with binomial and trinomial trees. With these models, a tree pricing program can be created that will value American and European options. Monte Carlo simulation is then discussed for pricing path-dependent securities. Trees can also be adapted to simulation asset price paths. Additional notes are included on option payoffs and mathematical derivations that underpin many characteristics of the models.

Actual implementation of the models is performed using Python and MATLAB and can be found here on Github. Market data on stock prices and options are obtained from Yahoo Finance and interest rate data is from the U.S. Treasury.

Contents

# 1 The Cox-Ross-Rubenstein (CRR) Model

The Cox-Ross-Rubenstein (CRR) is a market model for pricing equity derivatives and is often implemented with a binomial or trinomial tree. It was proposed in 1979, shortly after Myron Scholes and Fischer Black published their paper in 1973 on an option pricing formula that is regarded as the first completely satisfactory equilibrium option pricing model. CRR intended to provide a practical, simplified approach to pricing an option with a lattice-based method that could be straightforward to implement in practice. As will be discussed later, a lattice-based method is especially useful for considering American-style options that have the possibility of exercise before maturity; the Black- Scholes closed-form formula can only price European style options that can be exercised at maturity.

Within the binomial tree model, it is assumed that a stock price either moves up (u) or down (d) with certain risk neutral probabilities (p and $(1-p)$). This is a very simple approximation of asset price movement that implies that at time t, the stock price can only take on two values at $t + \Delta$. However, this simplification can produce quite accurate option prices with an adequate number of time periods. Visually, a one-period tree would look like:

$$S_t \qquad \overset{p}{\nearrow} \quad S_{t+\Delta} = S_t u$$
$$\underset{(1-p)}{\searrow} \quad S_{t+\Delta} = S_t d$$

$$t = 0 \qquad\qquad t = 1$$
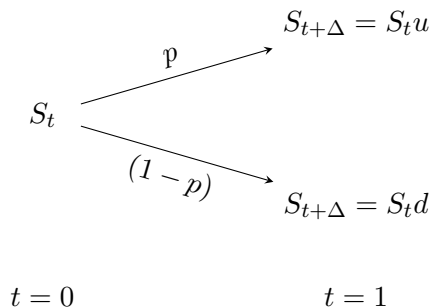
The initial asset price, S_t or S_0, is commonly inferred from market data. Throughout the entire tree, the three parameters (u, d, p) remain constant while the asset price evolves. It is also assumed that the risk-free rate for discounting, often inferred from Treasury data, is constant for the life of the option. In the market, most options have maturities of one year or less.

The nodes of the tree are designed to be *recombining*, which means that an up-then-down movement will ultimately yield the same asset price as a down-then-up movement. This effectively reduces the growth of the tree nodes as the number of steps increases to $n+1 \to O(n)$. A two-period tree would have the following form:

$$t = 0 \qquad\qquad t = 1 \qquad\qquad t = 3$$
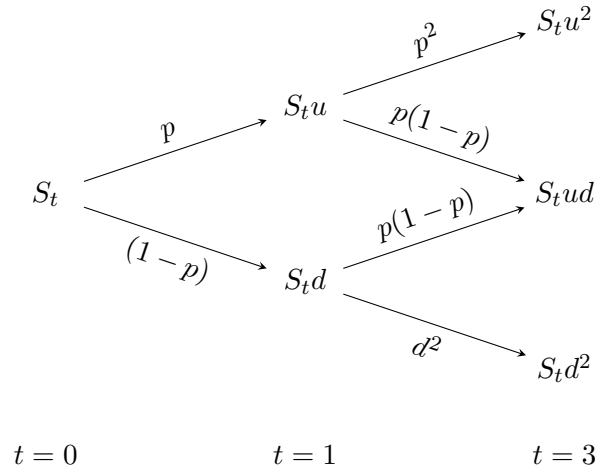
The recombining characteristic is now evident at $t = 2$ with node $S_t ud$, which could equivalently be written as $S_t du$. Once a tree is created, it is most useful pricing American and European options that are *not* path-dependent. In order for the CRR model to be produced, the following assumptions are made:

- The risk-free interest rate (r) is considered to be constant for the life of the option

- The volatility parameter (σ) is considered to be constant for the life of the option

- There are no transaction costs for buying or selling options

- The returns of the underlying asset are log-normally distributed

These are essentially the same assumptions that underlie the Black-Scholes formula for pricing European options. Formally, the last assumption regarding log-normal distribution of returns can be written as:

$$ln(S_{t+\Delta}) - ln(S_t) \sim \mathbb{N}\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta, \sigma\sqrt{\Delta}\right]$$
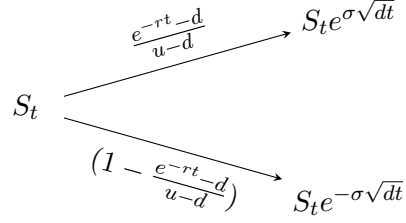
This assumption is utilized to derive equations for the up and down parameters of pricing trees.

1.1 Binomial Tree Parameters

The CRR model required parameters to be calculated from inputted data. parameters are:

$$u = e^{\sigma\sqrt{dt}} \qquad d = e^{-\sigma\sqrt{dt}} \qquad p = \frac{e^{-rt} - d}{u - d}$$

Using these expressions, a one-period tree would have the following form:

$$S_t \quad \xrightarrow{\frac{e^{-rt}-d}{u-d}} \quad S_t e^{\sigma\sqrt{dt}}$$

$$\xrightarrow{(1-\frac{e^{-rt}-d}{u-d})} \quad S_t e^{-\sigma\sqrt{dt}}$$

Once a tree is complete, a backward induction procedure can be utilized to determine an option price. This relies on financial theory regarding martingales and the risk-neutral measure, which enable one to calculate a price from the discounted expected payoffs of an option. The martingale risk neutral pricing equation at each node would be:

$$S_{i,j} = e^{-r\Delta}\mathbb{E}[S_{i,j}] = e^{-r\Delta}\left[pS_{i,j+1} + (1-p)S_{i+1,j+1}\right]$$

The notation (i, j) refers to rows and columns within a pricing tree that could be implemented as a two-dimensional array. The risk-neutral expectation of the price at a specific node is simply the probability-weighted sum of the two possible prices the asset can take at time $t + \Delta$. With a one-period tree, the pricing mechanism would look like:

$$e^{-r\Delta}\left[pS_{i,j+1} + (1-p)S_{i+1,j+1}\right] \quad \xrightarrow{p} \quad S_{i,j+1}$$

$$\xrightarrow{1-p} \quad S_{i+1,j+1}$$
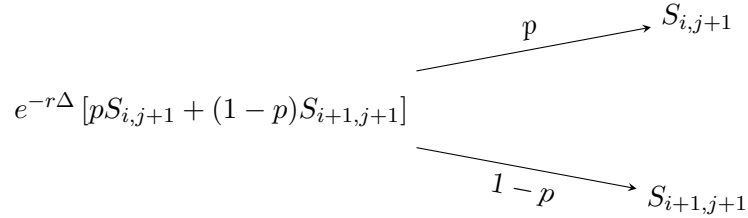
In this case, the asset S_t would refer to the option price, not the actual equity share price. Within models, at the end of tree, terminal payoffs must be calculated that will be based on the type of option being considered.

1.2 Trinomial Tree Parameters

The CRR model can also be implemented with a trinomial tree, which adds a third possible path the asset price can take, where the price is unchanged. This type of model adaptation was originally proposed by Boyle (1986). A one-period tree form is now:

$$S_{t+\Delta} = S_t u$$

$$p_u$$

$$1 - p_u - p_d \quad\longrightarrow\quad S_{t+\Delta} = S_t$$

$$S_t$$

$$p_d$$

$$S_{t+\Delta} = S_t d$$
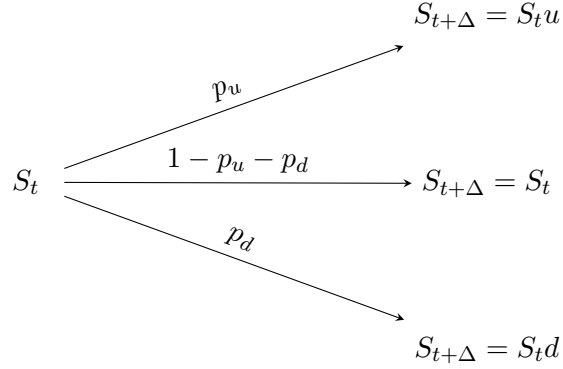
1.3 Monte Carlo Simulation

Lattice-based pricing is a useful tool for many types of derivatives, but Monte Carlo simulation is suitable when payoffs are path-dependent.

2 The Jarrow-Rudd (JR) Model

The JR is a particular case of the CRR in which probabilities are assumed to equal $1/2$, which implies certain dynamics for the tree. Similar to CRR, the JR model is often implemented with a recombining binomial tree.

2.1 Binomial Tree Parameters

Martingale Pricing condition:

$$S_{i,j} = e^{-r\Delta}\mathbb{E}\left[S_{i,j}\right] = e^{-r\Delta}\left[\frac{1}{2}S_{i+1,j} + \frac{1}{2}S_{i+1,j+1}\right]$$

3 Option Payoffs

3.1 European Calls & Puts

A call option is a derivative that grants its purchaser the right to buy a number of shares at a specific price at some future point in time. The purchase price is commonly referred to as the strike price or exercise price (often denoted as K).

4   A Tree Pricing Program

Programs to price options via binomial and trinomial trees are straightforward to implement in Python or MATLAB with a relatively small amount of code.

Tree pricing program has time complexity $O(n^2)$ and space complexity can be $O(n)$. The most straightforward way to approach implementation is to use an array to hold tree values. In practice, a single, one-dimensional array would be needed to perform calculations. For the sake of this documentation,

5   Example of Pricing with Trees

6   Example of Pricing with MC Simulation

The Euler discretization for MC simulation is the following:

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

This equation can be used

7 Mathematical Notes

7.1 The Feynman-Kac Result

The foundation of risk-neutral pricing is the Feynman-Kac result. Given a function $f(S,t)$ and pricing partial differential equation (PDE):

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

This embodies pricing function under the following dynamics for S:

$$dS = \mu(S,t)dt + \sigma(S,t)dZ^*$$

If the pricing function satisfies boundary condition $f(S,T) = g(S)$ where $g(...)$ is some payoff function then:

$$f(S,t) = \mathbb{E}^* \left[g(S_T)e^{-\int_t^T r_s ds} \right]$$

This is the Feynman-Kac result under stochastic interest rates, which is why the discount factor is part of the expectation and is not constant. Under Black-Scholes and CRR, interest rates are assumed to be constant.

7.2 Risk Neutral Probability Measure

If the CRR model has parameters $d < 1 + r < u$, then the model is considered both arbitrage-free and complete. Using a numeraire B_t, the asset price is written as:

$$S_t = B_t \mathbb{E}^Q \left[\frac{S_T}{B_T} \;\middle|\; \mathcal{F}_t \right]$$

The expectation $\mathbb{E}^*[...]$ is taken under the risk-neutral measure. Often, the selected numeraire is a zero coupon bond (ZCB) with continuous compounding. Assuming a constant interest rate, the pricing equation for a ZCB would be:

$$B(t,T) = e^{-r\Delta} \quad , \quad \Delta = T - t$$

At maturity, the price of a ZCB will be exactly equal to its face value, or \$1 according to the equation above. With the ZCB as the numeraire, the martingale pricing formula can be re-written as:

$$S_t = e^{-r\Delta}\mathbb{E}^Q \left[\frac{S_T}{1} \;\middle|\; \mathcal{F}_t \right] = e^{-r\Delta}\mathbb{E}^Q \left[S_T | \mathcal{F}_t \right]$$

This is the mathematical result that is implemented within binomial or trinomial trees, where the expectation is a weighted sum of probabilities the asset price will move up, down, or stay the same. In the case of a binomial tree with probability p, the pricing equation would be generally written as:

$$S_t = e^{-r\Delta}(pS_t u + (1-p)S_t d)$$

In the case of the Jarrow-Rudd model, probability p is considered to be equal to $1/2$:

$$S_t = e^{-r\Delta} \left(\frac{1}{2} S_t u + \frac{1}{2} S_t d \right)$$

With the trinomial model, a third term is added that allows the stock price to stay the same, or advance to a middle-oriented node. This adjustment gives the following equation:

$$S_t = e^{-r\Delta} (p_u S_t u + p_m S_t m + p_d S_t d)$$

7.3 Closed-Form Solution of the Black-Scholes PDE

To avoid discretization error, it is useful to derive a closed-form solution for dynamics of the stock price. The Black-Scholes SDE is given as:

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

With this equation, the strategy is to apply Itô's Lemma to the function $f(S_t) = ln(S_t)$. The formula for Itô's Lemma is the following:

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

Calculating the required derivatives for $f(S_t)$ to plug into Itô's Lemma:

$$\frac{\partial f(S_t)}{\partial t} = 0 \qquad \frac{\partial f(S_t)}{\partial S_t} = \frac{1}{S_t} \qquad \frac{\partial^2 f(S_t)}{\partial S_t^2} = -\frac{1}{S_t^2}$$

Plugging these results into the formula for Itô's Lemma gives the following:

$$dln(S_t) = \left(\frac{1}{S_t} \mu S_t - \frac{1}{2} \frac{1}{S_t^2} \sigma^2 S_t^2 \right) dt + \left(\frac{1}{S_t} \sigma S_t \right) dZ_t$$

$$dln(S_t) = \left(\mu - \frac{1}{2} \sigma^2 \right) dt + \sigma dZ_t$$

The distribution of $dln(S_t)$ can be inferred, assuming that μ and σ are constants:

$$dln(S_t) = \underbrace{\left(\mu - \frac{1}{2} \sigma^2 \right)}_{\text{constant}} dt + \underbrace{\sigma}_{\text{constant}} dZ_t \quad , \quad dZ_t \sim \sqrt{dt} \mathbb{N}(0, 1)$$

$$dln(S_t) = \left(\mu - \frac{1}{2} \sigma^2 \right) dt + \sigma dZ_t \sim \mathbb{N} \left[\left(\mu - \frac{1}{2} \sigma^2 \right) dt, \sigma^2 dt \right]$$

Discretizing the formula via t and $t + \Delta$ increments:

$$ln(S_{t+\Delta}) - ln(S_t) = \left(\mu - \frac{1}{2} \sigma^2 \right) \Delta + \sigma(Z_{t+\Delta} - Z_t)$$

$$ln \left(\frac{S_{t+\Delta}}{S_t} \right) = \left(\mu - \frac{1}{2} \sigma^2 \right) \Delta + \sigma(Z_{t+\Delta} - Z_t)$$

$$\frac{S_{t+\Delta}}{S_t} = exp\left[\left(\mu - \frac{1}{2}\sigma^2\right)\Delta + \sigma(Z_{t+\Delta} - Z_t)\right]$$

$$S_{t+\Delta} = S_t exp\left[\left(\mu - \frac{1}{2}\sigma^2\right)\Delta + \sigma(Z_{t+\Delta} - Z_t)\right]$$

This completes the derivation, but a slightly different formula is ultimately utilized for Monte Carlo simulation purposes. Applying a Euler discretization to the formula above provides the formula for MC simulation:

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

Where we have:

$$\epsilon \sim \mathbb{N}(0,1)$$

Under the risk-neutral measure, the expected return μ is assumed to be equal to the risk-less rate r.

7.4 Deriving Parameters for the CRR Model

7.4.1 Solving for Probability p

Assume that time t is the present period and time horizon is $\Delta = T - t$. The risk-neutral valuation of a stock price can be written as:

$$S_t = e^{-r\Delta}\mathbb{E}^*[S_{t+\Delta}]$$

Within the binomial tree, the expectation of $S_{t+\Delta}$ is the probability-weighted sum of an up and down movement in the tree, i.e. the only two values that S can become:

$$S_t = e^{-r\Delta}[p(S_t u) + (1 - p)(S_t d)]$$

This equation can now be rearranged to solve for p in the CRR market model:

$$S_t e^{r\Delta} = p(S_t u) + (1 - p)(S_t d)$$

$$e^{r\Delta} = pu + (1 - p)d$$

$$e^{r\Delta} - d = pu - pd$$

$$e^{r\Delta} - d = p(u - d)$$

$$p = \frac{e^{r\Delta} - d}{u - d}$$

7.4.2 Solving for Parameters u and d

First, recall the fundamental assumption of the CRR model that stock price changes are log-normally distributed:

$$ln(S_{t+\Delta}) - ln(S_t) \sim \mathbb{N}\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta, \sigma\sqrt{\Delta}\right]$$

10

7.5 Deriving the Log-Returns Expression

Deriving log-returns beginning with simple returns:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

$$r_t = \frac{P_t}{P_{t-1}} - 1$$

$$r_t + 1 = \frac{P_t}{P_{t-1}}$$

$$ln(r_t + 1) = ln\left(\frac{P_t}{P_{t-1}}\right)$$

$$ln(r_t + 1) = ln(P_t) - ln(P_{t-1})$$

Using historical price data, the log-return can be calculated on a daily basis as:

$$r_t = ln\left(\frac{P_t}{P_{t-1}}\right)$$

References

Add CRR mdoel, trinomial model, some powerpoints, the derivation.