Chapter 3 - Binomial Trees

Modelling Derivatives in C++

Diffusion processes can be approximated using binomial trees - a two-state lattice method. This can be used for a variety of European and American style derivatives.

1 Using binomial trees

Assuming that the underlying stock, with an initial value of S_0 can go up and down the next period by a factor of u and d respectively. We can find the price of the European call option with this explicit formula:

$$C = r^{-n}\left[\sum_{j=0}^{n}\left(\frac{n!}{j!(n-j)!}\right)p^j(1-p)^{n-j}\ \max(0, Su^jd^{n-j}-K)\right]$$

Where:

$$p = \frac{e^{r\Delta t}-d}{u-d}, 1-p = \frac{u-e^{r\Delta t}}{u-d}$$

Furthermore, if we write:

$$a = \left(log\frac{K}{Sd^n}/log\frac{u}{d}\right), p' = (u/r)p, 1-p' = (d/r)(1-p)$$

Then the call option formula can be simplified as:

$$C = S\Phi(\max(\lceil a\rceil, 0); n, p') - Kr^{-n}\Phi(\max(\lceil a\rceil, 0); n, p)$$

2 Cox-Ross-Rubinstein binomial tree

We then assume that $ud = 1$. The above formula can be simplified more, in that the following results (keeping only the linear terms) hold true:

$$u = 1+\sigma\sqrt{\Delta t}+\frac{1}{2}\sigma^2\Delta t, d = 1-\sigma\sqrt{\Delta t}+\frac{1}{2}\sigma^2\Delta t$$

and:

$$p = \frac{1}{2}+\frac{\mu}{2\sigma}\sqrt{\Delta t}, 1-p = \frac{1}{2}-\frac{\mu}{2\sigma}\sqrt{\Delta t}$$

The CRR model is the most commonly used version of the binomial model - though the original binomial model converges faster to an exact solution.

3 Jarrow-Rudd binomial tree

Not too relevant - similar exercise as above to simplify the binomial tree.

4 General tree

Not too relevant - similar exercise as above to simplify the binomial tree.

5 Dividend payments

Not too relevant - similar exercise as above to incorporate dividend payments into the binomial tree, and some conditions in which the price paths, and up and down probabilities to comply with.

6 American exercise

The American binomial option model reduces to a dynamic programming exercise, to discern whether an early payoff is optimal at each node.

7 Binomial tree implementation

This section concerns building a CRR model for an American option, where a function that takes in several inputs calculate an option price. For a non-dividend paying stock, there is no utility in exercising an American option early - this is due to the additional insurance it provides. As a result, its value will be the same as a European option.

8 Computing hedge statistics

Hedge statistics (e.g., delta, gamma) need to be approximated using finite differences under a binomial model. e.g., for delta:

$$\Delta \approx \frac{C_{1,1} - C_{1,0}}{S_{1,1} - S_{1,0}}$$

vega, rho and theta can all be approximated as follows:

$$\begin{aligned}
\nu &\approx \frac{C(\sigma + \Delta\sigma) - C(\sigma - \Delta\sigma)}{2\Delta\sigma} \\
\rho &\approx \frac{C(r + \Delta r) - C(r - \Delta r)}{2\Delta r} \\
\theta &\approx \frac{C(T + \Delta T) - C(T - \Delta T)}{2\Delta T}
\end{aligned}$$

9 Binomial model with time varying volatility

In practice, constant volatility models are rarely used - instead, we have to contend with the additional complexity of time varying volatility models. In this case, we replace μ, σ with μ_i, σ_i for time t_i. The binomial probabilities, and time change for each time point then becomes:

$$
\begin{aligned}
\Delta t_i &= \frac{-\sigma_i^2 \pm \sqrt{\sigma_i^4 + 4\sigma_i^2 \Delta x^2}}{2\mu_i^2} \\
p_i &= \frac{1}{2} + \frac{\mu_i \Delta t_i}{2\Delta x}
\end{aligned}
$$

Again, this becomes a binomial model

10 Two-variable binomial process

In this case, we can model the stocks through correlated GBMs, z_1, z_2:

$$
\begin{aligned}
dS_1 &= (r - q_1) S_1 dt + \sigma_1 S_1 dz_1 \\
dS_2 &= (r - q_2) S_2 dt + \sigma_2 S_2 dz_2
\end{aligned}
$$

Logging the above gives, $x_i := log S_i$:

$$
\begin{aligned}
dx_1 &= \mu_1 dt + \sigma_1 dz_1 \\
dx_2 &= \mu_2 dt + \sigma_2 dz_2
\end{aligned}
$$

Where $\mu_i = r - q_i - 0.5\sigma_i^2$.

Since we have 2 securities, we can write the up and down probabilities for both stocks (in all 4 cases as):

$$
\begin{aligned}
\Delta x_1 &= \sigma_1 \sqrt{\Delta t} \\
\Delta x_2 &= \sigma_2 \sqrt{\Delta t} \\
p_{uu} &= \frac{1}{4} \frac{\Delta x_1 \Delta x_2 + (\Delta x_2 \mu_1 + \Delta x_1 \mu_2 + \rho \sigma_1 \sigma_2) \Delta t}{\Delta x_1 \Delta x_2} \\
p_{ud} &= \frac{1}{4} \frac{\Delta x_1 \Delta x_2 + (\Delta x_2 \mu_1 + \Delta x_1 \mu_2 - \rho \sigma_1 \sigma_2) \Delta t}{\Delta x_1 \Delta x_2} \\
p_{du} &= \frac{1}{4} \frac{\Delta x_1 \Delta x_2 - (\Delta x_2 \mu_1 - \Delta x_1 \mu_2 + \rho \sigma_1 \sigma_2) \Delta t}{\Delta x_1 \Delta x_2} \\
p_{dd} &= \frac{1}{4} \frac{\Delta x_1 \Delta x_2 - (\Delta x_2 \mu_1 + \Delta x_1 \mu_2 - \rho \sigma_1 \sigma_2) \Delta t}{\Delta x_1 \Delta x_2}
\end{aligned}
$$

In the Python code - we implement a spread option of an additive binomial tree with 2 securities.

11 Valuation of convertible bonds

A convertible bond is a corporate debt security that can be converted into a pre-determined number of ordinary shares at the discretion of the bondholder. Mathematically, this can be represented as a bond component + an option (call or put depending on perspective).

Convertible bonds can be used to get around low credit ratings - and offer stability to the consumer with respect to income.

Some other important definitions:

1. Conversion ratio: Pre-determined exchangeable number of shares
2. Conversion price: Principal divided by conversion ratio (i.e., price paid per share upon conversion)
3. Parity: Market value of the underlying stock prices (i.e., conversion ratio $\times$ current stock price).
4. Conversion date: When the bondholder switches from the conventional bond to a stock.

The value of a convertible bond depends on a few factors:

1. Any options purchased alongside the convertible bond - e.g., a call provision and put provision (which allows the issuer/holder to purchase back or give back the bond to minimise downside risk for either party)
2. Stock price
3. Stock volatility
4. Dividend yield
5. Risk-free rate
6. Stock loan rate
7. Issuer's credit spread

Given the following parameters:

1. N: conversion ratio
2. S: stock price
3. C: call value
4. P: put value
5. I: accrued interest

6. H: holding value at a node
7. V_u: value of the option 1 period ahead (given an uptick in the stock price).
8. V_d: value of the option 1 period ahead (given a downtick in the stock price).
9. p_u: up probability
10. p_d: down probability
11. $p = \frac{1}{2}(p_u + p_d)$
12. r: risk-free rate
13. d: risky rate
14. $y = pr + (1-p)d$: credit-adjusted discount rate

The value of the convertible bond can be calculated at each node as:

$$\begin{aligned} V &= \max(NS, P+I,\ \min(H, C+I)) \\ H &= \frac{1}{2}\left(\frac{V_u + I}{1 + y_u \Delta t} + \frac{V_d + I}{1 + y_d \Delta t}\right) \end{aligned}$$

This is explored in more detail in the Python code provided.