Li Dong 661243168

Binomial Option Pricing

Final report for term project of Intro to Math of Finance

Introduction

Modern option pricing techniques are often considered among the most mathematically complex of all applied areas of finance. Because of its simplicity and convergence the binomial method has attracted the most attention. This report deals with binomial approximation methods for pricing European call and put options.

An option gives the holder the right to trade a specified quantity of an underlying asset at a fixed price at any time on or before a given date or expiration date. Since it is a right and not an obligation the owner of the option can choose not to exercise the option and allow the option to expire. In any option contract, there are two parties involved. An option holder that buys an option and an option writer that sells an option. The option's holder is said to take a long position while the option's writer is said to take a short position. The European option is options which are only exercisable at the expiry date of the option. The term European is confined to describing the exercise feature of the option, i.e., exercisable only on the expiry date. There are also other option types such as American options, Asian Options and so on, which will not be reviewed here because only European options are the focus of this report.

Methodology

In this report, European call and put options are priced using binomial approximation and for simplicity the following sections will omit the word "European". The method will be briefly interpreted in the following paragraphs.

A call option gives the holder of the option the right to buy the underlying asset by a certain price on a certain date, while a put option gives the holder of the option the right to sell the underlying asset by a certain price on a certain date. The payoff function for a call option depends on the price of the underlying asset at expiry T and the strike price K, while the payoff function for a put option depends on the price of the underlying asset at expiry T and the strike price K. Eq. (1) is for call option and Eq. (2) is for put option.

$$C(S_T, T) = (S_T - K)_+ \tag{1}$$

$$P(S_T, T) = (K - S_T)_+ \tag{2}$$

This method employs the lognormal model of asset pricing and does not consider dividend yield. The price of an option can be determined given the risk free yield r, the volatility σ, and the current price of the asset S_0. If S_0 is the price of a risky asset at time $t = 0$, after some time period T it can only become two distinct values: S_0u and S_0d, where u and d are real numbers such that $u > d$ and the existence of a risk-less asset with a constant yield r is assumed. Also no arbitrage argument has to be valid here as indicates by Eq. (3).

$$S_0 d < S_0 e^{rT} < S_0 u \tag{3}$$

Recursive Algorithm Interpretation

A key assumption has to be made here that between successive time levels the asset price either moves up by a factor $u > 1$ or moves down by a factor $d < 1$. An upward movement occurs with probability p and a downward movement occurs with probability $1 - p$. The values for u, d and p are given below:

$$u = \beta + \sqrt{\beta^2 - 1},$$

$$d = \frac{1}{u},$$

$$p = \frac{e^{rdt} - d}{u - d},$$

where $= \frac{1}{2}(e^{-rdt} + e^{(r+\sigma^2)dt})$, constant σ represents the volatility of the asset and dt is the unit period of time.

In general at time t_i, there are i possible asset prices which are given in Eq. (4). It is noted that the options expire after N periods at time $t_{N+1} = T$.

$$S_j^i = S_0 u^{j-1} d^{N-j+1}, \quad 1 \le j \le i \tag{4}$$

The values S_j^i for $1 \le j \le i$ and $1 \le i \le N+1$ form a binary tree. Since call and put options are basically very similar, only call options are displayed here as an example. Let K be the exercise price of the option, C_j^i denote the value of the call options at time $t = t_i$ and use Eq. (1):

$$C_j^{N+1} = max\left(S_j^{N+1} - K, 0\right). \tag{5}$$

The following Eq. (6) is implemented in code to calculate iteratively.

$$C_j^i = e^{-rdt}\left(pC_{j+1}^i + (1-p)C_j^{i+1}\right), \quad 1 \le j \le i, \quad 1 \le i \le N+1 \tag{6}$$

This code utilizes some MatLab subroutines from the internet and they are explained below together with some required parameters.

- *Get_Yahoo_Options_Data.m* [1] provides all the live options data including calls and puts for a given ticker. The output of this subroutine is used as the baseline to compare with the output from the binomial pricing algorithm. Also it can store all the strike prices in a matrix for later use.

- *get_last_trade.m* [2] provides the latest price of a given stock, i.e. S_0. The value is validated against Yahoo Finance [3].
- *hist_vol.m* [4] calculates the volatility σ of a given stock for a certain period. The value is validated against Morningstar [5].
- Risk-free rate r is obtained from [6].
- *main.m* is the subroutine that holds everything together and yield desired output.
- The structure of the code is greatly inspired by [7].

Results of Options Pricing Trial

The options of IBM are priced with this code as a test trial. This trial was conducted at 12:40 am, December 3rd, 2013. April 2014 was selected as the expiry time. There are around 105 trading days in total. Figure 1 is the output of the code with the iteration number set to 100. The continuous blue, green and red curves are extracted from the options data of IBM on Yahoo Finance, respectively standing for the last traded price, the bid price and the ask price. The circles are generated by the binomial pricing model which is trying to approximate the real options prices. From the figure, almost all circles fall on the green and red curves and the errors is minimal and can be further argued that the options data are for American options while the binomial pricing model is intended for European options. There are some points from the blue curvature deviating from the price produced by the model and even from the bid and ask prices. This is because the last price could be traded days ago, meaning they might be out of date data in some sense.

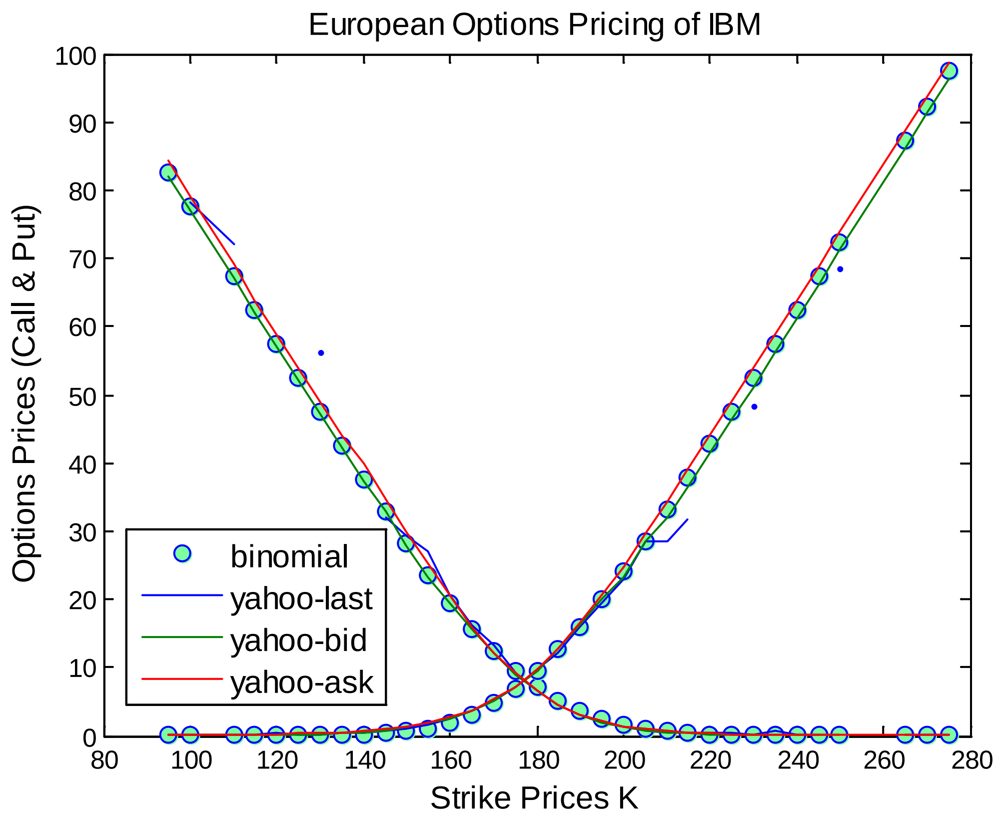

Figure 1. European options of IBM

References

[1] http://tradingwithmatlab.blogspot.com/2011/01/yahoo-options-data-web-scraper-in.html. Web visited on 12/1/2013.

[2] http://luminouslogic.com/stock-market-get-quote-matlab.htm. Web visited on 12/1/2013.

[3] http://finance.yahoo.com/. Web visited on 12/1/2013.

[4] http://www.mathworks.com/matlabcentral/fileexchange/24811-historical-volatility. Web visited on 12/1/2013

[5] http://www.morningstar.com/. Web visited on 12/1/2013.

[6] http://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=billratesAll. Web visited on 12/2/2013.

[7] http://www2.math.umd.edu/~petersd/424/. Web visited on 12/1/2013.