UNIVERSITEIT VAN AMSTERDAM

REPORT

Black-Scholes Model and Binomial Tree Methods

February 19, 2018

Students:
Sam Ferwerda
10446982

Daan van Ingen
10345078

Teacher:
dr. B.D. Kandhai

Course:
Computational Finance

Code:
5284COFI6Y

Introduction

In this report we will study the use of the Black-Scholes model and program a Binomial Tree using python. We will compare their results and see how the error decreases when we increase the depth of the tree. Of course this will increase the computing time because of the computational complexity. How complex our algorithm exactly is will be determined in question 1.2.
Furthermore, we will take a look at the differences between a European call and put option and a American call and put option. The difference is that in a American call option, the stock can be bought before the discussed date.
At last we will study hedge simulations, where we vary the frequency of the hedge adjustment and we will see that adjusting the hedge parameter more often will lead to a better estimation.

Exercise 1

Question 1.1

As mentioned before we have written our code using Python. With the help of class function we are able to create a Binomial Tree and calculate the option values using back-ward induction using the following formula.

$$f_{i,j} = e^{-r\delta t}(pf_{i+1,j+1} + (1-p)f_{i+1,j})$$

When using a tree of depth 50, the error between our value and the one determined by the Black-Scholes equations is $7.34 \cdot 10^{-3}$.
The second part of the question is about the error between the analytic solution and our model solution knowing that the volatility is different. We would expect that with increasing volatility, the error will enlarge. This is exactly what happens, as can be observed in the figure 1.

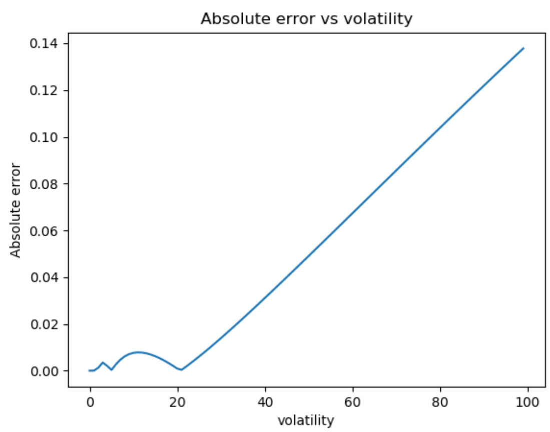

Figure 1: The error with different values of volatility

We do not yet know why there is a small error in the interval $[0, 20]$. For now we assume that it is the result of computer precision, however further research might enlighten this strange occurrence.

Question 1.2

We expected the error to be minimized when the tree is maximized. To prove this claim we have plotted the tree depth versus the absolute error between our model and the analytic values obtained by the Black-Scholes equations. The result is shown in figure 2 on the next page.

A logical consequence of the increasing tree depth is the time that our algorithm needs to calculate all layers. Therefore we are interested in the complexity of the algorithm. Which tells us how much more time the algorithm needs when a extra depth is added. It can be easily observed that every layer has one node more than the layer before him. This means that a layer of depth 5 has $\sum_{k=1}^{6} k$ nodes. The following mathematical formula can help us with calculating the number of nodes in the tree:

$$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1).$$

This means that the complexity of our algorithm is $\frac{1}{2}n^2 + \frac{1}{2}n$ or $O(n^2)$. In figure 2 on the right hand side, we can see how time increases versus depth.

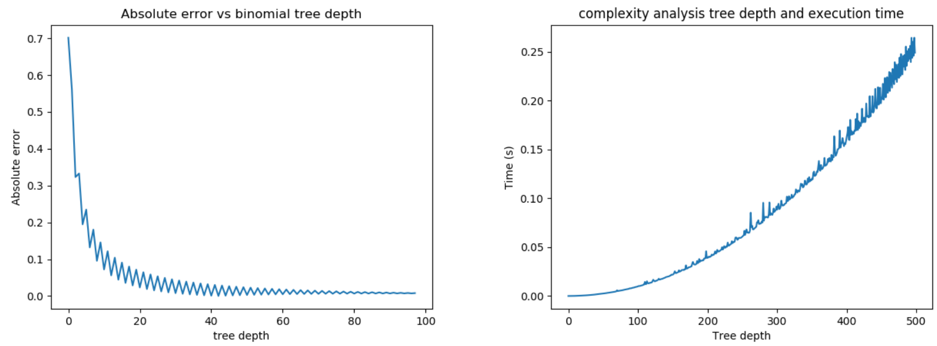

Figure 2: Left: Error vs depth. Right: Time vs depth

Question 1.3

The analytical values of the hedge parameter can be calculated using the normal cumulative distribution function $\Phi(x)$ and the value $d_1 = \frac{\ln(S/K)+(r+\sigma^2/2)\rho}{\sigma\sqrt{\rho}}$, while our model calculates the hedge parameter with the formula $\Delta = \frac{f_u - f_d}{S_u - S_d}$. Fortunately, the error between these two methods is really small. In the following figure we have calculated the hedge parameter with the use of our model and also with the analytical formula and some different values of volatility. The blue dots in this

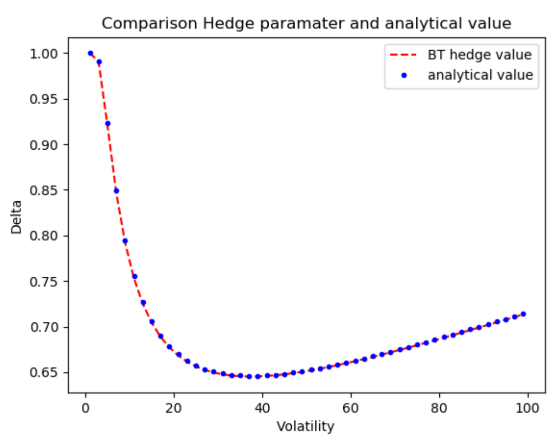

Figure 3: Delta versus Volatility for 2 different methods

graph are the analytical values and seem to laying on the red line which is the function of the delta using our model. It seems to be a great fit.

Question 1.4

We could not observe any difference between the American and European call and put options. This probably has to with the lack of advantage when buying or selling early. In figure 4, all the prices are plotted versus the volatility.
The only difference in our model will be that we want the $Max(f, S_{N,j} - K)$ instead of $Max(0, S_{N,j} - K)$ when using the call option and $Max(f, K - S_{N,j})$ instead of $Max(0, K - S_{N,j})$ when using the put option.

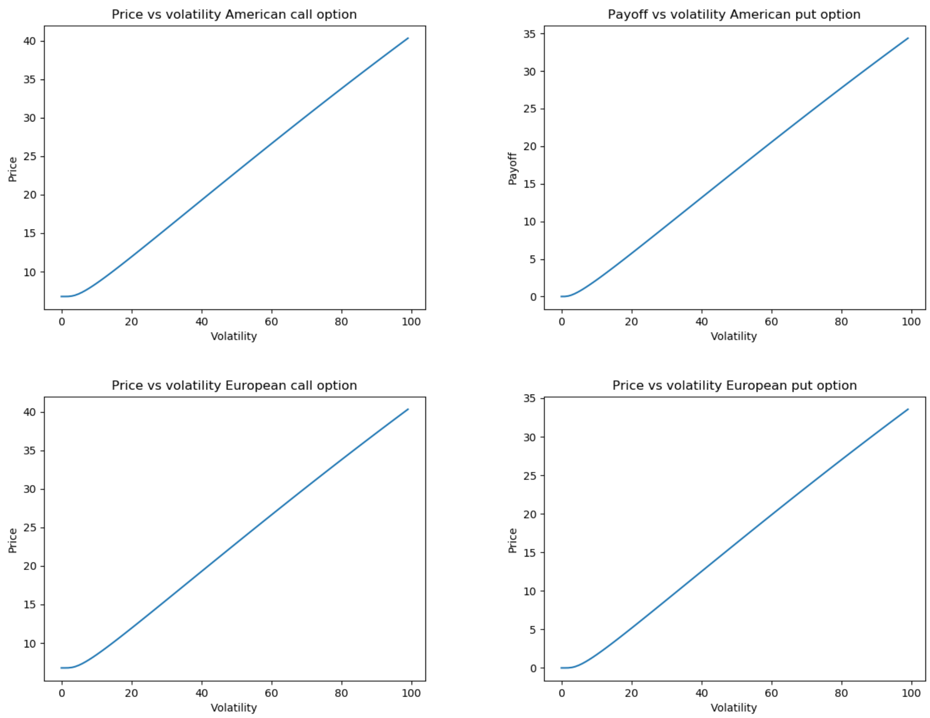

Figure 4: Call and put options for European and American

Exercise 2

Every value that we used will be the same in this exercise. We will however adjust the hedge parameter and notice that this closely relates with the standard deviation. In this study we have created 10000 samples to work with. Before we head to visuals, let us discuss what we think will happen when frequently adjusting the hedge parameter. One of the first things that should come to mind is that the more frequent you observe something, the more trustworthy it becomes. That is also exactly what we expect to happen when frequently adjusting the hedge parameter. See figure 5.

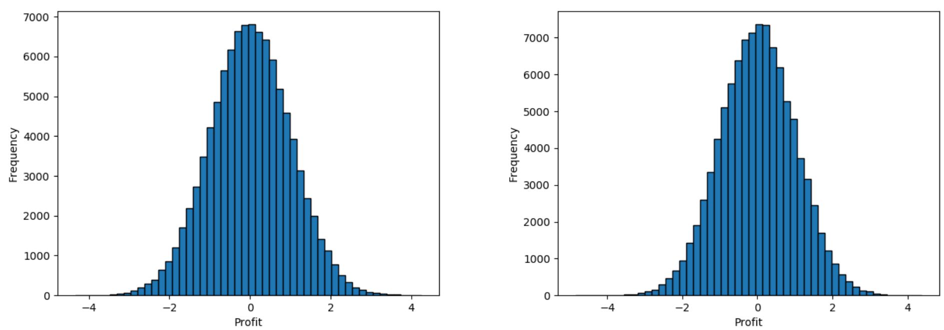

Figure 5: Daily and weekly hedge adjustment

Even though they seem to be similar, the histogram on the right from the weekly adjustment has a far larger standard deviation. Therefore, precision is achieved when we frequently adjust the hedge parameter. In the following graph, we have plotted the standard deviation versus the time between every adjustment (in days).

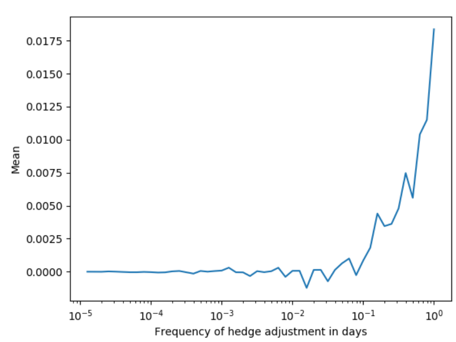

Figure 6: Increased precision with more frequent evaluations

Notice that the y-axes says mean, this is the mean of the standard deviation over all 10000 runs. However if we would have plotted the actual mean, it would look rather similar since it would converge to zero.