Universal Journal of Applied Mathematics 10(3): 43-48, 2022 http://www.hrpub.org
DOI: 10.13189/ujam.2022.100301

Numerical Approaches of Pricing European Options in the Cox-Ross-Rubinstein Models

Hai Phan[1], **Seonguk Kim**[2,*]

[1]Department of Computer Science, DePauw University, Greencastle, 46135, Indiana, United States

[2]Division of Natural Science, Applied Science, and Mathematics, Defiance College, Defiance, 43512, Ohio, United States

Received July 6, 2022; Revised October 31, 2022; Accepted November 14, 2022

Cite This Paper in the following Citation Styles

***(a)**: [1] Hai Phan, Seonguk Kim, "Numerical Approaches of Pricing European Options in the Cox-Ross-Rubinstein Models," Universal Journal of Applied Mathematics, Vol.10, No.3, pp. 43-48, 2022. DOI: 10.13189/ujam.2022.100301*

***(b)**: Hai Phan, Seonguk Kim (2022). Numerical Approaches of Pricing European Options in the Cox-Ross-Rubinstein Models. Universal Journal of Applied Mathematics, 10(3), 43-48. DOI: 10.13189/ujam.2022.100301*

Abstract The Cox-Ross-Rubinstein (CRR) market model is one of the simplest and easiest ways to analyze the option pricing model. CRR has been employed to evaluate a European Option Pricing (call options) model without complex elements, including dividends, stocks, and stock indexes. Instead, it considers only a continuous dividend yield, futures, and currency options. The CRR model is simple but strong enough to describe the general economic intuition behind option pricing and its principal techniques. Also, it gives us basic ideas on how to develop financial products based on current deviations and volatilities. The paper investigates the CRR model using numerical approaches with python code. It provides a practical event using the mathematical model to demonstrate the application of the model in the financial market. First, the paper provides a simple example to figure out the basic concept of the model. Only a two-period binomial model based on the introductory definitions of the call options makes us understand the concept more easily and quickly. Next, we used actual data on Tesla stock fluctuations from the Nasdaq website (See section 3). We developed the python code to make it easier to understand figures, including tables and graphs. The code allows us to visualize and simplify the model and output data. The code analyzes the stock data to evaluate the probability of the stock's price increasing or decreasing. Then, it used the CRR model to estimate all possible cases for the stock's prices and investigate the call and put option pricing. The code was based on the introductory code of binomial option pricing, but we improved it to get more information and provide more detailed results from the data. The detailed codes are provided in section 3 of the paper. As a result, we believe the CRR model is a fundamental formula, but the improved python code can suggest a new direction for evaluating the probability and investigating the value of the stocks. Also, we expect to develop the code to extend the Black Scholes Pricing model, increasing the number of periods.

Keywords Cox-Ross-Rubinstein Model, Binomial Tree Models, Securities, Derivatives, Options, European Call Option, European Put Option.

1 Introduction

Currently, economists have been using mathematical models to invest in analyzing phenomena in economics and finance [1]. For example, stochastic modeling is one of the forms of financial models used to help understand the circumstances [2, 3]. It presents data and predicts outcomes that account for certain levels of unpredictability or randomness. In addition, companies in many industries have employed stochastic modeling to improve their business practices and increase profitability [4].

The binomial option pricing model is an options valuation method developed in 1979 [5]. It uses an iterative procedure, allowing for the specification of nodes, or points in time, between the valuation date and the option's expiration date. It is simple but solid and effective in describing phenomena in the general economy and deciding on option pricing and its principal techniques.

Numerical Methods form an essential part of options valuation, especially in cases with no closed-form analytic formula [6, 7]. One of the popular numerical methods for option pricing is a binomial tree, the simplest and most widely used method. In particular, the CRR binomial tree is a discrete version of the Black-Scholes constant volatility process. Every numerical solution is just an approximation of the price. The primary purpose is to create the code that allows obtaining the value of a portfolio from the current data of stocks.

This paper is organized in the following. Section 2 introduces the theories for the binomial tree model describing how to estimate the value of an option price. Section 3 explains each code step and shows the tables and graphs illustrating the result. Then, according to the chart and table, we provide the main developments in section 4. Finally, we conclude in section 5.

2 Description of CRR Model

We firstly define the basic definitions of the option pricing based on the textbook, [8, 9, 10, 11].

Definition 2.1 *We define the European call and put option pricing models.*

1. *A call option is a contract by which at a definite time in the future, called the maturity date. The holder of the option may purchase from the option writer an asset known as the underlying asset for a definite amount known as the exercise price or strike price.*

2. *A put option is a contract by which at the maturity date. The holder of the option may sell to the option writer the underlying asset for a strike price.*

3. *The holder of a European option may only exercise it at the end of its life on the maturity date.*

2.1 Simplest CRR Model

The single period binomial model is the simplest CRR model. It is a good model for beginners to learn about mathematical finance. The model is short enough to make calculations while still a robust conceptual model to help beginners comprehend the financial markets. Single period binomial model requires the primary elements:

1. Single interval of time from $t = 0$ to $t = T$ where t is given by:

2. Continuously compounded interest rate r over the interval $[0, T]$.

3. At each point in time, S is assumed to either go 'up' by a fixed factor U by the probability p to become $S_t = SU$, or go 'down' by a fixed factor D by the probability $q = 1-p$ to become $S_t = SD$.

4. Exercise price (strike price) K at the maturity time T.

5. Market for derivatives (such as options) dependent on the value of the stock at the end of the period. The payoff of the derivative to the investor would be $f(SU)$ and $f(SD)$.

The precondition for the binomial model is

$$D < \exp(rT) < U, \tag{1}$$

interpreting that in a realistic financial situation, investment in the risky security may pay better than investment in a risk free bond. The probability to increase the stock price is computed as:

$$p = \frac{\exp(rT) - D}{U - D} \tag{2}$$

In a European call option, payoff value would be

$$f(S_T) = \max(0, S_T - K). \tag{3}$$

In a European put option, payoff value would be

$$f(S_T) = \max(0, K - S_T). \tag{4}$$

It follows from (2), (3), and (4) that the option price for simplest CRR model is calculated as:

$$\text{option price} = \exp(rT)[pf(SU) + (1-p)f(SD)]. \tag{5}$$

2.2 Options On N-Period Binomial Tree

The $N-$period binomial model uses a pair of integers (n, j) to indicate each node in the tree, with $n = 0, 1, \cdots, N$ and $j = 0, 1, \cdots, n$. Over one period of time, an origin node (n, j) leads to node $(n+1, j+1)$ with probability p or leads to node $(n+1, j)$ with probability $1-p$. The index J counts the number of up changes to that time, so $n - j$ is the number of down changes. The stock value at each time step is therefore given by:

$$S_t = S_{t-1} U^j D^{n-j}. \tag{6}$$

There are (n, j) paths that lead to node (n, j) so the probability of going from price S_{t-1} to price $S_t = S_{t-1} U^j D^{n-j}$ is:

$$p_{(n,j)} = \binom{n}{j} p^j (1-p)^{n-j}. \tag{7}$$

Then the option price f for N-period binomial model is calculated as:

$$\text{option price} = \exp(-rT) \sum_{j=0}^{N} p_{(n,j)} f(S_0 U^j D^{N-j}). \tag{8}$$

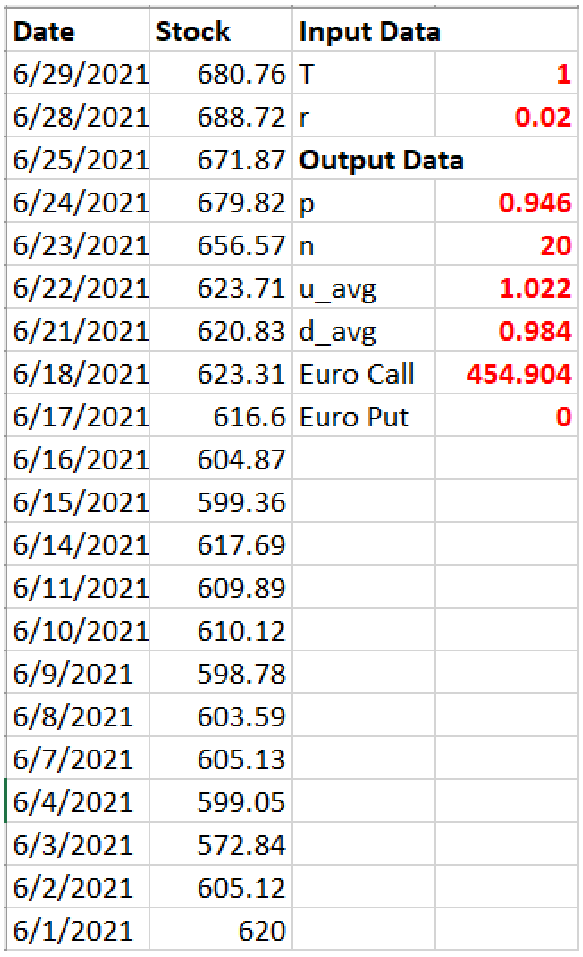
Remark 1 *From the figures and computations, we recognize that the options should be calculated by the future expected values. That is why we can say the option price model is determined by the conditional probability.*

3 Methodology with Python Code

We use and develop a Python code to analyze TSLA stock getting from the Nasdaq website, (https://www.nasdaq.com/market-activity/stocks/tsla/historical). The scope of the history for the stock is within one month of June 2021 (from June 1 to June 29). Number of time steps n = 20 since there are only 20 days in June 2021 that are updated with stock value.

First, the program works with input data from the downloaded .csv file from Nasdaq to calculate and create the binomial tree for stock and European options. It then calculates the average value for u factor u_avg and for d factor d_avg by dividing the total value of u and d factor throughout the interval of time by the number of S_U and S_D (See figure 1).

```
# Calculate the average up and down rate of stock
factor = []
u_count = d_count = 0
u_sum = d_sum = 0.0000
for i in range (length-1):
    x = input_stock[i+1] / input_stock[i]
    factor.append(round(x, 3))
    if (factor[i] > 1):
        u_sum += factor[i]
        u_count += 1
    else:
        d_sum += factor[i]
        d_count += 1
d_avg = d_sum / d_count
u_avg = u_sum / u_count
```



Date	Stock	Input Data	
6/29/2021	680.76	T	1
6/28/2021	688.72	r	0.02
6/25/2021	671.87	**Output Data**	
6/24/2021	679.82	p	0.946
6/23/2021	656.57	n	20
6/22/2021	623.71	u_avg	1.022
6/21/2021	620.83	d_avg	0.984
6/18/2021	623.31	Euro Call	454.904
6/17/2021	616.6	Euro Put	0
6/16/2021	604.87		
6/15/2021	599.36		
6/14/2021	617.69		
6/11/2021	609.89		
6/10/2021	610.12		
6/9/2021	598.78		
6/8/2021	603.59		
6/7/2021	605.13		
6/4/2021	599.05		
6/3/2021	572.84		
6/2/2021	605.12		
6/1/2021	620		

Figure 1. TSLA in June 2021 and the Input-Output data summary.

Second, we use Python to create a Python IDE-based graph showing Tesla stock trends over the time span of June 2021. It is based on the matplotlib.pyplot Python package (See Figure 2).

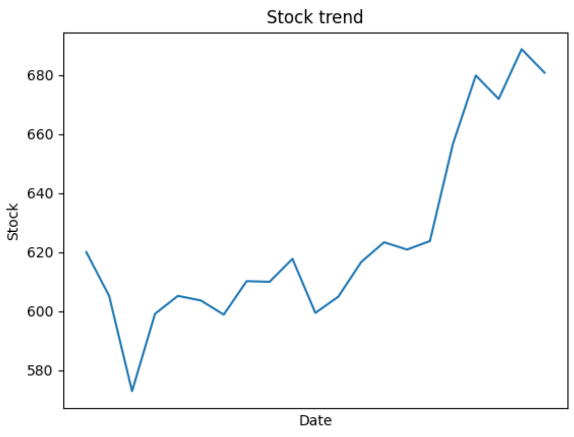


Figure 2. Stock Trend over June 2021.

Third, after getting the input from the $.csv$ file, the code is designed to calculate the average value for 'u' factor u_{avg} and for 'd' factor d_{avg} by dividing the total value of 'u' and 'd' factor throughout the interval of time by the number of SU and SD. The probability p is calculated as in formula (2):

$$p = \frac{e^{rT} - d_{avg}}{u_{avg} - d_{avg}}.$$

```
for i in range (length-1):
    x = input_stock[i+1] / input_stock[i]
    factor.append(round(x, 3))
    if (factor[i] > 1):
        u_sum += factor[i]%u_count += 1
    else:
        d_sum += factor[i]
        d_count += 1
d_avg = d_sum / d_count
u_avg = u_sum / u_count
```

The final step is to calculate stock trends over time and calculate European call and put options using formula (3), (4) and (5). It is designed for the computation for the European call option.

```
# Calculate Euro call payoff
for i in range (0, n+1):
    ECpayoff[i][n] = max(0, stock[i][n] - K)
    for j in range (n-1, -1, -1):
        for i in range (j, -1, -1):
            ECpayoff[i][j]
                = (p * ECpayoff[i+1][j+1]
            + (1 - p) * ECpayoff[i][j+1]) /
np.exp(-1 * r * T)
            finalECpayoff = ECpayoff[0][0]
```

The related code is on the github, GitHub https://github.com/haiphan2471/ binomial-option-pricing-model.

4 Results and Discussion

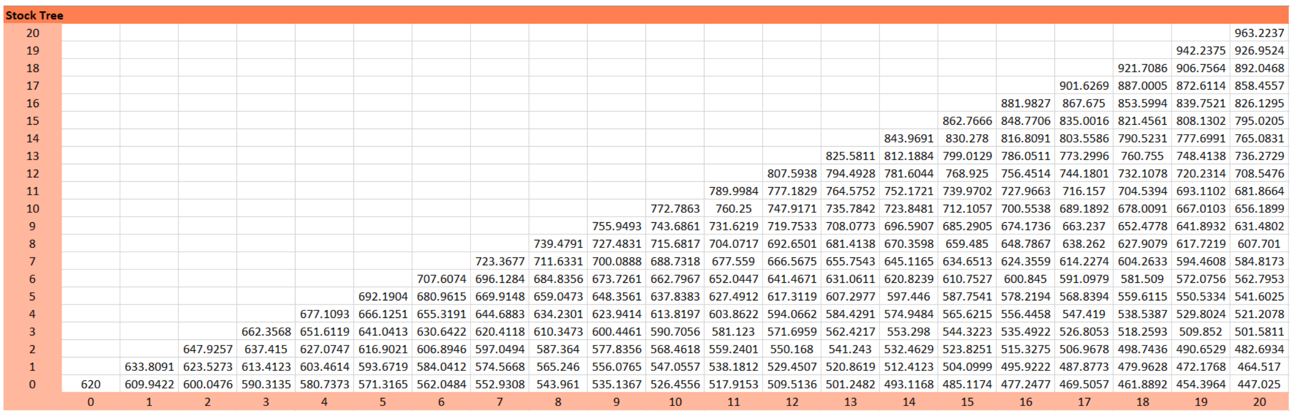
Visualizing the output data is the next step after all the binomial trees are computed, using tables to implement the option trees and the linear graph to implement the stock tree. For instance, Figure 1 shows that according to the data of TSLA stock for three days June 23-24-25, 2021, each day stock is \$656.57, \$679.82 and \$671.87, respectively to create a two-period binomial option model. Using $T = 1, r = 0.02$, and $K = \$656.57$, the Python code makes us create a .xlsx file and build tables to illustrate stock trends and the binomial models of two different option pricing tables. All tables will be placed in one worksheet called "Tables" (See Figure 1).

For stock, the computation moves forward from the first stock value to the stock at the last day. And now for option, backward computation is used, starting from the terminal values at the last day. The rule for option computation is from top to bottom and from right to left, so the first value should be used is the most upper right value. This value is cell (20, 20), together with the below value which is cell (19, 20), will help compute the value of cell (19, 19). This step is done repetitively until the most bottom-left value is computed, which is the final option value that we need.

To understand the tables (See Table 1 and Table 2), it is important to know why options are used. The reason is in a stock market, it makes sense as in the future, companies want to sell at a higher price or buy at a lower price than the current stock value. So option is the risk-minimizing method when companies can predict the future value to decide whether they want to sell or buy. Understanding this concept, when looking at Table 2 for European Call option, the final option value is \$454.9 (See left bottom in Table 2) while the initial stock value in Table 1 is \$620 with strike price at \$620 (See left bottom in Table 1). The option should not be exercised until in the future when it can bring profit to the company. Let's add \$454.9 + \$620 = \$1074.9. This means in the future the stock must be above \$1074.9 so the companies can get profit. The similar understanding can also apply to the three options below.

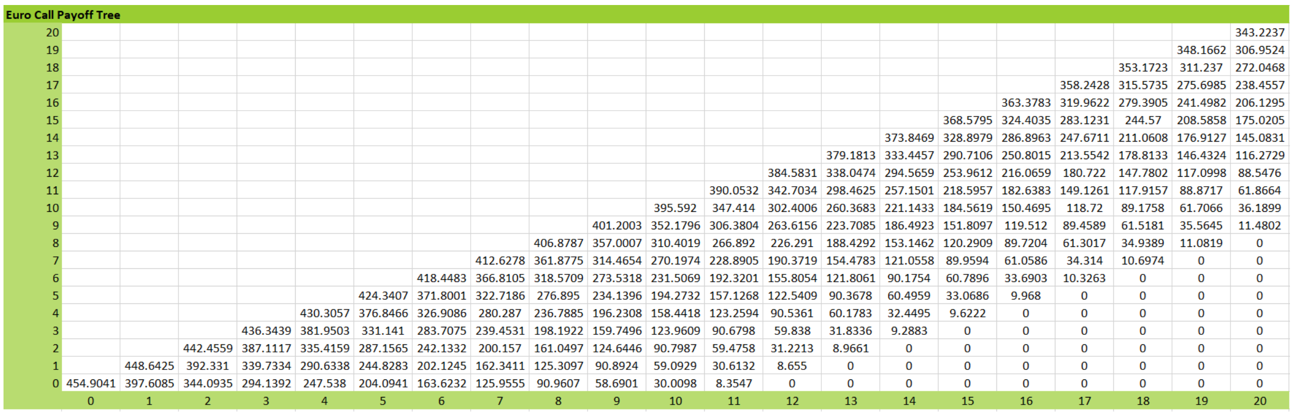
The details of the python code to create the excel files are below.

```
ws['O10'] = p
ws['O10'].font
= Font(color='FF0000', bold=True)
ws['O11'] = finalECpayoff
ws['O11'].font
= Font(color='FF0000', bold=True)
ws['O12'] = finalEPpayoff
ws['O12'].font
= Font(color='FF0000', bold=True)

#___print stock tree___(Orange)

align = Alignment(horizontal='center'
, vertical='center')

ws.cell(1,1, 'Stock Tree')
ws.cell(1,1).font = Font(bold=True)
for i in range(1, n+3): #title line
    ws.cell(1, i).fill
        = PatternFill(start_color='ff7f50',
    end_color='ff7f50', fill_type='solid')
    ws.cell(n+3, 1).fill
        = PatternFill(start_color='ffb79d',
    end_color='ffb79d', fill_type='solid')
for j in range(n+1):
  for i in range(j+1):
      if j == 1:
          for k in range(n+1):# vertical line
              ws.cell(n+1-k+1,j,k)
              ws.cell(n+1-k+1,j).fill
  = PatternFill(start_color='ffb79d',
      end_color='ffb79d', fill_type='solid')
      ws.cell(n+1-k+1,j).alignment = align

  ws.cell(n+1-i+1,j+2,round(stock[i][j],4))
  ws.cell(n+1-i+1,j+2).alignment = align

  ws.cell(n+3,j+2,j)# horizonal line
  ws.cell(n+3,j+2).fill
   = PatternFill(start_color='ffb79d',
     end_color='ffb79d', fill_type='solid')
  ws.cell(n+3,j+2).alignment = align


#___print Euro Call tree___(Green)


ws.cell(n+5,1, 'Euro Call Payoff Tree')
ws.cell(n+5, 1).font = Font(bold=True)
for i in range(1, n+3):
   ws.cell(n+5, i).fill
     = PatternFill(start_color='9ACD32',
   end_color='9ACD32', fill_type='solid')
ws.cell(2*n+7, 1).fill
        = PatternFill(start_color='b8dc70',
end_color='b8dc70', fill_type='solid')
for j in range(n+1):
    for i in range(j+1):
        if j == 1:
            for k in range(n+1):
             ws.cell(2*(n)+5-k+1,j,k)
             ws.cell(2*(n)+5-k+1,j).fill
             = PatternFill(
             start_color='b8dc70'
             , end_color='b8dc70'
             , fill_type='solid')
             ws.cell(2*(n)+5-k+1,j)

  ws.cell(2*(n)+5-i+1,j+2
        ,round(ECpayoff[i][j],4))
  ws.cell(2*(n)+5-i+1,j+2).alignment = align

  ws.cell(2*(n)+7,j+2,j)
  ws.cell(2*(n)+7,j+2).fill
  = PatternFill(start_color='b8dc70'
  , end_color='b8dc70', fill_type='solid')
   ws.cell(2*(n)+7,j+2).alignment = align

#___save the xlxs file___
wb.save('Binomial Option Pricing Model.xlsx')
```



Stock Tree																					
20																					963.2237
19																				942.2375	926.9524
18																			921.7086	906.7564	892.0468
17																		901.6269	887.0005	872.6114	858.4557
16																	881.9827	867.675	853.5994	839.7521	826.1295
15																862.7666	848.7706	835.0016	821.4561	808.1302	795.0205
14															843.9691	830.278	816.8091	803.5586	790.5231	777.6991	765.0831
13														825.5811	812.1884	799.0129	786.0511	773.2996	760.755	748.4138	736.2729
12													807.5938	794.4928	781.6044	768.925	756.4514	744.1801	732.1078	720.2314	708.5476
11												789.9984	777.1829	764.5752	752.1721	739.9702	727.9663	716.157	704.5394	693.1102	681.8664
10											772.7863	760.25	747.9171	735.7842	723.8481	712.1057	700.5538	689.1892	678.0091	667.0103	656.1899
9										755.9493	743.6861	731.6219	719.7533	708.0773	696.5907	685.2905	674.1736	663.237	652.4778	641.8932	631.4802
8									739.4791	727.4831	715.6817	704.0717	692.6501	681.4138	670.3598	659.485	648.7867	638.262	627.9079	617.7219	607.701
7								723.3677	711.6331	700.0888	688.7318	677.559	666.5675	655.7543	645.1165	634.6513	624.3559	614.2274	604.2633	594.4608	584.8173
6							707.6074	696.1284	684.8356	673.7261	662.7967	652.0447	641.4671	631.0611	620.8239	610.7527	600.845	591.0979	581.509	572.0756	562.7953
5						692.1904	680.9615	669.9148	659.0473	648.3561	637.8383	627.4912	617.3119	607.2977	597.446	587.7541	578.2194	568.8394	559.6115	550.5334	541.6025
4					677.1093	666.1251	655.3191	644.6883	634.2301	623.9414	613.8197	603.8622	594.0662	584.4291	574.9484	565.6215	556.4458	547.419	538.5387	529.8024	521.2078
3				662.3568	651.6119	641.0413	630.6422	620.4118	610.3473	600.4461	590.7056	581.123	571.6959	562.4217	553.298	544.3223	535.4922	526.8053	518.2593	509.852	501.5811
2			647.9257	637.415	627.0747	616.9021	606.8946	597.0494	587.364	577.8356	568.4618	559.2401	550.168	541.243	532.4629	523.8251	515.3275	506.9678	498.7436	490.6529	482.6934
1		633.8091	623.5273	613.4123	603.4614	593.6719	584.0412	574.5668	565.246	556.0765	547.0557	538.1812	529.4507	520.8619	512.4123	504.0999	495.9222	487.8773	479.9628	472.1768	464.517
0	620	609.9422	600.0476	590.3135	580.7373	571.3165	562.0484	552.9308	543.961	535.1367	526.4556	517.9153	509.5136	501.2482	493.1168	485.1174	477.2477	469.5057	461.8892	454.3964	447.025
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Table 1. Stock Tree.



Euro Call Payoff Tree																					
20																					343.2237
19																				348.1662	306.9524
18																			353.1723	311.237	272.0468
17																		358.2428	315.5735	275.6985	238.4557
16																	363.3783	319.9622	279.3905	241.4982	206.1295
15																368.5795	324.4035	283.1231	244.57	208.5858	175.0205
14															373.8469	328.8979	286.8963	247.6711	211.0608	176.9127	145.0831
13														379.1813	333.4457	290.7106	250.8015	213.5542	178.8133	146.4324	116.2729
12													384.5831	338.0474	294.5659	253.9612	216.0659	180.722	147.7802	117.0998	88.5476
11												390.0532	342.7034	298.4625	257.1501	218.5957	182.6383	149.1261	117.9157	88.8717	61.8664
10											395.592	347.414	302.4006	260.3683	221.1433	184.5619	150.4695	118.72	89.1758	61.7066	36.1899
9										401.2003	352.1796	306.3804	263.6156	223.7085	186.4923	151.8097	119.512	89.4589	61.5181	35.5645	11.4802
8									406.8787	357.0007	310.4019	266.892	226.291	188.4292	153.1462	120.2909	89.7204	61.3017	34.9389	11.0819	0
7								412.6278	361.8775	314.4654	270.1974	228.8905	190.3719	154.4783	121.0558	89.9594	61.0586	34.314	10.6974	0	0
6							418.4483	366.8105	318.5709	273.5318	231.5069	192.3201	155.8054	121.8061	90.1754	60.7896	33.6903	10.3263	0	0	0
5						424.3407	371.8001	322.7186	276.895	234.1396	194.2732	157.1268	122.5409	90.3678	60.4959	33.0686	9.968	0	0	0	0
4					430.3057	376.8466	326.9086	280.287	236.7885	196.2308	158.4418	123.2594	90.5361	60.1783	32.4495	9.6222	0	0	0	0	0
3				436.3439	381.9503	331.141	283.7075	239.4531	198.1922	159.7496	123.9609	90.6798	59.838	31.8336	9.2883	0	0	0	0	0	0
2			442.4559	387.1117	335.4159	287.1565	242.1332	200.157	161.0497	124.6446	90.7987	59.4758	31.2213	8.9661	0	0	0	0	0	0	0
1		448.6425	392.331	339.7334	290.6338	244.8283	202.1245	162.3411	125.3097	90.8924	59.0929	30.6132	8.655	0	0	0	0	0	0	0	0
0	454.9041	397.6085	344.0935	294.1392	247.538	204.0941	163.6232	125.9555	90.9607	58.6901	30.0098	8.3547	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Table 2. European Call Option Tree.

5 Conclusions

The binomial model that we investigated is using an iterative approach utilizing multiple periods. It is the widely adopted mathematical formula for valuing several kinds of options in economics. Thus, analyzing the model gives us insight into a stream of stocks or assets and estimates their values. Using Python code gave the methodology to explore the model and how to assess the pricing value.

REFERENCES

[1] Arlie O. Petters, Xiaoying Dong. An Introduction to Mathematical Finance with Applications, Springer, New York, 2016.

[2] J.-P. Bouchaud, D. Sornette, The black-scholes option pricing problem in mathematical finance: generalization and extensions for a large class of stochastic processes, Journal de Physique, vol.4, no.6, pp.863–881, 1994, DOI: 10.1051/jp1:1994233.

[3] Masaaki Kijima. AStochastic Processes with Applications to Finance, Chapman and Hall/CRC, New York, 2002.

[4] W. Paul, J. Baschnagel, Stochastic processes, Springer New York, 2013.

[5] John C. Cox, Stephen A.Ross, Mark Rubinstein. Option pricing: A simplified approach, Journal de Physique, vol.7, no.3, pp. 229-263, 1979, DOI: 10.1016/0304-405X(79)90015-1.

[6] P. Henry-Labordere, Automated option pricing: Numerical methods, International Journal of Theoretical and Applied Finance , vol.16, no.8, pp. 1-27, 2013, DOI: 10.1142/S0219024913500428

[7] Lars Stentoft, Value function approximation or stopping time approximation: a comparison of two recent numerical methods for American option pricing using simulation and regression, International Journal of Theoretical and Applied Finance , vol.18, no.1, pp. 65-120, 2014, DOI: 10.21314/JCF.2014.281

[8] Steven R. Dunbar, Mathematical Modeling in Economics and Finance: Probability, Stochastic Processes, and Differential Equations, American Mathematical Society, Rhode Island, 2019

[9] Amir Ahmad Dar, N. Anuradha, One Period Binomial Model: The risk-neutral probability measure assumption and the state price deflator approach , International Journal of Mathematics Trends and Technology , vol.43, no.4, pp. 246-255, 2017,

URL: http://www.ijmttjournal.org/2017/Volume-43/number-4/IJMTT-V43P531.pdf

[10] AA. A. Dar, N. Anuradha, Comparison: binomial model and black scholes model, Quantitative Finance and Economics, vol.2, no.1, pp. 230-245, 2017, DOI: 10.3934/QFE.2018.1.230

[11] AA. A. Dar, N. Anuradha, Studies on European call option of binomial option pricing model using Taguchi's L27 orthogonal array, Quantitative Finance and Economics, vol.7, no.13, pp. 234-249, 2020 DOI: 10.1504/IJIE.2020.104663