Qijie Li   526 Homework ID: A20355279

1. INTRODUCTION

This homework is to write two root finding methods and implement these methods in an example of Black-Scholes option pricing formula. The two root finding methods are

- Bisection Method
- Newton's Method

2. INTRODUCTION

2.1 Required functions. The assignment specications require the following four functions:

- **bsformula(callput, S0, K, r, T, sigma, q=0.).** This function return the option value, theta of option and vega of option.
- **bisect(target, f, start = None, bounds = None, tols = [0.001,0.010], maxtier =1000).** This function returns the root of function when the value of function is almost close to target value.
- **newton_Method(target, y, dy, x, tols=[0.001,0.01], maxtier=1000).** This function returns the root of function when the value of function is almost close to target value.

2.2. Implement functions. In problem 3, I need to implement two root find methods to find the volatility of target option. So I wrote a function **bsimpvol(callput, S0, K, r, T, price, q=0.0, priceTollerance=0.01, method='bisect', reportCalls=False)**, which will return volatility and callsTimes.

- **volatility** is the root that the option price from **bsformula** function is close to target option price.
- **callsTimes** is the number of times that **bsimpvol** function calls **bsformual** function.

3. TEST SUITE

To test whether these function can be called successfully and work correctly. I wrote the some test suite as follow:

- For **bsformula** function, I wrote **test_BS** program to test the whether this function can run successfully and correctly.
- For **bisect**, **newton_Method**, **bsimpvol** functions, I wrote some test codes start with **if __name ==’__main__’ :** below these functions.

After running this test suite, I got correct answers, which indicate these functions are correctly coded.

4. VALUE INPUTS

To implement **bsimpvol** functions to find the volatility of options. I try three types of European options to test the **bsimpvol** functions. They are:

(1) ATM option:

- For both call options and put options, underlying asset’s price is equal to strike price
    - Underlying asset price is 50, strike price is 50, risk-free rate is 0.1, mature time is 2, target price of option is 10, dividend yield is 0 and price tolerance is 0.01

(2) ITM option:

- For call options, underlying asset’s price is higher than strike price
    - Underlying asset price is 50, strike price is 40, risk-free rate is 0.1, mature time is 2, target price of option is 15 , dividend yield is 0 and price tolerance is 0.01
- For put options, underlying asset’s price is lower than to strike price
    - Underlying asset price is 50, strike price is 50, risk-free rate is 0.1, mature time is 2 target price of option is 6, dividend yield is 0 and price tolerance is 0.01

(3) OTM option:

- For call options, underlying asset’s price is lower than strike price

  - Underlying asset price is 50, strike price is 55, risk-free rate is 0.1, mature time is 2, target price of option is 6 , dividend yield is 0 and price tolerance is 0.01

- For put options, underlying asset's price is higher than to strike price
  - Underlying asset price is 50, strike price is 45, risk-free rate is 0.1, mature time is 2, target price of option is 6 , dividend yield is 0 and price tolerance is 0.01

5. PRPGRAM RESULT

Below are the result from my codes

ATM

Method	Root	The number of times calls **bsformula** function
newton method	0.15299735915449009	3
bisect method	0.15296740722656252	14
bisect method	0.15679196345677271	3
bisect method	0.15662927246093752	14

ITM

Method	Root	The number of times calls **bsformula** function
newton method	0.26379362766236569	2
bisect method	0.26373883056640623	15
bisect method	0.31048178924799064	2
bisect method	0.31042761230468752	14

OTM

Method	Root	The number of times calls **bsformula** function
newton method	0.10660806930917027	4

bisect method	0.10627862548828125	15
bisect method	0.47746926887688207	2
bisect method	0.47704248046874997	12

6. COMPARSON OF RESULT

Clearly, the number of times calls **bsformula** function when we use Newton’s Method is lesser when we use bisection method.

7. ANSWERS TO QUESTION POSED

This homework assignment posed one question: For which kinds of European options (i,e. ATM, ITM, OTM) is the performance more noticeable?

The answer is that, from my test result, Newton’s Method is more efficiently to find the root of function for three types of options.

8. FURTHER INVESTIGATION

There is one problem of Newton’s Method to find the root for **bsimpvol** function. In Newton’s Method, **x1= x-f /df** (**f** is function, **df** is the first order patical derivative of the function **f),** and volatility (in this assignment, x is volatility) often is a small number in financial markets, and **1/df** is the monotone decreasing function with the increase of volatility. When the start point of volatility is assign a kind of large number with a limited times of iteration, Newton’s method might not find a root after all iterations for the small value of **1/df**. For example, when the start point of volatility is1.0, underlying asset price is 45, strike price is 55, risk-free rate is 0.1, mature time is 2, dividend yield is 0, and price tolerance is 0.01. there is not root after all iterations if we use Newton’s method. But this problem will not happen if we give a proper start point when we use Newton’s Method.

9. CONCLUSION

Generally, if the start point is given properly the Newton’s Method can more quickly find function’s root. But if the start point is not available, bisection is a better safer method to find root of function.