APPLICATION OF HESTON'S STOCHASTIC VOLATILITY APPROACH IN NUMERICAL COMPUTATION OF THE EUROPEAN OPTIONS: PRELIMINARY DESIGN

Paul Ryan
SAINT CLOUD STATE UNIVERSITY
@stcloudstate.edu

Shivam Soni
SAINT CLOUD STATE UNIVERSITY
@stcloudstate.edu

RAQEEB ABDUL
SAINT CLOUD STATE UNIVERSITY
rasultanov@stcloudstate.edu

RENAT SULTANOV
SAINT CLOUD STATE UNIVERSITY
rasultanov@stcloudstate.edu

DENNIS GUSTER
SAINT CLOUD STATE UNIVERSITY
dcguster@stcloudstate.edu

Abstract

Financial forecasting is a very difficult field in part due to the volatility and complex interrelationships within economies. In fact, many prediction algorithms have fallen short of providing an accurate account of future financial trends. In our preliminary investigation, we used the Black-Scholes model to predict the theoretical price of a European stock option. The algorithm was implemented in java code, which will serve as springboard to refine the basic approach and explore alternative mathematical methodology that could be used to more accurately compute the price of a given option. In this preliminary investigation, we pulled data in spreadsheet format from the Yahoo financial site. Subsequent efforts will aim at pulling stock option prices in near real time from a financial site and feeding them into the java code.

Subsequent research by the authors' plan is twofold. First, to improve upon modeling the variability using variations of Heston's approach and second to take the original Black-Scholes equation and convert it to a series of partial differential equations.

Introduction

Financial forecasting is a very difficult field in part due to the volatility and complex interrelationships within economies. In fact, many prediction algorithms have fallen short of providing an accurate account of future financial trends. One of the bright spots in this field is the Black-Scholes approach [1], which defines the theoretical price of a European option. The goal of this paper is to refine the basic approach and to explore an alternative mathematical methodology that could more accurately compute the price of a given option. Further, a second goal is to develop software that can automate the calculation of such options and explore methodology to extract the data from the internet in near real time. Verifying the accuracy of the algorithm can be determined when the underlying asset is traded, on and the price of the option is set by, the market.

A main parameter in determining the value of an option is the expected volatility of the underlying asset. Fluctuations in asset prices directly affect (change) the value of the option. Therefore, to create successful algorithms the modeling of volatility is key. In this paper, the primary focus will be on to create a template using java code that will be used to integrate the Heston stochastic volatility model [2] into the Black-Scholes model. Specifically, Heston's approach generalizes the Black-Scholes (BSM) model and includes it as a special case. Heston's model can express the price of a process based on drift and non-constant volatility. The approach used by Heston is to model volatility as a diffusion process. In this application, the characteristic function models volatility instead of the probability density function. In the proposed approach, formulation of the algorithm can be generalized to other cases such as dividend payments, options and underlying assets denominated in foreign currencies. This paper will devise a java framework and test the basic Black-Scholes model using a cumulative standard normal probability distribution. In a subsequent paper, we will test several prediction algorithms using Heston's generalized approach to the BSM and compare accuracy to classical prediction algorithms on several different option dividend and currency models.

Literature Review

Before introducing Heston's approach a discussion of the Black-Scholes model (BSM) and associated terms are appropriate. The pricing of options is an extensively studied field. Early approaches ranged from simple statistical models to general equilibrium models. A main justification for the extensive study of options is their simple structure could lead to a general theory of contingent-claims pricing [9]. The specification of general option pricing theory is also a major step toward a theory of pricing a firm's liabilities, the term and risk structure of interest rates, and speculative markets [3]. The BSM relates the theoretical price of a call option to the value of the stock (S), the days until the option expires (T), the option strike price (K), the risk free rate (R), and the daily stock volatility (σ). The stock price refers to the value of a single stock on a given exchange. The day until the option expires refers to the date of maturity for the option. On this date, the buyer of the call option has the choice to either exercise the option or let it expire. The holder of the option will exercise the option if the stock price is greater than the strike price and will let the option expire if the stock price is less than the strike price. There are several assumptions underlying the BSM: (1) the short-term interest rate is known and is constant through time, (2) the stock price follows a random walk in continuous time and its variance is proportional to the square of the stock price, (3) the stock pays no dividends or other distributions, the option is European (it can only be exercised at maturity), (4) there are no transaction costs to buying/selling the stock or the option, (5) it is possible to borrow any fraction of the price of a security to buy or hold it, at the short-term interest rate [3]. To state succinctly the model prices call options than cannot be exercised early, on assets that pay no dividends (unless adjustments are made), with the underlying asset following a geometric Brownian motion with constant drift and volatility, and the interest rate remains constant as well. For the BSM only volatility of the underlying asset matters in the pricing of the option. The excepted rate of return on the stock is inconsequential for pricing options because BSM relates the level of Brownian motion, instead of the stock price, to the price of the option [4]. However, Brownian motion dictates that implied volatility would be constant through time, across strike price, and maturities [1]. Geske and Roll note the BSM model tend to exhibit biases when used to value American Call options [6]. This is consistent with the notion that the right to exercise an option early always has a nonnegative value [9]. The BSM model has computational biases because there is not a way to treat dividends and thus the probability of an early exercise is not handled properly. If the underlying asset does not pay any dividend, the Black-Scholes model still tends to misprice options due to variance (volatility) biases. There are many commonly used methods of option pricing models other than the Black-Scholes model such as constant elasticity of variance, stochastic volatility, and jump-diffusion models. In his 1995 working paper, “Testing Option Pricing Models” David Bates uses stock options, options on stock indexes and stock index futures, and options on currencies and currency futures to discuss and summarize empirical techniques and major conclusions from the literature. Bates states alternative models differ in their assumption of the distribution of option prices and the options underlying asset. There are two basic approaches for tests of consistency found in the literature: those that estimate distributional parameters from time series data and those that estimate model-specific parameters implicit in option prices and test distributional predictions for the underlying asset [2]. Several papers have attempted to examine the overall consistency of stock volatility with stock option prices. Black and Scholes, along with Latané and Rendleman find that high-volatility stocks tend to have high options prices [2]. Black and Scholes noted concern that high-volatility stocks tended to over predict and low-volatility stocks tended to under predict option prices linked to the asset. Many subsequent studies tried to explain the apparent discrepancy by accounting for early-exercise premiums and bid/ask-related biases. Event studies relating predictable volatility changes to specific one-time events have had mixed results [2]. Increases in implicit volatilities up until earnings calls (Patell and Wolfson 1979) and seasonal variations near the end of the year and into January (Maloney and Rogalski) have been noted by previous studies. Interestingly enough, predictable increases in stock volatility following stock splits were not reflected in option prices (Sheikh 1989). The non-stationary nature of stock volatilities is well known and may be remedied with a stochastic volatility model.

Steven Heston, in his paper, “A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options” developed an option pricing model for a

European call option on an asset with stochastic volatility. The main contribution of the paper is that the solution is in closed form and thus less computationally intensive than other stochastic volatility models. Stochastic volatility models are also advantageous because correlations between the option and the underlying asset do not need to be perfectly correlated [1]. Heston shows that if the volatility follows an Ornstein-Uhlenbeck process, then it follows that the correlation is ρ [7]. To state it mathematically:

$$dS_t = rS_t dt + \sqrt{V_t} S_t dW_t^1 \qquad (1)[5]$$

$$dV_t = a(\bar{V} - V_t)dt + \eta\sqrt{V_t} dW_t^2 \qquad (2)[5]$$

$$dW_t^1 dW_t^2 = \rho dt \qquad (3)[5]$$

Where S_t is the pricing of the underlying asset at time t, r is the risk free rate, V_t is the variance at time t, $\bar{V}$ is the historical variance, a is the mean-reversion speed, η is the volatility of the variance process, and dW_t^1, dW_t^2 are Weiner processes correlated with coefficient ρ. The mean-reversion attribute of the Heston model is important because it implies that volatility does not explode to infinity or go to zero, which is unusual in financial data. It also allows the calculation of dependence between asset returns and volatility [5]. He notes the implied variance from option prices may not equal the variance of spot returns given by the true process [7]. To estimate the true process, Heston argues, researchers could use the true spot-price process (a time series of spot prices) to compute the risk-premium parameter by using the average returns of options hedged against changes in the spot asset [7]. Alternatively, one could use parameters implied by option prices. The stochastic volatility model proposed by Heston also can deal with any variance biases mentioned above by linking the spot asset returns to option prices. A stock price can be modeled with stochastic volatility by first showing the stock price follows the diffusion process:

$$dX_t = \mu^P(X_t)d_t + \sigma(X_t)dW_t^P \qquad (2)[1]$$

where X_t is a vector of size m of state variables, W_t^P is the standard Brownian motion, $\mu^P(\cdot)$ is a function of X_t of size m, and $\sigma(\cdot)$ is a m × m matrix-valued function [1]. Here the stock price is a function of X_t so it follows$S_t = \mathrm{F}(X_t)$. AÏt-Sahalia and Kimmel note in practice the stock price or the logarithm of the stock price is a state variable inX_t. To satisfy the no arbitrage assumption AÏt-Sahalia and Kimmel rely on earlier theoretical grounds of a class of agents from Harrison and Kreps (1979). Under the "equivalent martingale measure", Q the state vector follows a similar process above:

$$dX_t = \mu^Q(X_t)d_t + \sigma(X_t)dW_t^Q \qquad (3)[1]$$

with the traded asset satisfying

$$dS_t = (r - \delta)(S_t)dt + \sigma_1(X_t)dW_t^Q \qquad (4)[1]$$

where r is the risk free rate (usually measured by short-term treasury yields, and δ are dividend payments. For simplicity, AÏt-Sahalia and Kimmel assume that δ is constant. From the equations above, an investment in a stock must have an instantaneous expected value of the risk free rate, its mean is only dependent on the stock, but the volatility can depend on any of the variables includingS_t. Further specification of the volatility parameter and payoff expectations the equation simplifies as follows:

$$\frac{\partial\xi(t,S_t)}{\partial t} + \frac{\partial\xi(t,S_t)}{\partial S_t}(r-\delta)S_t + \frac{1}{2}\frac{\partial^2\xi(t,S_t)}{\partial S_t}\sigma_s^2(S_t) - r\xi(t,S_t) = 0 \qquad (5)[1]$$

with the ultimate consequence being changes in derivative securities being perfectly correlated with changes in the stock price [10]. In a stochastic volatility model, the general assumptions relating to volatility do not hold. Namely, volatility is not just a function of the stock price but one of several state variables that need to be introduced.

A significant challenge to estimating stochastic volatility models is that the transition density of the state vector is rarely known and some state variables that are used to calculate volatility are rarely observed. AÏt-Sahalia and Kimmel point out in their paper, "Maximum likelihood estimation of stochastic volatility models" that estimation of stochastic volatility models is reduced to a filtering problem when just the asset prices are observed. The authors also argue the additional state variables can be extracted from option prices (e.g. at-the-money short maturity option). The paper develops a method of maximum likelihood estimation for both scenarios. Maximum likelihood estimation models are rarely used in finance because the variables are observed in discrete time and the theoretical models are in continuous time [1]. This reinforces the need for the idea to be able to pull stock values from the internet in near real time. Previous work had been done by AÏt-Sahalia (2001) to develop series approximations of the likelihood function at discrete time intervals. The effect of the approximation can be measured by calculating the volatility from the observed asset and option prices and comparing results to the implied volatility of an at-the-money-short maturity option [1]. A finding from AÏt-Sahalia and Kimmel is the error from calculating the implied volatility of an at-the-money short maturity option is often smaller than the one introduced from calculating the parameters and leads to efficiency gains. The authors show the feasibility of their results on four types of models: stochastic volatility model (Heston), a GARCH stochastic volatility model, and a constant elasticity volatility model using Monte Carlo simulations. The authors then use real world data from January 2nd, 1990 to September 30, 2003 further motivating their results. The authors find that the maximum likelihood estimation technique can be applied to the Heston model accurately and efficiently. The computation time ranges from a few minutes when volatility is unobserved and even less when a proxy for volatility is used.

Implementing option pricing models can be computationally difficult. Recent advances in computational power and option pricing theory have allowed researchers to implement option pricing models. Hurn et al. in their paper, "Estimating the Parameters of Stochastic Volatility Models Using Option Price Data", develop a maximum likelihood method for estimating the parameters of the stochastic volatility model. Their method combines the power of graphical processing units and parallel computing methods to manage the load of the pricing method. The data set used in the study is comprised of S&P index from January 2^{nd}, 1990 to June 30^{th} 2012 for 4,459,751 observations, over 5672 trading days, and 600,764 call and put options. Other studies have used S&P 500 index options because, unlike some indexes, the options are European options [10]. While the options are European there are dividend payments made on the underlying stock so an adjustment must be made[1] [10]. A common method to reduce the computational load in the non-parametric setting is to reduce the time dimension of the data set or reduce the cross section element (number of options). The use of particle filters in estimation, which requires Monte Carlo integration over unobserved volatility states, can take six months to complete a data set of 18,000 particles and 10 option prices on each day of the entire data set when estimating Heston's model [8]. The largest computational component when evaluating Heston's model is tied up in evaluating exponential, logarithmic and trigonometric functions [8]. Rearrangements in the pricing formula can reduce the number of trigonometric functions needed from four to two [8]. In the simulation experiments run by Hurn et al., for any call option with maturity T and strike price K once exponential function and two trigonometric function was needed to find the solution for its price. To implement Heston's model a researcher needs a way to decide which options to use in the estimation procedure. Since it cannot be guaranteed that the options are priced using Heston's model, one course of action, following Eraker (2004), is to draw a random option that is traded on the first day of the sample. Once the option is no longer traded another option is picked etc. [8]. Another option, which is used by Hurn et al., is to study the most liquid options. A measure of how well the particle estimation methods do is to measure its ability to track the VIX (Implied Volatility Index). Essentially, the VIX uses European put and call options to represent the average of the implied stock market volatility over the next 30 calendar days. The most significant contribution here is that modern computing hardware is powerful enough to estimate a large number of option prices using Heston's stochastic volatility model over a large number of days using GPUs. This ultimately a problem that will be addressed in future research. In the current paper, the goal is to create software using the basic equations and extract data from the internet in basic in batch. So this paper could be viewed a simply a proof of concept paper. Improving the equations and pulling the data in near real time are subsequent steps to the current paper.

Software Development Methodology

1.1 Background

The software development projects begins with an interface that allows the user to pick a data set which is downloaded into a standard EXCEL spreadsheet from the Yahoo stock center [11]. See the figure below.

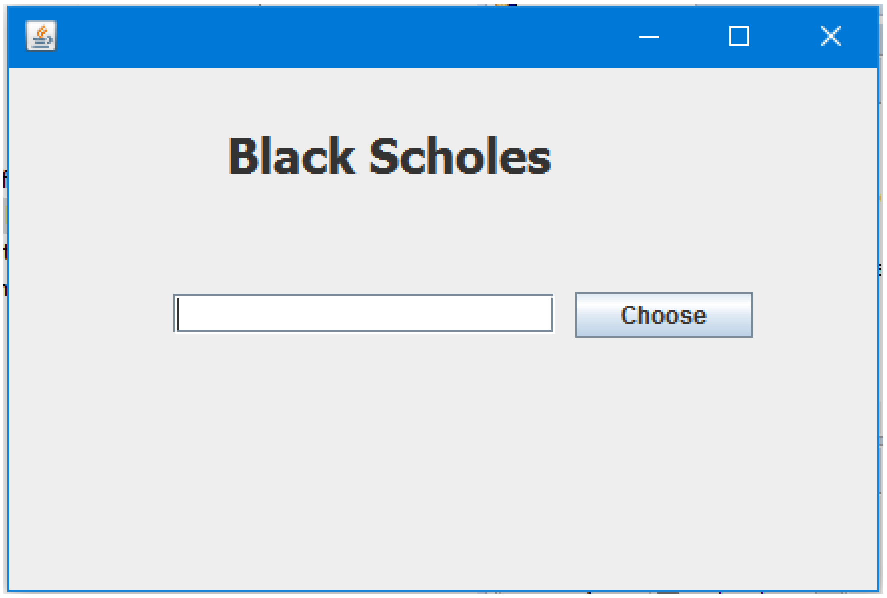

Figure 1: Home Screen

As stated earlier, the preliminary algorithm that was implemented involved calculating the theoretical value of a European call option. Java was selected as the developmental language because it is a C derivative, object oriented, has rich library options, and is the language of choice in the School of Business in which the authors' reside. This java application takes the input as defined from the drop box above as a .csv file, which in turn generates the call option values in .csv file. A sample csv file generated by the java application appears below

A	B	C	D	E	F
96.69	4	70	0.2572	0.0262	27.59967163
96.69	4	75	0.2572	0.0262	24.47892736
96.69	4	80	0.2572	0.0262	22.95420105
96.69	4	84	0.2572	0.0262	22.60269003
96.69	4	85	0.2572	0.0262	22.55883409
96.69	4	85.5	0.2572	0.0262	22.53553731
96.69	4	86	0.2572	0.0262	22.50936529

Figure 2: Sample CSV File

Specifically, the field description of csv file appears below:

Column A is **the Current Stock Price**

Column B is **days until expiration**

Column C is **Option Strike Price**

Column D is **Monthly Stock Volatility**

Column E is **risk free interest rate**

Column F is **Value of European call option (generated by the java application).**

1.2 Implementation

The Java application was developed using a GUI toolkit Java Swing. There are three main components within this application. The first is opening the file browser (which is used to browse and select the file). The second component allows parsing of the CSV File, the third then calculates the theoretical value of a call option. To implement this design three methods were created: openFileBrowser, parseCSV, and callBlackScholes. The figure below depicts the interrelationship pf these modules.

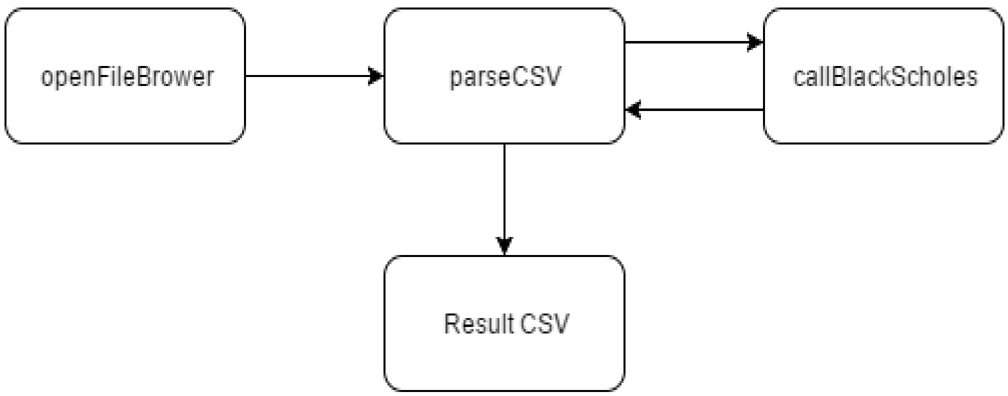

Figure 3: Java Swing Model

To clarify the purpose of each method a brief discussion and the source appear below.

1.2.1 openFileBrowser

By clicking the “choose” button the method is called. The objective of this function is to open the native file browser from the application. This will allow the user to choose the csv file. This will ultimately generate a URI (universal resource indicator) to locate the file, which is then passed to the parseCSV method. The source code for this method appears below.

```
private void openFileBrowser() {

fc = new JFileChooser();
FileNameExtensionFilter filter = new FileNameExtensionFilter("Comma Seperated Files",
"csv");
fc.setFileFilter(filter);
```

```
int returnVal = fc.showOpenDialog(frame);

if (returnVal == JFileChooser.APPROVE_OPTION) {
	System.out.println("You chose to open this file: " + fc.getSelectedFile().getName());
	textField.setText(fc.getSelectedFile().getPath());
	parseCSV(fc.getSelectedFile().getPath());
	}

}
```

1.2.2 parseCSV

This method is called after the file is chosen from the file browser. It passes one argument string which is the path indicator of the selected file. Then the file can be opened using FileReader and buffered using BufferedReader. The BufferedReader will allow efficient reading every line until the end of file is reached. As would be expected in a CSV file, each line is comma delimited, allowing easy storage in an array. Now these values are passed to the callBlackScholes method which returns the call option values. Finally, after the method has calculated all of the call option values they are stored in a CSV file. This strategy would make it easy for a typical financial end user to take advantage of this system. Below is the source code for parseCSV

```
private void parseCSV(String csvFile) {
	BufferedReader br = null;
	String line = "";
	String cvsSplitBy = ",";
	int i = 0;
	String sFileName = "C:\\Users\\riz\\Desktop\\bs1.csv";
	FileWriter writer = null;

	try
	{
		writer = new FileWriter(sFileName);
		br = new BufferedReader(new FileReader(csvFile));
		while ((line = br.readLine()) != null) {

		if (i != 0) { // use comma as separator
		String[] country = line.split(cvsSplitBy);


		double cv = callBS(Double.parseDouble(country[0]),
		Double.parseDouble(country[2]),
		Double.parseDouble(country[4]), Double.parseDouble(country[3]),
		Double.parseDouble(country[1]));

		writer.append(country[0]);
		writer.append(',');
```

```
                    writer.append(country[1]);
                    writer.append(',');
                    writer.append(country[2]);
                    writer.append(',');
                    writer.append(country[3]);
                    writer.append(',');
                    writer.append(country[4]);
                    writer.append(',');
                    writer.append(String.valueOf(cv));
                    writer.append('\n');

          }
          i++;
          }
          writer.flush();
             writer.close();

          } catch (FileNotFoundException e) {
          e.printStackTrace();
          } catch (IOException e) {
          e.printStackTrace();
          }
}
```

1.2.3 callBlackScholes

This method is used to generate the call option value. It evaluates five double arguments i.e. current stock price, option strike price, risk free interest rate, daily stock volatility and days until expiration. The result is returned as a double value. The figure below shows the mathematical equation for the Black-Scholes model used in this method.

To calculate the option price, we have create a method callBS

$$C = SN(d_1) - Ke^{-r\left(t/365\right)}N(d_2)$$

Figure 4: Black-Scholes Equation

Where,

$$d_1 = \frac{\ln(S/K) + \left(r/365 + \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

Figure 5: Variable d1 Solution Figure 6: Variable d2 Solution

To calculate the option price the program calls a method named callBS. The source code of the method is below.

```
public static  double callBS(double sp, double xp, double rr, double si, double t){

        double result =0, d1=0,d2=0;

        NormalDistribution nd = new NormalDistribution(0,1);

        d1= (Math.log(sp/xp) + ((rr/365)+((si*si)/2))*t) / (si * Math.sqrt(t));
        d2 = d1 - si * Math.sqrt(t);

        result = sp * nd.cumulativeProbability(d1) - xp * Math.exp(-rr * (t/365)) *
    nd.cumulativeProbability(d2);

        if(result<0)
            result = 0;

        return result;

    }
```

For calculating the cumulative standard normal probability distribution the java class NormalDistribution, which is contained in an open source library, is used. This Class has a method called cumulativeProbability that can be used to calculate the cumulative standard normal probability distribution. This modular approach will make it easy in subsequent versions of the project to integrate other distributions like the one used by Heston.

Java equivalent of d1 is below

```
d1= (Math.log(sp/xp) + ((rr/365)+((si*si)/2))*t) / (si * Math.sqrt(t));
```

Java equivalent of d1 is below

```
d2 = d1 - si * Math.sqrt(t);
```

Java equivalent of C is below

```
result = sp * nd.cumulativeProbability(d1) - xp * Math.exp(-rr * (t / 365)) *
nd.cumulativeProbability(d2);
```

Below is a sample output generated from the above java code above.

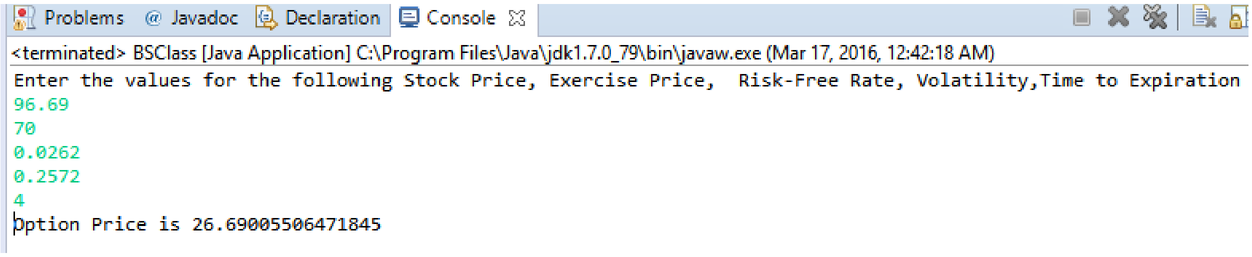

Figure 7: Sample Output

Discussion/Conclusions

Financial forecasting is a very difficult undertaking due to complex interrelationships within economies. While many algorithms and variations of those algorithms have been developed, many have fallen short in providing accurate predictions of future financial trends. One of the most successful algorithms in this area is the Black-Scholes approach. In our preliminary investigation, we used this approach to model the theoretical price of a European stock option. This algorithm was implemented in java code and will be used as springboard to further refine the basic approach and to explore alternative mathematical methodology that could be used to more accurately compute the price of a given option. The second goal of this project was to develop software that can automate the calculation of such options and explore methodology to extract the data from the internet in near real time. In this case, we pulled data in spreadsheet format from the Yahoo financial site. Subsequent efforts will be aimed at pulling the stock options in near real time from a financial site and feeding them into the java code.

While verifying the accuracy of the algorithm used herein was quite simple, as improvements in the algorithmic design occur the difficulty to verify increases. The authors’ plan is twofold. First, to improve upon modeling the variability using variations of Heston’s approach. Second, to take the original Black-Scholes equation and convert it to a series of partial differential equations.

References

1. AÏt-Sahalia, 1., & Kimmel, R. (207). Maximum likelihood estimation of stochastic volatility models. *Journal of Financial Economics, 83*, 413-452. Retrieved March 04, 2016.
2. Bates, D. M. (1995). Testing Option Pricing Models. *National Bureau of Economic Research*, 5129th ser. Retrieved March 03, 2016.
3. Black, F., & Scholes, M. (1993). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy, 81*(3), 637-654. Retrieved February 29, 2016.
4. Carr, P. (2002). FAQ's in Option Pricing Theory. 1-31. Retrieved 2016.
5. Crisostomo, R. (2014). An Analysis of the Heston Stochastic Volatility Model: Implementation and Calibration using Matlab. Retrieved February 28, 2016.
6. Geske, R., & Roll, R. (1984). On Valuing American Call Options with the Black-Scholes European Formula. *The Journal of Finance, 39*(2), 443-455. Retrieved March 03, 2016.
7. Heston, S. L. (1993). A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond Currency Options. *The Review of Financial Studies, 6*(2), 327-343.
8. Hurn, A., Lindsay, K., & McLelland, A. (2012). Estimating the Parameters of Stochastic Volatility Models using Option Prices. *NCER Working Paper Series*, 87th ser., 1-36. Retrieved March 01, 2016.
9. Merton, R. C. (1973). Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science, 4*(1), 141-183. Retrieved February 28, 2016.
10. Shu, J., & J. (n.d.). The Relationship Between Implied and Realized Volatility of S&P 500 Index. *WILMOTT Magazine*, (Technical Issue 4), 83-91.
11. Yahoo, (2016). http://finance.yahoo.com/stock-center/.