AMA 3021: Computational Finance

Business Project 2

Black-Scholes Solution by Finite Differences

Fynn McKay

(40099355)

Submission: 17th Dec 2015

School of Mathematics and Physics

Contents

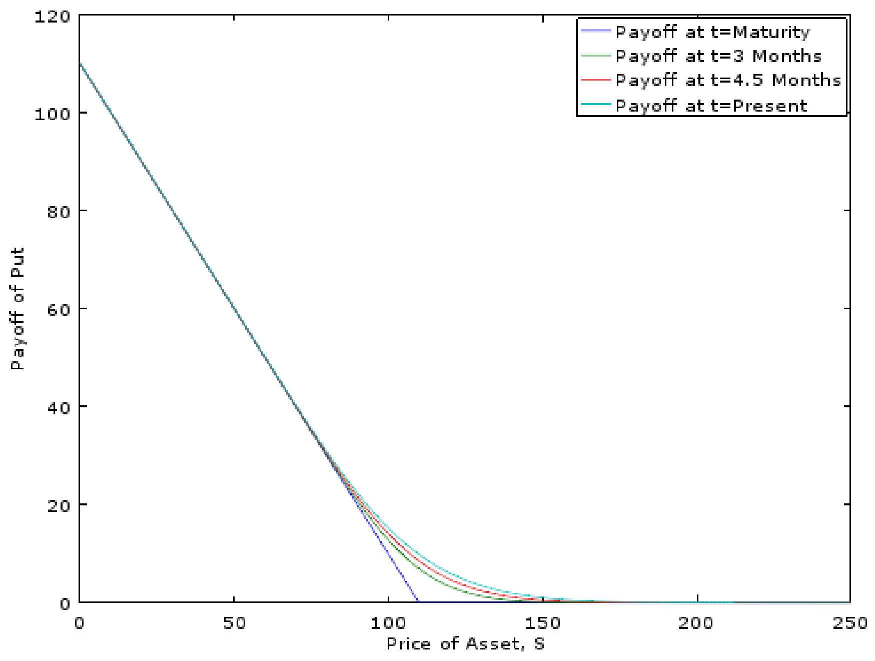
1. Executive Summary

Write a program (in MATLAB or C/C++) to calculate the Put option price p given data for the strike price X, risk-free interest rate r, volatility σ and time to expiry T.

This report was commissioned to solve the Black-Scholes Equation by iterative Finite Difference Methods in order to attain the fair price of an IBM European Put Option. The results were as follows;

Figure 1: Table of Put Option Values at Varying Strike Prices and Plot of Put Payoffs at different Time Intervals up to Maturity (For a Strike of X=$110)

Strike Price $X	European Put Option Price $P(t,S)
110	0.19236
120	0.96259
130	3.1222
140	7.3562
150	13.753
160	21.828
170	30.924
180	40.531



From the calculation of the above data, we have been able to conclude that as the strike price exceeded the original share price ($138.50), the payoff of the Put option increased as would be expected of a Put option at any point in time. More interestingly we have been able to highlight that as time moved further from maturity the Payoff curve became more rounded around the strike price representing the inherent uncertainty of share price fluctuations as we iterate back in time using Finite Differences and hence the resultant uncertainty in the payoff of the Put option at any point before maturity.

2. Introduction

2.1. Question

Write a program (in MATLAB or C/C++) to calculate the Put option price p given data for the strike price X, risk-free interest rate r, volatility σ and time to expiry T.

Do this by writing the Black-Scholes Equation as a finite-difference equation and then integrating backwards in time from the expiry date to find the Put price, given the current spot price.

Use the following IBM Put option figures to do so;

Current IBM spot price (As of November 28th 2015): S_0=£138.50
Risk-free interest rate: r=1.0% per Annum
Put option expiry: July 15th 2016
Current Volatility of IMB stock: σ=16%

2.2. Background Information

Before delving into the intricacies of the Black-Scholes (B-S) Formula and the iterative numerical methods required to solve it, it is important to add context to our problem to allow for a better grasp of the issue at hand.

2.2.1. Derivatives

A derivative is the name given to the group of securities which derive their price from the performance of one or more underlying assets. The process involves two or more parties entering into a contract that agrees terms for the potential exchanging of assets at or before some maturity T, depending on the fluctuations of the underlying assets price.

Derivatives themselves come in varying forms and levels of complexity however for the purpose of this project we shall be focusing on European/Vanilla options, more specifically longing European Put options. European options differ from their American/Asian counterparts in that they can only be exercised at maturity and hence tend to trade at a discount relative to their more flexible counterparts.

"Longing European Put options" hence refers to the process of buying the right (but not the obligation) to sell the underlying asset to a counterparty, for an agreed strike price X, at some point in the future T. Therefore by denoting the underlying asset's final price as S_t, we can state that at a maturity T, the payoff of a European option is described by;

$$\text{Payoff} = \begin{cases} \text{Max}[S_t - X, 0] & \text{for a Long Call Option} \\ \text{Max}[X - S_t, 0] & \text{for a Long Put Option} \end{cases} \tag{1}$$

2.2.2. Black-Scholes Origins

"Up until the time when Black and Scholes came up with their insight, the options world was full of uncertainty and risk - uncontrollable and unanalysable. Then, in a moment of tremendous clarity, Black and Scholes realised that two risky positions taken together could effectively eliminate risk itself. In that moment of brilliance they created, arguably, the most important equation in the history of modern day finance." **Stan Jonas, MD FIMAT USA**

Published in 1973, "The Pricing of Options and Corporate Liabilities"[1] is the crowning achievement of both Fischer Black and Myron Scholes and was the basis on which Myron Scholes and Robert Merton were awarded the 1997 Nobel Prize for Economic Sciences.

In their paper they introduced a Mathematical model that allowed for calculation of the fair price of a European option at any point up to and including its maturity. Using a no arbitrage assumption they derived the following PDE in terms of time to maturity t, and the underlying asset price S, which models the progression of a European option's price V, prior to expiration;

(2) $$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

Notably, due to the simplicity of European options we can derive analytical, Closed Form Solutions for the above Black-Scholes PDE, something which cannot be done for more exotic, complex options (American/Asian).

The following is the Closed Form Solution for a European Put option;[1]

(3) $$P(t,S) = Xe^{-rt}N(-d_2) - SN(-d_1)$$

where N(.) is the cumulative distribution function of the standard normal distribution and d1 and d2 are variables which will be described later in our code.

2.2.3. Assumptions of Black-Scholes[2]

Whilst some (or arguably all) of the following assumptions can be disputed to some degree, they are still fundamental in our ability to understand the basics of the Black-Scholes Model and will be important to know when we later look at the derivation of the model itself.

They are as follows;

- No arbitrage opportunities exist within the market
- Delta Hedging is done continuously
- Assets award no dividends and no transaction costs exist
- Risk-free interest rate (r) is a function of time and individuals are free to lend and borrow at the rate r as they please
- Returns on underlying stocks are log-normally distributed
- Markets are efficient

2.3. Other Applications[2]

Whilst we will mainly be looking at how to apply the Black-Scholes PDE to Vanilla options, the equation has a number of other applications outside of options pricing.

Examples of such additional applications include;

- Guarantees and Insurance contracts: Both give the right (but not obligation) to be exploited under certain conditions. Hence those who purchase or are in possession of either contracts could be considered to hold a kind of option, which Black-Merton-Scholes' methodology has allowed the pricing of in many cases.
- Investment Decision Flexibility: The flexibility of energy suppliers (in commodities such as coal and oil) to stop and start production in accordance with market demand is another example of how B-S can be applied elsewhere. For certain investors the ability to accurately price said flexibility is fundamental in their investment strategy and B-S has allowed a platform from which such prices can be derived.

Notably, the B-S Model can also still be used in the pricing of more exotic options such as those of the aforementioned Asian and American variety however, due to their ability to be exercised at any point in time, the coding and iterative requirements for the calculation of their fair price is considerably more demanding.

3. Mathematical Theory and Numerical Methods

3.1. Black-Scholes Model

In the following section we will look at the derivation of the Black-Scholes Equation based on the assumptions made earlier.

3.1.1. Derivation of the Black-Scholes Model[3]

We will assume that the change in a stock's price over time can be modelled as a stochastic differential equation i.e. Geometric Brownian Motion[4];

(4)
$$dS_t = \mu_d S_t dt + \sigma S_t Z\sqrt{dt}$$

where μ_d denotes drift rate, σ denotes volatility of the asset and Z denotes a normally distributed random variable.

For ease of further calculation we shall replace our stochastic element $Z\sqrt{dt}$ with a single variable dW such that;

(5)
$$dS_t = \mu_d S_t dt + \sigma S_t dW$$

Hence if we look to estimate the value of an option at any time t before maturity, we must understand how V, the price of an option, evolves as a function of S and t.

As V is a time-dependent function of a stochastic process we can hence use Itô's lemma to find its differential;

(6)
$$dV = \left(\mu S \frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right) dt + \sigma S \frac{\partial V}{\partial S} dW$$

Now consider constructing a portfolio consisting of 2 positions at time t: Short one (unspecified) option and long $\frac{\partial V}{\partial S}$ shares in the underlying security. This position is described as a delta-hedge portfolio and is used to reduce the randomness in our PDE to 0. It can be written;

(7)
$$\Pi = \frac{\partial V}{\partial S} S - V$$

By considering a small change in time, the resultant profit/loss due to changes in our underlying security is described as;

(8)
$$\Delta\Pi = \frac{\partial V}{\partial S} \Delta S - \Delta V$$

Similarly we can discretize our previous 2 equations **(5)** and **(6)** by replacing our differentials with deltas, hence we write;

(9)
$$\varDelta S_t = \mu_d S_t \varDelta t + \sigma S_t \varDelta W$$
$$\varDelta V = \left(\mu S \frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\varDelta t + \sigma S \frac{\partial V}{\partial S}\varDelta W$$

Subbing into **(8)** our expression becomes;

(10)
$$\varDelta \Pi = \frac{\partial V}{\partial S}(\mu_d S_t \varDelta t + \sigma S_t \varDelta W) - \left(\mu S \frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\varDelta t + \sigma S \frac{\partial V}{\partial S}\varDelta W$$

Which reduces down to;

(11)
$$\varDelta \Pi = \left(-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\varDelta t$$

At this point we can notice 2 things. Our $\varDelta W$ has cancelled out and hence effectively removed the uncertainty associated within our portfolio. This in turn implies that our portfolio must have a return equal to other riskless instruments on the market otherwise an arbitrage situation would arise, violating one of our initial assumptions.

Hence assuming a risk-free rate of return r over a time period $\varDelta t$, we can write;

(12)
$$\varDelta \Pi = r\Pi\varDelta t$$

Therefore if we fill in equations **(7)** and **(11)** we have;

(13)
$$\left(-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\varDelta t = r\left(\frac{\partial V}{\partial S}S - V\right)\varDelta t$$

Finally by simplifying we arrive at our previously stated Black-Scholes PDE;

(14)
$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

3.2 Finite Differences Method[4]

The finite differences method is a form of numerical analysis used to iteratively solve certain forms of ordinary and partial differential equations. We will hence be using it in order to solve the B-S PDE for the fair price of a European Put option.

We begin by first creating a grid formed from discretizing both our S and t values, then define the boundary conditions of a European Put to later be implemented in the code. From here we then look to use Taylor approximations for the derivatives in our model so that we are able to iteratively solve for the price of our European Put in MATLAB and finally we redefine the B-S Model in terms of our newly derived Taylor approximations.

3.2.1. Formation of Finite-Difference Grid

Our first step involves splitting both our time and price intervals such that;

(15) **Discretize Time:**

$$0 \leq t \leq T$$

$$t = 0, t_1, \ . \ . \ . \ . \ , t_i, \ . \ . \ . \ . \ , t_N$$

$$t_i = i\delta t \quad , \quad i = 0,1,2, \ . \ . \ . \ . \ , N$$

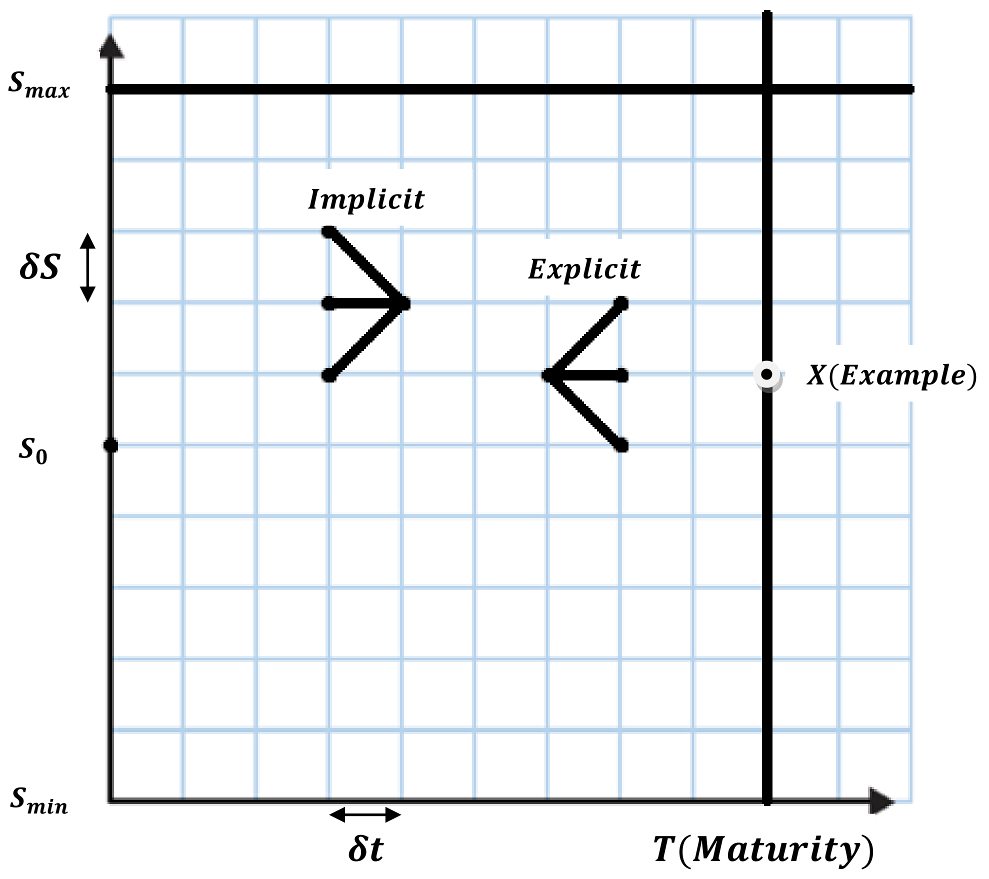
(16) **Discretize Price:**

$$S_{min} \leq S \leq S_{max}$$

$$S = 0, S_1, \ . \ . \ . \ . \ , S_{j-1}, S_j, S_{j+1}, \ . \ . \ . \ . \ , S_M$$

$$S_j = j\delta S \quad , \quad j = 0,1,2, \ . \ . \ . \ . \ , M$$

Figure 2: Example of a Finite-Difference Grid



By dividing our price and time into sufficiently small intervals we can form the above grid which will be combined with our initial boundary conditions and our Taylor approximations to integrate backwards over in order to iteratively solve our differential equation.

3.2.2. Put Boundary Conditions

There are a number of boundary conditions for European Put options which we must consider in order to allow ourselves the ability to code our Finite Solutions Algorithm in the next section. I will provide no derivation of said boundaries as they pertain more to common sense than mathematical theory.

Instead, I will simply state the boundaries below with a small explanation;

(17)

- $P(T,S) = Max[X - S_t, 0]$ (as previously stated in **(1)**)
- $P(t,0) = Xe^{-rt}$: Max[X-0, 0] = X for X>0. Hence, at any time t, to attain the present value of X we discount it continuously by discount factor e^{-rt} .
- $P(t, S_t \to \infty) = 0$: As S_t tends to infinity our strike price X becomes comparatively small hence tending towards Max[-S_t,0]=0 since negative share prices are impossible.

3.2.3. Framing our Intervals and Boundaries is terms of grid notation

We will first reiterate our grid and boundary conditions;

(18)
$$Discretised\ Time : \{0, \delta t, 2\delta t, \ \ . \ . \ . \ . \ , N\delta t\} \quad where \quad N\delta t = T$$
$$Discretised\ Price : \{0, \delta S, 2\delta S, \ \ . \ . \ . \ . \ , M\delta S\} \quad where \quad M\delta S = S_{max}$$

We will also denote the value of our Put function as the above variables change;

(19)
$$P_{i,j} = P(i\delta t, j\delta S) \quad where \quad i = 0,1, \ \ . \ . \ . \ . \ , N \quad and \quad j = 0,1, \ \ . \ . \ . \ . \ , M$$

And hence our previous boundary conditions (17) expressed in terms of our grid are;

(20)
$$P_{i,0} = Xe^{-r(N-i)\delta t} \quad where \quad i = 0,1, \ \ . \ . \ . \ . \ , N$$
$$P_{i,M} = 0 \quad where \quad i = 0,1, \ \ . \ . \ . \ . \ , N$$
$$P_{N,j} = Max[X - (j\delta S), 0] \quad where \quad j = 0,1, \ \ . \ . \ . \ . \ , M$$

3.2.4. Finite Differences PDE approximations

In our problem we are asked to solve the B-S PDE which we have chosen to do implicitly. We hence require our above boundary conditions as well as the following derivative terms in order to solve our problem backwards in time, starting at maturity and working backwards toward present time.

This is done by replacing the partial derivatives in our PDE with Taylor expansion approximations near the points of interest.

For example we can derive the Forward (and Backward) Difference approximation's to 1st Order DE by first considering the Taylor Approximation;

(21)
$$\frac{\partial P_{i,j}}{\partial t} = \frac{P_{i+1,j} - P_{i,j}}{\delta t} - \frac{\delta t}{2!}\frac{\partial^2 P_{i,j}}{\partial t^2} + \ \ . \ . \ .$$

And then by gathering and later discarding our higher order terms giving us;

(22) $$Forward : \frac{\partial P_{i,j}}{\partial t} \approx \frac{P_{i+1,j} - P_{i,j}}{\delta t} + O(\delta t)$$

(23) $$Backward : \frac{\partial P_{i,j}}{\partial t} \approx \frac{P_{i,j} - P_{i-1,j}}{\delta t} + O(\delta t)$$

By a similar process we can also derive our Central Difference approximation to 1st Order DE by again looking at the Taylor expansions;

(24) $$\frac{\partial P_{i,j+1}}{\partial S} = P_{i,j} + \frac{\delta S}{1!}\frac{\partial P_{i,j}}{\partial S} + \frac{(\delta S)^2}{2!}\frac{\partial^2 P_{i,j}}{\partial S^2} + \ . \ . \ .$$

(25) $$\frac{\partial P_{i,j-1}}{\partial S} = P_{i,j} - \frac{\delta S}{1!}\frac{\partial P_{i,j}}{\partial S} + \frac{(\delta S)^2}{2!}\frac{\partial^2 P_{i,j}}{\partial S^2} - \ . \ . \ .$$

Then subtracting and later discarding higher order terms producing;

(26) $$Central : \frac{\partial P_{i,j}}{\partial S} \approx \frac{P_{i,j+1} - P_{i,j-1}}{2\delta S} + O(\delta S)^2$$

And hence by a similar process as the above derivation only this time by adding, we can derive the 2nd Order Central Difference approximation giving us;

(27) $$Central : \frac{\partial^2 P_{i,j}}{\partial S^2} \approx \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{(\delta S)^2} + O(\delta S)^2$$

Equipped now with the approximations for our derivatives, we can rewrite the B-S equation **(14)** in terms of our newly derived backwards and central difference expressions;

(28) $$\frac{P_{i+1,j} - P_{i,j}}{\delta t} + \frac{1}{2}\sigma^2 (j\delta S)^2 \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{(\delta S)^2} + r(j\delta S)\frac{P_{i,j+1} - P_{i,j-1}}{2\delta S} - rP_{i,j} = 0$$

Finally, by algebraic rearranging, we end up with the formula we shall program in the next section to attain our results;

(29) $$P_{i+1,j} = \alpha_j P_{i,j-1} + \beta_j P_{i,j} + \gamma_j P_{i,j+1}$$

$$where \quad \alpha_j = \frac{1}{2}\delta t(rj - \sigma^2 j^2) \quad \beta_j = 1 + (\sigma^2 j^2 + r)\delta t \quad \gamma_j = -\frac{1}{2}\delta t(rj + \sigma^2 j^2)$$

With this expression, along with our aforementioned grid and boundary conditions, we are now equipped to iteratively solve the Black-Scholes equation by way of the Implicit Finite-Differencing method using the previously stated information on IBM Put options.

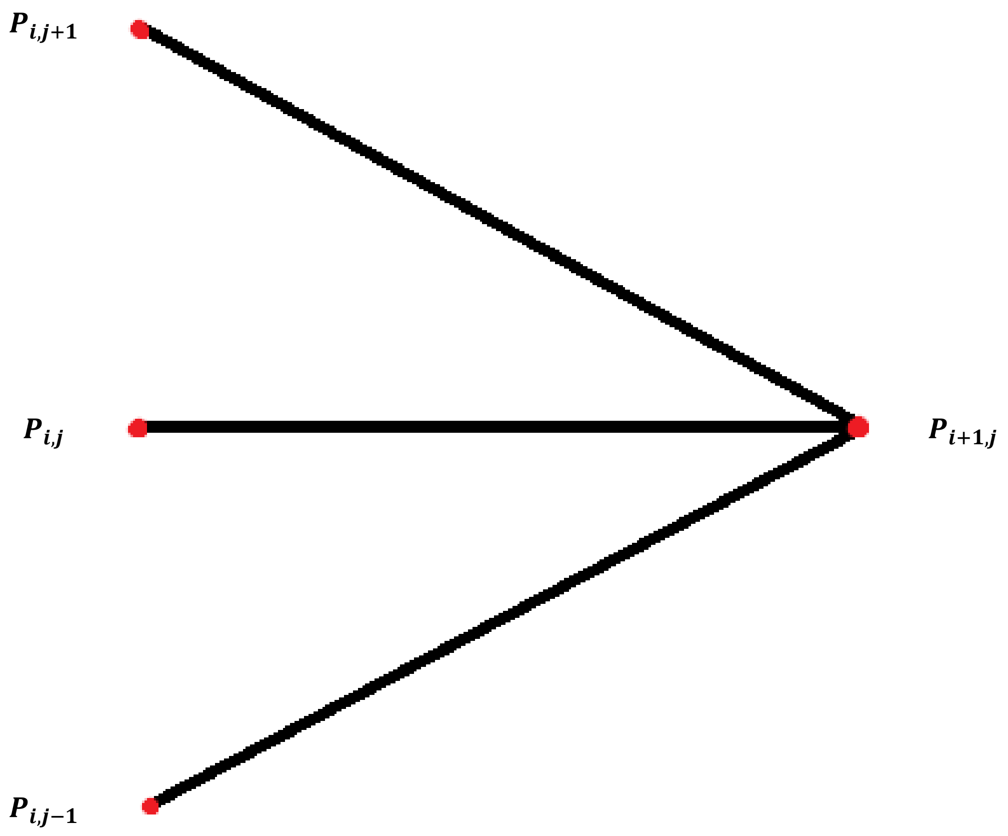
Worth noting is that for each time step we will have a set of M-1 equations with M-1 unknowns to solve for, as will be seen in the following matrices.

3.3 Implementation of Matrix Notation

Since we know that by definition the Implicit Difference method solves our set of unknowns backwards in time, we can see that equation (29) essentially states that we will implicitly calculate 3 unknowns further back in time using one known value in the current time frame.

This is easier understood by considering the following diagram and Matrix notation.

Figure 3: Implicit Finite Difference Method as a Trinomial Tree[5]



Due to the iterative intensity of the Implicit Finite Differences method, the use of some form of programming is a fundamental necessity to finding a correct solution to our problem.

Hence for the purposes of implementing equation **(29)** into MATLAB we shall rewrite it in terms of the following Matrix Notation;

(30) $$AP_i = P_{i+1} - C_i$$

Where P$_i$ represents a matrix of Put solutions at time *iδt;*

(31) $$P_i = \begin{pmatrix} P_{i,1} \\ P_{i,2} \\ \vdots \\ P_{i,M-2} \\ P_{i,M-1} \end{pmatrix}$$

C represents a matrix containing 2 edge conditions, notably where the second will always equal 0;

(32) $$C_i = \begin{pmatrix} \alpha_1 P_{i,0} \\ 0 \\ \vdots \\ 0 \\ \gamma_{M-1} P_{i,M} \end{pmatrix}$$

Finally A represents a diagonal, square (by necessity) matrix of constants from equation (29) with dimensions (M-1)x(M-1);

(33) $$A= \begin{pmatrix} \beta_1 & \gamma_1 & 0 & \cdots & 0 & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \cdots & 0 & 0 \\ 0 & \alpha_3 & \beta_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{M-1} & \beta_{M-1} \end{pmatrix}$$

And hence following the formation of these Matrices we will invert our Matrix of constants A to solve for our Put price backwards in time from P$_{i+1}$;

(34) $$P_i = A^{-1}(P_{i+1} - C_i)$$

3.4 Algorithm and Code

The methodology behind the following code is as follows;

- Set up initial variable values as well as parameters necessary for forming our grid of Put prices. Arbitrarily choose a sensible number of price steps and time steps over which we seek to integrate. Create 2 variables ‘Sgrid’ and ‘Tgrid’ which mirror the function of equations **(18)** noting that ‘Tgrid’ will move backwards in time starting at maturity as previously detailed.
- Create matrix of different Put prices at varying S and t i.e. ‘solngrid’, initialise as a matrix of zero’s and add in boundary conditions from equations **(20)**, setting values for the initial row of the matrix (S=0), final column of the matrix (t=T i.e. maturity) and the final row of the matrix(S=Smax hence P(t,S)=0). Also code variables for α_j, β_j and γ_j as detailed in **(29)**.
- Create the tri-diagonal matrix A **(33)** by creating 3 diagonalised matrices of size (M-1)x(M-1), with alphas offset down 1 position, betas taking the leading diagonal and gamma’s being offset up one position as seen above, and sum them to form Matrix A.

- Finally create Matrix C, ensuring it has the number of rows equal to the number of columns in A and implement in into our for loop. The for loop, starting at maturity and S=0, will use values of $P_{i+1,j}$ to calculate values for $P_{i,j}$ for all possible prices before “taking a step” backwards in time and repeating this process. By doing this for all values of t up to current time we will have created a grid of all possible Put option prices, with the first column of the “solngrid” representing what fair price of P(t,S) should be at the current time for all S.

```
%% PLEASE NOTE: Indexes will be displaced by one compared to the above sets
of matrices due to MATLAB conventions i.e. a_1=a(2) for j=0:M etc.
S=138.50;                        %% Initial Share Price
r=0.01;                          %% Risk-Free Interest Rate
sigma=0.16;                      %% Volatility
X=110;                           %% Example Strike Price
T=0.632876712;                   %% Years until Expiry

%% Grid Parameters %%

M=250;                           %% Asset Grid points
N=2310;                          %% Time Grid points
Szero=0;                         %% Specify Minimum Share Price
Smax=250;                        %% Specify Maximum Share Price

%% Grid Setup and Boundary Conditions %%
j=0:M;                               %% Set up j vector
dt=T/N;                              %% Construct time step
ds=Smax/M;                           %% Construct price step
solngrid=zeros(M+1,N+1);             %% Initialise Grid i.e. 251x2311 matrix
Sgrid=0:ds:Smax;                     %% Positive, equally spaced price steps
                                     %% i.e. 251 elements
Tgrid=T:-dt:0;                       %% Negative(Backward), equally spaced
                                     %% time steps i.e. 2311 elements
solngrid(1,:)=X*exp(-r*Tgrid);       %% Boundary Condition : Price=0
solngrid(:,end)=max(X-Sgrid,0);      %% Boundary Condition : Payoff of Put
solngrid(end,:)=0;                   %% Boundary Condition : Price tending
                                     %% to "infinity"

alpha=(1/2)*dt*(r*j-sigma^2*j.^2);     %(29) function alpha
beta=1+(sigma^2*j.^2 +r)*dt;           %(29) function beta
gamma=-(1/2)*dt*(r*j+sigma^2*j.^2);    %(29) function gamma

%% Construction of Soln Matrix %%

A=diag(alpha(3:M),-1)+diag(beta(2:M))+diag(gamma(2:M-1),1);
% Here we create Matrix (33) with betas on the leading diagonal, alphas
% offset down(-1) and gammas offset up (+1)
Ainv=inv(A);                    % Create inverse of A to test stability
normi=norm(Ainv,inf);           % Stability Test
C=zeros(size(A,2),1);  %Create matrix of 0's w/1 column and number of rows
                        =no. of columns in A
for i=N:-1:1 %For loop solves our M-1 eqns for every time grid point N=2310
    C(1)=alpha(2)*solngrid(1,i); % first element of C
    C(end) = gamma(end)*solngrid(end,i); % Will always be zero as
                                          previously stated hence irrelevant
    solngrid(2:M,i)=A\(solngrid(2:M,i+1)-C); %Inverted matrix soln for P_i
end
EuropeanPutOptionPrice=interp1(Sgrid,solngrid(:,1),S) %Finally interpolate
between ds intervals to get exact price for any possible S value 0 upto 250
```

4. Results and Discussion

In this section we will look to analyse the figures that our code has produced for the fair price of an IBM European Put Option as well as consider how these figures compare against the markets opinion of what a fair price should be.

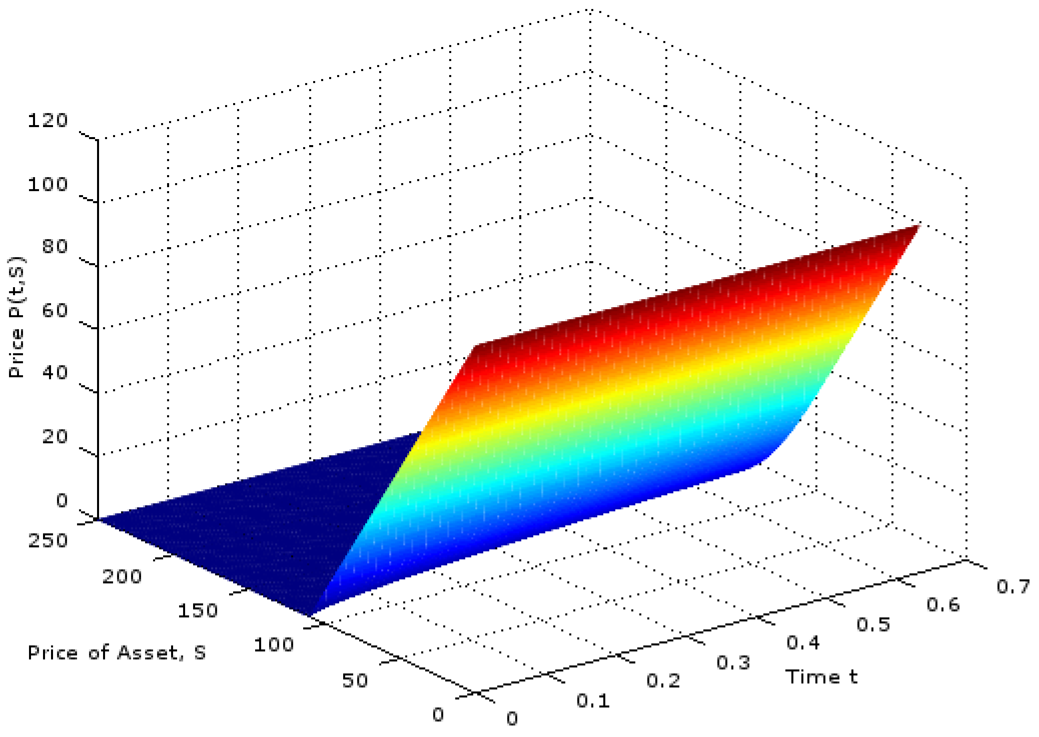
We will also look to consider where some error may lie within our approximation and how we might/have combated .

4.1 Tabulated Figures and Graphs

Given the initial condition's laid out for our task, we can now use our code to calculate the fair price of an IBM European Put option P(t,S), over a variety of different strike prices. In the displayed code I opted to show the algorithm for a strike price X=$110, however any of the other given strike price's would be equally valid.

Below is a plot of the progression of P(t,S) as time and share price vary across our predesignated ranges, again with strike price X=$110. Refer to **Executive Summary** or **Appendix (3)** for cross-plots.

Figure 4: Progression of European Put Option over t and S (Strike X=$110)



```
mesh(Tgrid,Sgrid,solngrid)      %3-D plot of Solution Grid
xlabel("Time t")                %Plot labels
ylabel("Price of Asset, S")
zlabel("Price P(t,S)")
```

Furthermore, below is a table of tabulated Put values at varying strike prices computed using the given data in our task sheet;

Table 1: European Put option prices for IBM Corp. shares expiring 15/07/ 2016

Strike Price $X	European Put Option Price $P(t,S)
110	0.19236
120	0.96259
130	3.1222
140	7.3562
150	13.753
160	21.828
170	30.924
180	40.531

A number of observations can be made on both **Figure 4** and **Table 1** which verify that our solution for the B-S PDE is at least within the realms of possibility.

Figure 4 initially takes the form of a long Put payoff plot (see **Appendix (2)**) at expiry and gradually deforms as we use Taylor approximations to plot further back in time and hence introduce higher levels of uncertainty regarding share price, as would be expected.

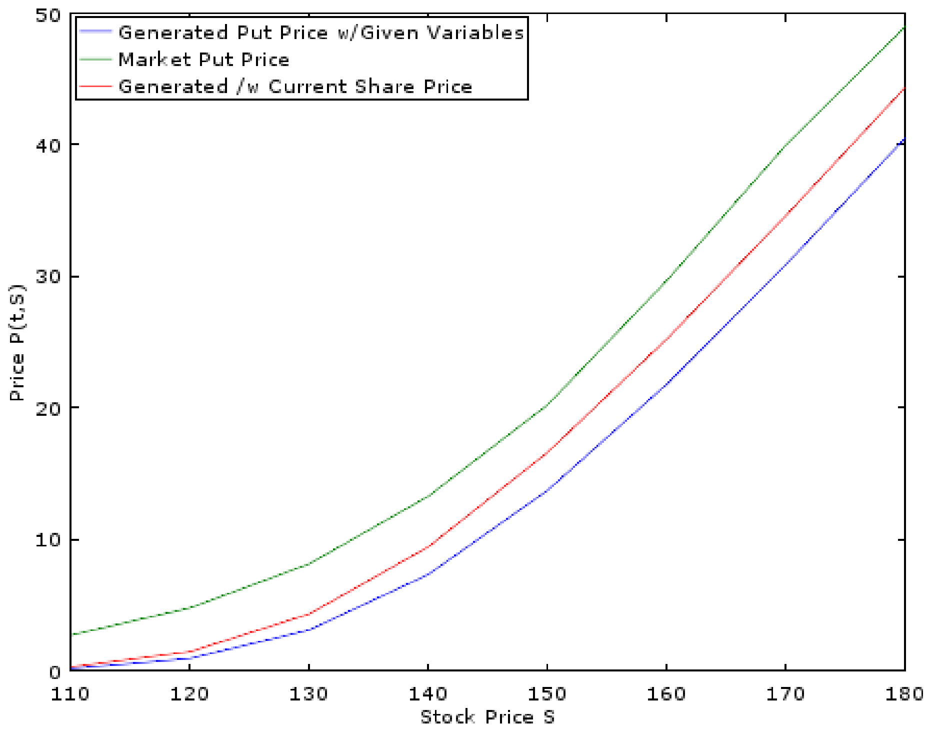
Additionally, the values for P(t,S) in **Table 1** begin small but get exponentially bigger as our strike price rises in value. Logically, the option to sell an asset in the short term future for a value considerably less than it is currently worth (given that the asset has proven to be not particularly risky (σ=16%) and is associated with a blue chip company) is not something investors would be willing to spend much money on hence driving P(t,S) down. By a similar form of logic, the option to sell the aforementioned asset for considerably more than it is currently worth is something that investors would be willing to pay money for hence driving P(t,S) up. Both of these facts are reflected in the above table.

Furthermore if we compare our figures (using current and given Share Price) to the "Market's opinion" of what a fair value of an IBM European Put option is we get the following table and graph;

Table 2: Comparison of Calculated P(t,S) vs. Market P(t,S)

Strike Price $X	European Put Option Price $P(t,S)	Market Put Option Price $P(t,S)
110	0.19236	2.73
120	0.96259	4.80
130	3.1222	8.15
140	7.3562	13.30
150	13.753	20.25
160	21.828	29.70
170	30.924	40.00
180	40.531	49.00

Figure 5: Comparison of Generated Put Price for the given Share Price ($138.50)/Current Share Price ($134.57) and Market Put Price



We can see that in spite of what seems to be some kind of anomaly that differentiates the Markets opinion from our iterative approach, the general trend of our prices are virtually identical to that of the Markets, again lending credence to the assumption that our model is at least somewhat correct. It is worth noting, however, that this is somewhat surprising given that all 3 of our estimations (Calculated P(t,S), "True" P(t,S) and Outside Source P(t,S)) in the next section all agree, suggesting that some alteration of the B-S method of option pricing is used in the Markets. We will discuss this further in our conclusion. (NB Appendix **(1)** attempts to recreate above Market Put Price/Conditions)

4.2 Numerical Accuracy[6]

As with any form of numerical analysis, it is important to consider the accuracy of the algorithms we have coded and the data we have produced.

In particular we should be careful to ensure that our algorithms are stable, that error is kept to a minimum and by extension that our output is converging.

4.2.1. Stability

Standard matrix algebra dictates that for an equation of the form of **(30)**, specifically for a matrix such as A, stability is achieved if and only if;

(35) $$||A^{-1}||_{\infty} \leq 1$$

where $||A||_{\infty}$ denotes the infinity norm.

For any MxN matrix, the infinity norm is defined as;

(36)
$$||A||_\infty = \max\left(\sum_{j=1}^{N}|A_{1j}|, \sum_{j=1}^{N}|A_{2j}|, \; . \; . \; . \; , \sum_{j=1}^{N}|A_{Mj}|,\right)$$

It is worth noting that for both the Implicit (which we have used) and the Crank-Nicholson method of finite differences, stability is guaranteed. It is only for the Explicit method that instability can occur due to the inappropriate choosing of variables, in particular the size of the time grid.

Despite this it is easily demonstrated within MATLAB that our data is stable, hence the following;

```
A=diag(alpha(3:M),-1)+diag(beta(2:M))+diag(gamma(2:M-1),1);
%Creation of our tri-diagonal matrix of alpha/beta/gammas
Ainv=Inv(a)                 %Inverse for Infinity Norm
normi=norm(Ainv,inf);       %Test for Stability

>> Project2
normi=1.000      %Proof of Stability
```

4.2.2. Convergence

Another element worth considering is the rate at which our algorithm converges, which is directly related to the truncation error we introduced to our model whenever we opted to use Taylor approximations for our PDE's. As such, our algorithm converges at the rates of O(δt) and O(δS)^2, the 2 sets of higher order terms we discarded in order to approximate our derivatives in **(22)** and **(26)/(27)** respectively.

To calculate the truncation error in our data we shall take the Closed Form Solution for a European Put option in equation **(3)** as the "true" value of a Put and compare it to the figures our algorithm has produced.

The following is the algorithm required to code equation **(3)** for varying strike prices;

```
%% Closed Form Solution for a European Put

d1=1/(sigma*T^0.5)*(log(S/X)+(r+((sigma^2)/2))*T);  %d1 variable
d2=d1-(sigma*T^0.5);                                %d2 variable
n1=normcdf(-d1);    % Normal CDF's
n2=normcdf(-d2);
p=n2*X*exp(-r*T)-n1*S;    %Put Price
```

Additionally, for peace of mind, I will also include figures sourced from an online model[7] so as to have examples of an outside source to compare our results against. However it is worth noting that the outside source only allows the entering of variables to 4 d.p. and hence our Closed Form Solution is more accurate.

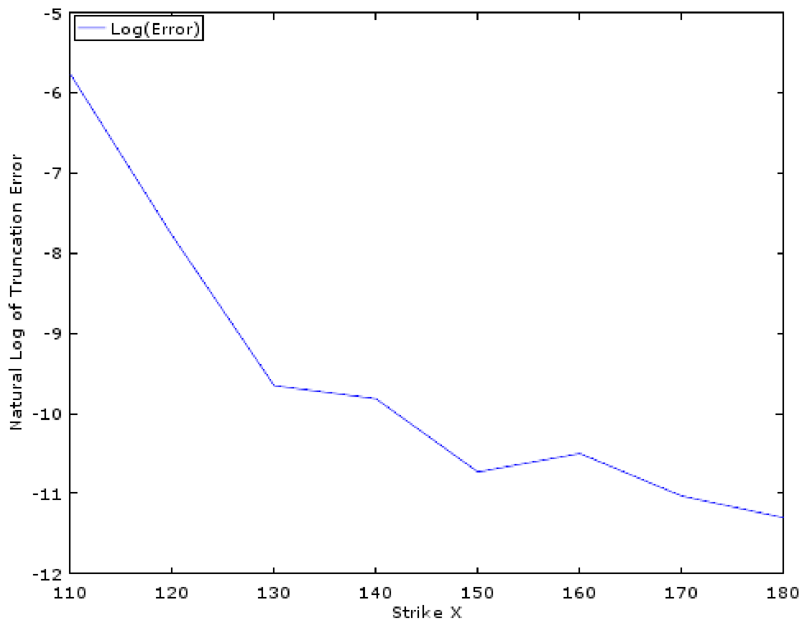
If it is evident that our results contain only a small amount of truncation error and thus are relatively quick to converge, it will hence be logical to conclude, considering both our error and expectations of data in the previous section, that the results we have calculated fulfil the purpose of this assignment.

Table 3: Comparison of Calculated P(t,S) vs. "True" P(t,S) vs. Outside P(t,S)

Strike Price $X	European Put Option Price $P(t,S)	Closed Form Solution $P(t,S)	Outside Source $P(t,S)
110	0.19236	0.19178	0.19180
120	0.96259	0.96220	0.96230
130	3.1222	3.1224	3.1225
140	7.3562	7.3566	7.3567
150	13.753	13.752	13.752
160	21.828	21.827	21.827
170	30.924	30.923	30.923
180	40.531	40.530	40.530

Figure 3: Truncation Error as % of Calculation; Table and Log(Error) Plot

Strike Price $X	European Put Option Price $P(t,S)	Truncation Error Size	Truncation Error Size %
110	0.19236	0.0006	$3.1192\text{x}10^{-3}$
120	0.96259	0.0004	$4.1555\text{x}10^{-4}$
130	3.1222	0.0002	$6.4057\text{x}10^{-5}$
140	7.3562	0.0004	$5.4376\text{x}10^{-5}$
150	13.753	0.0005	$2.1813\ \text{x}10^{-5}$
160	21.828	0.0006	$2.7488\ \text{x}10^{-5}$
170	30.924	0.0005	$1.6169\ \text{x}10^{-5}$
180	40.531	0.0005	$1.2336\ \text{x}10^{-5}$



Hence by finding the difference between the Calculated Put Option Price and the "True" Put Option Price, we have attained values for the Truncation Error in our data. From the above figures and plot it is clear that, due to such small values for Truncation Error, the figures we have computed are extremely accurate and convergent and hence we conclude our results are of sound credibility.

5. Conclusion

5.1 Reiteration of Main Findings

The object of this project was to calculate the fair European Put Option Price for IBM shares expiring in July of 2016. Whilst there are a number of points you may argue to question the validity of the B-S Model in allowing for this calculation, the following figures are the best estimate for a fair price given the information we were initially provided.

Table 4: European Put option prices for IBM Corp. shares expiring 15/07/ 2016

Strike Price \$X	European Put Option Price \$P(t,S)
110	0.19236
120	0.96259
130	3.1222
140	7.3562
150	13.753
160	21.828
170	30.924
180	40.531

5.2 Closing Remarks: Drawbacks of B-S and What Next?

The problem itself has been fairly rudimentary. We attempted to estimate the fair price of a European Put Option by solving the Black-Scholes Partial Differential Equation via Finite Difference Methods given a set of initial values for the various variables involved.

At no point during our exploration of the B-S formula, understanding of Finite Differences, framing of our question in Matrix Notation or the process of coding our solution algorithm did we encounter any bugs or issues we were unable to resolve or understand. The only notable exception is the difference between our Calculated Put Price and the Market Put Price which may be due to varying interest rates or the use of different models. Regardless, we have adequately verified that the figures we have calculated are correct.

We have proved that the algorithm we have used is stable and that the error within our model is infinitesimally small and we were even able to verify the validity of our figures both through internal checks in the form of the Closed Form Put Solution, and external checks in the form of Black-Scholes online calculators. With that being said there are still notable areas with which concern might arise.

The B-S model is based off of a no-arbitrage assumption which underpins the equation we can derive in **(12)**. Whilst there are many schools of thought which argue that arbitrage and alpha in general is a fictitious anomaly (mainly advocates of the Efficient Market Hypothesis), many Hedge Fund managers and long term value investors would argue the contrary. Hence, for a topic which has no clear consensus, the no-arbitrage assumption (and by extension the efficient market assumption) is one that must be treated with care.

Secondly, the geometric Brownian motion model implies that the series of first differences of the log prices must be uncorrelated. But for the S&P 500 as a whole, observed over several decades, daily from 1 July 1962 to 29 Dec 1995, there are in fact small but statistically significant correlations in the differences of the logs at short time lags, thus leading to another challenge of a key assumption. [8]

In addition to the figures and concepts we were able to derive given the starting data and time frame we had to work with, there are a number of other areas I would wish to explore had more time existed following the conclusion of this project.

American/Asian Options – In this report we have tackled the simplest form of option on the market i.e. European Options. As previously mentioned this simplicity is derived from the defining feature of European Options – that they can only be executed at maturity. For more exotic forms of options i.e. American/Asian or different combinations of option strategies i.e. Straddles, Protective/Covered etc. is would be convenient to write an algorithm that contained a set of functions which we could switch between at any given time to cover an array of potential pricing requirements. This would be considerably more difficult to program as it would involve an iterative approach to calculating the inverse of Matrix A **(33)**, however given more time I am confident it could be done.

Crank-Nicholson – In the pursuit of the most accurate figure for option prices, we also could have adopted the Crank-Nicholson Finite Differencing Method which incorporates both forward and backward difference approximations to essentially get the best approximation of both the implicit and explicit methods. The key feature of the Crank-Nicholson method is its faster rate of convergence than that of both Implicit and Explicit methods, however due to our truncation error being so small and hence our convergence already being so fast, I suspect adopting the Crank-Nicholson method would produce figures that differ by only an extremely small margin when compared to the figures we have already produced.

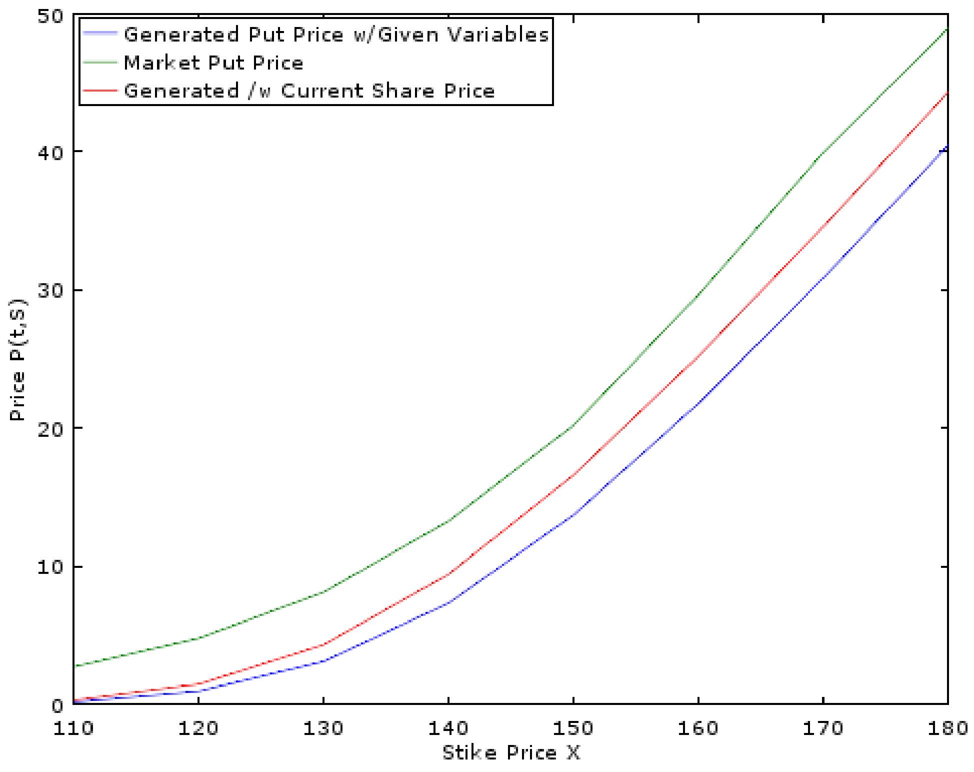
Whilst our figures may not agree with market data, this is indicative of a different approach being used and not of our figures being incorrect, as verified by multiple sources. As can be seen in Appendix **(1)**, by altering our given variables to better reflect current market conditions, we can bring our Put Prices more in line with market opinion. I suspect that the differing prices are derived from the manner in which the B-S method is incorporated into the world of options trading. Whilst B-S is good as a guideline it would seem that the markets approach the model with an air of caution particularly towards the consistency of volatility and the initial assumption of Geometric Brownian motion. This is to some extend illustrated in the following quote from Clifford Hurvich, Professor at the Stern School of Business in New York;[8]

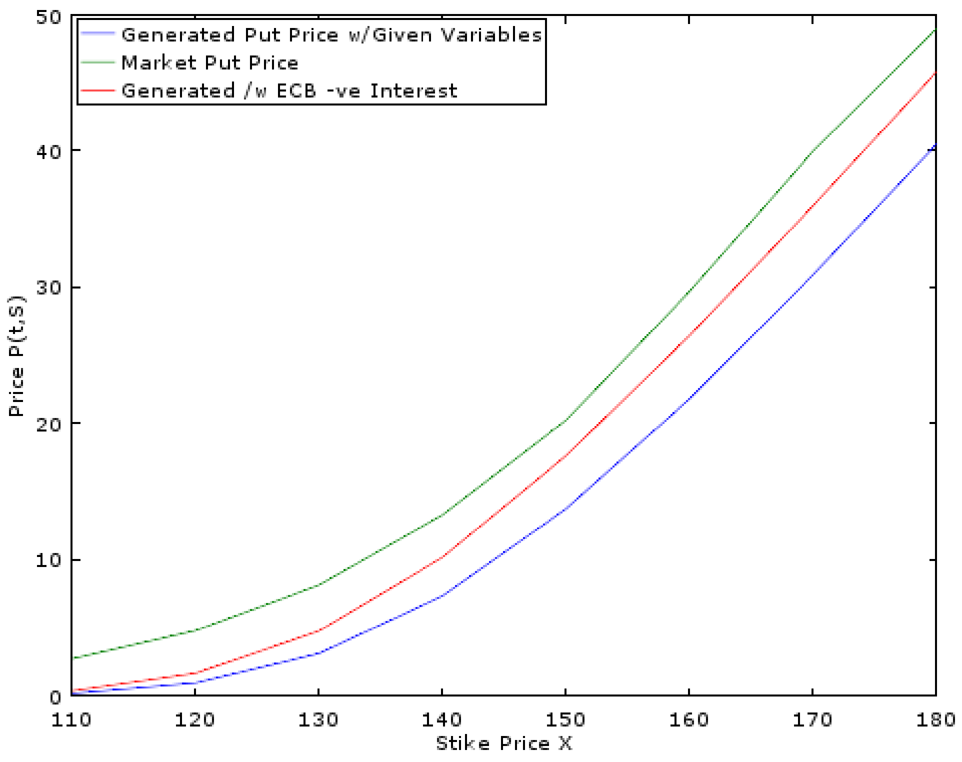
"There is the question of constancy of the variance of [B-S]. In practice, it is often found that for financial time series, after taking logs (if needed) and first differences, the level of volatility (i.e., fluctuation) seems to change with time. Often, periods of high volatility follow immediately after a large change (often downward) in the level of the original series. It may take quite some time for this heightened volatility to subside. For example, the plot of differences of the logs of the S&P 500 shows very long periods of high volatility interspersed with periods of relative calm."

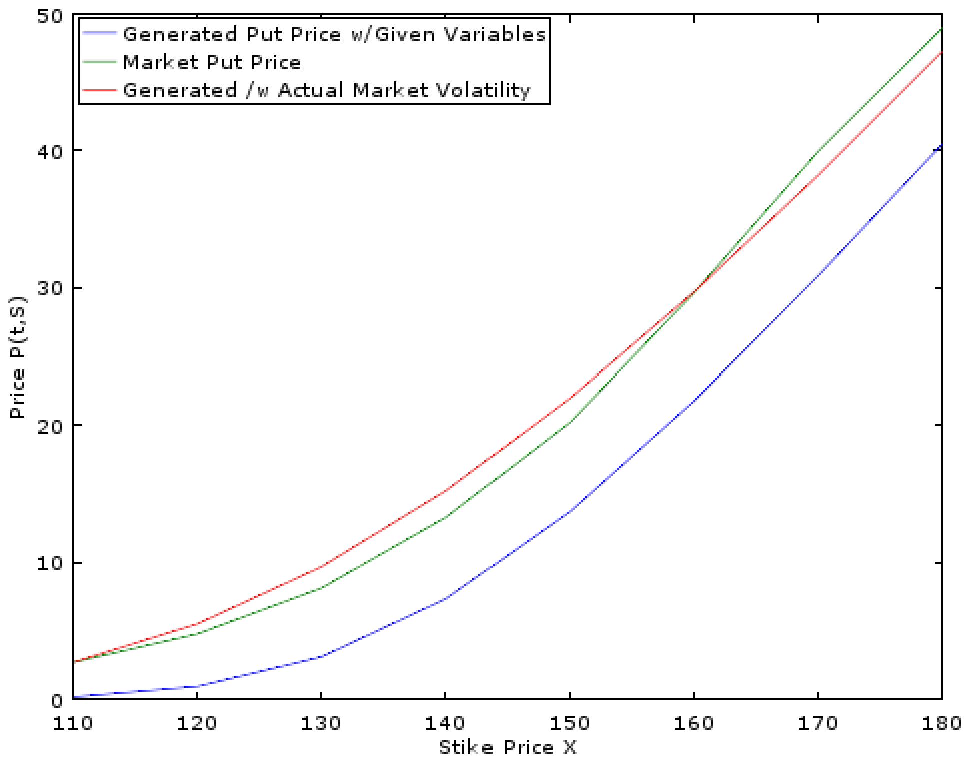
In conclusion however, we can now say with relative certainty that we have calculated the fair price of an IBM European Put Option, expiring July 15th 2016, according to the Black-Scholes Model.

6. Appendix

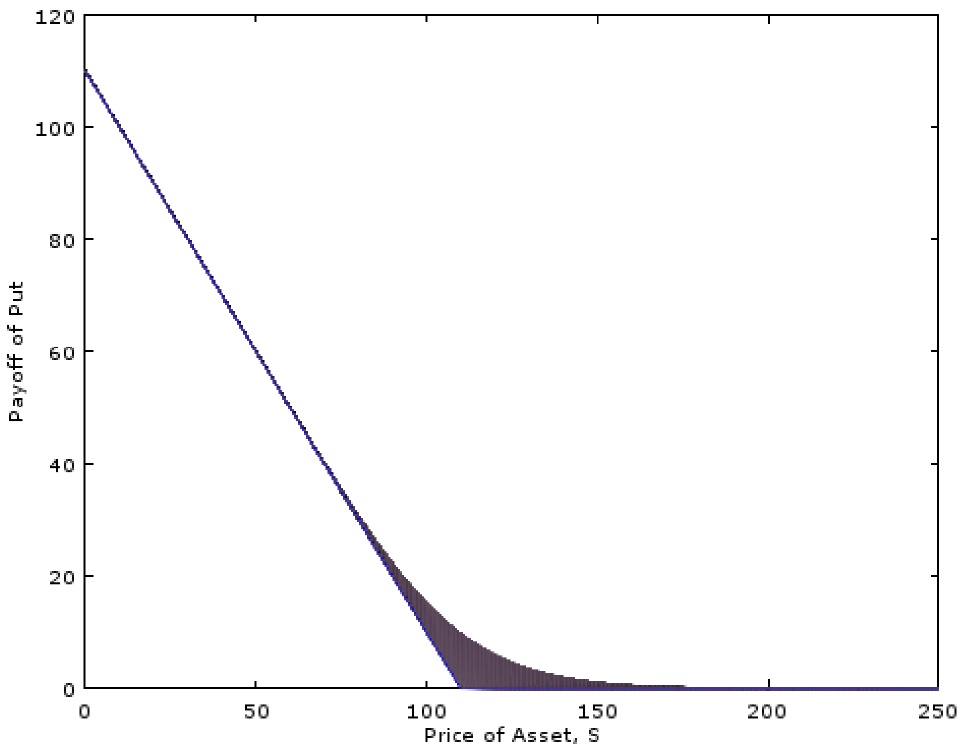
Point 1 - Graphs created by changing input variables to mirror Market conditions (Note each consecutive graph contains previous changes)

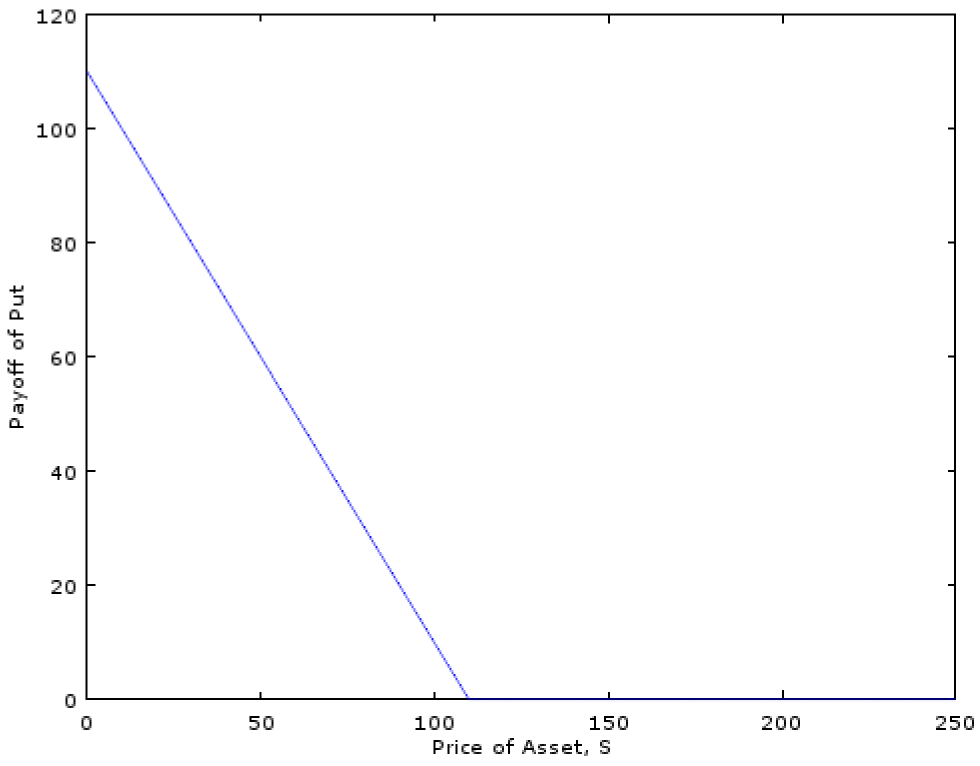




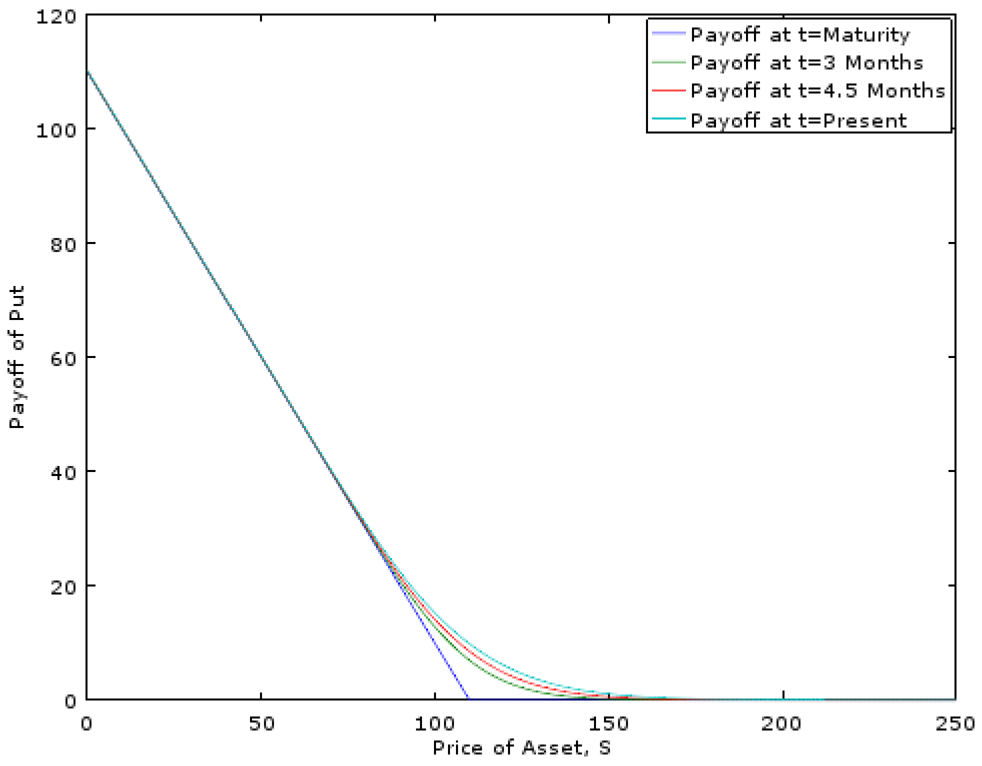


Point 2 - Comparison of sideways plot of Put Price Solution to standard Put payout at maturity (X=$110)





Point 3 - Plot of Put Prices at different Time Intervals (X=$110)



Code for above plots;

```
X=110:10:180; %Varying Strike Price
Y=[0.1923 0.9625 3.122 7.356 13.75 21.82 30.92 40.53]; % Original SP
Z=[2.73 4.80 8.15 13.30 20.25 29.70 40.00 49.00]; %Actual
A=[0.33143 1.4745 4.3382 9.4524 16.637 25.250 34.641 44.382]; % Current SP
B=[0.38767 1.6683 4.7768 10.187 17.641 26.458 35.995 45.845]; % interest=-0.3%
C=[2.7313 5.5346 9.7054 15.243 22.001 29.753 38.258 47.298]; % sigma=26%
K=110
S=0:250; %For Put Plot
D=max(K-S,0); %For Put Plot
plot(Sgrid,solngrid(:,end),Sgrid,solngrid(:,1155),Sgrid,solngrid(:,578),Sgrid,
solngrid(:,1)) %Prices for various time steps
```

7. References

[1] - *Essential's of Investment Chapter 16, Section 3 : Black-Scholes Option Valuation by Bodie, Kane and Marcus*

[2] - *Fisher Black & Myron Scholes The Pricing of Options and Corporate Liabilities. The Journal of Political Economy. Vol 81, Issue 3, 1973.*

[3] - *Fundamentals of Futures and Options Markets by John C. Hull - Chapter 13*

[4] - *AMA3021 Chapter 5 Random Processes, Chapter 6 Diffusion Processes* - Jim McCann

[5] - Goddard Consulting. *Option Pricing Using The Implicit Finite Difference Method. Available at: http://www.goddardconsulting.ca/option-pricing-finite-diff-implicit.html*

[6] - *AMA2004 Numerical Analysis Chapters 1 and 4* - Myrta Gruening

[7]- MoneyZine. *Black-Scholes Calculator. Available at : http://www.money-zine.com/calculators/investment-calculators/black-scholes-calculator/*

[8] - Clifford Hurvich, Stern University. *Some Drawbacks of Black Scholes. Available at: http://people.stern.nyu.edu/churvich/Forecasting/Handouts/Scholes.pdf*