2012

# Developing Real-Time Options Pricing

John Stamey
*Coastal Carolina University*, jwstamey@coastal.edu

Kenneth Small
*Coastal Carolina University*, ksmall@coastal.edu

Andrew Whalley
*Coastal Carolina University*, adwhalle@coastal.edu

# DEVELOPING REAL-TIME OPTIONS PRICING

**John Stamey**
Coastal Carolina University
jwstamey@coastal.edu

**Kenneth Small**
Coastal Carolina University
ksmall@coastal.edu

**Andrew Whalley**
Coastal Carolina University
adwhalle@coastal.edu

**Abstract**

This paper describes a real-time web-based implementation of the Black-Scholes Options Pricing Model. The initial implementation was done in PHP on a Linux server. A mobile implementation was done using the E*Trade PHP SDK. The implementation is interesting as stock prices, historical quotations, and risk-free interest rates were all available using free web services. Critical Success Factors for the implementation are also presented.

***Keywords***

Financial technology, web application development, PHP, options, Black-Scholes, mobile applications

## INTRODUCTION

Hedge funds and risk management have become more and more important in today's financial markets. Skills in the design and implementation of mathematical models for the pricing of derivatives, assessment of risk, or predicting market movements are becoming increasingly desirable. The Black-Scholes options pricing model (Black and Scholes, 1973) has long been a standard mathematical model used in financial engineering. One problem with implementing this model is the gathering of data necessary for calculation of put option and call option pricing. In this web-based and mobile implementation, data for stock prices, historical quotations, and risk-free interest rates are all found to be available using free web services. In this paper, we describe how the data is gathered and computed for both a traditional web-based system and a mobile application.

## IMPLEMENTATION CONSIDERATIONS

The traditional Black-Sholes options pricing model requires the following inputs:

- Stock price
- Option strike price
- Options price ( with which to compare the results of the Black-Scholes model)
- Volatility measure (standard deviation) – historical volatility obtained from price movements over the last 25 days of trading
- Time to expiration – number of days remaining on the stock option
- Risk-free interest rate – frequently obtained from the one-year T-bill rate

Additionally, the normal cumulative distribution function is needed in the Black-Scholes computation.

### Stock Quotation Information

Sources for stock quotation information were identified for the data needed in this online implementation. These included:

- free data from websites such as finance.yahoo.com, which can be delayed from 15-20 minutes;
- data usually available without charge from a personal brokerage account, which has access to stock and options prices in real-time; and,
- data at a cost from paid providers such as Bloomberg.com.

For the initial version of the system, using stock quotes were obtained from `finance.yahoo.com`. In PHP, the following code will take form input of a stock symbol and use finance.yahoo.com to return a CSV (comma-separated value file) of

```
$symbol = urlencode( trim( substr(strip_tags($symbol),0,7) ) );
$yahooCSV = "http://finance.yahoo.com/d/quotes.csv?s=
```

```
        $symbol&f=sl1d1t1c1ohgvpnbaejkr&o=t";
```

It was straightforward to parse string returned (for ticker symbol SLV)

```
SLV, 32.85, 5/16/2011, 4:00pm, -1.541, 33.82, 34.46, 32.76, 87513664,
34.391, iShares Silver Tr, 32.73, 32.74, 0, 16.94, 48.35, N/A
```

into an array in PHP:

```
list($quote['symbol'], $quote['last'], $quote['date'],
       $quote['timestamp'], $quote['change'], $quote['open'],
       $quote['high'], $quote['low'], $quote['volume'], $quote['previousClose'],
       $quote['name'], $quote['bid'],
       $quote['ask'], $quote['eps'], $quote['YearLow'], $quote['YearHigh'],
       $quote['PE']) = fgetcsv($csv, ',');
```

#### Volatility

Historical volatility calculations require data which is also available from `finance.yahoo.com`**.** The URL:

```
http://ichart.finance.yahoo.com/table.csv?s=SLV
```

will conveniently return the following CSV file (below we display only the first five rows):

```
Date, Open, High, Low, Close, Volume, Adj Close
5/13/2011, 34.43, 34.94, 33.11, 34.39, 117598800.00, 34.39
5/12/2011, 32.98, 34.69, 31.97, 33.32, 183600000.00, 33.32
5/11/2011, 36.48, 36.59, 34.18, 34.39, 153114200.00, 34.39
5/10/2011, 37.24, 37.9, 37.05, 37.52, 92253500.00, 37.52
5/9/2011, 36.17, 37.04, 35.8, 36.98, 109221900.00, 36.98
```

#### Risk-Free Interest Rate

The risk-free interest rate is available from `treasury.gov`. We are fortunate that an RSS feed is available to pull the previous close for use in this program.

```
$xmlFile
= 'http://data.treasury.gov/feed.svc/DailyTreasuryYieldCurveRateData?'.
'$filter=month(NEW_DATE)%20eq%20'.$month.'%20and%20year(NEW_DATE)%20eq%20'.
$year.'%20and%20day(NEW_DATE)%20ge%20'.$day;
```

Where `NEW_DATE` allows the use of simple date parameters `{month, day, year}` to pull the date of choice. Returned will be T-bill rated for `{one month, 3 months, 6 months, 1year,...,30 years}`.

#### Time To Expiration

The time to expiration is the number of days until the expiration of the options being modeled. This number can be easily computed by functions available in the implementation language (PHP), or can be manually entered for special situations.

#### Computing the Normal Cumulative Distribution Function CDF

The Normal Cumulative Distribution Function **(**CDF) was implemented a Maclaurin series when values converged to within .000001, matching tables in well-known statistics textbooks. The code used was:

```
function normalCDF($z) {
        $r = 100;
        $delta = .000001;
        $c = 1/sqrt(2*3.14);
        for($n=0 ; $n<=100 ; $n+=1) {
            $normal = $normal + (pow(-1,$n) *
                    pow($z,(2*$n+1))/(pow(2,$n)*(fact($n))*(2*$n+1)));
```

```
            if(abs($r*$c-$normal*$c) < $delta)
                return $normal*$c+.5;
            $r = $normal;
        }
        return $normal*$c+.5;
    }
    function fact($x){
        if ($x <= 1)
            return 1;
        else
            return ($x * fact($x-1));
    }
```

## FINAL WEB IMPLEMENTATION

Users input information into a web form as seen in Figure 1. The resulting computations may be seen in Figure 2. The URL for this system is http://www.softwareengineeringonline.com/Black-ScholesV1.

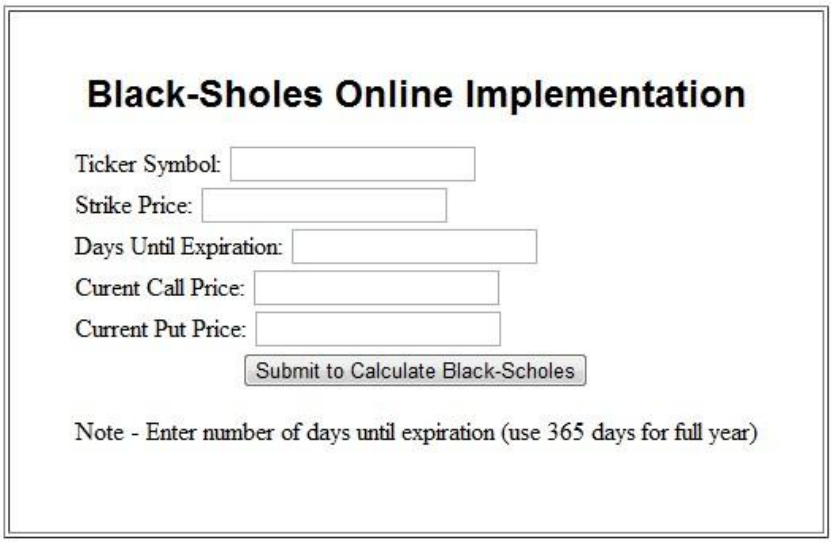



**Figure 1. Initial form for data input**

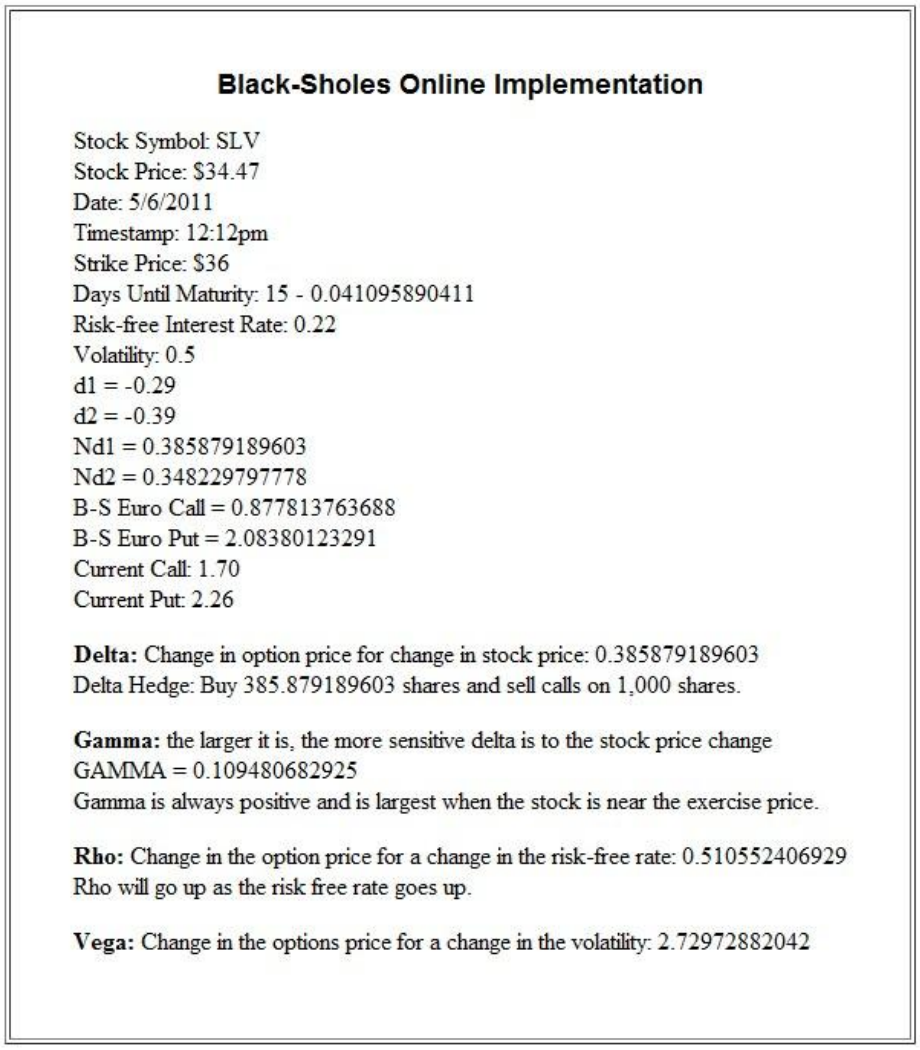
## MOBILE IMPLEMENTATION

Recently several brokerage houses have created APIs for use by individuals and companies to facilitate the development of automated trading systems. Brokerage firms who have released APIs for individuals to create automated trading systems include:

- E*Trade, www.etrade.com (Java, C#, PHP)
- Interactive Brokers, www.interactivebrokers.com (Java, C++, ActiveX, DDE for Excel)
- Pinnacle Capital Markets, www.pcmtrading.com (Java, Win32, FIX, C++, Visual Basic, VB.Net, C#, Visual C++)
- MB Trading, mbtrading.com (VB.Net, Excel VBA)

These APIs can be used to develop a wide range of applications that include:

- Fully automated trading applications: APIs used to monitor price and volume information, and to place orders
- Algorithmic trading applications: APIs used to collect price and volume day on a regular basis to populate a database to determine the best algorithm to space out orders
- Account-related applications to monitor portfolios: Polling or mobile data input is used in conjunction with data from APIs to monitor price and volume, compute statistics and trading metrics.



**Black-Sholes Online Implementation**

Stock Symbol: SLV
Stock Price: $34.47
Date: 5/6/2011
Timestamp: 12:12pm
Strike Price: $36
Days Until Maturity: 15 - 0.041095890411
Risk-free Interest Rate: 0.22
Volatility: 0.5
d1 = -0.29
d2 = -0.39
Nd1 = 0.385879189603
Nd2 = 0.348229797778
B-S Euro Call = 0.877813763688
B-S Euro Put = 2.08380123291
Current Call: 1.70
Current Put: 2.26

**Delta:** Change in option price for change in stock price: 0.385879189603
Delta Hedge: Buy 385.879189603 shares and sell calls on 1,000 shares.

**Gamma:** the larger it is, the more sensitive delta is to the stock price change
GAMMA = 0.109480682925
Gamma is always positive and is largest when the stock is near the exercise price.

**Rho:** Change in the option price for a change in the risk-free rate: 0.510552406929
Rho will go up as the risk free rate goes up.

**Vega:** Change in the options price for a change in the volatility: 2.72972882042

**Figure 2. Resulting Black-Scholes Computation**

For the mobile implementation of the Black-Scholes computation, the E*Trade Financial PHP Software Development Kit was used to access the E*Trade API. Also, we used the jQuery Mobile Framework to ensure standards compliance among all top mobile phone brands such as Apple, Blackberry and Android. The application relies heavily on HTML 5, JavaScript and PHP for performance and functionality.

Three basic categories of functions were found to be available in the API. They correspond to the basic functionality of accessing account data, viewing quotes and placing orders:

- Accounts library
  - List of user accounts
  - Account balances
  - Account positions
  - Get alerts (that have been set up by the user)
- -Market library
  - Get options chain
  - Product lookup (details about a company)
  - Get options expiration dates
  - Get quotes

- -Orders library
    - Get orders list
    - Preview equity orders
    - Place equity orders
    - Preview options orders
    - Place options orders
    - Change equity orders
    - Change options orders
    - Cancel orders

The mobile implementation involved setting up a configuration file, setting up authentication, and finally accessing the API for stock quotes. A configuration file, `config.php`, was placed in the `Common/` folder to provide basic configuration for the software development kit (SDK). Specifically, this file provides URLs for accessing the API functions, and allows selection of server response to be either XML of JSON (JavaScript Object Notation) is a lightweight data-interchange format.

Authentication is provided by OAuth (www.oauth.net) using the following five-step procedure:

- Instantiate an object $A of the class etOAuthConsumer(a,b) where a is your key and b is an additional secret code
- Instantiate an object $B of the class etOAuth using object A as a parameter
- Using the method GetRequestToken() of the etOAuth class, obtain the authorization token
- Using the method GetAuthorizeURL() of the etOAuth class, obtain the authorize URL from which access will be requested
- Obtain the access token with the GetAccessToken() method of etOAuth

The code to implement this is quite straightforward in PHP

```
$A = new etOAuthConsumer(a,b);
$B = new etOAuth($A);
$token = $B->GetRequestToken();
$auth_url = $B->GetAuthorizeURL();
$access_token = $B->GetAccessToken('VERIFIER CODE');
```

The getQuote() API function was used to retrieve quote detail in XML (the option selected in our application. Screen shots of the mobile application may be seen in Figure 3. The URL for this system is http://m.softwareengineeringonline.com/Black-ScholesV1.

### Critical Success Factors

We now examine the Critical Success Factors (CSFs) of the mobile application presented based on the approach of Buellingen and Woerter (2004). For the four CSFs in this model:

- Transmission rate - Whereas many mobile applications require a high degree of data transfer to and from a web server, this application does not. The economy of data transferred to and from the server is under 2k in each direction. While 3G and 4G networks are abundant, this application will work on a Wi-Fi network, making it independent of the transmission rate.
- Personalization - Users are allowed to input the parameters necessary for computing the Black-Scholes pricing model for a designated option. There are no restrictions, except that the options contract should be European, or an ETF (Electronically Traded Fund) which has the same lack of dividend as the European Options model.
- Data security and IT security - The second (2.0) release of this application will use a secure SSL connection (https) to the server, helping in the authentication of data received and data transmitted back to the mobile application. Such security is critical for the use of applications that perform real-time financial modeling.
- User-friendliness - This application requires a minimum of inputs from the user, reducing time for input and increasing the timeliness of responses from the server.

### Conclusions and Future Work

Computational upgrades to Version 2.0 will include:

- As the Black-Scholes model produces prices for European options, Version 2.0 will use Black's Adjustment for American Options. (Black, 1975; Greske and Roll, 1984).

- Calculation of volatility will be improved using the exponentially weighted average. (Risk Metrics, 1996).
- An optional calculation using the Binomial Options Pricing Model (Cox, Ross and Rubinstein, 1979) specifically for American Options.

Version 2.0 will use the PhoneGap SDK (`www.phonegap.com`) to create a mobile application that can be downloaded directly to mobile devices. PhoneGap will take a web based application and create the necessary binaries across all mobile platforms ensuring rapid development and deployment.

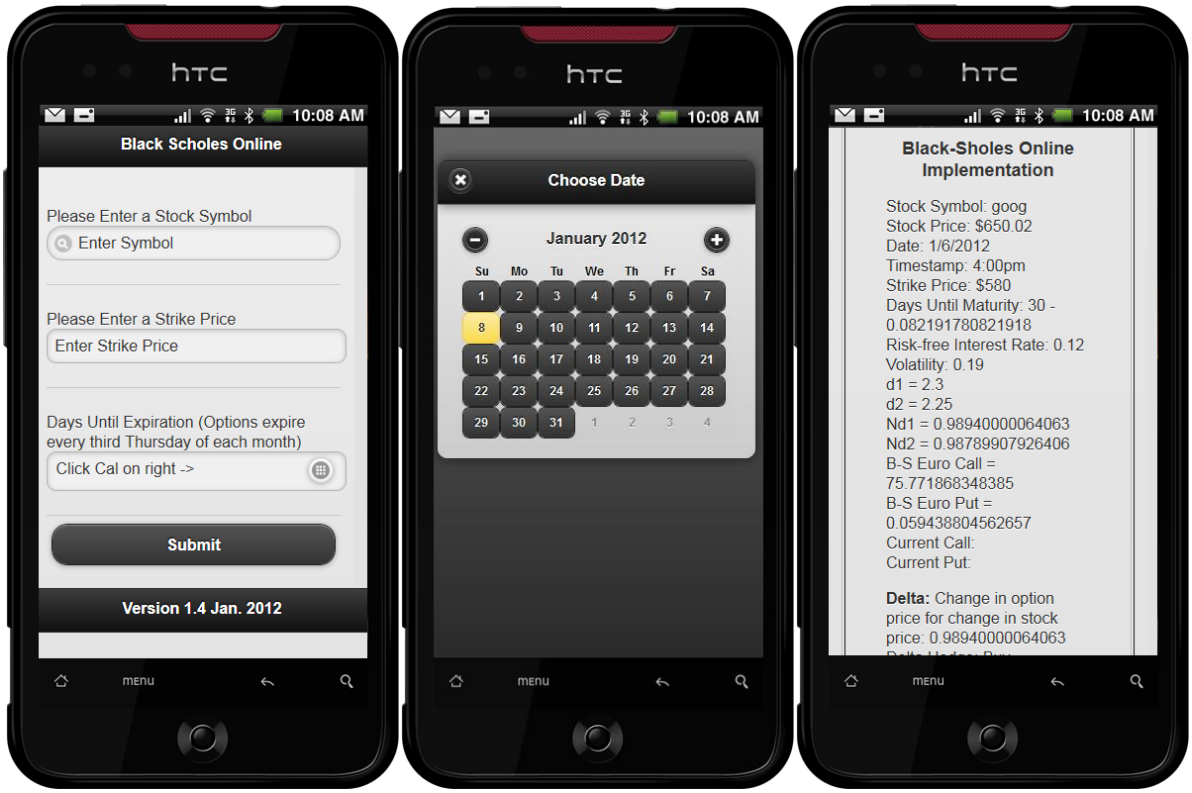



**Figure 3. (Left) Black-Scholes Mobile Application Home Page; (Center) Date Chooser from HTML 5; (Right) Mobile Black-Scholes Calculation**

## REFERENCES


1. Black, F. (July-August 1975) Fact and fantasy in the use of options, *Financial Analysts Journal, 31(36-41),* 61-72.
2. Black, F. and Scholes, M. (May - Jun., 1973) The pricing of options and corporate liabilities, *The Journal of Political Economy*, 81(3), 637-654.
3. Buellingen, F. and Woerter (December 2004) Development perspectives, firm strategies and applications in mobile commerce, *Journal of Business Research, 57(12)*, pp. 1402-1408.
4. Cox, J.C., Ross, S.A. and Rubinstein, M. (1979) Option pricing: A simplified approach, *Journal of Financial Economics*, 7(3), 229-263.
5. E*Trade Financial (2011) *PHP-SDK for Etrade APIs (Developer Beta).*
6. Geske, R. and Roll, R. (Jun., 1984) On valuing American call options with the Black-Scholes European formula, *Journal of Finance, 39(2),* 443-455
7. Risk Metrics (1996) JP Morgan/Reuters Risk metrics technical document, 4th Edition. p. 78.