

# Secant and Newton method about Finding Black-Scholes Implied Volatility

Jingjie Zhou

September 5, 2014

## 1 INTRODUCTION

In numerical analysis, the secant method is a root-finding algorithm that uses a succession of roots of secant lines to better approximate a root of a function  $f$ . The secant method is a finite difference approximation of Newton's method. Black-Scholes model uses a variety of inputs to derive a theoretical value for an option. the implied volatility of an option contract is that value of the volatility of the underlying instrument which, when input in Black-Scholes will return a theoretical value equal to the current market price of the option. The value of an option depends on its implied volatility.

## 2 FUNCTIONS USED

### 2.1 REQUIRED FUNCTIONS.

The assignment specifications require the following 6 functions:

- `bs( callput, S0, K, r, T, sigma, q=0.)` for using callput type, underlying option price, strike, risk-free rate, time to maturity, implied volatility and Annual Dividend Yield on black-scholes model. This function returns three variables: option price, delta and vega.
- `secantsolve(target, targetfunction, start=None, bounds=None, tols=[0.01,0.01], maxiter=100)` for using secant method to get root(implied volatility). This function returns an array: bounds, start point and root.
- `newton(target, targetfunction, start=None, bounds=None, tols=[0.01,0.01], maxiter=100)` for using Newton method to get root(implied volatility). This function returns an array: bounds, start point and root.
- `newton(target, targetfunction, start=None, bounds=None, tols=[0.01,0.01], maxiter=100)` for using Newton method to get root(implied volatility). This function returns an array: bounds, start point and root.
- `bsimpvolsec( callput, S0, K, r, T, price, q=0., priceTolerance=0.01, reportCalls=False )` for using secant method to find black-scholes implied volatility. This function returns an array: bounds, start and implied volatilities.

- `bsimpvolnewton( callput, S0, K, r, T, price, q=0.,priceTolerance=0.01, reportCalls=False )` for using newton method to find black-scholes implied volatility. This function returns an array: bounds, start and implied volatilities.

### 3 HELPER FUNCTIONS

`f(sigma)` is a new function which only the volatility is an unknown variable, the other variable become known. This function return the option price. We use this new function to find it's root, which is the implied volatility.

### 4 TEST SUITE

In Question 1, I defined two test using matlab black-scholes caculator's result to my BS formula program. It all went through well.

In Question 2, I used `test_secant.py` to test my Secant method, all seven test pass, which means my `secantsolve(target, target_function, start = None, bounds = None, tols = [0.01, 0.01], maxiter = 100)` is correct.

In the Question3, see the figure 1 on the top of this page. I captured the result of line 50-80 to compare the volatilities. Comparing the implied volatilities that I computed to the CSVfile's volatilities, they are not close, Which I can not figure out why.

### 5 RANGE OF INPUTS

I set up bounds as  $[-0.5, 2]$ , start at 0.5 and make the tolerance as  $[0.0, priceTolerance]$  in both secant and newton method to find the implied volatility.

### 6 ANALYSIS

Compare these two method, secant and newton, secant always make more calls to the function `bsformula`. That means secant method takes more time. It is reasonable. Using secant method, we need two points in each step while using newton method, we only need one point. The implied volatilities are really closed using these two method. But there still remain a lot of questions that I do not understand.

### 7 FUTHER INVESTIGATION

These two method both have advantages and disadvantages. Secant method need more known variables but it is much easier to compute the function. Newton method, on the other hand, is much faster and require less known variables, but it needs derivative. We need to use computer tools to help us get function

derivatives. The secant method can be thought of as a finite difference approximation of Newton's method. The recurrence formula of the secant method can be derived from the formula for Newton's method.

## 8 QUESTION

1. When I first finished my BS.py. I can go through my test. But when I finished the last question, add and edit two definition of `secantsolve()` and `newton()`, BS.py can not went through my test, I can not figure out why.

2. The implied Volatility is not close to the given volatility on CSVfile. I do not think my function is wrong. So What's the reason for that?

## 9 CONCLUSION

Both secant and newton method are good ways to find the black-scholes implied volatilities. If we have our computer by hand, we can use both these two method, otherwise, we'd better use secant method since it is much more easier to calculate.