Programing A Binomial Tree to Price Options

There are some basic ingredients for implementing option pricing using a binomial tree

1. Installation and other preparation
2. Take inputs for option pricing
3. Output option price
4. Calculate binomial tree parameters
5. Calculate and store prices of underlying stock
6. Calculate option price as intrinsic value of option at last time step
7. Create a loop to move back though time and price option by discounting future option prices
8. Evaluate early exercise for American option
9. Determining early exercise threshold

Installation and other preparation

Your program is a set of instructions for the computer. The instructions are written using different rules for different programing languages. The computer needs to be able to understand your instructions. For this you need to install software on your computer that can understand the instructions in your program. Some of

Take inputs for option pricing

Inputs can be provided by the user by typing in parameters at the time software is run. Alternatively, the input values can be stored in advance in a computer file. We will not consider files here. The inputs needed are:

s	Price of the Underlying Stock
k	Strike Price
T	Time to Maturity in Years
V	Volatility of the Return on Underlying Stock
R	Continuously Compounded Risk-free Rate
Y	Continuously Compounded Dividend Yield on Stock
Cp	1 for Call, 0 for Put
Am	1 for American, 0 for European
N	Number of Steps in Binomial Tree

Your main program can prompt user for these inputs. However, if you use repl.it, there is already a main program to prompt user for these inputs that passes test values of parameters to your function to evaluate your function. So you need to write a function that accepts these parameters as inputs. For example, in Python, you can define

```
def binomial_value(s, k, t, v, r, y, cp, am, n):
```

Output option price

You can print the option price to standard output (user screen). However it is preferable for that your function return the calculated option price. This value can be used by a main program that interacts with the user. if you use repl.it, there is already a main program that takes the output of your function to evaluate your program.

Calculate binomial tree parameters

To create binomial tree, use the inputs to determine the up and down movement of stock prices in each step, and the probability of up movement. Specifically, the two methods recommended are:

Cox Ross Rubinstein Specification

$$\Delta t = \frac{T}{n}.$$

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

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

$$q = \frac{e^{(r-y)\Delta t} - d}{u - d},$$

$$discount = e^{-r\Delta t}.$$

Jarrow Rudd Risk Neutral Specification

$$\Delta t = \frac{T}{n}.$$

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

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

$$q = \frac{e^{(r-y)\Delta t} - d}{u - d},$$

$$discount = e^{-r\Delta t}.$$

Note: These methods may fail in special cases of $\sigma = 0$ or $t = 0$ or $u = d$. Take care of those situations specifically. If $\sigma = 0$ or $t = 0$, instead of using the normal CRR formulas for u and d, use $u = d = e^{(r-y)\Delta t}$. In both methods, before calculating q, check if u=d. If so, don't use formula for q. Instead choose q=0.5.

Calculate and store prices of underlying stock

Create two two-dimensional arrays, one for stock price and one for option price. For example, StockPrice[i][j] and OptionPrice[i][j] where index i stands for time and ranges from 0 to n or 1 to n+1 depending on the programing language. Here, I will use Python convention and assume it ranges from 0 to n. The second index stands for different possible stock prices at a fixed time and again ranges from 0 to n or 1 to n+1 depending on the programing language.

$t = 0, S = S_0$	$t = \Delta t, S = S_0 u$	$t = 2\Delta t, S = S_0 u^2$					$t = n\Delta t, S = S_0 u^n$
	$t = \Delta t, S = S_0 d$	$t = 2\Delta t, S = S_0 ud$					$t = n\Delta t, S = S_0 u^{n-1} d$
		$t = 2\Delta t, S = S_0 d^2$					$t = n\Delta t, S = S_0 u^{n-2} d^2$
							$t = n\Delta t, S = S_0 u^2 d^{n-2}$
							$t = n\Delta t, S = S_0 u d^{n-1}$
							$t = n\Delta t, S = S_0 d^n$

Fill in stock prices. Ignore cells in array that are not needed. For example, at time t = 0, there is only one stock price, the initial stock price and other cells are irrelevant. You will need a nested double loop for this. The first loop considers different time points (the first index of array) and the second index considers the different up and down movements at that time point (the second index of array). For example, your code might achieve something like this:

For first index i of array ranging from 0 to n

    For second index j of array ranging from 0 to i

        Set StockPrice[i][j] equal to $S_0 u^j d^{(i-j)}$

Calculate option price as intrinsic value of option at last time step

To determine option price, we move back in time. We first start with the option price at maturity. Option price at maturity equals intrinsic value. This can be achieved with a single loop across all prices at last time step. For example,

    For second index j of array ranging from 0 to n

        Set OptionPrice[n][j] equal to max(0,StockPrice[n][j] – k) if Option is Call

        Set OptionPrice[n][j] equal to max(0,k - StockPrice[n][j]) if Option is Put

Create a loop to move back though time and price option by discounting future option prices

Next you will start a backward propagation in time where at any given time point i and stock price s, the option price is calculated using the option prices already calculated at time point i+1 and stock prices su and sd. These two future option prices are multiplied with their risk neutral probabilities, the two terms are added and then discounted back one period. The code would appear something like following:

For first index i of array ranging from n-1 to 0

For second index j of array ranging from 0 to i

Set OptionPrice[i][j] equal to (q* OptionPrice[i+1][j] + (1-q)* OptionPrice[i+1][j+1]) * discount

Evaluate early exercise for American option

The above logic should work for European options. However, for American options, we also need to check if early exercise results in greater value. The above code can be changed as follows:

For first index i of array ranging from n-1 to 0

For second index j of array ranging from 0 to i

Set OptionPrice[i][j] equal to (q* OptionPrice[i+1][j] + (1-q)* OptionPrice[i+1][j+1]) * discount

Set OptionPrice[i][j] equal to max(OptionPrice[i][j],StockPrice[i][j] – k) if Option is American Call

Set OptionPrice[i][j] equal to max(OptionPrice[i][j],k - StockPrice[i][j]) if Option is American Put

Finally, the function should return OptionPrice[0][0] as the current option price.

Determining early exercise threshold

Start this step after your option pricing function is working well.

For call options, when stock price is very low, option is out-of-money and intrinsic value of the option is less than the value of the option. In this case, the option should not be exercised early. When stock price is slightly above exercise price, option is in-the-money but the intrinsic value of the option is less than the value of the option and early exercise is still not optimal. When the stock price is far above the exercise price, early exercise is optimal and option value equals intrinsic value. You have to find the stock price threshold, let us call it B, such that option value = intrinsic value if stock price > B and option value > intrinsic value if stock price < B.

For put options, when stock price is very high, option is out-of-money and intrinsic value of the option is less than the value of the option. In this case, the option should not be exercised early. When stock price is slightly below exercise price, option is in-the-money but the intrinsic value of the option is less than the value of the option and early exercise is still not optimal. When the stock price is far below the exercise price, early exercise is optimal and option value equals intrinsic value. You have to find the stock price threshold, let us call it B, such that option value > intrinsic value if stock price > B and option value = intrinsic value if stock price < B.

You can find the early exercise boundary B by starting with a high value H and a low value L of stock price. These must be on two opposite sides of B. That means that for call option, option value = intrinsic value if stock price = H and option value > intrinsic value if stock price = L. For put option, option value > intrinsic value if stock price = H and option value = intrinsic value if stock price = L.

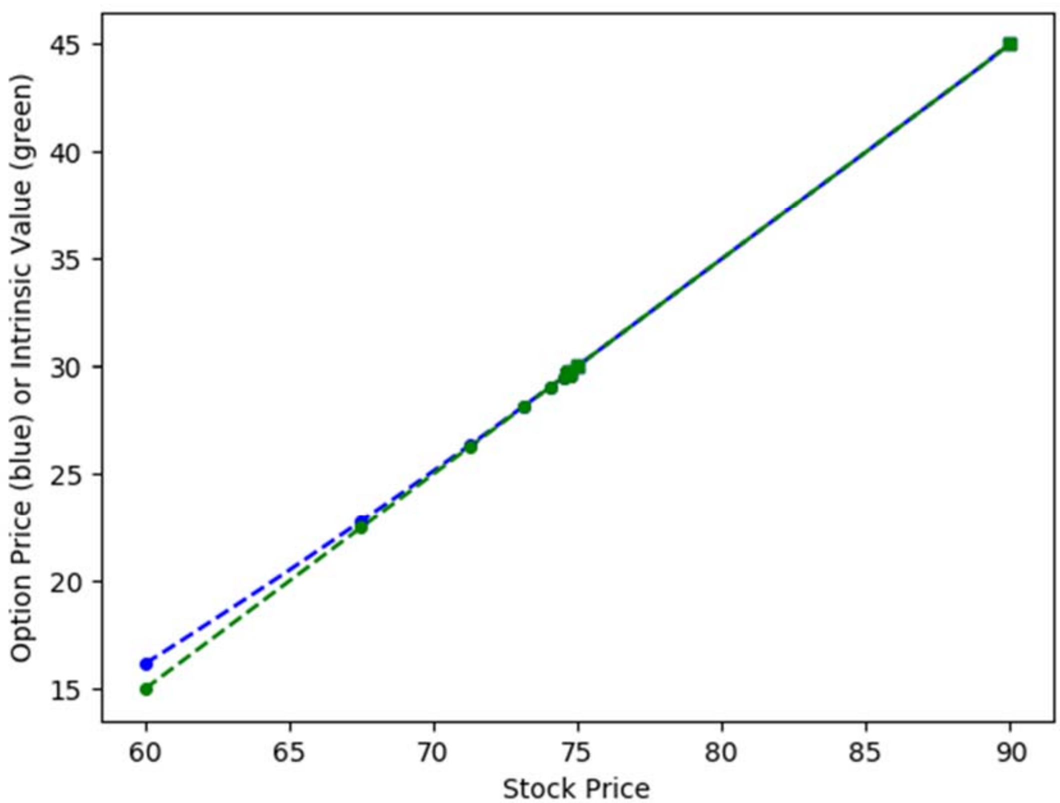
Now you have to find B inside the range (L,H). One method is binary search. Check option value and intrinsic value at stock price of (H+L)/2.

If option value = intrinsic value, then for call option reset H downwards to (H+L)/2 and for put option reset L upwards to (H+L)/2. If option value > intrinsic value, then for call option reset L upwards to (H+L)/2 and for put option reset H downwards to (H+L)/2. Repeat the step with new smaller range (L,H). Eventually, the range (L,H) will become very small and you can choose the midpoint of that range as the boundary B.

The following two examples illustrate the method with the help of graphs.

Example 1

1. First consider a call option with strike price of 45. We know that the option will never be exercised below 45 so early exercise threshold must be above 45. We start with a wide range of stock prices in which to search for this threshold. To keep the graph reasonable, we start with the stock price range of 60 to 90. You might start with a wider range. We use blue color for plotting option price and green color for plotting intrinsic value. We use dots as markers when stock price is less than the exercise threshold and squares as markers when stock price is more than the exercise threshold.



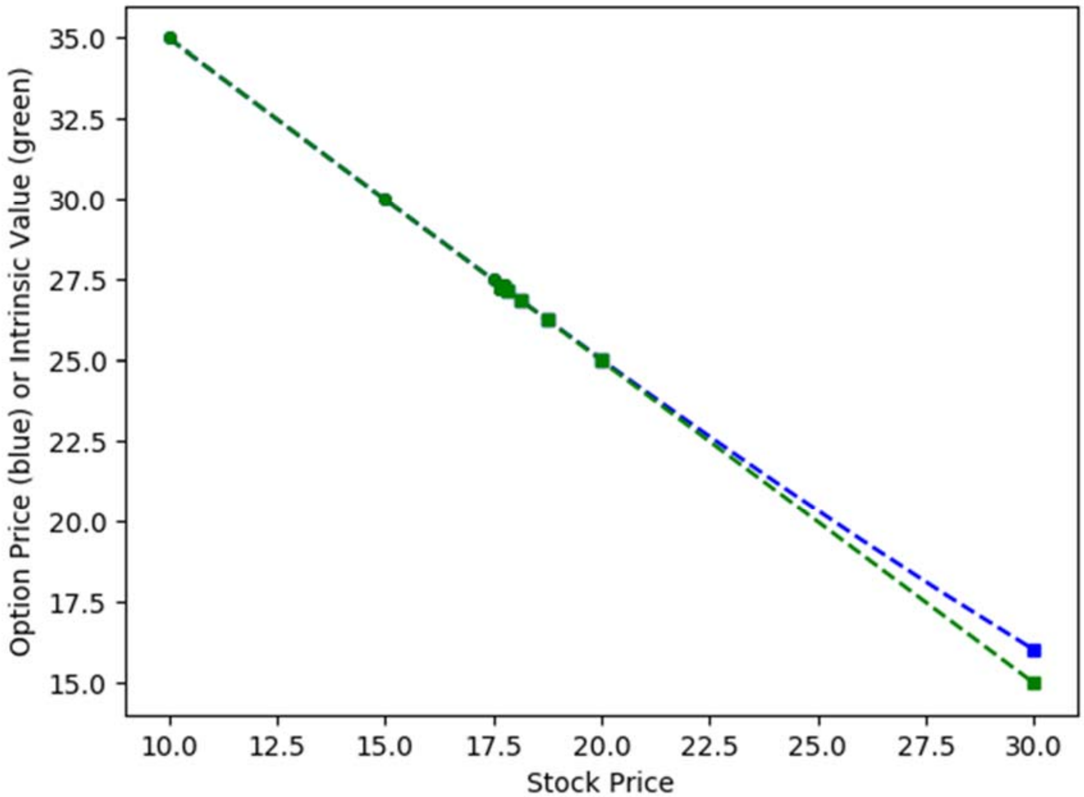
2. We determine the option price and the intrinsic value when stock price is 60. The option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. We also determine the option price and the intrinsic value when stock price is 90. The option price (blue square) is equal to the intrinsic value (green square) so early exercise is

optimal at this price (you can see only one square in the graph as one square is hidden beneath the other). The threshold must lie somewhere between 60 and 90.

3. We try the midpoint stock price of 75. At stock price of 75, the option price (blue square) is equal to the intrinsic value (green square) so early exercise is optimal at this price. That means the threshold for early exercise must lie between 60 and 75.
4. We try the midpoint stock price of 67.5. At stock price of 67.5, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 67.5 and 75.
5. We try the midpoint stock price of 71.25. At stock price of 71.25, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 71.25 and 75.
6. We try the midpoint stock price of 73.13. At stock price of 73.13, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 73.13 and 75.
7. We try the midpoint stock price of 74.07. At stock price of 74.07, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 74.07 and 75.
8. We try the midpoint stock price of 74.54. At stock price of 74.54, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 74.54 and 75.
9. We try the midpoint stock price of 74.77. At stock price of 74.77, the option price (blue square) is equal to the intrinsic value (green square) so early exercise is optimal at this price. That means the threshold for early exercise must lie between 74.54 and 74.77.
10. We try the midpoint stock price of 74.66. At stock price of 74.66, the option price (blue dot) is higher than the intrinsic value (green dot) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 74.66 and 74.77.
11. Continuing this way, we can stop when the range becomes very small. The range converges to a stock price of about 74.68

Example 2

1. Consider a put option with strike price of 45. We know that the option will never be exercised above 45 so early exercise threshold must be below 45. We start with a wide range of stock prices in which to search for this threshold. To keep the graph reasonable, we start with the stock price range of 10 to 30. You might start with a wider range. We use blue color for plotting option price and green color for plotting intrinsic value. We use dots as markers when stock price is less than the exercise threshold and squares as markers when stock price is more than the exercise threshold.



2. We determine the option price and the intrinsic value when stock price is 10. The option price (blue dot) is equal to the intrinsic value (green dot) so early exercise is optimal at this price (you can see only one dot in the graph as one dot is hidden beneath the other). We also determine the option price and the intrinsic value when stock price is 30. The option price (blue square) is greater than the intrinsic value (green square) so early exercise is not optimal at this price. The threshold must lie somewhere between 10 and 30.
3. We try the midpoint stock price of 20. At stock price of 20, the option price (blue square) is greater than the intrinsic value (green square) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 10 and 20.
4. We try the midpoint stock price of 15. At stock price of 15, the option price (blue dot) is equal to the intrinsic value (green dot) so early exercise is optimal at this price. That means the threshold for early exercise must lie between 15 and 20.
5. We try the midpoint stock price of 17.5. At stock price of 17.5, the option price (blue dot) is equal to the intrinsic value (green dot) so early exercise is optimal at this price. That means the threshold for early exercise must lie between 17.5 and 20.
6. We try the midpoint stock price of 18.75. At stock price of 18.75, the option price (blue square) is greater than than the intrinsic value (green square) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 17.5 and 18.75.

7. We try the midpoint stock price of 18.13. At stock price of 18.13, the option price (blue square) is greater than the intrinsic value (green square) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 17.5 and 18.13.
8. We try the midpoint stock price of 17.82. At stock price of 17.82, the option price (blue square) is greater than the intrinsic value (green square) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 17.5 and 17.82.
9. We try the midpoint stock price of 17.66. At stock price of 17.66, the option price (blue dot) is equal to the intrinsic value (green dot) so early exercise is optimal at this price. That means the threshold for early exercise must lie between 17.66 and 17.82.
10. We try the midpoint stock price of 17.74. At stock price of 17.74, the option price (blue square) is greater than the intrinsic value (green square) so early exercise is not optimal at this price. That means the threshold for early exercise must lie between 17.66 and 17.74.
11. Continuing this way, we can stop when the range becomes very small. The range converges to a stock price of about 17.70.