SOFTWARE APPLICATION FOR COMPUTING OPTIONS

Mairiza, Dewi
Faculty of Computer Science, University of Indonesia, Depok  
mairiza@cs.ui.ac.id

ABSTRAK

Kata Kunci: Penjualan dan Pembelian Options, Sensitivity Variables, Estimasi Harga dan Keuntungan

1. Introduction
There are a lot of problems regarding options trading in Indonesia. One of them is the spin-off process of two public organizations (PT. Semen Padang and PT. Semen Gresik) that becomes a national problem. Some people have different arguments whether the options should or should not be exercised. By then, options trading are of interest of many traders. Options trading are not as popular as stocks trading, since many traders and buyers have difficulties in understanding the conduct of options trading. The difficulties include problems in making decisions to buy, sell, and exercise call or put options.

The Black-Scholes and Merton’s Jump Diffusion models are put and call options pricing models which are based on uncertainty in stock market behaviours. Using this model, the options traders (organizations or companies) can estimate their profit and the time to buy, sell, and exercise their options. There are four groups of users:
1. Call writers/sellers: They sell call options if they estimate the stock pricing in the stock market is stable or decreases.
2. Call buyers: They buy call options if they estimate the stock pricing in the stock market increases in short term.
3. Put writers/sellers: They sell put options if they estimate the stock pricing in the stock market is stable or increases.
4. Put buyers: They buy put options if they estimate the stock pricing in the stock market decreases.

To make options traders and buyers easier in estimating their profit in the uncertainty options market, a tool implementing options trading concept is a necessity. Such software application can provide useful information to traders before they sell or buy options.

2. Constraint Program
There are some constraints involved in this application:
1. All of constraints in Black-Scholes and Merton’s Jump Diffusion Model are in effect.
2. Only for European Options.
3. Interest rates are constant.

3. Implementation of Black-Scholes Model
Call option of the Black-Scholes model is calculated using this equation:
\[ c = S \cdot N(d1) - X \cdot e^{-rt} \cdot N(d2) \]  
where:
\[ c \] = call option premium pricing  
\[ N(.) \] = cumulative normal density function  
\[ d1 = \ln \left( \frac{S}{X} \right) + \left( r + 0.5 \sigma^2 \right) t \]  
\[ d2 = d1 - \sigma \sqrt{t} \]  
\[ r \] = interest rates (in %, each year)  
\[ X \] = exercise price of call option  
\[ S \] = stock price  
\[ e \] = natural number (2.71828...)  
\[ t \] = option lifetime before expired (year)  
\[ \sigma \] = standard deviation of stock return (volatility)

Put Option in the Black-Scholes model is calculated using this equation:
\[ p = X \cdot e^{-rt} \cdot N(-d2) - S \cdot N(-d1) \]  
where:
\[ p \] = put option premium pricing
\[ N(.) \] = cumulative normal density function
\[ d1 = \frac{\ln ( S / X ) + ( r + 0.5 \sigma^2 ) t}{\sigma \sqrt{t}} \]
\[ d2 = d1 - \sigma \sqrt{t} \]
\[ r \] = interest rates (in %, each year)
\[ X \] = exercise price of call option
\[ S \] = stock price
\[ e \] = natural number (2.71828...)
\[ t \] = option lifetime before expired (year)
\[ \sigma \] = standard deviation of stock return (volatility)

Sensitivity variables involves in the model are:

Table 1. Formula of Black-Scholes Sensitivity Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Call Option</th>
<th>Put Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>( \Delta C / S = N(d1) )</td>
<td>( \Delta P / S = N(d1) - 1 )</td>
</tr>
<tr>
<td>GAMMA</td>
<td>( \Gamma C = \partial \Delta C / \partial S = S N'(d1) )</td>
<td>( \Gamma P = \partial \Delta P / \partial S = S N'(d1) - 1 )</td>
</tr>
<tr>
<td>THETA</td>
<td>( \Theta C = - \delta S N(d1) - \frac{S N'(d1)}{2} \sqrt{t} - r X e^{-rt} N(d2) )</td>
<td>( \Theta P = - \delta S N(d1) + \frac{S N'(d1)}{2} \sqrt{t} + r X e^{-rt} N(-d2) )</td>
</tr>
<tr>
<td>VEGA</td>
<td>( \nabla C = \partial \Delta C / \partial \sigma = S \sqrt{t} N'(d1) )</td>
<td>( \nabla P = \partial \Delta P / \partial \sigma = S \sqrt{t} N'(d1) )</td>
</tr>
<tr>
<td>RHO</td>
<td>( \rho C = \partial \Delta C / \partial \delta = X t e^{\delta t} N(d2) )</td>
<td>( \rho P = \partial \Delta P / \partial \delta = -X t e^{\delta t} N(-d2) )</td>
</tr>
</tbody>
</table>

4. Implementation of Merton’s Jump Diffusion Model

Call Option for the Merton’s Jump Diffusion model is calculated using the following equation:

\[ c^M = e^{-\delta t} S N(d1^M) - X e^{-rt} N(d2^M) \]  

Put Option for the Merton’s Jump Diffusion model is calculated using the following equations:

\[ p^M = X e^{-rt} S N(-d2^M) - S e^{-\delta t} N(-d1^M) \]

Sensitivity variables involves in the model are:

Table 2. Formula of Merton’s Jump Diffusion Sensitivity Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Call Option</th>
<th>Put Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>( \Delta C / S = e^{\delta t} N(d1^M) )</td>
<td>( \Delta P / S = e^{\delta t} [ N(d1^M) - 1 ] )</td>
</tr>
<tr>
<td>THETA</td>
<td>( \Theta C = - \delta (c / \sigma) = - \frac{S N'(d1^M) \sigma e^{\delta t}}{2 \sqrt{t}} )</td>
<td>( \Theta P = - \delta (c / \sigma) = - \frac{S N'(d1^M) \sigma e^{\delta t}}{2 \sqrt{t}} )</td>
</tr>
<tr>
<td>VEGA</td>
<td>( \nabla C = \partial \Delta C / \partial \sigma = e^{\delta t} N(d2^M) )</td>
<td>( \nabla P = \partial \Delta P / \partial \sigma = e^{\delta t} N(d2^M) )</td>
</tr>
<tr>
<td>RHO</td>
<td>( \rho C = \partial \Delta C / \partial \delta = S (c / \sigma) N(d1^M) e^{\delta t} )</td>
<td>( \rho P = \partial \Delta P / \partial \delta = -S (c / \sigma) N(d1^M) e^{\delta t} )</td>
</tr>
<tr>
<td>GAMMA</td>
<td>( \Gamma C = \partial \Delta C / \partial \delta = S N'(d1^M) )</td>
<td>( \Gamma P = \partial \Delta P / \partial \delta = S N'(d1^M) )</td>
</tr>
</tbody>
</table>

5. Software Application

The Black-Scholes and Merton’s Jump Diffusion models have been implemented in the Matlab 5.3 environment. By using this application, buyers and sellers of call and put options can estimate the options pricing and the value of sensitivity variables as a judgment in deciding whether the buy or sell options can be proceeded. The output also gives a graph representation of options pricing in market price. By using this graph, the options sellers can estimate the minimum and maximum of options pricing. This can give them the estimation of the maximum profit from the options selling transaction. The options buyers also can estimate the minimum and maximum of options pricing. This can give them the estimation of the maximum profit from the options buying transaction.

\[ a : \text{ range of minimum and maximum of call option pricing when stock price is } $100. \text{ In this range, buyers and sellers of call option can estimate call options pricing that will give them maximum benefits.} \]
Figure 1. Call Pricing as a Function of Stock Pricing

Figure 2. Put Pricing as a Function of Stock Pricing

b: range of minimum and maximum of put option pricing when stock price is $100. In this range, buyers and sellers of put option can estimate put options pricing that will give them maximum benefits.

Figure 3. User Interface of Software Application
Explanation of picture:
a: menu to choose model (Black-Scholes or Merton’s Jump Diffusion)
b: input menu, users have to enter the value of variable to find the price of options, sensitivity variables, and graph representation. The inputs are: Stock price (S), Exercise price (X), interest rates, option’s life (year), volatility, and dividend rates (only for Merton’s Jump Diffusion model).
c: help menu.
d: reset the value.
e: enter, to find solution.
f: display of output; price of put and call option, value of sensitivity variables for put and call option.
g: graph representation of put option, to estimate minimum and maximum of put option price.
h: graph representation of call option, to estimate minimum and maximum of call option price.
i: history, to save the record of simulation.
j: save as menu, to save history to new file.
k: exit menu, exit from application.

Table 3. Execution Time of Black-Scholes Model

<table>
<thead>
<tr>
<th>No</th>
<th>S ($)</th>
<th>X ($)</th>
<th>t (year)</th>
<th>R (%)</th>
<th>$\sigma$</th>
<th>Second (Black-Scholes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80</td>
<td>100</td>
<td>1</td>
<td>9</td>
<td>0.40</td>
<td>0.4400</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>100</td>
<td>180/365</td>
<td>8</td>
<td>0.30</td>
<td>0.3900</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>45</td>
<td>6/12</td>
<td>10</td>
<td>0.45</td>
<td>0.6100</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>50</td>
<td>3/12</td>
<td>6</td>
<td>0.40</td>
<td>0.3300</td>
</tr>
<tr>
<td>5</td>
<td>110</td>
<td>100</td>
<td>1</td>
<td>9</td>
<td>0.40</td>
<td>0.6100</td>
</tr>
<tr>
<td>6</td>
<td>500</td>
<td>550</td>
<td>3/12</td>
<td>6</td>
<td>0.75</td>
<td>0.4300</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>0.80</td>
<td>0.4400</td>
</tr>
<tr>
<td>8</td>
<td>990</td>
<td>900</td>
<td>2</td>
<td>10</td>
<td>0.25</td>
<td>0.3800</td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>150</td>
<td>4/12</td>
<td>3</td>
<td>0.45</td>
<td>0.3300</td>
</tr>
</tbody>
</table>

Average time of execution program for Black-Scholes model is 0.4088888 second.

Table 4. Execution Time of Merton’s Jump Diffusion Model

<table>
<thead>
<tr>
<th>No</th>
<th>S ($)</th>
<th>X ($)</th>
<th>t (year)</th>
<th>R (%)</th>
<th>$\sigma$</th>
<th>$\delta$</th>
<th>Second (Merton’s Jump Diffusion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80</td>
<td>100</td>
<td>1</td>
<td>9</td>
<td>0.40</td>
<td>0.03</td>
<td>0.4400</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>100</td>
<td>180/365</td>
<td>8</td>
<td>0.30</td>
<td>0.08</td>
<td>0.4900</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>45</td>
<td>6/12</td>
<td>10</td>
<td>0.45</td>
<td>0.03</td>
<td>0.3800</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>50</td>
<td>3/12</td>
<td>6</td>
<td>0.40</td>
<td>0.10</td>
<td>0.3800</td>
</tr>
<tr>
<td>5</td>
<td>110</td>
<td>100</td>
<td>1</td>
<td>9</td>
<td>0.40</td>
<td>0.09</td>
<td>0.3900</td>
</tr>
<tr>
<td>6</td>
<td>500</td>
<td>550</td>
<td>3/12</td>
<td>6</td>
<td>0.75</td>
<td>0.50</td>
<td>0.3800</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>0.80</td>
<td>0.3</td>
<td>0.4400</td>
</tr>
<tr>
<td>8</td>
<td>990</td>
<td>900</td>
<td>2</td>
<td>10</td>
<td>0.25</td>
<td>0.01</td>
<td>0.4900</td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>150</td>
<td>4/12</td>
<td>3</td>
<td>0.45</td>
<td>0.02</td>
<td>0.3800</td>
</tr>
</tbody>
</table>

Average time of execution program for Merton’s Jump Diffusion model is 0.4188888 second.

Here are some advantages that can affect options traders and buyers when using this software application:
1. They can make a simulation of options pricing before deciding sell or buy options.
2. They can estimate the minimum and maximum price of option that will give them maximum profit.
3. They also find sensitivity variables as a judgment in deciding to buy or sell option.
4. There is a history menu, that saves all simulations that have already been done, and users also can put data from history if they need.
5. Users can choose options model (Black-Scholes or Merton’s Jump Diffusion) in their simulation, so that they can compare the output to find the best decision.

6. Summary
Options trading as one of many hot issues in Indonesia need a software application that helps sellers and buyers to make a decision in options selling or buying. By using this application, users can simulate stock price, interest rates and others to find the best options pricing that give maximum profit based on sensitivity variables as a judgment. Software application has been developed in Matlab environment, and has average time execution for Black-Scholes model = 0.4088888 second and for Merton’s Jump Diffusion model = 0.4188888 second.
Bibliography