Project 1 Report

Hoa H. Duong & Ran Cao

MATH 347 – Financial Mathematics

October 9, 2018

1. Introduction

1.1. Situation and goals

When we think about financial derivative, we usually hope to be able to figure out its value prior to expiration or settlement date T. In order to evaluate the financial derivative price, we describe its value as a function of the underlying asset price and time.

In this project, we seek to demonstrate three primary derivative valuation approaches that we have talked about in class: (1) Binomial Tree model, (2) Monte-Carlo Simulation, and (3) Analytic Model, i.e. the Black-Scholes formula. We aim to illustrate and analyze the convergence of these three approaches through a European put option. We study the convergence pattern of the Binomial Tree method with different number of steps or paths, ranging from 30 to 39, then 300 to 309. In addition, we construct a histogram of the Monte-Carlo Simulation simulated price and perform statistical analysis to show the statistical properties of these simulated price. We then compare how fast the Binomial Tree method converges and how good the Monte-Carlo simulation method is, using the number calculated using the Black-Scholes formula as a benchmark.

Additionally, we also show and compare the value of an American put option with that of a European put option that has the same parameters.

1.2. Parameters

For all of our three approaches, we are using a European put option with the following properties:

- The underlying asset is a stock that follows Geometric Brownian Motion.
- The current stock price, S_0, is \$60.
- The stock's annual volatility, σ, is expected to be 15%.
- The risk-free rate, r_f, is 5% per annum with continuous compounding.
- The strike price of the option, K, is \$62.
- The time to maturity of the option, T, is 6 months, i.e. 0.5 year.

1.3. Assumptions

a. Binomial Tree model

For the Binomial Tree model to work, we have some assumptions:

- We live in a two-state economy, where stock prices will either go up with u or go down with d.
- u, d, and r_f are known and constant.
- Securities are divisible.
- The stock or underlying asset pays no dividend.
- Short sales are allowed.
- No transaction cost.
- The payoff in the up state scenario, f_u, and the payoff in the down state scenario, f_d, are known.
- No arbitrage.

b. Black-Scholes model

For the Black-Scholes model to work, we have some assumptions:

- The underlying asset follows the Geometric Brownian Motion:

$$dS = \mu S dt + \sigma S dZ$$

- Payoff at the expiration date, f_T, is known.
- Short sales are allowed.
- No transaction cost.
- Securities are divisible.
- The stock or underlying asset pays no dividend (note that we can also make the model works if the underlying asset pays dividends, as demonstrated later).
- Continuous trading.
- σ and r_f are known and constant.

1.4. Hypothesis and expected results

For the Binomial Tree method, we expect the result to be closer to the true value calculated by the Black-Scholes formula as the number of steps increases. This means that the difference between the price given by the Binomial Tree model and the price calculated by the Black-Scholes formula decreases as the number of steps increases, so the Binomial Tree model prices converge to the Black-Scholes price.

For the Monte-Carlo Simulation, we hypothesize that the more times we run the simulations, i.e. the higher our number of observations for the price and the smaller our confidence interval for the true value of the put option price is. This means that if we keep running the simulation multiple times, eventually, we will likely be confident enough to say that the price produced by our simulation is not different from the price calculated by the Black-Scholes formula.

2. The Binomial Tree Model

2.1. The 30-period Binomial Tree – part a1

We use the EXCEL file named "Project1_347.xlsx" to derive the 30-period Binomial Tree and approximate the put value. From the time to maturity, T, and the number of steps, n, we calculate the time interval, Δt, using the formula: $\Delta t = \frac{T}{n}$

To calculate the movement of the stock, u and d, we use the following formula:

$$u = e^{\sigma\sqrt{\Delta t}} \text{ and } d = e^{-\sigma\sqrt{\Delta t}} = \frac{1}{u}$$

We then build the stock tree using S_0, u, and d.

The price in the up scenario at column n^{th} is calculated by the formula:

$$S_{u\_n} = S_0 \times u^n$$

The price in the down scenario at column n^{th} is calculated by the formula:

$$S_{d\_n} = S_0 \times d^n$$

The price at any intermediate node at row i^{th} and column j^{th} is:

$$S_{i,j} = S_{i-1,j-1} \times d$$

After building the stock tree, we build the last column in the derivative tree using the payoff function for a put option $Payoff = Max\ (K - S_T, 0)$. We calculate the risk-neutral probability as followed:

$$q = \frac{e^{r_f \Delta t} - d}{u - d}$$

Using the risk-neutral probability and the last column payoff, we utilize the backward induction method to derive the derivative tree.

The payoff at any intermediate node at row i^{th} and column j^{th} is:

$$f_{i,j} = e^{-r\Delta t} \left[q \times f_{i,j+1} + (1 - q) \times f_{i+1,j+1}\right]$$

Then, the put price is f_0, the value at the first column and first row of the derivative tree. We got this value to be 2.788714233.

2.2. Number of steps from 30 to 39 – part a2

We use Python program named "Project1.py" to define and run the function "EuropeanPutOption(s, sigma, t, r, n, k)" that takes $s = S_0$, $sigma = \sigma$, $t = T$, $r = r_f$, n, and $k = K$ to be the parameters and run the simulation as specified in section 2.1 with different number of steps. Table 1 shows the put values for each number of steps.

Table 1. European put option value for 30 to 39 steps

Number of steps	Put option price
30	2.7871
31	2.8035
32	2.7840
33	2.8037
34	2.7812
35	2.8037
36	2.7786
37	2.8034
38	2.7761
39	2.8030

Using the data above, we plot the put option price and the number of steps.

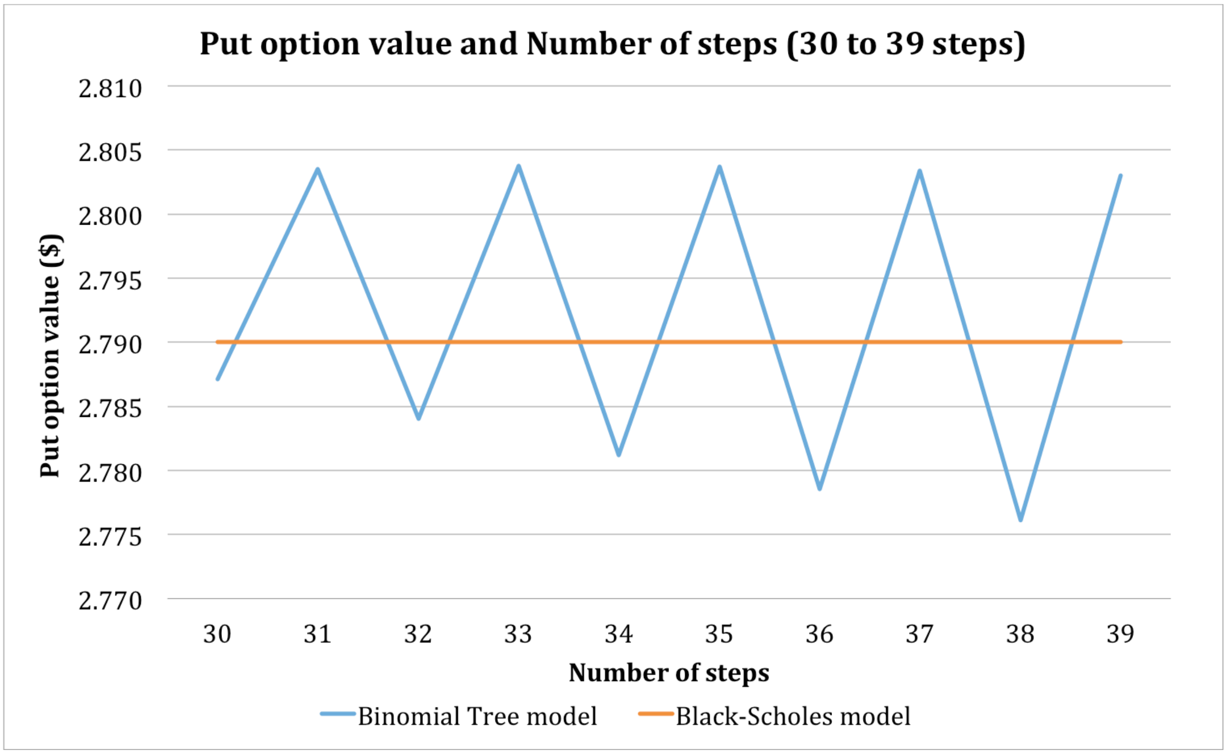


Figure 1. The pattern of the put option value with the number of steps

We see from Figure 1 that as the number of steps increases, the put option price calculated by the Binomial Tree model does not converge to that calculated by the Black-Scholes model. In fact, the price bounces above and below the benchmark.

2.3. Number of steps from 300 to 309 – part a3

We use the same program "Project1.py" to run the simulation with a higher number of steps. We create a table and a graph for to show the results of the pattern.

Table 2. European put option value for 300 to 309 steps

Number of steps	Put option price
300	2.7904
301	2.7892
302	2.7903
303	2.7893
304	2.7902
305	2.7893
306	2.7902
307	2.7894
308	2.7901
309	2.7895

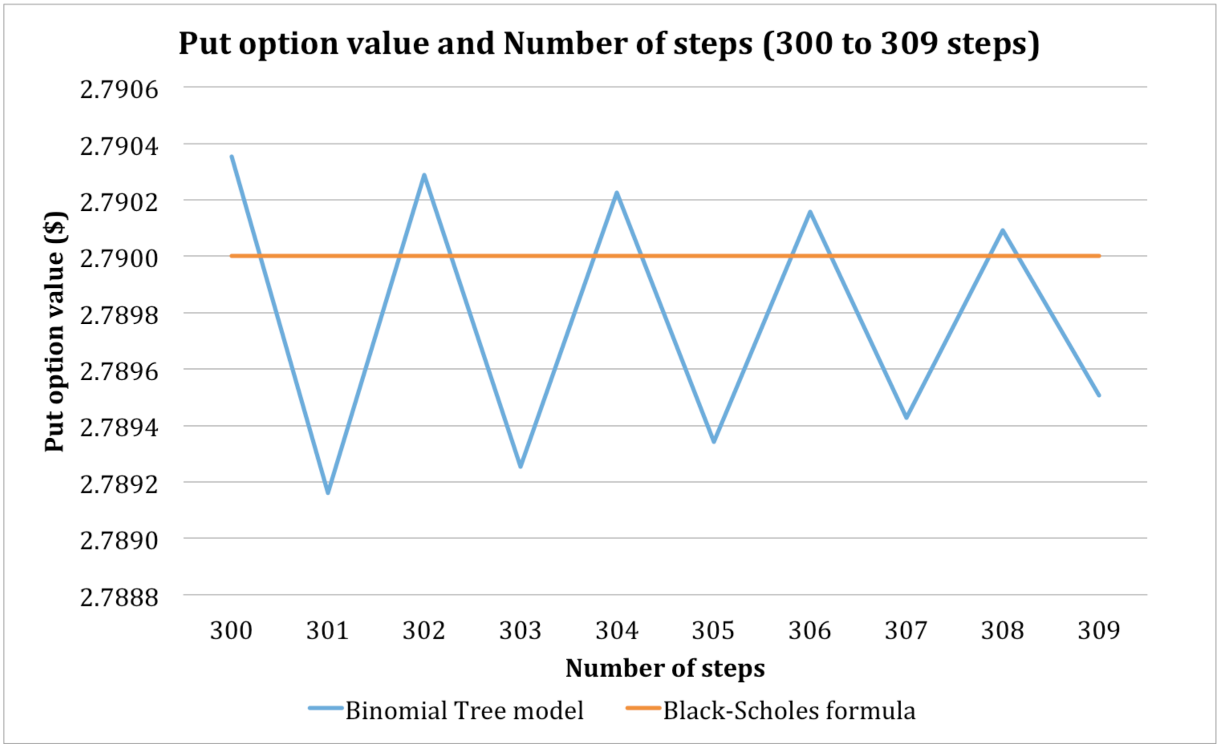


Figure 2. The pattern of the put option value with the number of steps

As the range for number of steps increases to 300 to 309, we start to see the convergence. We see clearly that even though the put option value still bounces above and below the value given by the Black-Scholes formula, the difference between the two models decreases as the number of steps increases. Similar to what happened in Figure 1, the put option values do not converge monotonically. Interestingly, though, note that in Figure 1, the put option value bounces above the value given by the Black-Scholes formula with an odd number of steps (31, 33, 35, 37, 39) but the pattern reverse in Figure 2. Indeed, in Figure 2, the put option bounces above under an even number of steps (300, 302, 304, 306, 308).

Additionally, when we look at the range of the two graphs, we see that in Figure 1, with 30 to 39 steps, the prices given by the Binomial Tree model is within \$0.015 away from the benchmark price given by Black-Scholes. However, in Figure 2, the range is much smaller, being only approximately \$0.0009 away from the benchmark price. Figure 3 shows the difference in magnitude between the two ranges of steps.

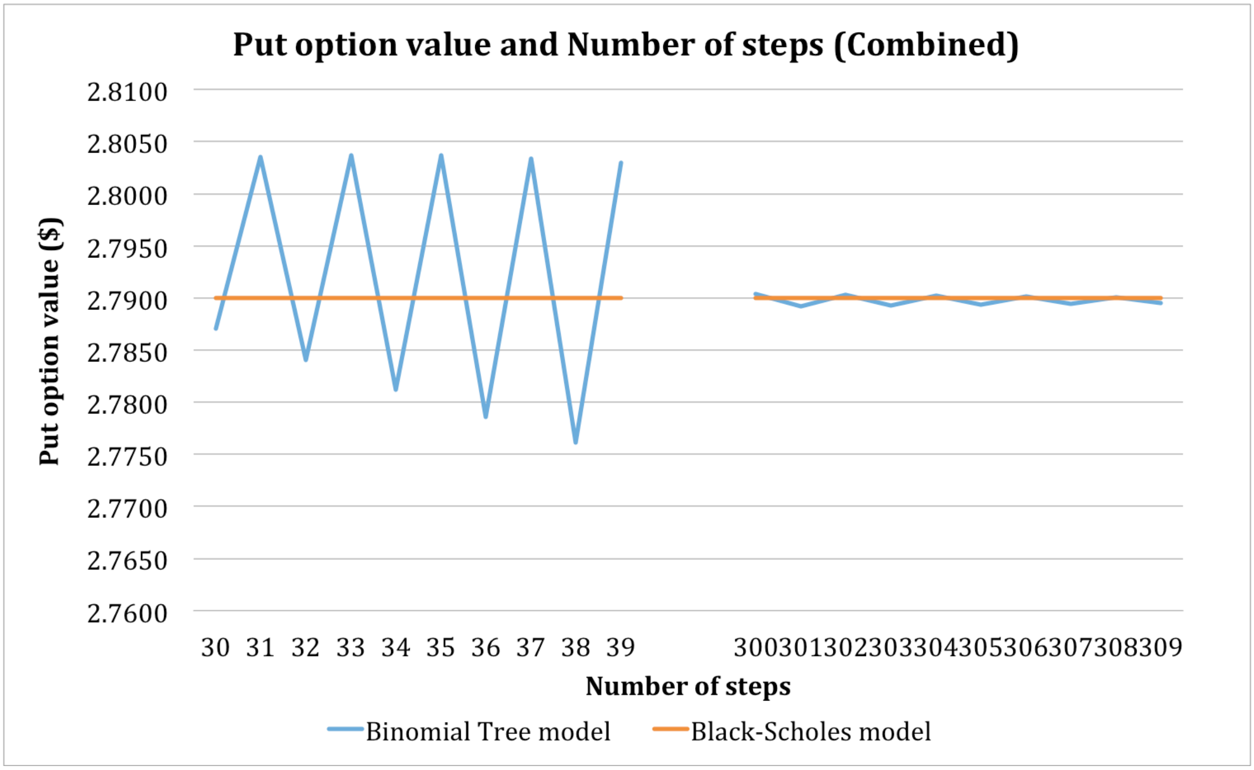


Figure 3. Comparison between two ranges of steps

From Figure 3, it is obvious that the higher number of steps, the more "accurate" our price from the Binomial Tree model will be. "Accurate" in this case means that it is close to the Black-Scholes model results. This suggests that the price calculated by the Binomial Tree model will eventually converge to the price calculated by the Black-Scholes model.

2.4. Adjusting the model for American put option – part a4 + bonus

In the previous sections, as we were calculating the price for a European put option, we know that the option will only be exercised at the end. In this section, we will adjust the model to reflect the property of an American put option, i.e. to be exercised at any point before the expiration date. However, our model has a restriction on the time that the option can be exercised. Instead of being able to exercise at any point, we only allow for intermediate exercise at nodes, which means it can only be exercised at $m \times \Delta t$, for any $m \le n$. Note that in this case, as the number of steps, n, increases, there are more nodes where the American put can be exercised, and thus the model closer replicates the actual property of the American put option.

In order to adjust for this property, in building the derivative tree, at every node, we compare the payoff from keeping the option until the end and immediate exercise. More specifically, the payoff at any intermediate node at row i^{th} and column j^{th} is calculated as followed:

$$Payoff\ from\ keeping\ till\ the\ end = f_1 = e^{-r\Delta t}\left[q \times f_{i,j+1} + (1-q) \times f_{i+1,j+1}\right]$$

$$Payoff\ from\ immediate\ exercise = f_2 = Max\ (K - S_{i,j}, 0)$$

$$f_{i,j} = Max\ (f_1, f_2)$$

The function "AmericanPutOption(s, sigma, t, r, n, k)" in "Project1.py" takes in the same parameter as the "EuropeanPutOption(s, sigma, t, r, n, k)" function, but makes the aforementioned adjustment. Using this program, we are able to find the American put option values for different number of steps, similar to previous sections.

Using this technique, we can run the program to estimate the American put option using different number of steps, similar to previous sections. Table 3 and Table 4 show how the American put option value changes as we change the number of steps.

Table 3. American put option value for 30 to 39 steps

Number of steps	Put option price
30	3.049
31	3.063
32	3.047
33	3.063
34	3.046
35	3.062
36	3.045
37	3.061
38	3.044
39	3.061
Average	3.054

Table 4. American put option value for 300 to 309 steps

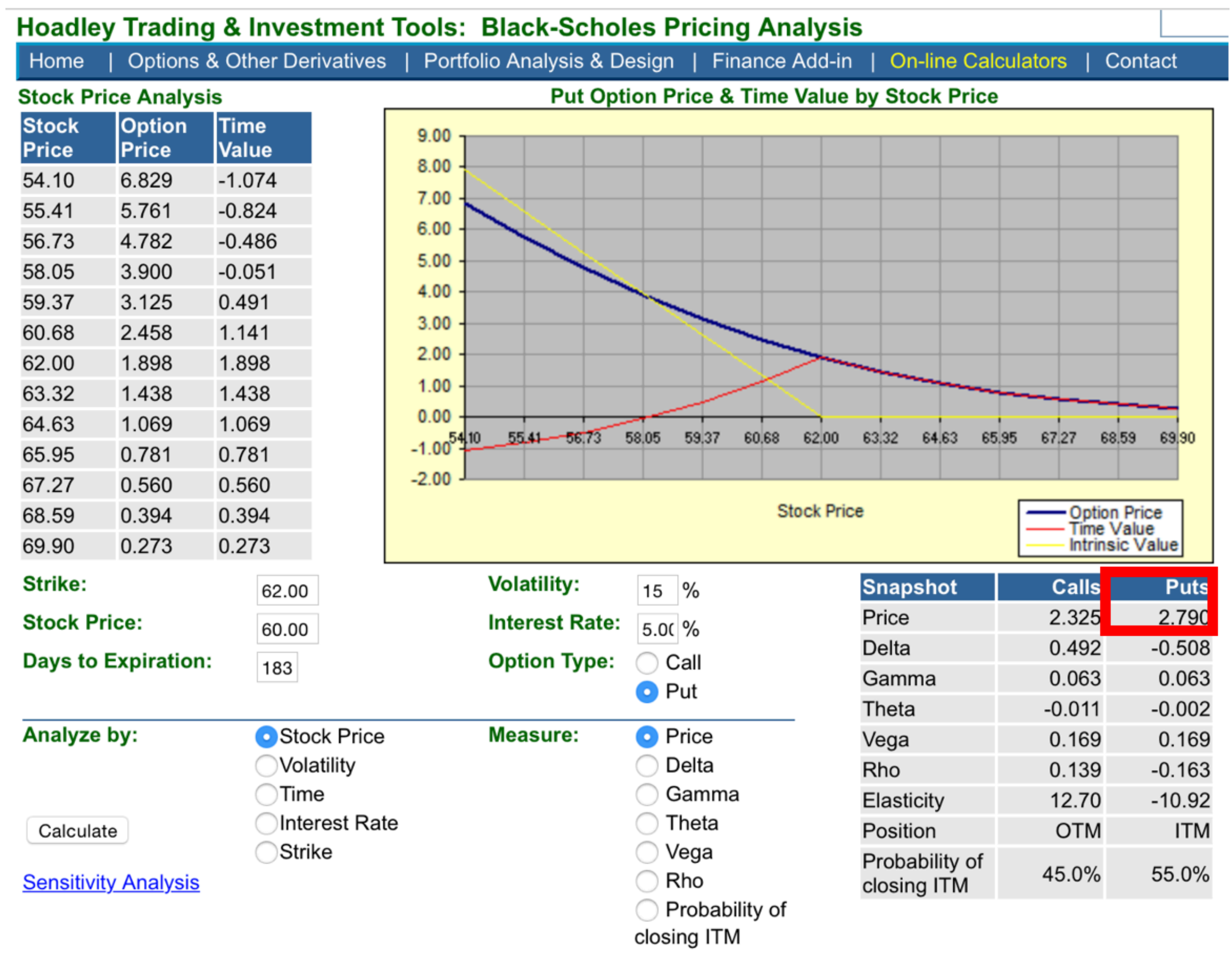
Number of steps	Put option price
300	3.0513
301	3.0509
302	3.0512
303	3.0509
304	3.0512
305	3.0510
306	3.0511
307	3.0510
308	3.0511
309	3.0510
Average	3.0511

We see that as expected, the American put option price is higher than the European put option price, due to the flexibility of exercise time.

3. The Analytic Model (Black-Scholes formula)

3.1. The Black-Scholes output – part b1

We got the put price of 2.790 from an online calculator (hoadley), with no dividend paid. The following graph shows how we chose the parameters with the corresponding put price highlighted in the red box in Figure 4 below.



Stock Price	Option Price	Time Value
54.10	6.829	-1.074
55.41	5.761	-0.824
56.73	4.782	-0.486
58.05	3.900	-0.051
59.37	3.125	0.491
60.68	2.458	1.141
62.00	1.898	1.898
63.32	1.438	1.438
64.63	1.069	1.069
65.95	0.781	0.781
67.27	0.560	0.560
68.59	0.394	0.394
69.90	0.273	0.273

Snapshot	Calls	Puts
Price	2.325	2.790
Delta	0.492	-0.508
Gamma	0.063	0.063
Theta	-0.011	-0.002
Vega	0.169	0.169
Rho	0.139	-0.163
Elasticity	12.70	-10.92
Position	OTM	ITM
Probability of closing ITM	45.0%	55.0%

Figure 4. Put Price without Dividends

(Source: https://www.hoadley.net/options/optiongraphs.aspx?)

If we drop the "With no dividend paid" assumption, by switching "without dividends" to "with dividends" while keeping other parameters the same, the put price increases to 2.839, as figure 5 shows below. Even with a small dividend paid (0.1), the put price is much higher than the numbers we got from Figure 4, from the previous section (Binomial Tree), and from the upcoming section (Monte-Carlo simulation, will be discussed later). Therefore, we showed that Binomial Tree and Monte-Carlo simulation only worked under the condition that no dividends were paid. When we put a higher dividend amount, the put price is even higher. This makes sense, since that means the payoff in the future will be higher, and the expected payoff should also be higher.

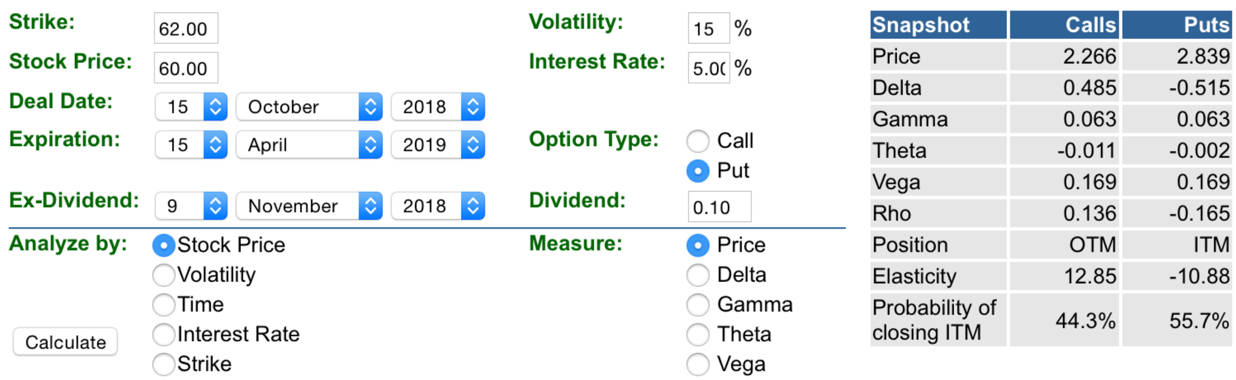


Figure 5. Put Price with Dividends

(Source: https://www.hoadley.net/options/optiongraphs.aspx?divs=Y)

3.2. A story about the Black-Scholes model – part b2

In 1970s, economists and mathematicians started to research financial markets by looking for inspiration in physics, and they believed that the market changes were similar to the erratic movement of molecules and atoms. The most likely paths in market changes, especially the changes in stock price, could be modeled and managed by using statistics, just like the motion of atoms and molecules. This is what Black-Scholes does.

The Black-Scholes-Merton model is a mathematical model for the dynamics of a financial market containing derivative investment instrument. According to name of the model, three great scientists, Fischer Black, Myron Scholes and Robert C. Merton made extraordinary contribution in establishing and developing model, led to a boom in options trading, and provided mathematical legitimacy to the CBOE and other options market around the world.

The model was initiated by Fischer Black and Myron Scholes from in 1970s, they deduced the Black-Scholes formula, which provided a theoretical estimate price of European style options, from the partial differential equation in the model, known as the

Black-Scholes equation and finally published an article entitled 'The Pricing of Option and Corporate Liabilities' in the Journal of Political Economy. Later on, Robert C. Merton expanded the mathematical understanding of the options pricing model and coined term 'Black-Scholes option pricing model'. Scholes and Merton received Nobel Memorial Prize in Economic Sciences in 1997 and Black was not ineligible of the prize because he passed away in 1995, but he was still mentioned as a great contributor to the model.

However, just like a saying, 'All models are wrong, but some of them are very useful' (Box, Empirical Model-Building and Response Surfaces), the Black-Scholes model is very useful, but its deduction is still in theoretical and its assumptions are very ideal. Therefore, the model could underestimate some extreme moves, yielding heavy tail risk and the assumption of constant volatility could lead to a volatility risk. In summary, the result of using the Black–Scholes model differs from real world prices because of simplifying assumptions of the model.

Currently, the model is still widely used, and options market participants made some adjustment and correction to its ideal assumptions and applied the model more suitable to current rapidly changing market.

4. The Monte-Carlo Simulation

4.1. A single Monte-Carlo simulation – part c1

We use R to estimate the put option value by projecting 10,000 trajectories then averaging the 10,000 put payoffs. We then discount the average back by $e^{-rT} = e^{-0.05\times0.5}$. The R-code can be found in the file named "Monte-Carlo.R"

The put value calculated via Monte-Carlo simulation is 2.791445, and this result is close to what we got in section 2 (Binomial Tree model) and section 3 (Black-Scholes model).

4.2. Convergence analysis – part c2

After running the Monte-Carlo simulation with the R-code in the "Monte-Carlo.R" file 10 times, we graphed the put prices in Figure 6 below. The mean value for the put option is 2.789662 and the standard deviation is 0.03617486. We see that this mean value is only $0.0003 away from the value given by the Black-Scholes model. Thus, we can say that the Monte-Carlo simulation result also converges quite nicely to the Black-Scholes model result.

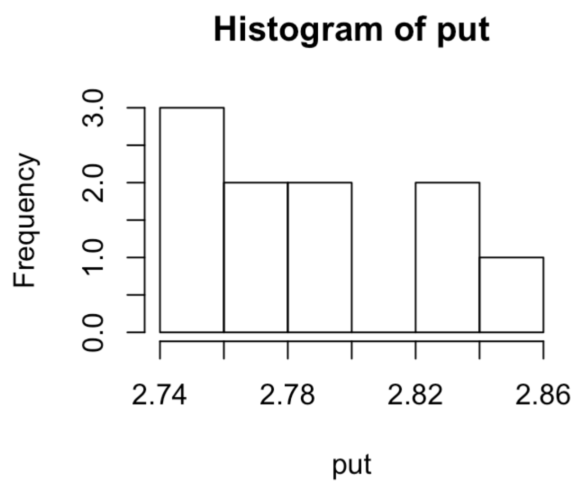


Figure 6. Generating Put Value Through Monte-Carlo simulation

From the figure, we can see that the value ranges from 2.74 to 2.86, but the majority of the time it ranges from 2.76 to 2.84. We also obtain the 95% confidence interval for the true value of put option as demonstrated in Figure 7. We see that the 95% confidence interval is (2.763785, 2.815540), which means that we are 95% confident that the true put option value lies between 2.763785 and 2.815540. Note that the value given by the Black-Scholes formula (2.79) also lies in this range.

```
> t.test(put)

        One Sample t-test

data:  put
t = 243.86, df = 9, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.763785 2.815540
sample estimates:
mean of x
 2.789662
```

Figure 7. 95% confidence interval

After getting the put prices from Monte-Carlo simulation, we conducted a one-sample t-test and the result is shown in Figure 8. Since we are using the value calculated from the Black Scholes model as a benchmark, our hypotheses are:

$$H_o: \mu - 2.79 = 0$$

$$H_A: \mu - 2.79 \neq 0$$

```
> t.test(put-black_scholes)

        One Sample t-test

data:  put - black_scholes
t = -0.029503, df = 9, p-value = 0.9771
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.02621544  0.02554044
sample estimates:
 mean of x
-0.0003375
```

Figure 8. One Sample T-Test on Put Prices simulated by Monte-Carlo Simulation

Here, we have the p-value is 0.9771, which is very high, so we fail to reject the null hypothesis. This means that either the null hypothesis is actually correct and that's why we can't reject it, i.e. the difference between the put option values estimated by Monte-Carlo simulation and the put option value given by the Black-Scholes model is actually 0, or we simply do not have enough data and observations to support the rejection.

4.3. Computing times for Monte-Carlo simulation and Binomial Tree model – part c3 + bonus

After running previous simulation 10 times, we run it again 10 times. The computing times are recorded and shown in histogram in Figure 9, in minutes scale. The mean value is 2.355118 minutes while the standard deviation is 0.4260592 minutes.

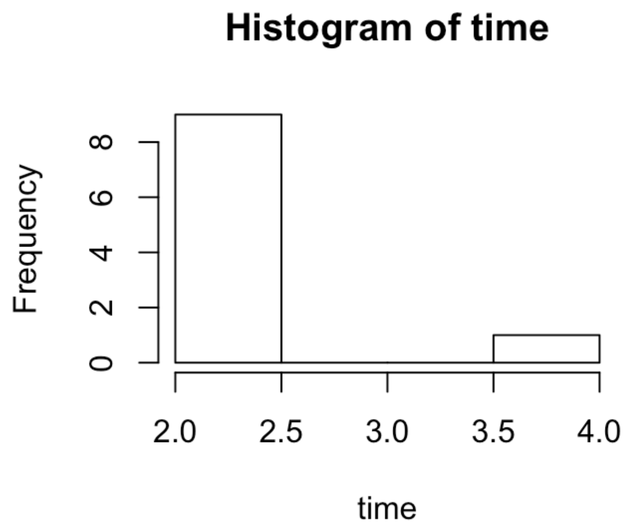


Figure 9. Histogram of The Computing Time with Monte-Carlo Simulation

Normally, the computing time lies in the range of 2.0-2.5 minutes, but there is one time when the computing time is really long, 3.538606, which probably would be an outlier since my laptop was running other programs. However, in section 2, it takes 1.5 seconds to run the Binomial Tree program. The reason is that the number of periods in this Monte-Carlo simulation is too big (10,000 trajectories), whereas we only use 30 steps for the Binomial Tree program. Therefore, we tried two ways to compare the computing time for Monte-Carlo simulation with the Binomial Tree method.

- After changing the number of steps, n, to be 10,000 in section 2 with the Binomial Tree model, the computing time for the Binomial Tree model is 3.24 minutes. It takes longer to execute Binomial Tree methods than running Monte-Carlo simulation.
- Change $n = 300$ in the Monte-Carlo simulation, to match up the same parameter in section 2.3 – part a3, and Figure 10 below shows the spread of computing time after 10 runs, scaled in seconds. Here we have a mean computing time of 3.998215 seconds and standard deviation of 0.410527. We see that the computing time is much shorter compared to when $n = 10{,}000$, but it is still larger than the computing time in section 2.3 with $n = 300$, being only 1 – 2 seconds.

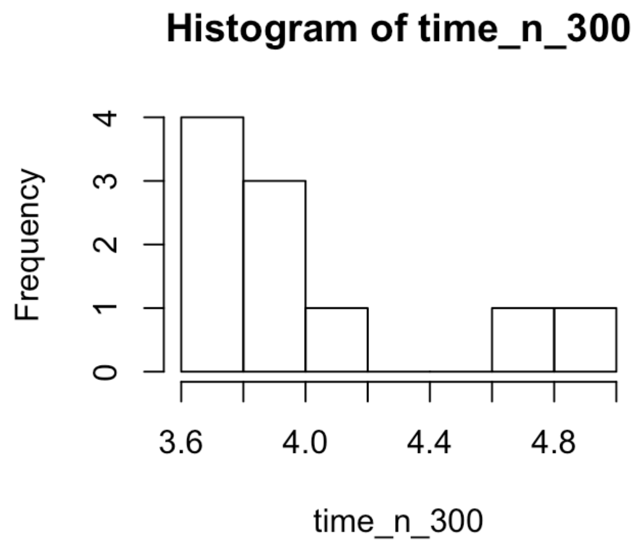


Figure 10. Computing time for Monte-Carlo Simulation when $\mathrm{n} = 300$

5. Conclusion

Through this project, we conclude that all three approaches of pricing a European put option converge and give us the very similar results. The put option value ranges from 2.7892 to 2.7904 using the Binomial Tree model with around 300 steps, and from 2.7638 to 2.8155 using the Monte-Carlo simulation with 10,000 trajectories. The Black-Scholes formula gives us a value of 2.79, which falls into the range of both the Binomial Tree model and the Monte-Carlo simulation.

Additionally, as the number of steps increases for the Binomial Tree model, the put option value gets closer and closer to 2.79, the value given by the Black-Scholes formula. On a similar note, the statistical analysis shows the Monte-Carlo simulation also converges to 2.79.

6. Changing parameters – bonus

In this section, we examine the relationship between the European put price and the parameters that define the option, such as time to maturity, T, strike price, K, interest rate, r_f, and volatility, σ. We hypothesize the following:

- As time to maturity increases, put option value increases, since there is a longer time for the price to change and therefore higher change for it to fall down a below the strike price and earn higher payoff.
- As strike price increases, the put option value increases, since there might be a bigger difference between the strike price and the market price at the expiration date, so the holder earns more.
- As the interest rate goes up, investing money in the stock is more profitable than shorting a put, and therefore the price of the put option will fall.
- As volatility increases, the put option value increases, since the market is more volatile, so there is a higher chance that the price will fall below the strike price and earn higher payoff.

For each of these hypotheses, we only change the value of one parameter while keeping others the same. In each subcase, we evaluate the changing variable with two values, one is bigger than the default value and another is smaller than it. We use both the Binomial Tree model with 300 steps – the "Project1.py" program and the Black-Scholes model – the "hoadley website" for comparison and assurance.

6.1. Put option value and Time to maturity

Using the "Project1.py" program, we run the program with different parameters and report the results. Table 5 shows the how the put option value changes as time to maturity increases.

Table 5. European put option value and time to maturity

Time to maturity	Binomial Tree model	Black-Scholes model
3 months	2.494	2.489
6 months	2.790	2.790
1 year	3.067	3.068

Similar to before, the Binomial Tree model and the Black-Scholes model results are very close to each other. To examine the trend, Figure 11 shows the relationship between the put option value and the time to maturity.

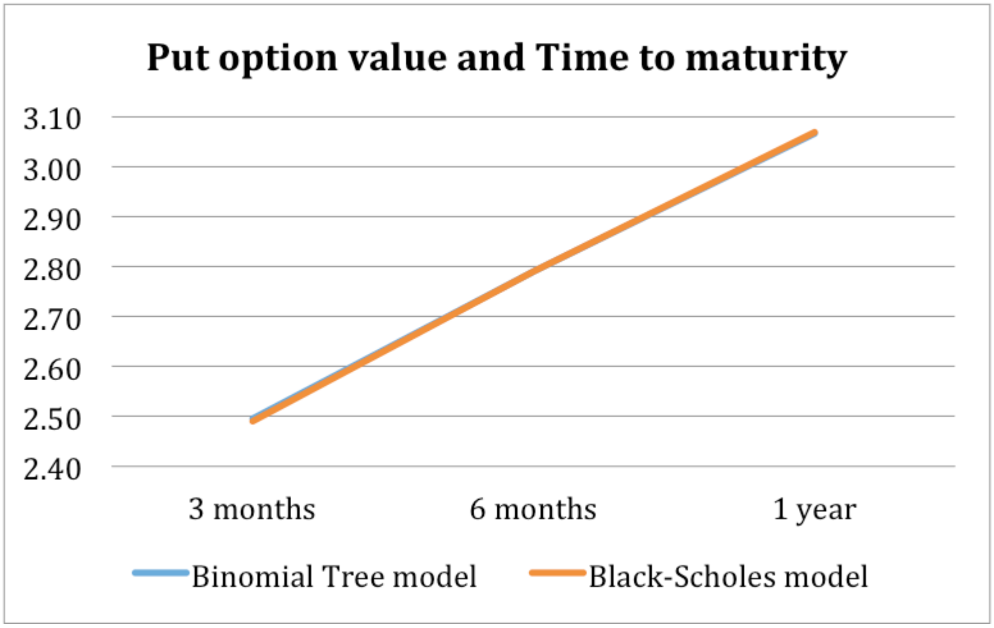


Figure 11. Relationship between put option value and time to maturity

As we hypothesized, the put option value increases as time to maturity increases.

6.2. Put option value and Strike price

Table 6 shows the how the put option value changes as strike price increases.

Table 6. European put option value and strike price

Strike price	Binomial Tree model	Black-Scholes model
61	2.283	2.283
62	2.790	2.790
65	4.651	4.650

We see that the strike price positively affects put price. The payoff that an investor gets equals to the difference between the strike price and the stock price. Hence, with a high strike price, an investor has a higher chance to get a higher payoff. A higher put price is set to cancelling the advantage of a high strike price. Figure 12 shows this relationship more clearly.

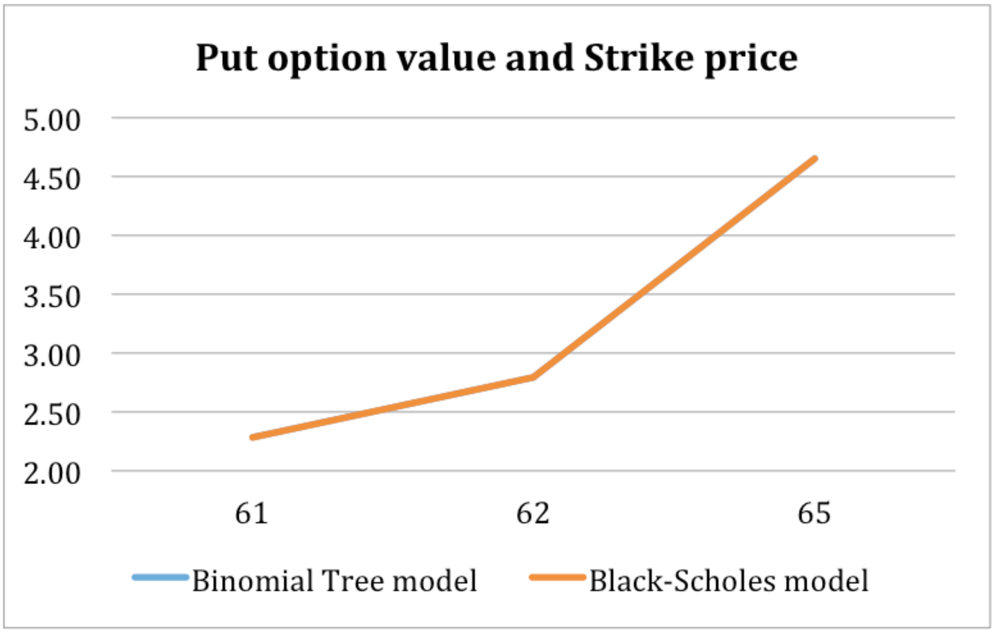


Figure 12. Relationship between put option value and strike price

6.3. Put option value and Interest rate

Table 7 shows the how the put option value changes as the risk-free interest rate increases.

Table 7. European put option value and interest rate

Interest rate	Binomial Tree model	Black-Scholes model
2%	3.319	3.320
5%	2.790	2.790
8%	2.320	2.319

An inverse relationship between the put price and the interest rate can be seen here. According to the Table above, put price is 3.32 when interest rate is 2%; it goes down to 2.79 when interest rate goes up to 5%; and it still expresses a downside trend when interest rate increased to 8%. This relationship is demonstrated visually further in Figure 13. This shows evidence supporting our hypothesis.

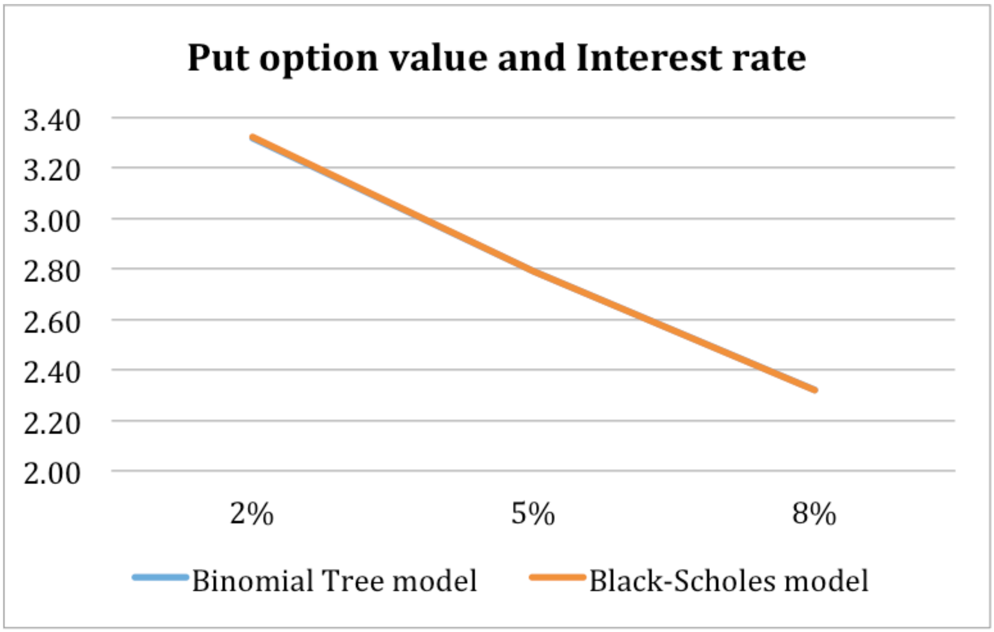


Figure 13. Relationship between put option value and interest rate

6.4. Put option value and Volatility

Table 8 shows the results of the put option value with different volatility figures.

Table 8. European put option value and volatility

Volatility	Binomial Tree model	Black-Scholes model
5%	1.103	1.103
15%	2.790	2.790
25%	4.484	4.485

As Table 8 above has shown, put price equals to 1.103 when volatility is 5%, and it changes to 2.790 when volatility goes up to 15%, and keeps going up when the volatility becomes 25%, while keeping interest rate, strike price, and days to expirations constant. A clear trend is shown that put price increases as the volatility increases, which makes sense since a higher volatility gives a buyer more downside benefits and a higher put price could offset the dominant position. Figure 14 further illustrates this relationship.

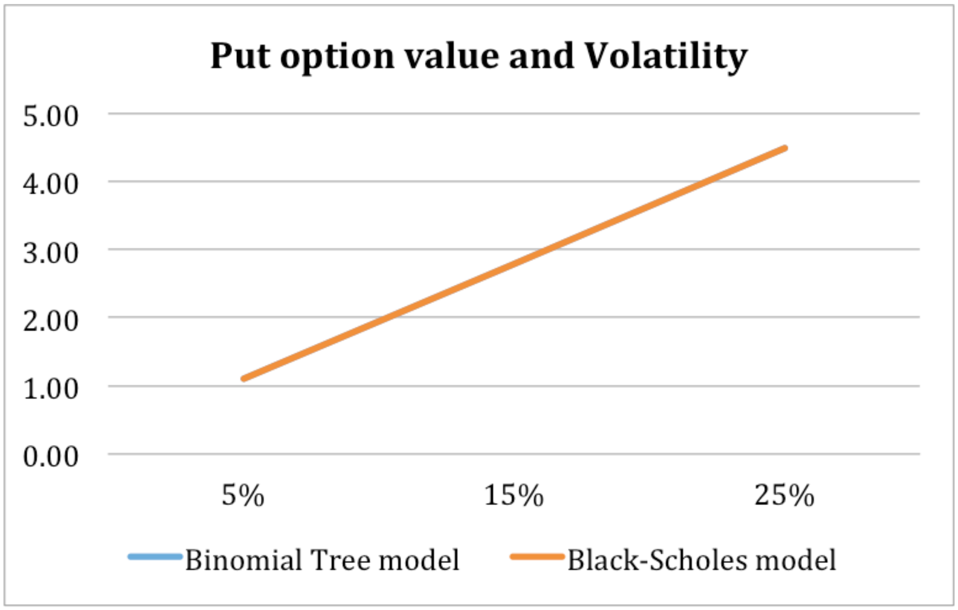


Figure 14. Relationship between put option value and volatility

6.5. Conclusion

In conclusion, we show that the European put price is positively related to the time to maturity, T, the strike price, K, and the volatility, σ. However, it is negatively related to the risk-free interest rate, r_f.

7. Reference

Udemy. "The History of the Black-Scholes Formula." *Priceonomics*, priceonomics.com/the-history-of-the-black-scholes-formula/.

Box, George E. P.; Norman R. Draper (1987). Empirical Model-Building and Response Surfaces, p. 424, Wiley. ISBN 0471810339.

MacKenzie, Donald (2006). An Engine, Not a Camera: How Financial Models Shape Markets. Cambridge, MA: MIT Press. ISBN 0-262-13460-8.

Bodie, Zvi; Alex Kane; Alan J. Marcus (2008). Investments (7th ed.). New York: McGraw-Hill/Irwin. ISBN 978-0-07-326967-2.

Mandelbrot & Hudson, 2006. pp. 9–10.