Homework 4: MATH 548

Zachary Denis André Scialom
A20497400

18 November 2021

Contents

1 Presentation of the zip file

First of all, let's present the content of the zip file. Therefore, in this section, the data and certain choices will be described and explained. Then, the functions will be described so that one can understand its purpose.

1.1 Data chosen

At first, all the data files have been obtained thanks to the Python file called $option - data.py$. There are 5 files for the data.
For what I considered as a "short" expiration (9 trading days), two files have been used:

- data2.csv
- data3.csv

data2.csv contains the real call options prices of Apple taken on the 8th of November for an expiration on the 19th of November. On the 8th of November, $S_0 = 150.44$ at close. Based on the volume observed, I decided to study strike prices from 145 to 175 (16 values).
data3.csv contains the real put option prices of Apple with the same dates as above. Based on the volume observed, I decided to study strike prices from 136 to 155 (17 values).

Then, what I considered as a "long" expiration (70 trading days), two files have been used:

- data4.csv
- data5.csv

data4.csv contains the real call options prices of Apple taken on the 15th of November for an expiration on the 18th of February 2022. On the 15th of November, $S_0 = 149.87$. Based on the volume observed, I decided to study strike prices from 140 to 190 (11 values).
data5.csv contains the real put option prices of Apple with the same dates as above. Based on the volume observed, I decided to study strike prices from 120 to 170 (11 values).

I also decided to choose $r = 0.1\%$ in all my scripts. At last, another file called $AAPL.csv$ has been used especially to do the method of estimation which is based on the ratio $\frac{S_j}{S_{j-1}}$. This file contains the stock prices of Apple from the 9th of August 2021 to the 5th of November 2021.

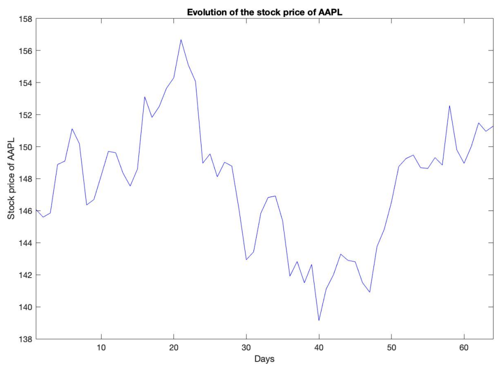

Figure 1: Evolution of AAPL stock prices over the last 4 months

1.2 Functions

One of the main priorities of this project was to find the coefficients u and d. To do so, two methods can be used.

1.2.1 Method 1

The first methods consists in calibrating the parameters. By trying different values for u, we try to minimize the root-mean squared error. The optimal parameter is the one which minimizes the RMSE. Here are the different files related with this method:

- long-expiration-calls.m
- long-expiration-puts.m
- short-expiration-call.m
- short-expiration-put.m

In other words, we are looking for:

$$\arg\min_x \sqrt{\sum_{i,j} |ModelPriceCall(u,d,T_j,K_k) - MarketPrice(Call,T_j,K_k)|^2}$$

These are 4 different main scripts. There are just the data studied and the choices related to the strike prices that vary from one file to another.
When I was calibrating, I always considered $d = \frac{1}{u}$.

1.2.2 Method 2

The second method consists in estimating the parameters u and d with accurate expressions. Indeed, we have:

$$\hat{\mu} = (\frac{1}{N}\sum_{j=1}^{N}(\frac{S_j}{S_{j-1}} - 1)\frac{1}{\Delta t} \tag{1}$$

And,

$$\hat{\sigma} = \sqrt{(\frac{1}{N-1}\sum_{j=1}^{N}(|X_j - \bar{X_N}|^2)\frac{1}{\Delta t}} \tag{2}$$

At last:

$$u = 1 + \hat{\mu}\Delta t + \hat{\sigma}\sqrt{\Delta t} \tag{3}$$
$$d = 1 + \hat{\mu}\Delta t - \hat{\sigma}\sqrt{\Delta t} \tag{4}$$

Here are the files related to that method:

- main-method-2.m
- volatility-estimation.m
- drift-estimation.m
- optimal-u-d.m

The file main-method-2 is the main script where I estimate the parameters u and d by using the other functions listed above that respectively enable to obtain an estimation of σ, μ and at last u and d.

1.2.3 Pricing

Obviously, obtaining u and d is important, but then, functions to price the options must be developed. That is what I did using the two methods seen in class.
Thus, in the file $pricing - option - 0.m$, I have used the method where we find H_0 and H_1 for every omega. It appeared that I was limited by this method (explanations below). That is why I also did the method with the risk-neutral probabilities that works very well. This method of pricing is contained in the file $risk - neutral - pricing.m$.
To be able to price, I also did a function $binomial - tree.m$ that enables to build the binomial tree with the right coefficients and time steps.
When I obtained my results with the second method (estimation), usually d was not equal to $\frac{1}{u}$. Therefore, I developed similar functions where d was added as an input.
At last, another function has been realized to obtain forward prices, called $forward - prices.m$. It has been tested in a small script called $test - forward.m$. It seems to work, giving the hedging strategy.

2 Presentation of the results

2.1 Call options: Long expiration

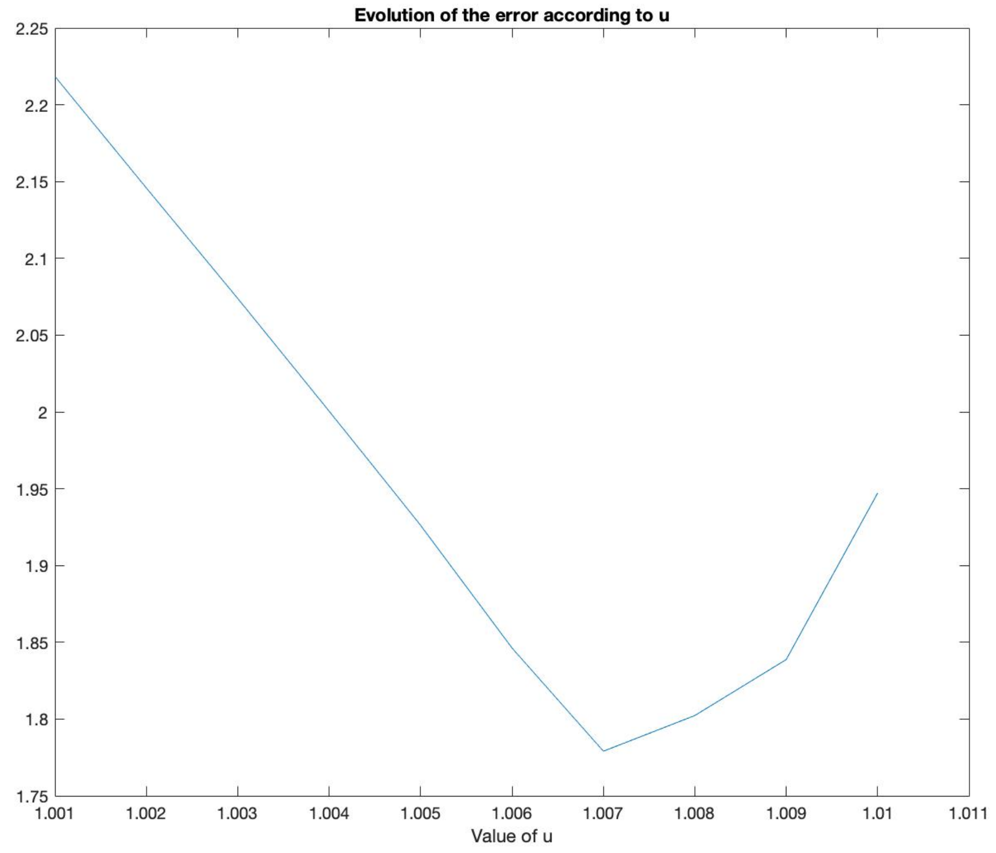

Figure 2: Evolution of the RMSE according to u: comparing real prices with model prices

On the figure 1, we can clearly see that the optimal u that must be chosen must be equal to 1.007.

These results have been obtained with the first method of "calibrating" the parameter u (Engineering way). When I use the second method of estimation, I obtain the following results: $u = 1.013$ and $d = 0.9883$.
The drift estimation gives me $\mu = 0.0025$ and $\sigma = 0.0245$.
The method that gives me the best result in terms of "being the closest" to the real market value is the method of calibrating even though the second method gives satisfying results. Maybe I should have taken more months of Apple stock prices to have a higher volatility and a higher drift.

Then, the evolution of price of the European call option has been ploted according to the strike prices chosen. Obviously, I chose to plot this curve with the optimal $u = 1.007$ found above.
This declining curve makes sense as the higher is the strike price, the more the option is out-of-the money.

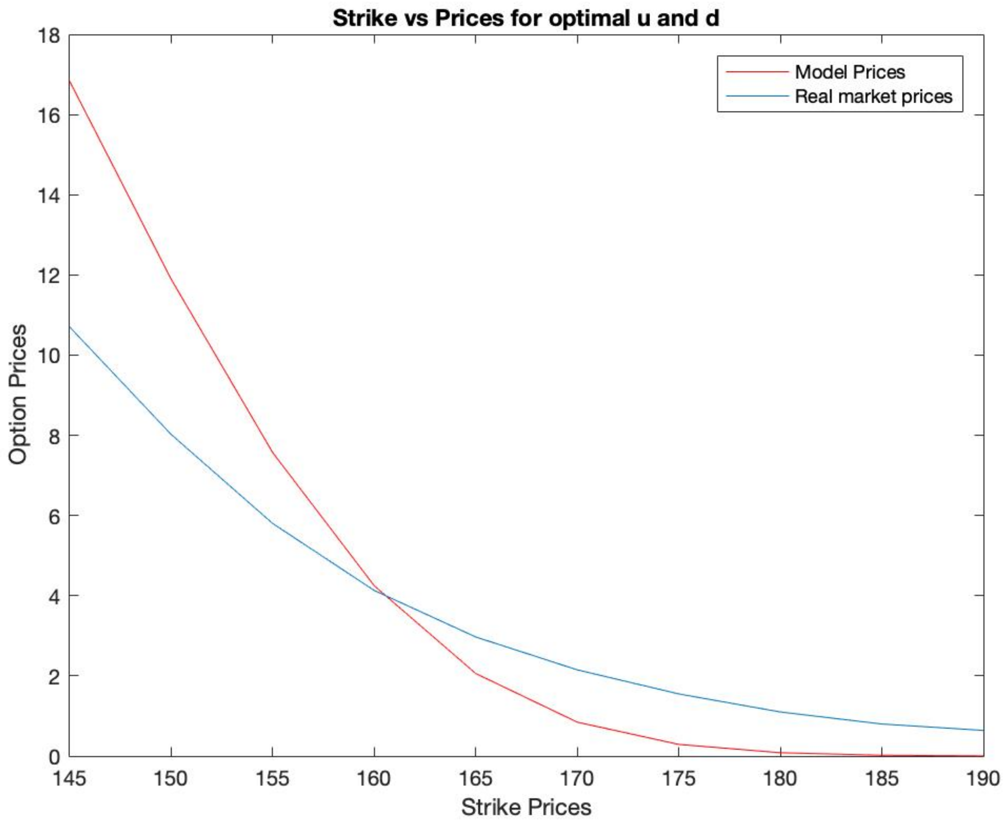

Figure 3: Strike prices vs Call Option prices obtained

2.2 Put options: Long expiration

The same work has been realized with put options. Here, we can see that the optimal u is equal to 1.02. For that value, the RMSE has been minimized effectively.

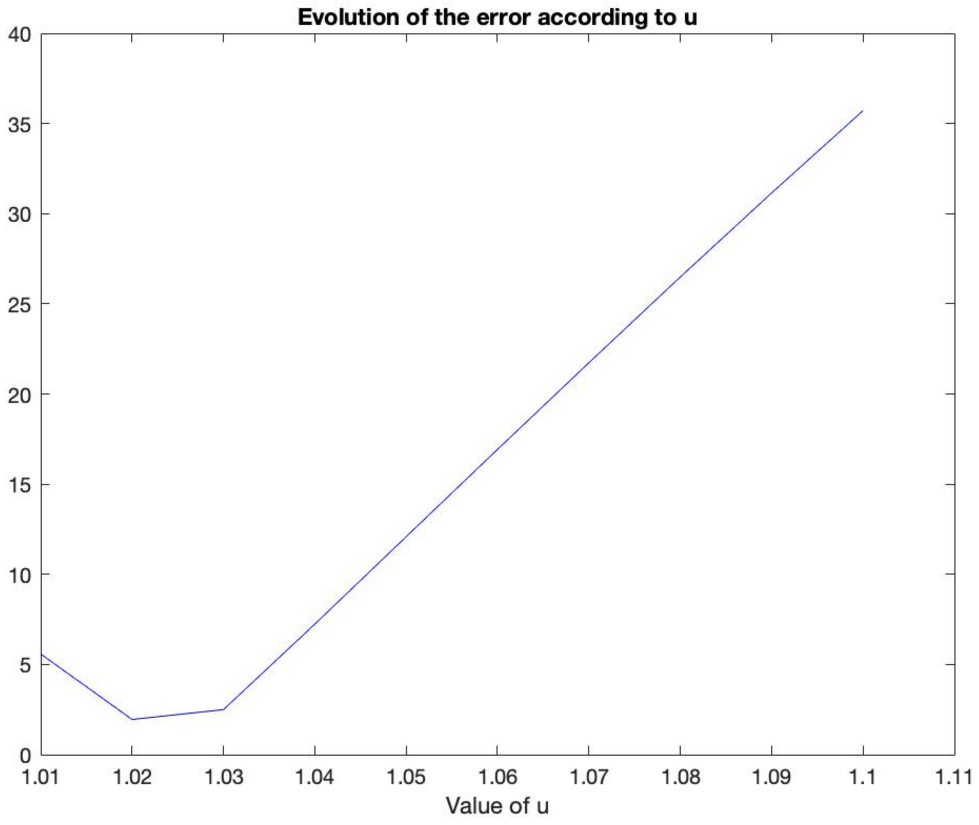

Figure 4: Evolution of the RMSE according to u

Then, the evolution of the price of the put options according to the strike prices chosen has been ploted. Once again, this curve makes sense. Above 150, the option clearly gets in-the-money. Therefore, it gets more expensive.

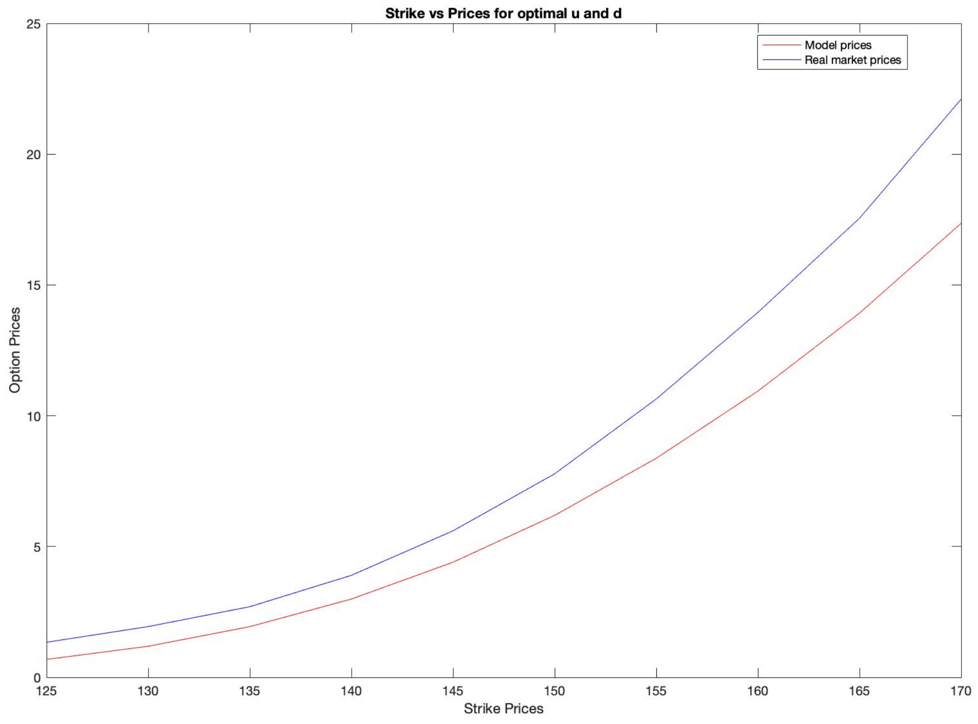

Figure 5: Strike prices vs Put Option prices

Generally, these results seem to make sense. The value of u obtained varies according to the method chosen (estimation or calibration) but there is not a significant difference between the two methods to obtain the optimal parameters. The risk-neutral probability pricing method has been used for long expiration.
The value of u is around 1.01 which makes sense as it would mean an a variation of 1% daily. Also, with the second method, d is close to be equal to $\frac{1}{u}$.

At last, it is not exactly equal to the real market. Many factors could justify those differences. The model remains an approximation of reality. Then, I maybe should have considered continuous compounding for the pricing. Obviously, the code is not optimal and can be improved. Moreover, the pricing for American options must be done (next HW): it may give better results as they are more used in real life.

2.3 Some results on a short expiration

As it has been explained above, two methods have been tried to price the options. With the risk-neutral probabilities, the code runs very well and it is fast with a good accuracy. However, it was not my first choice.
The first method I used consisted in finding what we called H_0 and H_1 in class. Therefore, this method works also very well but only for short expiration, meaning a low time step in my code. For higher time-steps, it genuinely is time-consuming. Nonetheless, this method enables to obtain H_0 and H_1 that I denoted by $sol1$ and $sol2$ in my code. I chose to not put them in outputs as it already takes a lot of time without them.
Thus, here is a comparison of the results obtained with the two functions (call options, T=9).

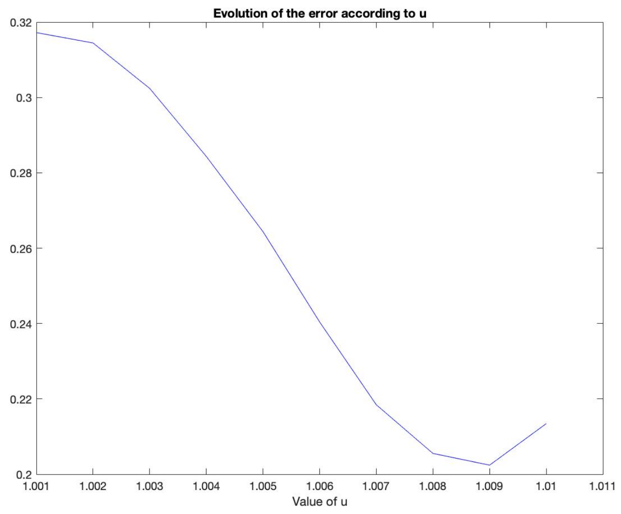

Figure 6: Evolution of the RMSE according to u: Risk-neutral method

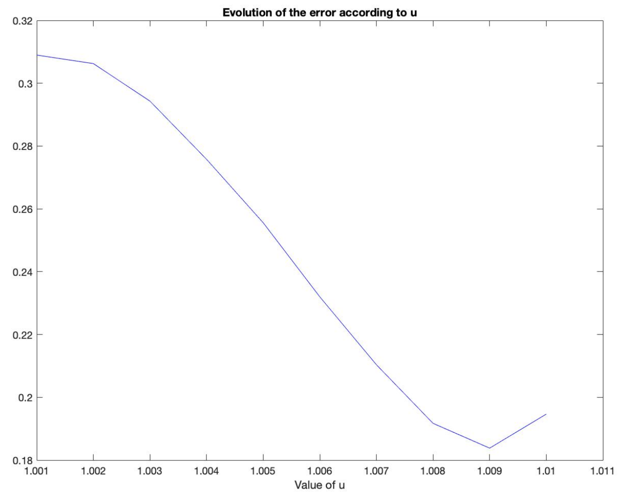

Figure 7: Evolution of the RMSE according to u: H_0 and H_1 method

The method where I find H_0 and H_1 is even more accurate than the risk-neutral pricing method. Also, it enables to obtain the hedging strategy at each time. Again, here $u_{optimal} = 1.009$ which makes sense.

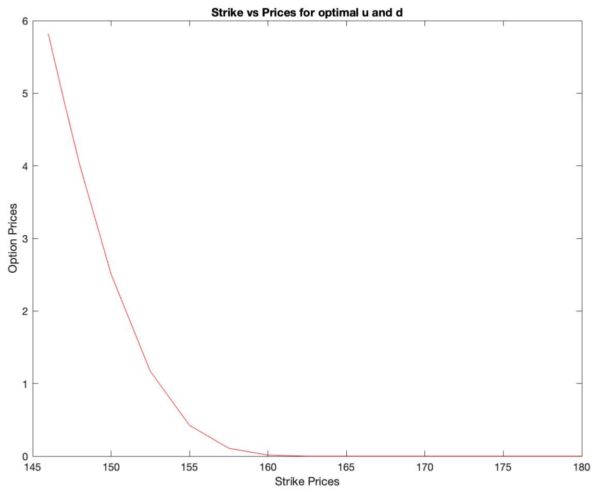

Figure 8: Strike prices vs Call Option prices: Risk-neutral method

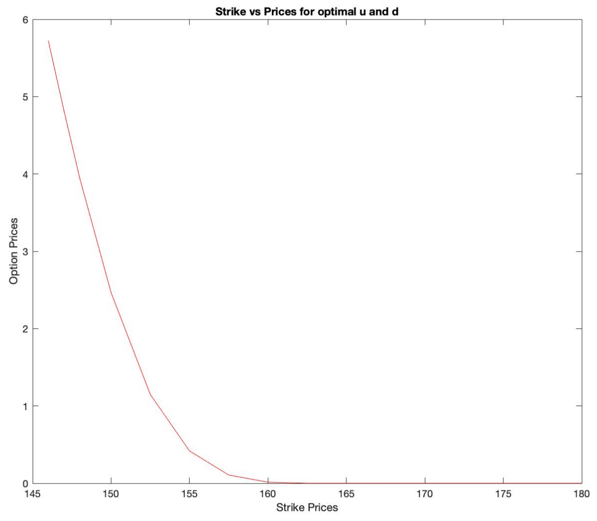

Figure 9: Strike prices vs Call Option prices: H_0 and H_1 method

3 Code explanation

For the estimating method, I mainly followed what we did in class. I computed the different expressions needed.

For the second method, I needed to test different values for u and for different strikes. This is why I created a variable called $calibrate - option - price - call$. Each column of this variable corresponds to a value of u and each row represents a strike price.

Then, for my pricing functions, first I did a function called $binomial - tree$ that enables me to obtain the binomial tree according to u and T.
To do so, I take all the lines of the previous column thanks to $Binomialtree(1 : i - 1, i - 1)$ that I multiply by u. That enables me do treat the multiplication by d as a "simple" case. Also, it enables me to regroup the values when they are similar (linear construction). Initially, I choose $BinomialTree$ as a $T \times T$ matrix filed with Nan values because then, it simplifies the process of pricing backwards $($obtaining the right index of the column studied$)$.

Then, the pricing can be divided in two parts:

- First, I obtain the payoff according to the "type" of the option.
- Then, I take the last column of the pricing and put it in a list
- At last, I do the pricing by going backwards through the tree. Every new price is placed in the pricing list.

At last, the function that enables to obtain the hedging strategy for a forward contract, $forward - prices$, has been mostly developed similarly as for pricing call and put options. Only the payoff is different. That time I put the hedging strategy as an output (H0 and H1) as well as the forward price $($at time t=0$)$. The variable S_0 is the price at time t=0 of a stock or any other underlying assets that could be the object of a forward contract. The hedging strategy is obtained from a binomial tree built with the parameters u and d precised as inputs.