University Of Waterloo

ACTSC 971

Finance 2

Executive Summary on A Study of Numerical Solution on Black-Scholes Model

Name:
Yifei Deng

14 April, 2020

1 Objective

The purpose of the technical report is to understand how different finite difference algorithms can be applied to price European type of option under the classical Black-Scholes equations. The final goal of the technical report is to determine which algorithm is optimal to implement when pricing an option in the financial market under different scenarios.

2 Summary

Generally, a total of four finite difference algorithms: explicit, semi-implicit, fully-implicit and Crank-Nicolson are introduced, derived and investigated in the technical report. The technical report also contains detailed validations on each algorithm by a series of numerical analyses. Furthermore, numerical results for comparison on accuracy and computational cost among all algorithms when pricing a European type of option is also presented in the report. As a result, both the numerical results and the validations in the technical report showed that these finite difference algorithms are capable of pricing the European option price in an accurate and sufficient way.

3 Introduction

In reality, there are many options like the exotic option in the financial market that can not be priced directly by using a formula. To effectively solve this problem, the finite difference method is introduced to find the price of this kind of option. The finite difference method is a form of numerical analysis used to iteratively solve the Black-Scholes equations for the fair price of a European-type option by finitely dividing the price and time of the equation with equal size and approximating the option price discretely.

4 Evaluation

In this section, we will summarize the performance of each algorithm on pricing a European-type option in the aspect of accuracy, computational cost, and limitations based on the numerical results summarized from the technical report. Before proceeding to the first part of this section, we need to denote some notations for illustrative purposes. For simplicity, M is referring to the number of price steps equally divided in stock price and N is referring to the number of time steps equally divided in time. Naturally, the stock price can grow unbounded. However, to save computational cost and for us to indicate our area of interest, we choose a constant, namely, the largest stock price with 3-4 times the value of the strike price. Therefore, the stock price from zero to the largest stock price will be the area of our interest for pricing the targeted option.

4.1 Accuracy

Firstly, we will discuss the accuracy among the algorithm on pricing a European-type option. By looking at Table 1., where the second column is the exact solution for the European put option and the remaining column is the absolute difference between the exact value of the put option and the numerical results from each algorithm. The higher the absolute difference the lower the accuracy. We can see from the Table 1.both explicit, fully-implicit and Crank-Nicolson algorithm perform well on pricing accurately the put option. Whereas, for the computational cost, the explicit algorithm is more efficient than the other two algorithms. As for the semi-implicit algorithm, it is both lower in accuracy and higher in computational cost. Moreover, comparing all the algorithms with the traditional approach for pricing the option with Monte-Carlo simulation, we could see the merits of finite difference method to have more accurate and stable numerical solutions for pricing a European-type option.

largest stock price = 150 , time to maturity = 3 years, Strike price = 50, risk-free interest rate = 0.05, volatility = 0.25, M = 500, N = 50000

Stock Price	Black Scholes (PUT) (Exact Solution)	Explicit (PUT) (M = 500, N = 5e4) 0.396048 seconds	Seme-Implicit (PUT) (M = 500, N = 5e4) 18.151862 seconds.	Fully Implicit (PUT) (M = 500, N = 5e4) 16.484444 seconds.	Monte_carlo (PUT) (path = 1e7) 3.417380 seconds.	Crank-Nicolson (PUT) (M = 500, N = 5e4) 19.718434 seconds.
10	33.0363	0.0000	0.0000	0.0000	0.0004	0.0000
15	28.0619	0.0000	0.0011	0.0001	0.0007	0.0000
20	23.2277	0.0000	0.0042	0.0001	0.0016	0.0001
25	18.7363	0.0000	0.0089	0.0001	0.0030	0.0001
30	14.7739	0.0000	0.0137	0.0000	0.0029	0.0000
35	11.4386	0.0000	0.0171	0.0000	0.0016	0.0000
40	8.7340	0.0000	0.0188	0.0000	0.0008	0.0000
45	6.6021	0.0000	0.0189	0.0001	0.0012	0.0000
50	4.9565	0.0000	0.0179	0.0001	0.0057	0.0000
55	3.7047	0.0000	0.0162	0.0000	0.0028	0.0000
60	2.7621	0.0000	0.0142	0.0000	0.0023	0.0000
65	2.0575	0.0000	0.0121	0.0000	0.0015	0.0000
70	1.5328	0.0000	0.0102	0.0000	0.0003	0.0000
75	1.1430	0.0000	0.0084	0.0000	0.0014	0.0000
80	0.8539	0.0001	0.0068	0.0001	0.0002	0.0001
85	0.6392	0.0002	0.0054	0.0002	0.0008	0.0002
90	0.4797	0.0003	0.0042	0.0003	0.0007	0.0003
		↑	↑	↑	↑	↑
		Absolute Difference	Absolute Difference	Absolute Difference	Absolute Difference	Absolute Difference

Table 1. Accuracy comparison among all algorithms for European put option, where Column 2 is the exact solution

4.2 Limitations

On the one hand, the limitation of the explicit algorithm is that we need to be cautious about the choice of the number of times, price steps and largest stock price. As an illustration of this limitation, according to Figure 1., as we choose the wrong number of time steps, the option price gets exploded in either a negative or positive direction. In the financial market, when pricing options, this is certainly not the phenomenon we desire. The solution for this problem is simple, as I have derived the constraint formula from the technical report, we can manually pick the finer number of steps so that the solution will not get exploded.

strike price = 60, largest stock price =100, risk-free interest rate = 0.05,
Time to maturity = 1, Volatility = 0.2, M = 100

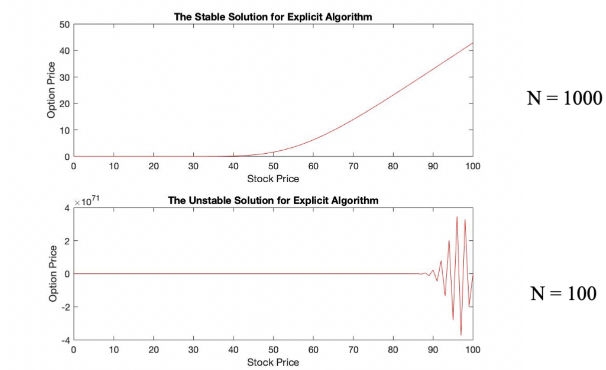

Figure 1. Illustration for the statement that choice of the size for time steps, price steps and largest stock price matters for Euro Call

On the other hand, the limitations for the remaining algorithms are quite similar, to obtain higher accuracy of the price of the option, we need to have a large number of time steps, and at the same time, by dividing more pieces for the stock price and time, one has to increase the execution time. In this case, I give a typical example in Figure 2. for the Crank-Nicolson algorithm. The percentage error in this the vertical axes of the figure is a measure of the difference in percentage between the exact option value with the numerical option value. And the higher the percentage error the lower the accuracy of the method. As we can see with a small number of time steps and, there will be a low accuracy measure of option value. As we increase the area of interest and number of time steps the problem will be resolved easily, but as a sacrifice, we will need more execution time.

Typical Example of Crank-Nicolson Algorithm
Time to maturity = 1, risk-free interest rate = 0.12, Strike price = 100, Volatility= 0.10

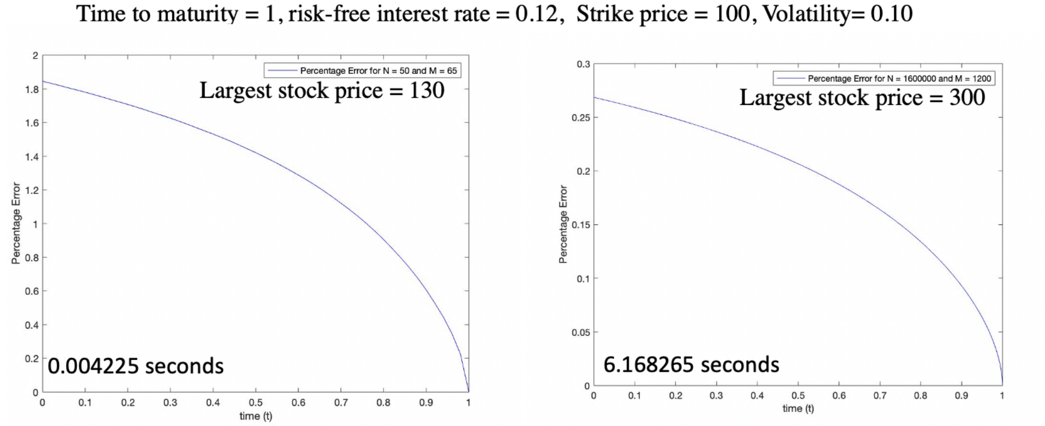

Figure 2. Illustration of the solution for the limitation for Crank-Nicolson scheme for European call option

4.3 Computational Cost

According to Figure 2., we observe that when the number of time steps and price steps increases, the execution time will increase accordingly. Among all algorithms, Crank-Nicolson has the highest computational cost while the explicit method has the lowest computational cost on pricing a European call option. Meanwhile, both fully implicit and Crank-Nicolson algorithms will have similarly higher computational cost to obtain an accurate option price.

largest stock price = 150 , time to maturity = 3 years, strike price = 50,
risk-free interest rate = 0.05, volatility = 0.25

N	M	Explicite Method	Semi-implicit Method	Fully Implicit Method	Crank-Nicolson Method
100	1000	0.016201132	0.058299668	0.027857408	0.035009212
200	2000	0.021118678	0.452126413	0.454776955	0.568210267
300	3000	0.064795756	2.049929684	1.98251907	2.149419295
400	4000	0.129020036	5.034479106	4.94288876	6.223492484
500	5000	0.225636164	11.79788101	11.43453437	13.96208422
600	6000	0.334697979	26.47826721	22.20137643	27.00717863
700	7000	0.461299751	36.41938679	33.40580349	40.53211332
800	8000	0.592552204	56.623195	50.2304602	60.47547852
900	9000	0.75606638	73.34828497	73.17481938	87.00846146
1000	10000	0.853447122	99.8562171	100.6259627	119.0020807

Table 2. Execution time (in seconds) among all algorithm for a European call option

Overall, the semi-implicit algorithm is both lower in accuracy and higher in computational cost. The fully-implicit algorithm shows a similar computational cost with lower accuracy for pricing the option compares to the Crank-Nicolson algorithm. At the same time, the explicit finite difference algorithm has lower computational cost and a fairly accurate approximation to the option price. Thus, explicit and Crank-Nicolson finite difference algorithms are the favorable two among the four algorithms on pricing an option.

5 Recommendations & Conclusion

Based on the results from the previous section, we can make recommendations for different scenarios. When pricing an option in the financial market, the explicit finite difference algorithm is recommended if one considers a lower computational cost and good accuracy on pricing the option. Alternatively, one is recommended to implement the Crank-Nicolson finite difference algorithm if he or she prefers the high accuracy of the option price and does not care about the long waiting time. As a conclusion, we have summarized the performance of each algorithm on pricing a European-type option in the aspect of accuracy, computational cost, and limitations. In the end, results showed that these finite difference algorithms are capable of pricing the European option price in accurately and stably. Among all the algorithms, explicit and Crank-Nicolson finite difference algorithms are more reliable on pricing an option in the financial market.