European Option Pricing for Non Dividend Paying Stocks

Implementation

Version 1.0

Vivek Rauniyar

Monday 30th September, 2019

Document Version Control

Version	Summary of Changes	Author	Date
1.0	First Edition of the documentation of the implementation of Black Scholes Model for European Option Pricing for Non Dividend Paying Stocks	Vivek Rauniyar	2019-09-30

Table of Contents

1.0 Project Scope

This project aims at system implementation of European Option Pricing for Non Dividend paying stocks and their Greek calculation for user inputted pricing parameters. It also involves calibration of Volatility Smile based on user inputted Implied Volatility/ Strike grid. The programming language used is C++ and the platform used is Dev C++. Before we get into the detailed description of the implementation, the immediate next section introduces us to Options and usage of Black Scholes Merton formula in option pricing. Also there is a mention of the closed form solution for calculating option greeks.

2.0 Option Product

Options are a type of derivatives. Options are financial contracts that give the buyer the right, but not the obligation, to either buy or sell an amount of some underlying asset at a pre-determined price at or before the contract expires.

2.1 Type of Options

Options are of two types – Call and Put.

Call Option gives the buyer the right, but not the obligation, to buy an amount of some underlying asset at a pre-determined price at or before the contract expires.

Put Option gives the buyer the right, but not the obligation, to sell an amount of some underlying asset at a pre-determined price at or before the contract expires.

Based on whether the option can be exercised at or before the contract expires, options can be classified as European, American and Bermudan. European options can be exercised only at the expiry date. American option can be exercised any day before the expiry date. Bermudan option can be exercised on specific dates prior to the expiry date of the option.

This project deals with European options – their pricing and risk measurement.

How to price European options will be covered in the next section on Black Scholes Merton model.

2.2 Black Scholes Merton Model

Black-Scholes Merton is a pricing model used to determine the theoretical value for a call or a put option based on six variables such as volatility of stock, underlying stock price, time to expiry, strike price of the option, dividend yield and risk-free rate. This model is used to determine the price of a European option, which simply means that the option can only be exercised on the expiration date. Black Scholes Model does not consider dividend yield in option pricing while the Black Scholes Merton model considers dividend yield.

2.3 Black-Scholes Merton Formula Parameters

According to the Black-Scholes option pricing model (its Merton's extension that accounts for dividends), there are six parameters which affect option prices:

S_0 = underlying price ($ per share)

X = strike price ($ per share)

σ = volatility (% p.a.)

r = continuously compounded risk-free interest rate (% p.a.)

q = continuously compounded dividend yield (% p.a.)

t = time to expiration (% of year)

Note: In many resources you can find different symbols for some of these parameters. For example, strike price is often denoted K (instead of X being used in this documentation), underlying price is often denoted S (without the zero), and time to expiration is often denoted T – t (difference between expiration and now).

2.4 Black-Scholes Merton Call and Put Option Price Formulas

Call option (*C*) and put option (*P*) prices are calculated using the following formulas:

$$C = S_0\, e^{-qt} * N(d_1) - X\, e^{-rt} * N(d_2)$$

$$P = X\, e^{-rt} * N(-d_2) - S_0\, e^{-qt} * N(-d_1)$$

… where *N(x)* is the standard normal cumulative distribution function.

The formulas for *d1* and *d2* are:

$$d_1 = \frac{ln(\frac{S_0}{X}) + t\,(r - q + \frac{\sigma^2}{2})}{\sigma\sqrt{t}}$$

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

2.5 Black-Scholes Merton Formulas for Option Greeks

Below you can find formulas for the most commonly used option Greeks. Some of the Greeks (gamma and vega) are the same for calls and puts. Other Greeks (delta, theta, and rho) are different.

In several formulas you can see the term:

$$\frac{1}{\sqrt{2\pi}} * e^{\frac{-d_1{}^2}{2}}$$

… which is the standard normal probability density function.

Delta

Delta measures the exposure of option price to movement of underlying stock price.

$$Call\ delta = e^{-qt} * N(d_1)$$

$$Put\ delta = e^{-qt} * \ (N(d_1) - 1)$$

Gamma

Gamma measures the exposure of the option delta to the movement of the underlying stock price.

$$Gamma = \frac{e^{-qt}}{S_0\ \sigma \sqrt{t}} * \frac{1}{\sqrt{2\pi}} * e^{\frac{-d_1{}^2}{2}}$$

Theta

Theta measures the exposure of the option price to the passage of time.

$$Call\ theta =$$

$$= \frac{1}{T}\left(-\left(\frac{S_0\ \sigma\ e^{-qt}}{2\sqrt{t}} * \frac{1}{\sqrt{2\pi}} * e^{\frac{-d_1{}^2}{2}}\right) - r\ X\ e^{-rt}\ N(d_2) + q\ S_0\ e^{-qt}\ N(d_1)\right)$$

$$Put\ theta =$$

$$= \frac{1}{T}\left(-\left(\frac{S_0\ \sigma\ e^{-qt}}{2\sqrt{t}} * \frac{1}{\sqrt{2\pi}} * e^{\frac{-d_1{}^2}{2}}\right) + r\ X\ e^{-rt}\ N(-d_2) - q\ S_0\ e^{-qt}\ N(-d_1)\right)$$

… where T is the number of days per year (calendar or trading days, depending on what you are using).

Vega

Vega measures the exposure of the option price to changes in volatility of the underlying

$$Vega = \frac{1}{100} S_0 \, e^{-qt} \sqrt{t} * \frac{1}{\sqrt{2\pi}} * e^{\frac{-d_1{}^2}{2}}$$

Rho

Rho measures the exposure of the option price to changes in risk free rate

$$Call\,rho = \frac{1}{100} X\,t\,e^{-rt} * N(d_2) \quad Put\,rho = -\frac{1}{100} X\,t\,e^{-rt} * N(-d_2)$$

3.0 Project Implementation

The programming language used for the implementation of this project is C++ and the IDE used is Dev C++. Object Oriented concepts like Classes, Inheritance has been used in this program. The program is divided into three modules – Option Pricing, Option Greeks generation and Volatility Smile Fitting. The implementation of the above modules will be described in fair details in the next section.

3.1 Program Structure

Since the project implements option pricing for non dividend paying stocks, it means the parameter "q" which is the continuously compounded dividend yield is considered as 0. As per the Black Scholes Merton formula simplifies to the Black Scholes version which is implemented in the code base for this project. Hence, the code considers only five pricing parameters - Spot Price (denoted as "S" in the code), Strike Price (denoted as "K" in the code), Volatility (denoted as "v" in the code), risk free rate (denoted as "r" in the code), time to expiry (denoted as "t" in the code).

The program uses "Option" instrument as a base class and CallOption and PutOption as a child class. The Options class has the following public functions based on their common applicability to both Call and Put Option pricing.

1> Function Name: **d_j** - This function performs calculation of d1 and d2 parameters as described in the black scholes merton pricing model in the previous sections. This function takes in parameters – 1 or 2 based on whether the requirement is calculating d1 or d2, Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns d1/d2 to the calling function.

2> Function Name: **norm_cdf** – This function calculates an approximation to the cumulative distribution function for the standard normal distribution. This is a recursive function. This piece of code has been taken from quantlib website. This function takes in parameters – d1 or d2 calculated previously with appropriate signage and returns N(x) as described in the model section to the calling function.

3> Function Name: **gamma** – This function calculates the option Greek "Gamma" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option gamma to the calling function. Since the formula is common for both call and put option, It has been incorporated as a public function in base class "Option".

4> Function Name: **vega** – This function calculates the option Greek "Vega" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option vega to the calling function. Since the formula is common for both call and put option, It has been incorporated as a public function in base class "Option".

5> Function Name: **Vol_Smile_Fitting** – This function arrives at the best fit for Volatility Smile based on user inputted Implied Volatility/Strike grid. The best fit is arrived at assuming a quadratic relationship between Implied Volatility and "Strike/Spot" ratio by minimising the standard error between the user inputted implied volatility and fitted volatility and returns a pointer to the array of calibrated parameters - Convexity, Skew and a Constant to the calling function. The module expresses smile surface (volatility as a function of Strike) as "Volatility = Convexity * Strike^2 + Skew * Strike + Constant" and then tries to arrive at best fit by changing the parameters Convexity, Skew, Constant by a fixed amount (tweak amount) in each iteration. Standard error is calculated for each iteration as per the below formula. The iteration for which the standard error is minimal, is considered as the best fit for the volatility smile and can be used to quote implied volatility for range of strikes. Please note that module calibrates well within a close range of ATM Strike. However, for deep out of/ in the money strikes, the fit may not be that optimal meaning the quadratic behaviour breaks at higher strikes. However, the module does fairly well for close to ATM Strikes. The following boundary values and tweak amount are used for arriving at the best fit by running a nested loop for each of the below smile parameters.

Smile Parameter	Lower Boundary	Upper Boundary	Tweak Amount
Convexity	-10	10	0.01
Skew	-10	10	0.1
Constant	-10	10	0.1

For each iteration within the nested loop, Standard Error is calculated using the following formula:

$$\sum_{1}^{Strike\ count}((Implied\ Vol - Calibrated\ Vol)^{\wedge}2 * (\frac{1}{\sqrt{Absolute\ distance\ from\ ATM\ Strike}}))$$

Where,

Implied Vol is user inputted Volatility for a certain Strike

$$Strike\ Count = (Highest\ Strike - Lowest\ Strike)/5$$

$$Calibrated\ Vol = Convexity * Strike^2 + Skew * Strike + Convexity$$

$$Absolute\ Distance\ from\ ATM\ Strike = |Strike - ATM\ Strike|$$

Mean standard error (MSE) for each iteration is calculated as the following:

$$MSE = \frac{\sqrt{Standard\ Error}}{Strike\ Count}$$

The values of the parameters Convexity, Skew and Constant for the iteration in which MSE is the lowest is considered as the best fit smile parameters and is used to arrive at final Calibrated Volatility (Fitted Volatility). The tweak amount in each iteration can be lowered to arrive at better fits however at the cost of increased computation time. Hence the user needs to balance between accuracy of the fit and compute time to arrive at best fit.

The CallOption class inherits the properties of the Option class and has the following public functions.

1> <u>Function Name:</u> **Call_Opt_Pricer** – This function calculates the option premium for a Call Option. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns call option premium to the calling function. Since the formula is applicable for call option only, it has been incorporated as a public function in child class "CallOption".

2> <u>Function Name:</u> **delta** – This function calculates the option Greek "Delta" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option delta to the calling function. Since the formula is applicable for call option only, it has been incorporated as a public function in child class "CallOption".

3> Function Name: **theta** – This function calculates the option Greek "Theta" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option theta to the calling function. Since the formula is applicable for call option only, it has been incorporated as a public function in child class "CallOption".

4> Function Name: **rho** – This function calculates the option Greek "Rho" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option rho to the calling function. Since the formula is applicable for call option only, it has been incorporated as a public function in child class "CallOption".

The PutOption class inherits the properties of the Option class and has the following public functions.

1> Function Name: **Put_Opt_Pricer** – This function calculates the option premium for a Put Option. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns put option premium to the calling function. Since the formula is applicable for put option only, it has been incorporated as a public function in child class "PutOption".

2> Function Name: **delta** – This function calculates the option Greek "Delta" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option delta to the calling function. Since the formula is applicable for put option only, it has been incorporated as a public function in child class "PutOption".

3> Function Name: **theta** – This function calculates the option Greek "Theta" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option theta to the calling function. Since the formula is applicable for put option only, it has been incorporated as a public function in child class "PutOption".

4> Function Name: **rho** – This function calculates the option Greek "Rho" as per the methodology described in the model section. This function takes in parameters – Spot Price, Strike Price, Volatility, risk free rate, time to expiry and returns option rho to the calling function. Since the formula is applicable for put option only, it has been incorporated as a public function in child class "PutOption".

3.2 Program Execution

This program performs 3 functions. It performs Equity Vanilla Option Pricing, generates option greeks and performs volatility smile fitting based on user inputted Vol/Strike grid. The following section takes you through the execution steps for each of these modules.

Equity Vanilla Option Pricing and Greeks Generation happen together in one module while Volatility Smile Fitting happens in another module. The program when run initially asks for which of the above two module the user wants to run via the command prompt. The user needs to select "P" for Option Pricing and Greek Generation and "V" for Volatility Smile Fitting (See below screenshot – Fig 1).

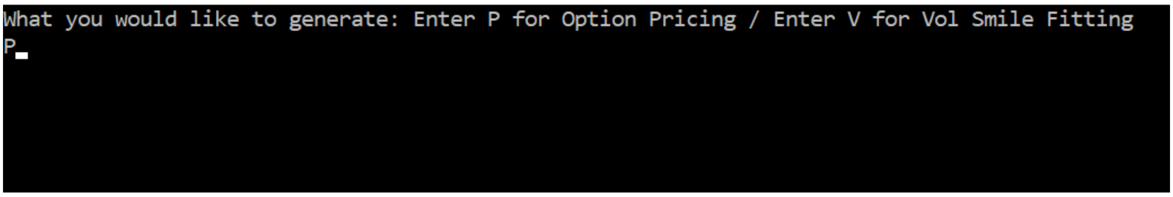

Fig 1: Module Selection Screen. The user selects "P" (Module 1).

3.2.1. Module 1: European Option Pricing for Non dividend paying stocks and Greeks Generation

Once the user selects "P" as shown in Fig 1, the program jumps to Module 1 asking user inputs for the Option pricing and its greeks generation as shown in Fig 2 below. The user needs to type the 6 inputs as shown in Fig 2 and hit enter.

```
What you would like to generate: Enter P for Option Pricing / Enter V for Vol Smile Fitting
P

******************************************

 Enter Option Pricing Parameters

******************************************


Enter Spot Price:
100

Enter Strike of Option:
120

Enter Risk free Interest Rate (in fraction):
0.05

Enter Volatility of Underlying (in fraction):
0.20

Enter Option Expiry (in years):
2

Which Option do you want to price: Enter C for Call/ P for Put
C
```

Fig 2: User providing inputs for Option pricing

The code calculates the option premium and option greeks based on user inputs and prints on console as shown in Fig 3 below. The program ends at this stage.

```
****************************************************
Underlying Price:              100
Strike Price:                  120
Risk-Free Rate:                0.05
Volatility:                    0.2
Maturity:                      2
Option Type:                   C
Option Premium:               $7.92821


Option Greeks are as below

******************************
Delta:              0.440529
Gamma:              0.015575
Vega:               0.557909
Rho:                0.722493
Theta:              -0.0125912

--------------------------------
Process exited after 13.62 seconds with return value 0
Press any key to continue . . .
```

Fig 3: Output of Module 1 showing Option Premium and Option Greeks based on user input

3.2.2. Module 2: Volatility Smile Fitting

Once the user selects "V" as input in Fig 1, the program jumps to Module 2 asking user inputs for the Volatility Strike grid as shown in Fig 4 below. The user needs to input the ATM Strike, Highest and Lowest Strike and the corresponding implied volatility for each strike and hit enter. It is advised that user input all the above parameters in multiples of 5.

```
What you would like to generate: Enter P for Option Pricing / Enter V for Vol Smile Fitting
V
Enter ATM Strike
100
Enter Lowest Strike for the Strike Range for the Smile Grid. Please enter multiples of 5.
80
Enter Highest Strike for the Strike Range for the Smile Grid. Please enter multiples of 5.
120

Enter Implied Vol in % for the Strike  : 80
20

Enter Implied Vol in % for the Strike  : 85
17

Enter Implied Vol in % for the Strike  : 90
15

Enter Implied Vol in % for the Strike  : 95
12

Enter Implied Vol in % for the Strike  : 100
11

Enter Implied Vol in % for the Strike  : 105
12

Enter Implied Vol in % for the Strike  : 110
14

Enter Implied Vol in % for the Strike  : 115
16

Enter Implied Vol in % for the Strike  : 120
18
```

Fig 4: User inputting Implied Volatility against Strike Grid in Module 2

Once inputs as shown in Fig 4 are entered, the program invokes the Volatility_Smile_Fitting module which produces the results as shown in Fig 5 below.

```
*******************************************************
The best fit estimates for the parameters are the below
*******************************************************
Convexity (a.k.a VoV)                  :1.82
Skew (a.k.a. Correlation)              :-3.7
Constant                               :2
Mean Squared Standard error            :0.00148681

*********************************************************************************

Relative Strike        Implied Vol             Fitted_Vol
*********************************************************************************
     0.8                 0.2                     0.2048
     0.85                 0.17                     0.16995
     0.9                 0.15                     0.1442
     0.95                 0.12                     0.12755
     1                 0.11                     0.12
     1.05                 0.12                     0.12155
     1.1                 0.14                     0.1322
     1.15                 0.16                     0.15195
     1.2                 0.18                     0.1808

--------------------------------
Process exited after 538.4 seconds with return value 0
Press any key to continue . . .
```

Fig 5: Output of Module 2 showing calibrated smile parameters and comparison between implied volatility and fitted volatility

4.0 Test Cases

The following test cases were executed to show that the program achieves its intended purpose and gives sensible output.

Case I: Applicable to Module 1 i.e., Option Pricing and Greek Generation.

The output of Module 1 which is option pricing and Greek generation was compared to similar metrics calculated using a web based tool (http://www.option-price.com/index.php). The results of the two closely match. Here is an example showing comparison for the following Options along with screenshot of results obtained from option-price.com website shown in Fig 6.

Table 1: External Source of Option Price and Greeks

Trade Economics	Option Premium ($)	Delta ($/unit change in Underlying)	Gamma ($/unit change in Delta)	Theta ($/unit change in time to expiry)	Rho ($/unit change in RFR)	Vega ($/unit change in Volatility)
Option: Call Strike: $120 Spot: $100 Expiry: 2 yr RFR: 5% Vol: 20%	7.928 (External) 7.928 (Module 1)	0.441 (External) 0.441 (Module 1)	0.014 (External) 0.014 (Module 1)	-0.013 (External) -0.013 (Module 1)	0.722 (External) 0.722 (Module 1)	0.558 (External) 0.558 (Module 1)
Option: Put Strike: $120 Spot: $100 Expiry: 2 yr RFR: 5% Vol: 20%	16.509 (External) 16.509 (Module 1)	-0.559 (External) -0.559 (Module 1)	0.014 (External) 0.014 (Module 1)	0.002 (External) 0.002 (Module 1)	-1.449 (External) -1.449 (Module 1)	0.558 (External) 0.558 (Module 1)

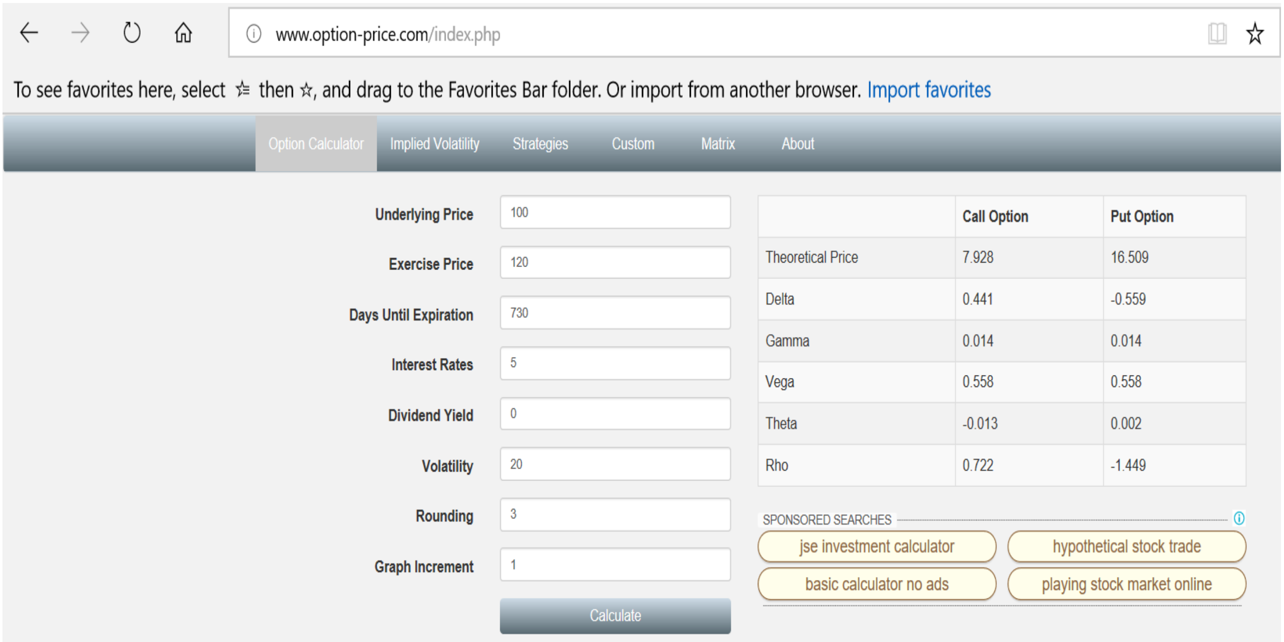

Fig 6: Screenshot of results from option-price.com website

Case II: Applicable to Module 2 i.e., Volatility Smile Fitting

Case II a: Flat Volatility Smile test

In this case, the users inputs a flat implied volatility smile surface across a range of strikes. The output is expected to be a flat fitted volatility surface as well. This is indeed the case as shown in the screenshot below.

```
*******************************************************
The best fit estimates for the parameters are the below
*******************************************************
Convexity (a.k.a VoV)                 :-1.68889e-013
Skew (a.k.a. Correlation)             :-1.87905e-014
Constant                              :0.2
Mean Squared Standard error           :4.40196e-014

***************************************************************************

Relative Strike        Implied Vol             Fitted_Vol
***************************************************************************
     0.8                0.2                      0.2
     0.85                0.2                      0.2
     0.9                0.2                      0.2
     0.95                0.2                      0.2
     1                0.2                      0.2
     1.05                0.2                      0.2
     1.1                0.2                      0.2
     1.15                0.2                      0.2
     1.2                0.2                      0.2

-------------------------------
```

Fig 7: Fitted Volatility Surface is flat which is same as Flat Implied Volatility Surface

<u>Case II b: Calibrated Smile Parameters are close to similar exercise done in MS Excel</u>

```
********************************************************
The best fit estimates for the parameters are the below
********************************************************
Convexity (a.k.a VoV)                   :1.82
Skew (a.k.a. Correlation)               :-3.7
Constant                                :2
Mean Squared Standard error             :0.00138889

********************************************************************

Relative Strike        Implied Vol             Fitted_Vol
********************************************************************
     0.8                0.2                      0.2048
     0.85                0.17                      0.16995
     0.9                0.15                     0.1442
     0.95                0.13                      0.12755
     1                0.11                     0.12
     1.05                0.12                      0.12155
     1.1                0.14                     0.1322
     1.15                0.16                      0.15195
     1.2                0.18                     0.1808

--------------------------------
```

Fig 8: Comparison of Fitted Volatility versus User Inputted Implied Volatility

	Strike	Strike/ ATM	**Implied Volatility** (in fraction)
	80	0.8	0.2
	85	0.85	0.17
	90	0.9	0.15
	95	0.95	0.13
ATM	100	1	0.11
	105	1.05	0.12
	110	1.1	0.14
	115	1.15	0.16
	120	1.2	0.18

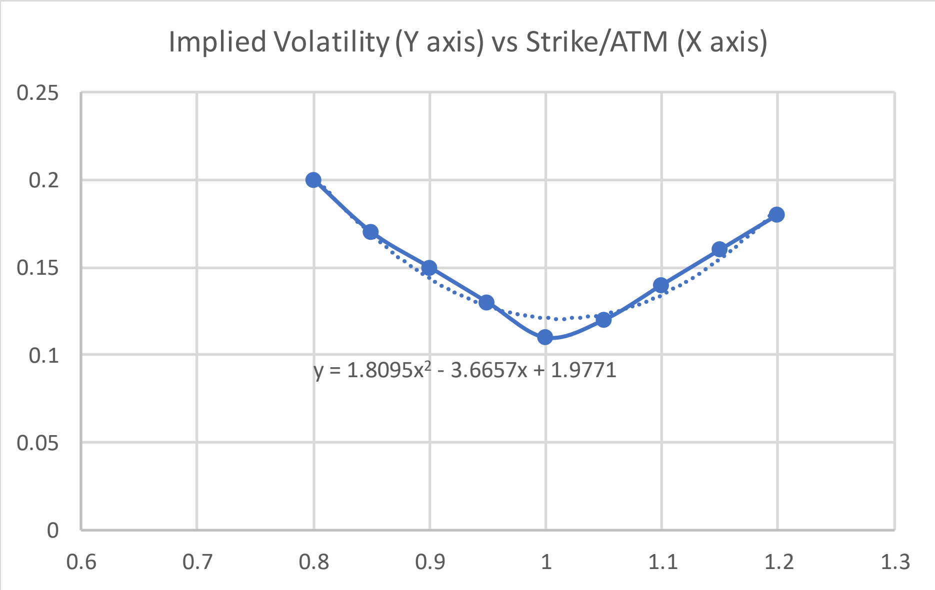

Fig 9: Smile Parameters obtained from Volatility Surface Fitting in MS Excel

MS Excel gives the following Smile Parameters:

Convexity : 1.8095

Skew : -3.6657

Constant : 1.9771

Which compares closely to ones obtained from Module 2

Convexity : 1.82

Skew : -3.7

Constant : 2.0

The difference can be further reduced if the tweak amount is reduced further for all the three smile parameters to something like 0.0001 in Module 2. However, this will increase the computation significantly. The user depending upon his acceptable margin of error can change the tweak amount to arrive at better fits.

5.0 References

1. Option Pricing Website: http://www.option-price.com/index.php
2. Quantlib Website: https://www.quantlib.org/
3. Black Scholes Pricing: https://www.macroption.com/black-scholes/