

# An Analysis of Option Pricing Models

Charles Vu

Computer Systems Lab

June 12, 2003

## **Abstract**

The stock options market is a trillion dollar industry and many models exist which attempt to price stock derivatives. Previously, most attempts at analyzing the merits of each option model have used theoretical economic analysis. However, no attempt has been made to analyze how the models perform in the actual options market. The goal of this project is to perform an empirical analysis of the option pricing models.

# Contents

<b>1</b>	<b>What is a stock option?</b>	<b>2</b>
<b>2</b>	<b>Previous Research on Pricing Models</b>	<b>2</b>
<b>3</b>	<b>Black-Scholes Model</b>	<b>3</b>
3.1	No dividends or cash flows are generated by the underlying asset . . . . .	3
3.2	Only one interest rate is input, implying a flat yield curve . . . . .	3
3.3	Black-Scholes formula . . . . .	4
<b>4</b>	<b>Binomial Tree Model</b>	<b>5</b>
4.1	Derivation . . . . .	5
<b>5</b>	<b>Process for Testing Models</b>	<b>7</b>
<b>6</b>	<b>Limitations of my Project</b>	<b>8</b>
<b>7</b>	<b>Code</b>	<b>10</b>
7.1	website.phtml . . . . .	10
7.2	php.phtml . . . . .	13
7.3	stocks.phtml . . . . .	16
7.4	quotes.phtml . . . . .	19
7.5	opm.cpp . . . . .	22

# 1 What is a stock option?

An option is a contract that gives a buyer the right, but not the obligation, to purchase shares of a stock at a predetermined price on or before a certain date. An option is a security, just like a stock or a bond, and these contracts may be traded just like other securities. For example, if Microsoft (MSFT) is at \$50 on October 1st, the price of the option to purchase a share of MSFT for \$50 at anytime before October 15th might cost me \$2. If MSFT is still at \$50 on October 15, I will lose my \$2. However, if MSFT is at \$60 on October 15th, I can still purchase the shares at \$50, and therefore make a  $60 - 50 = \$10$  profit, a good profit since I only put \$2 at risk. Further, if MSFT goes down to \$40, or lower, I still only lose the \$2 I spent for the option. If I had simply bought shares of MSFT, I would have been down \$10, or more. If used correctly, stock options can provide an effective alternative to traditional stocks.

## 2 Previous Research on Pricing Models

But the question remains, is the option I bought worth \$2, or is it worth more than \$2, or less? Many models have been proposed to answer this question. Heretofore, these models have been evaluated using economic analysis, using traditional economic theory. However, there has not been an analysis of how these models perform under actual market conditions. Considering that the purpose of option pricing models is to assist firms in determining a suitable price to purchase and sell options, it would be wise to analyze how option pricing models perform in the same markets from which they will be utilized. The goal of this project

is to perform this numerical analysis, to determine which option pricing model performs best in the actual options market.

### **3 Black-Scholes Model**

By far, the most popular option pricing model is the Black-Scholes model. Developed in 1973 by Fischer Black and Myron Scholes, this was the first legitimate model to price stock options. Nearly all modern models are rooted in the Black-Scholes model, with some variations. Black and Scholes make several assumptions regarding the option, namely,

#### **3.1 No dividends or cash flows are generated by the underlying asset**

This means that this model is only accurate if the underlying stock does not pay any dividends. Many stocks do pay dividends, and therefore, the model should be modified for such stocks, due to the fact that it would be preferable to own the underlying stock when dividends are paid, as opposed to merely owning the right to purchase options and not receiving the dividend payment. In the interest of this project, only stocks which do not pay dividends will be used for testing.

#### **3.2 Only one interest rate is input, implying a flat yield curve**

The underlying assumption behind the concept of stock options is that one does not have the ability to create a situation where one can make money regardless of whether the underlying

asset moves up or down. Having the ability to purchase a combination of options such that one would make money regardless of the movement of the stock price is called arbitrage. Most financial analysts believe stock options should be priced on the *no-arbitrage principle*. The easiest way to arbitrage is to borrow money against an interest rate. One could either earn interest with a Treasury bond, or one could spend it on purchasing an option. The Black-Scholes Model is based only on the interest rates of Treasury bonds, and we will treat options as such.

### 3.3 Black-Scholes formula

The Black-Scholes model is as follows:

$$V_{\text{call}} = SN(d_1) - E^{-r(T-t)}N(d_2) \quad (1)$$

where

$$d_{1;2} = \frac{\ln(S/E) + (r - D \pm \frac{\sigma^2}{2})(T - t)}{\sigma\sqrt{T - t}} \quad (2)$$

The derivation of Black-Scholes is rather complex. I will not be studying the theoretical validity of this model; rather, I will look at whether it accurately prices options in the real market. If one is interested in the derivation of the Black-Scholes formula, see <http://www.lifelong-learners.com/opt/SYL/s4node10.php3>

## 4 Binomial Tree Model

The other model that was tested was the Cox, Ross, & Rubenstein binomial tree model. Consider a MSFT 50 call when the stock is at trading at \$50 10 days before expiration. For each of the ten remaining days prior to expiration, we assume that the stock can move in two directions: up 2% or down 2%. We then repeat this for each day until we reach the expiration date in 10 days, and the tree will consist of 1024 leaves, all possible outcomes of the stock price movement in the 10 days prior to expiration. At this point, calculate the option value at expiration for each of these combinations. One can then work backwards, calculating the value of each parent node by using the following formula:

$$f = E^{-rT} [pf_u + (1 - p)f_d] \quad (3)$$

where

$$p = \frac{E^{-rT} - d}{u - d} \quad (4)$$

### 4.1 Derivation

The derivation of this formula can be found in Hull's Options, Futures, and Other Derivatives (5th edition, pp. 202-203).

However, let us consider a numerical example so that we can gain an intuitive under-

standing of the binomial tree. Consider a stock XYZ which is trading at \$50 and a call option with strike price \$55. In 3 months, the stock price can be one of two values, \$60 or \$40. Our goal is to create a riskless portfolio, that is, one which has the same value regardless of whether the stock is at \$60 or \$40 in 3 months. We will do this by selling one call option and purchasing  $\Delta$  shares. So, fastforwarding 3 months, we calculate the value of the portfolio under the two possible conditions and making them equal to each other, thereby making it riskless.

$$60\Delta - 5 = 40\Delta \tag{5}$$

$$20\Delta = 5 \tag{6}$$

$$\Delta = 1/4 \tag{7}$$

Thus, our riskless portfolio contains .25 shares and one short call option. At expiration, this portfolio will equal

$$60 \times 0.25 - 5 = 10 \tag{8}$$

Since this portfolio is riskless, its return should therefore be equal to the risk-free interest rate, which we will assume for the purposes of our example to be 12%. Therefore, the original investment should be worth



$$10e^{-rT} = 10e^{-0.12 \cdot 3=12} = \$9.70 \quad (9)$$

Thus, our original investment should be \$9.70, which will grow to \$10 regardless of the movement in the stock price. Calculating the cost of our original investment and setting it equal to \$9.70 will give us the option price,  $f$ :

$$50 \times 0.25 - f = \$9.70 \quad (10)$$

$$12.50 - f = \$9.70 \quad (11)$$

$$f = \$2.80 \quad (12)$$

The value of the option is the value of the root node in this binomial tree.

## 5 Process for Testing Models

The first step towards evaluating the option pricing models is to obtain the necessary historical stock and option data from an internet finance website. Once the data is retrieved, it can be analyzed. This "data mining" process was done using a PHP script. The PHP script would go and retrieve the source code of a given stock, strip all of the HTML tags, and then search and copy the portions of the webpage that contained the

desired data. At first, it can be used as an alternate place to get stock quotes. See <http://www.tjhsst.edu/cvu/techlab/stocks.phtml>

The next step was to program the option pricing models. This was fairly simple for Black-Scholes, as it is simply an explicit formula, albeit long and complex. The binomial tree model was also simple to code, as it is a classic example of a recursive tree. The source code for these two models are in the appendices.

## 6 Limitations of my Project

There are several variables inherent in the study of this topic which makes it very difficult to study option pricing models effectively. First, while in theory, we can assume perfectly random market movement, in theory, this is not the case. We can not assume that markets move randomly day-by-day; actually, markets tend to move in trends. Markets may move up five days in a row, followed by moving down 7 days in a row. While in the long-run, markets will have roughly the same number of up days and down days, in the short-term, markets like to move in trends. In other words, tomorrow's stock price is at least somewhat dependent to today's stock movement, a fact that all the option models discount.

Hull[1] noted that many problems exist when trying to empirically test Black-Scholes and other option pricing models. One has to account for the fact that while all option pricing models assume that markets are perfectly efficient; in fact, they often do not. Also, stock price volatility, a key component in nearly all option pricing models, is very difficult to measure. Most formulas just assume the variance in the stock price to be the implied

volatility; in truth, that is not always the case. Also, it is difficult to ensure that stock and option prices are synchronous; that is, that the last option trade corresponds with the last stock trade. For example, the last option trade of the day for MSFT 50 calls may occur at 1:00 PM, when MSFT is trading at \$50, while the last stock trade may occur at 4:00 PM, when the stock is higher or lower, which would effect how the option pricing models would theoretically price the option.

Black and Scholes (1972) originally tested whether their model would work in practice. They would purchase undervalued options and sell overpriced options. In the long-term, they indeed did make money, but if one took into account transaction costs, they concluded that only market-makers would have the ability to profit from this; the market seems to be efficient enough to avoid such arbitrage opportunities.

## References

- [1] Chicago Board of Options Exchange Learning Center

<http://www.cboe.com/LearnCenter/cboeeducation/CourseList.html>

- [2] A Black-Scholes formula for vanilla and binary options

<http://www.lifelong-learners.com/opt/SYL/s4node10.php3>

- [3] Option Pricing Models

<http://www.hoadley.net/options/BS.html>

[4] Information on Option Calculation Models

<http://www.jereseach.com/Opti-Calc/Models.htm>

[5] Hull, John. Introduction to Futures and Options. Upper Saddle River, N.J. : Prentice Hall, 1998.

[6] Hull, John. Options, Futures, and other Derivatives. Upper Saddle River, N.J. : Prentice Hall, 2003.

[7] Cox, J., S. Ross, and M. Rubenstein, "Option Pricing: A Simplified Approach," *Journal of Financial Economics*, 7 (October 1979), 229-64.

[8] Black, F., and M. Scholes, "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, 81 (May/June 1973), 637-659

## 7 Code

### 7.1 website.phtml

```

<HTML>

<HEAD>


<body bgcolor = "#5cb1c6">

<CENTER>

<font size = "6" color="#000000">

<b> Website PHP Program</b><br>


<font size = "4">


<I> By Charles Vu </I> <br><br>

</CENTER>


<font size = "3">

Enter the value(s) to iterate the selected equation. <BR>

<form action = "php.phtml" METHOD = POST>

Website:<input type = "text" NAME = "myfiles"><BR>

<BR>

<input type = "submit" VALUE = "Show Results">

<BR><BR>

</form>

```

</BODY>

</HTML>

## 7.2 php.phtml

```

<html>

    <head>

        <title>Example</title>

    </head>

    <body>


<?php

$myfiles = $myfiles.rawurlencode($stadt);

$datei = fsockopen("bigbrother.tjhsst.edu", 8002, &$errno, &$errstr);

if( !$datei )

{

echo "proxy not available !";

fclose($resultfile);

exit();

} else {

fputs($datei,"GET $myfiles/ HTTP/1.0\n\n");

$stock;

while (!feof($datei))

{

```



```
$name = fgets($datei, 500000);

// $num = strpos("MSFT", $name);

// $num2 = strpos("Chart", $name);

// $stock =

print("$name");

}

}

//print("$stock");

//fscanf($fp, "%s", $name);

?>

</body>

</html>
```

### 7.3 stocks.phtml

```
<HTML>

<HEAD>


<body bgcolor = "#5cb1c6">

<CENTER>

<font size = "6" color="#000000">

<b> Stock Quotes Program</b><br>


<font size = "4">


<I> By Charles Vu </I> <br><br>

</CENTER>


<font size = "3">

Enter a stock symbol (i.e. MSFT) <BR>

<form action = "quotes.phtml" METHOD = POST>

Website:<input type = "text" NAME = "myfiles"><BR>

<BR>

<input type = "submit" VALUE = "Show Results">

<BR><BR>

</form>
```

</BODY>

</HTML>

## 7.4 quotes.phtml

```

<html>

    <head>

        <title>Example</title>

    </head>

    <body>


<?php

$myfiles = "http://finance.yahoo.com/q?s=msft&d=v1";

$myfiles = $myfiles.rawurlencode($stadt);


$datei = fsockopen("bigbrother.tjhsst.edu", 8002, &$errno, &$errstr);

if( !$datei )

{

echo "proxy not available !";

fclose($resultfile);

exit();

} else {

fputs($datei,"GET $myfiles/ HTTP/1.0\n\n");

$stock;

$name2 = "";

while (!feof($datei))

```

```

{
    $name = fgets($datei, 1);

    if ($name == "<")
    {
        while($name = fgets($datei, 1) != ">" && !feof($datei))
        {}

        print(" ");

    }

    print("$name");
}

}

print("$name2");

//print("$stock");

//fscanf($fp, "%s", $name);

?>

</body>

</html>

```

## 7.5 opm.cpp

```
#include <iostream.h> #include <math.h> #include <stdlib.h> #include <iomanip.h> #include
<string.h>

#define pi 3.141592653

class option public: /**FUNCTIONS**//

void DisplayInfo(); void defineOption(ifstream myFile); void BlackScholes(); float nd(float
d1); void sdeviation(ifstream myFile);

/**IMPORTANT VARIABLES**//

char name[5]; //ticker symbol float stock; //stock price float strike; //strike price float
expiration; //years until expiration float rate; //interest rate float volat; //volatility

/**AUXILLARY VARIABLES**// float mean; //mean of stock data float sd; //sd of
stock data float BSprice; //black-scholes option price ;

void option::DisplayInfo() cout<<"Stock Name: "<<name<<endl; cout<<"Stock Price: "<<stock<<endl;
cout<<"Expiration Date: "<<expiration<<endl; cout<<"Interest Rate: "<<rate<<endl; cout<<"Volatility:
"<<volat<<endl; cout<<"Black-Scholes Price: "<<BSprice<<endl;

void option::defineOption(ifstream myFile) myFile << name << stock << strike << expira-
tion << rate << volat;

void option::BlackScholes() float d1 = (log(stock / strike) + (rate + pow(volat,2) / 2)
* expiration)/(volat*sqrt(expiration)); float d2 = d1 - volat * sqrt(expiration); //cout<<d1<<
"<<d2<<endl; BSprice = stock * nd(d1) - strike * exp( -1.0 * rate * expiration) * nd(d2);

float option::nd(float d1) //the standard normal cumulative distribution function float
```



```

score; //loop variable (z-score) float CDF = 0.0; //value of the standard normal cumulative
distribution function for(float score=-3.0; score<1; score+=.00001) //integral, -3 to z-score
of normal curve CDF += .00001* 1/sqrt(2*pi*1) * exp(-1*(score)*(score)/(2*1*1)); //sd =
1 return CDF + .0013499672; //integral from -infinity to -3 = .015

```

```

void option::sdeviation(ifstream myFile) //standard deviation of the 30-day stock data
int x; //for loop variable float sum_of_array = 0.0; //sum of 30-day stock data float sum_of_residuals =
0.0; //sum of residuals for sd calculation float price_array[30]; //array of stock data mean = 0.0; //mean, for
0; x < 30; x++) myFile >> price_array[x]; sum_of_array += price_array[x]; mean = sum_of_array/30.0; for(x
0; x < 30; x++) float difference = mean - price_array[x]; if(difference < 0) //absolute value difference
sqrt(sum_of_residuals/30.0); volat = sd/stock; //volatility = standard deviation/stock price

```

```

int main() option a; cout << "Input values: "; cin << a.stock << a.strike << a.expiration <<
a.rate << a.volat; a.BlackScholes(); //calculates the Black-Scholes option value cout << "Option
Price: " << a.BSprice << endl; return 0;

```