Numerical Methods for solving the Black-Scholes Equation and its Applications

Rachael Paciello (Applied Mathematics)
Noah McDonald (Applied Mathematics)
Advisor: Andrea Dziubek
With help from Tural Sadigov

May 2018

Contents

1 List of Figures

List of Figures

2 Definitions in Financial Mathematics

2.1 Arbitrage

The simultaneous buying and selling of securities, currency, or commodities in different markets or in derivative forms in order to take advantage of differing prices for the same asset. Essentially ”free-money”

2.2 Black-Scholes Model

Black-Scholes Model gives a theoretical estimate of the price of European-style options and shows that the option has a unique price regardless of the risk of the security and its expected return

2.3 Call Option

An option to buy assets at an agreed price on or before a particular date

2.4 Derivative Security

Financial security with a value that is reliant upon or derived from an underlying asset or group of assets

2.5 Dividends

A sum of money paid regularly (typically quarterly) by a company to its shareholders out of its profits (or reserves)

2.6 European Option

A European option may be exercised only at the expiration date of the option, i.e. at a single pre-defined point in time. An American option the on other hand may be exercised at any time before the expiration date

2.7 Geometric Brownian Motion

Continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion (also called a Wiener process) with drift.

2.8 Hedge

A transaction that reduces the amount of risk of an investment

2.9 Implied Volatility

The implied volatility of an option contract is that value of the volatility of the underlying instrument which, when input in an option pricing model (such as Black–Scholes) will return a theoretical value equal to the current market price of the option

2.10 Ito's Lemma

Used to determine the derivative of a time-dependent function of a stochastic process. It performs the role of the chain rule in a stochastic setting, analogous to the chain rule in ordinary differential calculus
Let $X_t, t \in R_+$ be an Ito Process $X : \Omega \times R_+ \to R$ and $f : C^2(R \times R_+ \times R_+)$. Then the stochastic process $f_t := f(X_t, t)$ is also an Ito process which satisfies $\partial(f_t) = (\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial x} + \frac{b^2\partial^2 f}{2\partial t^2})dt + dW_t\frac{\partial f}{\partial x}$ where W_t is a Wiener process

2.11 Ito Process

A stochastic process X_t satisfying the equation $dX_t = a(X_t, t)dt + b(X_t, t)dW_t$ is said to be an Ito Process

2.12 Portfolio

A range of investments held by a person or organization

2.13 Risk Free Rate

The risk-free rate represents the interest an investor would expect from an absolutely zero risk investment over a specified period of time

2.14 Security Trading

Trading securities is a category of securities that includes both debt and equity securities, and which an entity intends to sell in the short term for a profit that it expects to generate from increases in the price of the securities

2.15 Short Selling

The sale of a security that is not owned by the seller or that the seller has borrowed. Short selling is motivated by the belief that a security's price will decline, enabling it to be bought back at a lower price to make a profit

2.16 Strike Price

The price fixed by the seller of a security after receiving bids in a tender offer, typically for a sale of bonds or a new stock market issue

2.17 Transaction Cost

Transaction costs are expenses incurred when buying or selling a good or service

2.18 Volatility

A variable in option pricing formulas showing the extent to which the return of the underlying asset will fluctuate between now and the option's expiration. Volatility can either be measured by using the standard deviation or variance between returns from that same security or market index. Commonly, the higher the volatility, the riskier the security

2.19 Wiener Process

A continuous-time stochastic process which has characteristics

1. $W_0 = 0$
2. W has independent increments: for every $t > 0$, the future increments $W_{t+u} - W_t, u \geq 0$ are independent of the past values $W_s, s < t$
3. W has Gaussian increments: $W_{t+u} - W_t$ is normally distributed with mean 0 and variance $u, W_{t+u} - W_t \sim \aleph(0, u)$
4. W has continuous paths: With probability 1, W_t is continuous in t.

3 Introduction to the Black-Scholes Equation

The purpose of this project is to look at the famous Black-Scholes equation and give an analysis on the numerical methods to solve the equation. Although an exact solution for the price of European options is available, the prices of more complicated derivatives such as American options must employ numerical methods. This project will focus on explicit and implicit numerical methods for solving partial differential equation (PDE) and compare the two methods to the analytic solution to the Black-Scholes equation. This will give us a better understanding of the equation and the methods to solve the equation.

The Black-Scholes equation is rooted in and was derived for the European stock market. In the financial world mathematics occupies a large role in how we make financial risks and investments. Mathematical models and equations help us make said financial decisions, the Black-Scholes is one such equation. The Black-Scholes equation is a partial differential equation used in the stock market, the equation gives a theoretical projection of the price of European-style options, specifically European-style call option in our case. The equation shows that any call option has a unique value and does not effect the risk of the security and its expected return. The equation is given as:

$$rf = \frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \tag{1}$$

Where f is the price of the option as function with respect to the stock price S and time t, r is the interest rate of the said stock , and σ represents the volatility of the stock.

This report will focus on the solution to the equation for European call options. It will present a derivation of the Black Scholes model, then introduce the analytical solution by transforming said equation into the well known diffusion equation and the boundary value problems for a European call will be solved. Then it will introduce the finite difference method for solving PDE's and employ explicit and implicit approaches to approximating the solution. Finally a brief analysis of the accuracy of each approach is provided.

4 Derivation

4.0.1 Brownian Motion/Wiener Process

As one might expect, stocks tend to follow an exponential pattern. In a perfect world, their growth could be measured using a simple first order differential equation

$$\frac{dx}{dt} = \mu x \tag{2}$$

with solution

$$X(t) = X_0 e^{\mu t} \tag{3}$$

This is shown graphically below

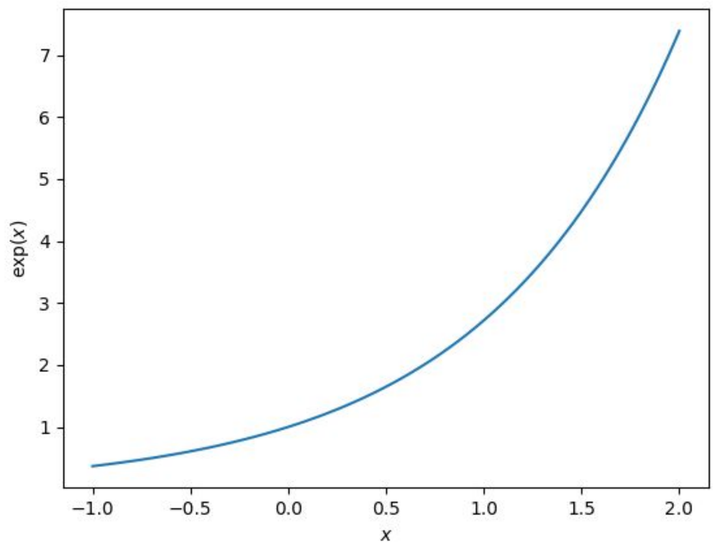


Figure 1: e^x

However, this is not sufficient in the real world, as there is "noise" that needs to be taken into account that will effectively prevent a smooth curve from forming. This "noise" will have a normal distribution called Y, $Y \sim Normal\ (\mu = 0, \sigma^2)$. We will write this as σZ where $Z \sim Standard\ Normal(\mu = 0, \sigma^2 = 1)$ and σ is volatility. Our new equation becomes

$$dx = \mu x + \sigma Z \tag{4}$$

We will now refer to Z as dW, as it will follow the Wiener Process or Brownian Motion. A Wiener process is a particular type of stochastic process where only the present state of the process is relevant for predicting the future.dW is related to dt by the equation

$$dW = \epsilon\sqrt{dt} \tag{5}$$

where ϵ is a random sample from a standardized normal distribution. A graph representing a Wiener Process in 1D, 2D, and 3D is shown below

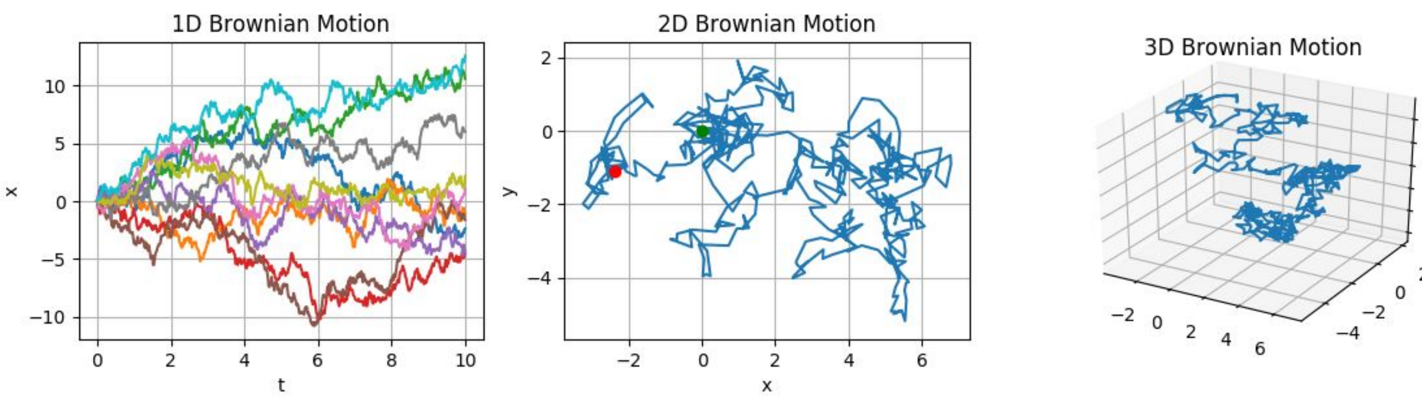


Figure 2: Brownian Motion

Now we must apply our model to the financial world. Let $S(t)$ represent stock price at time t, $S(t)$ be proportional to dt and dW, μ representing the expected return on the stock (assumed to be constant) and σ representing the volatility (also assumed to be constant).

$$dS = \mu S dt + \sigma S dW \tag{6}$$

4.0.2 Ito's Lemma

The price of a stock option is a function of the underlying stock's price and time. This concept of the behavior of functions of stochastic variables lead to an important discovery by mathematician, Ito of Japan in 1951. Consider a continuous and differentiable function G of variable x. If Δx is a small change in x and ΔG is the resulting small change in G then it is known that

$$\Delta G \approx \frac{dG}{dx}\Delta x \tag{7}$$

In other words, ΔG is approximately equal to the rate of change of G with respect to x multiplied by Δx. For a continuous function of two variables, x and t, the resulting analogous equation is

$$\Delta G \approx \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t \tag{8}$$

For a more precise approximation of ΔG the Taylor series expansion can be used

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + \frac{\partial G}{2\partial x}\Delta x^2 + \frac{\partial^2 G}{\partial x \partial t}\Delta x \Delta t + \frac{\partial G}{2\partial t}\Delta t^2 + ... \tag{9}$$

Taking the limit as Δx and Δt approach zero we obtain

$$\Delta G = \frac{\partial G}{\partial x}dx + \frac{\partial G}{\partial t}dt \tag{10}$$

since Δx and Δt are independent of each other and higher order terms can be eliminated beacuse of their minimal contribution. Since the Black Scholes equation is one that follows a stochastic process, we will extend equation (10) to to cover such functions. Suppose that variable x follows the general Ito process in equation (4) rewritten as

$$dx = a(x,t)dt + b(x,t)dW \tag{11}$$

Using equation (5), we can rewrite our previous equation as Taylor series expansion

$$\Delta x = a(x,t)\Delta t + b(x,t)\epsilon\sqrt{\Delta t} \tag{12}$$

From this equation, it can be clearly seen that Δx and Δt are related. Now by squaring our equation and dropping arguments we obtain

$$\Delta x^2 = a^2\Delta t^2 + b^2\epsilon^2\Delta t + 2ab\Delta t^{\frac{3}{2}} \tag{13}$$

Once again, higher order terms of Δt can be ignored because of their minute contribution. By analogy we can rewrite equation (9) to

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + \frac{\partial G}{2\partial x}\Delta x^2 + \frac{\partial^2 G}{\partial x \partial t}\Delta x \Delta t + \frac{\partial G}{2\partial t}\Delta t^2 + ... \tag{14}$$

At this point, we will substitute our Δx^2 into equation (14):

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + \frac{\partial^2 G}{\partial x^2}b^2\epsilon^2\Delta t + ... \tag{15}$$

This equation brings to light an important difference in equations (9) and (14). When limiting arguments where used in equation (9), second order terms Δx^2 were ignored. From equation (15) it can be seen that we cannot ignore our Δx^2 term since it is of component Δt.
At this point we must examine what is happening with our ϵ^2 term.The variance of a standard normal distribution is 1.0. This means that

$$E(\epsilon^2) - [E(\epsilon)]^2 = 1 \tag{16}$$

where E denotes the expected value. Since $E(\epsilon) = 0$, it follows that $E(\epsilon^2) = 1$ from equation (16). The expected value of $\epsilon^2\Delta t$ is therefor Δt. The variance of $\epsilon^2\Delta t$ is shown to be $2\Delta t^2$ below:

$$\text{Var}[\epsilon^2\Delta t] = \Delta t^2 \ \text{Var}[\epsilon^2] = \Delta t^2 \ (\text{E}(\epsilon^4) - \text{E}(\epsilon^2)^2) = \Delta t^2 \ (3-1) = 2\Delta t^2 \equiv 0 \tag{17}$$

Because the variance has component Δt^2, it is essentially equal to 0. With no variance, everything gets accumulated around the mean, Δt.

In Figure 3, we show this relationship.

$\epsilon^2 \Delta t$ becomes non-stochastic and equal to its expected value of Δt as it tends to 0. Therefore, the term $b^2\epsilon^2\Delta t$ from equation (15) equals $b^2 dt$ as Δt tends to 0. Taking the limits as Δx and Δt approach 0 in equation (14) and using our previous result, we obtain Ito's Lemma:

$$dG = \frac{\partial G}{\partial x}dx + \frac{\partial G}{\partial t}dt + \frac{\partial^2 G}{2\partial x^2}b^2 dt \qquad (18)$$

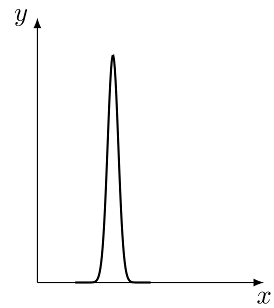


Figure 3: A narrow Gaussian.

4.0.3 Assumptions

The following assumptions are necessary to our derivation of the Black Scholes model:

- The stock price follows the process developed in equation (5) with μ and σ constant
- The short selling of securities with full use of proceeds is permitted
- There are no transaction costs or taxes
- There are no dividends during the life of the derivative security
- There are no arbitrage opportunities
- Security trading is continuous
- r is constant and the same for all maturities

4.0.4 Putting it all together

Now we are ready to derive the Black Scholes Equation. We assume that the Stock Price S follows the process discussed in equation (6): $dS = \mu S dt + \sigma S dW$. Suppose f is the price of a derivative security contingent on S. The variable f must then be some function of S and t. Applying Ito's Lemma to our equation we obtain

$$df = (\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{\partial^2 f}{2\partial S^2}\sigma^2 S^2)dt + \frac{\partial f}{\partial S}\sigma S dz \qquad (19)$$

Recall from Ito's Lemma section that the Wiener process underlying f and S are the same, i.e dz in equations (6) and (19) are the same. By choosing a suitable portfolio, the Wiener process can be eliminated. Consider the following portfolio in which we buy $\frac{1}{\Delta}$ shares of our option where $\Delta[0,1]$ and we sell one stock, i.e short one stock, and long an amount of $\frac{\partial f}{\partial S}$ shares:

- -1 : Derivative security

- $+\frac{\partial f}{\partial S}$: shares

We will define Π as the value of the portfolio. Therefore

$$\Pi = -f + \frac{\partial f}{\partial S} \tag{20}$$

Since this equation does not involve dz the portfolio must be risk less during time dt as there is no randomness. The assumptions listed previously imply that the portfolio must instantaneously earn the same rate of return as other short term risk free stocks in order to avoid an arbitrage. It follows then that

$$d\Pi = r\Pi dt \tag{21}$$

where r is the risk free interest rate. Combining and substituting equations (6), (19), and () we obtain

$$rf = \frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \tag{22}$$

Equation (19) is the Black Scholes differential equation.

4.1 Boundary and Initial Conditions

For a call option with payoff (Stock Price - Strike Price) boundary conditions are as follows:

$$V(s,t) \approx 0 \text{ for } s \text{ very small}$$
$$V(s,t) \approx s \text{ for } s \text{ very large}$$

with initial condition

$$V(s,0) = max(\text{Payoff}, 0)$$

5 Analytical Solution

Beginning with the Black Scholes equation

$$rf = \frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \tag{23}$$

supplemented with the terminal and boundary conditions in the case of a European call

$$F(S,T) = max(S-K,0)$$
$$F(0,t) = 0$$
$$F(S,t) \sim S \text{ as } S \to \infty$$

The option value is defined over the domain $0 < S < \infty$ and $0 \leq t \leq T$. In order the analytically solve this PDE, we will first transform it into the well known diffusion equation. We will begin by making the substitution $u = e^{-rt}f$, which comes from the fact that it is the portfolio value discounted by the interest rate r that is a martingale (more specifically Brownian motion). Using the product rule on $f = e^{-rt}u$ we can derive the PDE that the function u must satisfy:

$$rf = re^{rt}\,u = re^{rt}\,u + e^{rt}\frac{\partial u}{\partial t} + rfe^{rt}\frac{\partial u}{\partial f} + \frac{1}{2}\sigma^2 f^2 e^{rt}\frac{\partial^2 u}{\partial f^2} \tag{24}$$

or more simply:

$$0 = \frac{\partial u}{\partial t} + rf\frac{\partial u}{\partial f} + \frac{1}{2}\sigma^2 f^2 \frac{\partial^2 u}{\partial f^2} \tag{25}$$

Now we will make a change of variables. It can be observed that the underlying process of variable f, is a geometric Brownian motion (as described in the derivation) so that $\log f$ describes some Brownian motion with a drift. It is well known in physics that $\log f$ should satisfy some form of the diffusion equation. Therefore we will let $y = logf$ and $\tau = T - t$.

The boundary condition of the Black Scholes equation is given as the terminal state, and so the evolution of the system must be constructed backwards. On the other hand, the heat equation describes temperature changing in forwards time, and so we must use substitution to reverse time. The coefficient $\frac{\partial u}{\partial t}$ of equation (24) compared to the standard heat equation coefficient $0 = \frac{-\partial u}{\partial t} + \frac{\partial u}{\partial S}$. Since

$$\frac{\partial u}{\partial \tau} = -\frac{\partial u}{\partial t}\,, \quad \frac{\partial u}{\partial f} = \frac{\partial u}{\partial y}\frac{dy}{df} = \frac{1}{f}\frac{\partial u}{\partial y}$$

and

$$\frac{\partial^2 u}{\partial f^2} = \frac{\partial}{\partial f}\left(\frac{1}{f}\frac{\partial u}{\partial y}\right) = -\frac{1}{f^2}\frac{\partial u}{\partial y} + \frac{1}{f^2}\frac{\partial^2 u}{\partial y^2}$$

substituting into equation (24) we find

$$0 = -\frac{\partial u}{\partial \tau} + (r - \frac{1}{2}\sigma^2)\frac{\partial u}{\partial y} + \frac{1}{2}\sigma^2\frac{\partial^2 u}{\partial y^2} \tag{26}$$

To simplify this equation, we want to cancel the first partial derivative with respect to y (unless $r = \frac{1}{2}\sigma^2$). In order to do so, we must take into account the drift of the Brownian motion. To cancel the drift, we make the substitutions:

$$z = y + (r - \tfrac{1}{2}\sigma^2)\theta\,, \quad \theta = \tau$$

Under our new coordinate system, (z, θ) we have the following relation amongst vector fields:

$$\tfrac{\partial}{\partial z} = \tfrac{\partial}{\partial y}\,, \quad \tfrac{\partial}{\partial \theta} = -(r - \tfrac{1}{2}\sigma^2)\tfrac{\partial}{\partial y} + \tfrac{\partial}{\partial \tau}$$

leading to equation (25) becoming

$$0 = -\frac{\partial u}{\partial \theta} - (r - \frac{1}{2}\sigma^2)\frac{\partial u}{\partial z} + (r - \frac{1}{2}\sigma^2)\frac{\partial u}{\partial z} + \frac{1}{2}\sigma^2\frac{\partial^2 u}{\partial z^2} \tag{27}$$

or

$$\frac{\partial u}{\partial \theta} = \frac{1}{2}\sigma^2\frac{\partial^2 u}{\partial z^2}\,, \quad u = u(\theta, z) \tag{28}$$

which is one form of the well known diffusion equation with initial condition $\max(e^{\frac{k+1}{2}x} - e^{\frac{k-1}{2}x}, 0)$:

The original function f is recovered with:

$$f(T, S) = u(T - t, logS + (r - \frac{1}{2}\sigma^2)\theta) \tag{29}$$

The fundamental solution to the previously given PDE(28) is given by:

$$D_\theta(z) = \frac{1}{\sqrt{2\pi\sigma^2\theta}}e^{\frac{-z}{2\sigma^2\theta}} \tag{30}$$

Derivation was found using the Fourier transform; the solution for u with the initial condition u_0 is given by the convolution:

$$u(\theta, z) = u_0 * D_\theta(z) = \frac{1}{\sqrt{2\pi\sigma^2\theta}}\int_\infty^\infty exp(-\frac{(logS + (r - \frac{1}{2}\sigma^2)\theta - j)^2}{2\sigma^2\theta})dj \tag{31}$$

Finally we put this in terms of the original function f:

$$f(T, S) = \frac{1}{\sqrt{2\pi\sigma^2\theta}}\int_\infty^\infty exp(-\frac{(logS + (r - \frac{1}{2}\sigma^2)\theta - j)^2}{2\sigma^2\theta})dj, \tag{32}$$

Thus with some mathematical manipulation we are give:

$$u(\theta, z) = Ke^{(z + \frac{1}{2}\theta\sigma^2}N(d_1) - KN(d_2), \tag{33}$$

Where N is the standard normal cumulative distribution function, and likewise

$$d_1 = \frac{1}{\sigma\sqrt{\theta}}[(z + \frac{1}{2}\sigma^2\theta) + \frac{1}{2}\sigma^2\theta] \tag{34}$$

$$d_2 = \frac{1}{\sigma\sqrt{\theta}}[(z + \frac{1}{2}\sigma^2\theta) - \frac{1}{2}\sigma^2\theta] \tag{35}$$

Thus changing u, z, and θ back into the original set of variables give the previously stated solution to the Black-Scholes equation. The asymptotic condition can now be seen.

$$\lim_{z\to\infty} u(z,\theta) = Ke^{(z)} \tag{36}$$

This condition gives back S when reverting to the original set of coordinates. This is made possible by

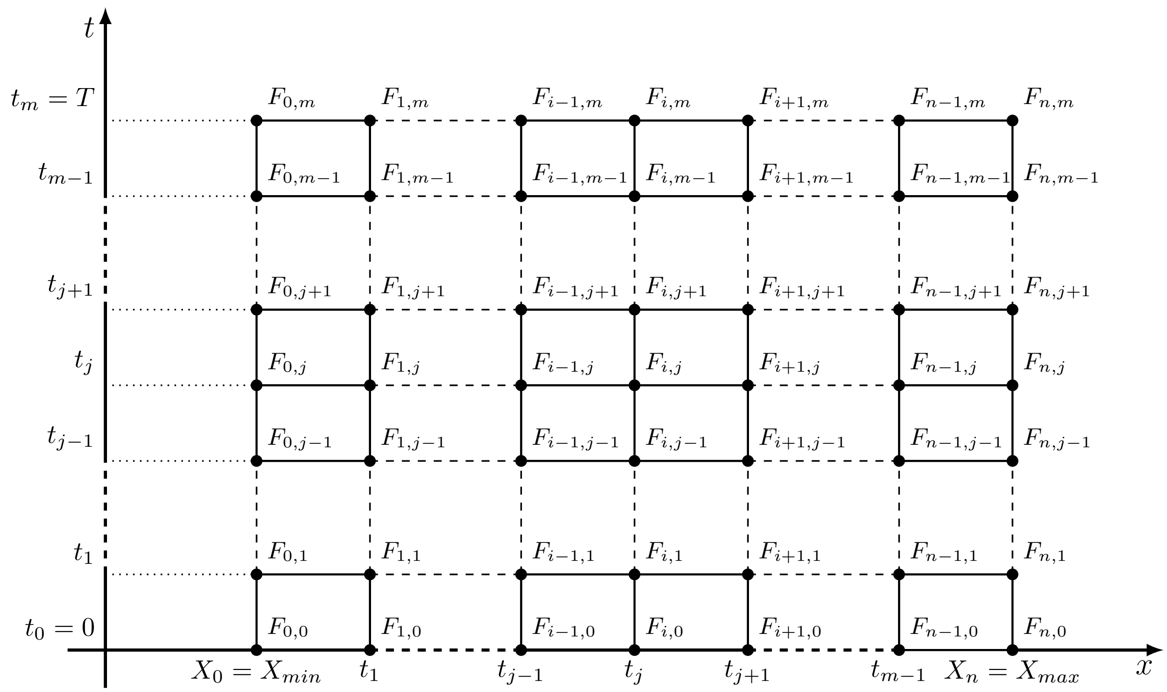
$$\lim_{z\to\infty} N(z) = 1. \tag{37}$$

6 Numerical Methods - Finite Differences

A prominent numerical method used to solve partial differential equations is finite difference methods. This method approximates values of solutions at certain rectangular mesh points by replacing the partial derivatives in the PDE by finite difference approximations and then solves the resulting system of equations.
Since in numerical computations we can only find finitely many numbers, we will try to compute a table of the approximate value of the solution (Let's call it f). For this we need to fix the minimum and maximum values of x we are interested in (x_{min} and x_{max}), the number m of sub intervals of the time period $[0, T]$, and the number of sub intervals n we use in the x direction. We will denote

$$\Delta t = \frac{T}{m}, \text{ and } \Delta x = \frac{x_{max} - x_{min}}{n}$$
$$\text{and define } t_k = k\Delta t,\ k = 0, ..., m;\ x_i = x_{min} + i\Delta x,\ i = 0, ..., n$$

Our aim is to find the approximate values $f_{ij} = f(S_i, t_j)$ In other words, we want to form a $(m+1) \times (n+1)$ table of approximate values (denoted by F_{ij}. The notations are illustrated below:



The values of f at $t = T$ are given by the terminal condition . The values corresponding to $x = x_{min}$ and $x = x_{max}$ are boundary conditions discussed in earlier chapters. In order to find the middle values, we will have to make use of our Black Scholes equation where derivatives are replaced by numerical differentiation formulas. From textbooks of numerical methods we can find the following approximate differentiation rules for a sufficiently smooth (meaning enough times continuously differentiable) function f:

$$f'(z) = \frac{f(z+h) - f(z)}{h} \tag{38}$$

$$f'(z) = \frac{f(z) - f(z-h)}{h} \tag{39}$$

$$f'(z) = \frac{f(z+h) - f(z-h)}{2h} \tag{40}$$

$$f''(z) = \frac{f(z-h) - 2f(z) + f(z+h)}{h^2} \tag{41}$$

The first formula is called forward difference approximation, the second is backward difference approximation and the third is the central difference approximation of the derivative

6.1 Explicit Finite Difference Method

After taking into account the terminal condition we have $(n-1)\cdot m$ empty spaces to fill. After applying the boundary conditions, we will need to use our PDE to derive $(n-1)\cdot m$ additional equations for our unknown values. In order to get an explicit method for our backwards parabolic equation we start by writing our Black Scholes equation at the points $(x_i, t_j), i = 1, ..., (n-1), j = 1, ..., m$:

$$0 = \frac{\partial f}{\partial t}(x_i,t_j) + rS(x_i,t_j)\frac{\partial f}{\partial S}(x_i,t_j) + \frac{1}{2}\sigma^2 S^2(x_i,t_j)\frac{\partial^2 f}{\partial S^2}(x_i,t_j) - rf(x_i,t_j) \tag{42}$$

To approximate the partial derivatives of f in the previous equation we will use its values at the following grid points

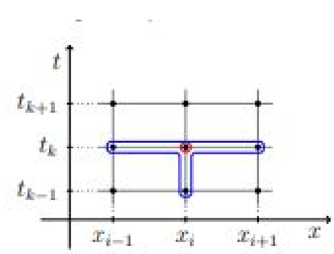


Figure 4: Grid Points

Using our finite difference methods to obtain estimates for our partial derivatives we get

$$\frac{\partial f}{\partial t}(x_i,t_j) = \frac{f_{i,j} - f_{i-1,j}}{\Delta t} \tag{43}$$

$$\frac{\partial f}{\partial S}(x_i, t_j) = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} \tag{44}$$

$$\frac{\partial^2 f}{\partial S^2}(x_i, t_j) = \frac{f_{i,j+1} - 2f_{i,j} + 2f_{i,j-1}}{\Delta S^2} \tag{45}$$

After substituting our equations into our PDE, we obtain

$$rf_{i,j} = \frac{f_{i,j} - f_{i-1,j}}{\Delta t} + rS\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} + \frac{1}{2}\sigma^2 S^2 \frac{f_{i,j+1} - 2f_{i,j} + 2f_{i,j-1}}{\Delta S^2} \tag{46}$$

and after solving for $f_{i-1,j}$ we get

$$f_{i-1,j} = \frac{1}{2}\Delta t(\sigma^2 j^2 - rj)\ f_{i,j-1} + 1 - \Delta t(\sigma^2 j^2 + r)\ f_{i,j} + \frac{1}{2}\Delta t(\sigma^2 j^2 + rj)\ f_{i,j+1} \tag{47}$$

or more simply

$$f_{i-1,j} = a\ f_{i,j-1} + b\ f_{i,j} + c\ f_{i,j+1} \tag{48}$$

where

$$a = \tfrac{1}{2}\Delta t(\sigma^2 j^2 - rj)$$

$$b = 1 - \Delta t(\sigma^2 j^2 + r)$$

$$c = \tfrac{1}{2}\Delta t(\sigma^2 j^2 + rj)$$

The following figure is a pictorial representation of equation (48)

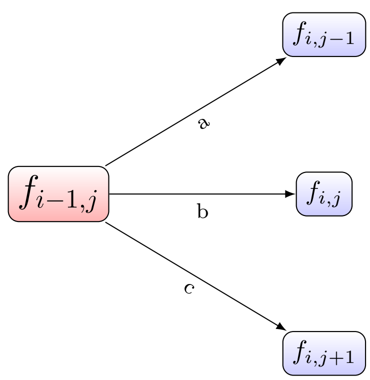


Figure 5: Trinomial Tree of Explicit Finite Differences.

In our option pricing framework,figure 2 shows that given the value of the option at boundary conditions, all interior points can be calculated using backwards induction. In other words, given the option payoff at expiry nodes, then the prices Δt before expiry can be calculated. Then at those prices, the value $2\Delta t$ before expiry can be calculated and so on. Working iteratively backwards through time until the option price for grid nodes $t = 0$ (i.e today) can be

calculated.
We will now formulate matrices from equation (48).

$$F_{i-1} = AF_i + K_i \text{ where } i = N, ..., 2, 1 \tag{49}$$

$F_i =$

$$\begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ \vdots \\ f_{i,M-1} \end{bmatrix}$$

$K_i =$

$$\begin{bmatrix} a_1 f_{i,0} \\ 0 \\ \vdots \\ 0 \\ C_{M-1} f_{i,M} \end{bmatrix}$$

$A =$

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 \\ 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{M-1} & b_{M-1} \end{bmatrix}$$

We will implement matrix methods later on in this paper using Python and evaluate the results.

6.2 Implicit Finite Difference Method

Using our finite difference methods to obtain estimates for our partial derivatives we get

$$\frac{\partial f}{\partial t}(x_i, t_j) = \frac{f_{i,j} - f_{i+1,j}}{\Delta t} \tag{50}$$

$$\frac{\partial f}{\partial S}(x_i, t_j) = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} \tag{51}$$

$$\frac{\partial^2 f}{\partial S^2}(x_i, t_j) = \frac{f_{i,j+1} - 2f_{i,j} + 2f_{i,j-1}}{\Delta S^2} \tag{52}$$

After substituting our equations into our PDE, we obtain

$$rf_{i,j} = \frac{f_{i,j} - f_{i+1,j}}{\Delta t} + rS\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} + \frac{1}{2}\sigma^2 S^2 \frac{f_{i,j+1} - 2f_{i,j} + 2f_{i,j-1}}{\Delta S^2} \tag{53}$$

and after solving for $f_{i-1,j}$ we get

$$f_{i+1,j} = \frac{1}{2}\Delta t(\sigma^2 j^2 - rj)\ f_{i,j+1} + 1 - \Delta t(\sigma^2 j^2 + r)\ f_{i,j} + \frac{1}{2}\Delta t(\sigma^2 j^2 + rj)\ f_{i,j+1} \quad (54)$$

or more simply

$$f_{i+1,j} = a\ f_{i,j-1} + b\ f_{i,j} + c\ f_{i,j+1} \quad (55)$$

where

$$a = \tfrac{1}{2}\Delta t(rj - \sigma^2 j^2)$$

$$b = 1 - \Delta t(\sigma^2 j^2 + r)$$

$$c = \tfrac{1}{2}\Delta t(-\sigma^2 j^2 - rj)$$

The following figure is a pictorial representation of equation (55)

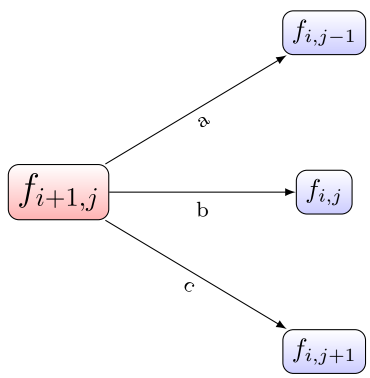


Figure 6: Trinomial Tree of Implicit Finite Differences.

In the option pricing framework,figure 3 shows that given the value of the option at boundary conditions, all interior points can be calculated using forward induction. In other words, given the option payoff at expiry nodes, then the prices Δt before expiry can be calculated. Then at those prices, the value $2\Delta t$ before expiry can be calculated and so on. Working iteratively forwards through time until the option price for grid nodes $t = T$ (i.e final expiration time) can be calculated.

We will now formulate matrices from equation (55).

$$BF_i = F_{i+1} + K_i \text{ where } i = N-1, ..., 2, 0 \quad (56)$$

$F_i =$

$$\begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ \vdots \\ f_{i,M-1} \end{bmatrix}$$

$K_i =$

$$\begin{bmatrix} -a_1 f_{i,0} \\ 0 \\ \vdots \\ 0 \\ -C_{M-1} f_{i,M} \end{bmatrix}$$

$B =$

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 \\ 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{M-1} & b_{M-1} \end{bmatrix}$$

7 Error in Numerical Methods

The basic idea of using finite difference methods is approximating the differential operator by replacing the derivatives in the equation with differential quotients. The error between the numerical solution and the exact solution is determined by the error that is committed by going from a differential operator to a difference operator. This error is called the discretization error or truncation error. The term truncation error reflects the fact that a finite part of a Taylor series is used in the approximation.The truncation error approximation is given by

$$E = f'(z) - \text{ finite difference approximation}$$

The common way of calculating E is to

1. Expand $f(z)$ in a Taylor series around the point where the derivative is evaluated
2. Insert this Taylor series into error approximation formula
3. Collect terms that cancel and simplify the expression

Using the finite differences methods previously mentioned, we obtain the following truncation errors:

$$\text{Forward Difference: } O(h) = \frac{f(z+h) - f(z)}{h} - f'(z) \tag{57}$$

$$\text{Backward Difference: } O(h) = \frac{f(z) - f(z-h)}{h} - f'(z) \tag{58}$$

$$\text{Central Difference: } O(h^2) = \frac{f(z+h) - f(z-h)}{2h} - f'(z) \tag{59}$$

$$\text{Second Order Central Difference: } O(h^4) = \frac{f(z-h) - 2f(z) + f(z+h)}{h^2} - f''(z) \tag{60}$$

7.1 Stability and Convergence

An important question to consider when using any numerical algorithm is *when is it stable*? And if it is stable, *when does it converge*? From standard matrix algebra it is known that a matrix of the form given in equation (48) is stable if and only if

$$||A||_\infty \leq 1$$

If the infinity norm of our matrix is less than 1 then successive values of F_i in equation (49) get smaller and smaller and hence the algorithm converges, or is stable. It can be shown that for certain combinations of $r, \sigma,$ and Δt (and

therefore values for a, b, and c) the infinity norm of A will be greater than 1. Consequently, unless the grid size (particularly in the time axis) is chosen appropriately the explicit finite difference method can be unstable, and hence un-useful for option pricing.
From standard matrix algebra it is known that a matrix of the form given in equation (55) is stable if and only if

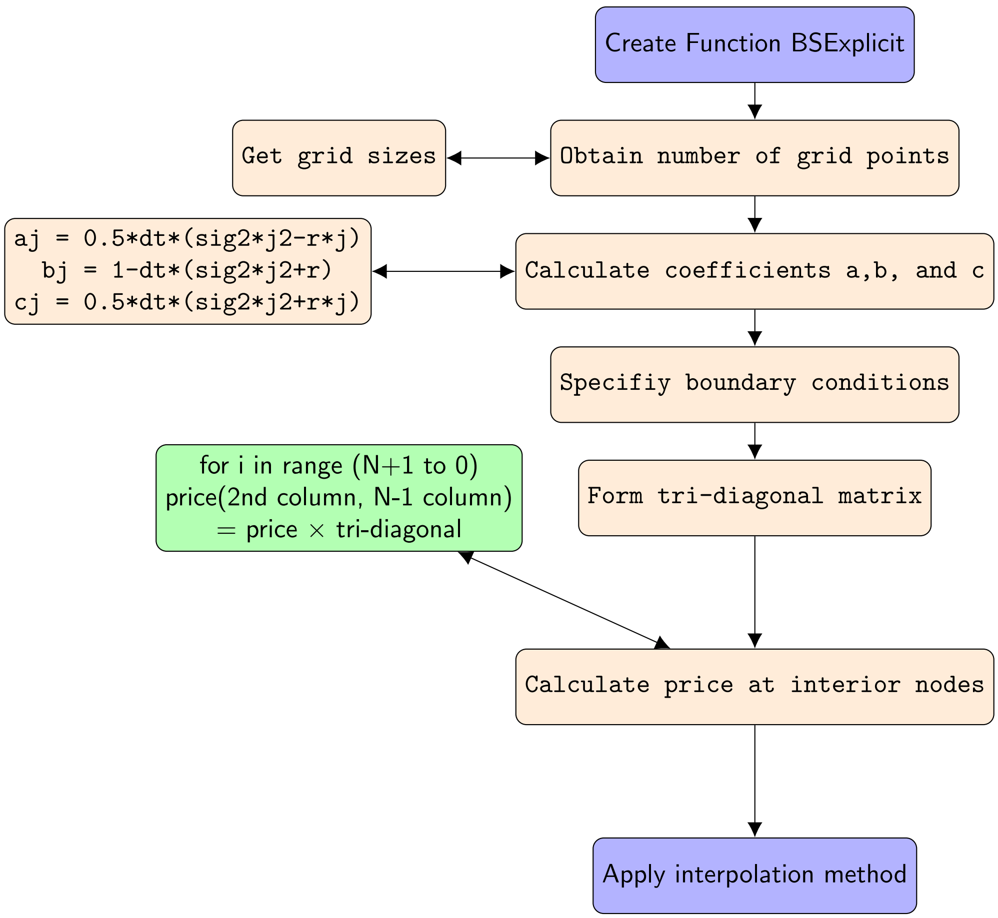
$$||B^{-}1||_{\infty} \leq 1$$

If the infinity norm of our matrix is less than 1 then successive values of F_i in equation (56) get smaller and smaller and hence the algorithm converges, or is stable. It can be shown that the infinity norm of $B^{-}1$ is always less than 1 for all combinations of $r, \sigma,$ and Δt (and therefore values for a, b, and c) the infinity norm of A will be greater than 1. The implicit method is therefore guaranteed to be stable.

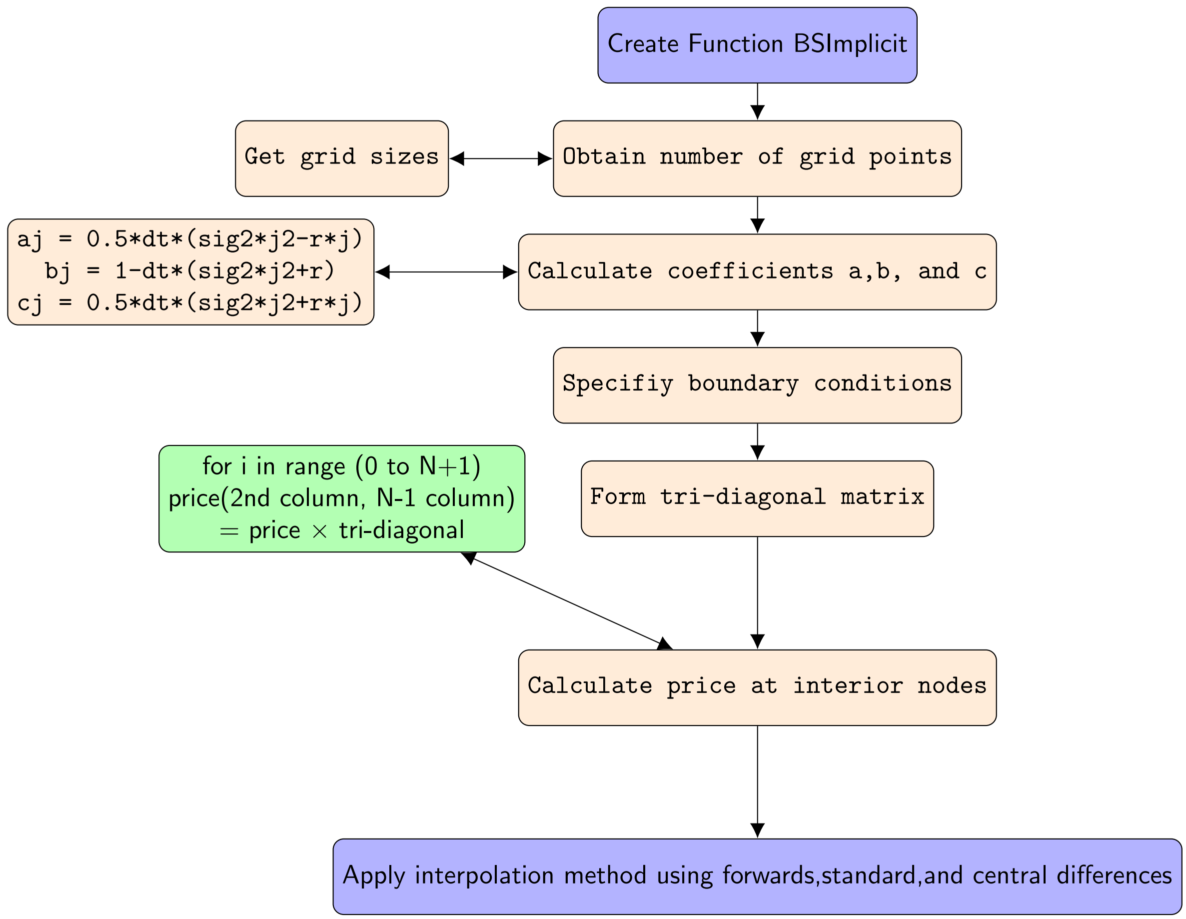
The rate of convergence for both explicit and implicit algorithm is directly related to the truncation error. They both converge at the rate of $O(h)$ and $O(h^2)$. A disadvantage of using the implicit method is that A disadvantage of the implicit method is that it requires the inverse of a matrix to be calculated, and the inverse of a matrix is (computationally) an expense operation to perform. Fortunately, for tri-diagonal matrices such as B, fast inversion algorithms are available.

8 Algorithm and Programming

8.1 Explicit Method



8.2 Implicit Method



8.3 Exact Solution

The following code snippet was used to calculate the exact solution:

```
import math
import numpy as np
from scipy import stats
from scipy.stats import norm
def exactSolution(r,sigma,price,T,E):
    d1 = (np.log(price/E) + T * (r+sigma**2/2)) / sigma*np.sqrt(T)
    d2 = d1 - sigma*np.sqrt(T)
    return price*np.exp(-0 * T)*norm.cdf(d1)-E*np.exp(-r*T)*norm.cdf(d2)
```

9 Results

9.1 Explicit results

In comparison to the exact solutions, our explicit results turned out very well. Later on in this paper we show comparison of our results. The following abbreviations were used:

- T = time
- E = strike price
- r = risk free rate
- σ = volatility
- NAS = Asset steps (the higher the more accurate, but more time consuming)

Our results are what we expected, with the most sensitive factors being volatility and risk free rate (respectively). However, as noted before the explicit method can be unstable. If we set a large time step we begin to see unreadable results, as shown in figure 13.

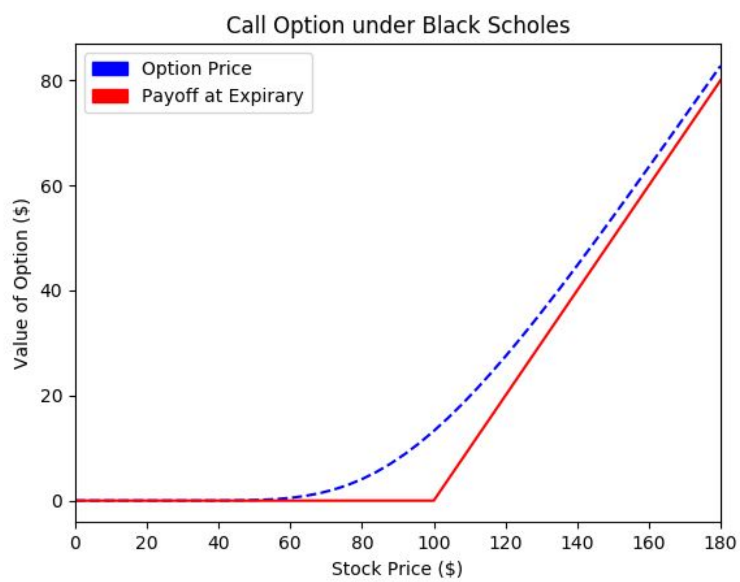


Figure 7: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 45$

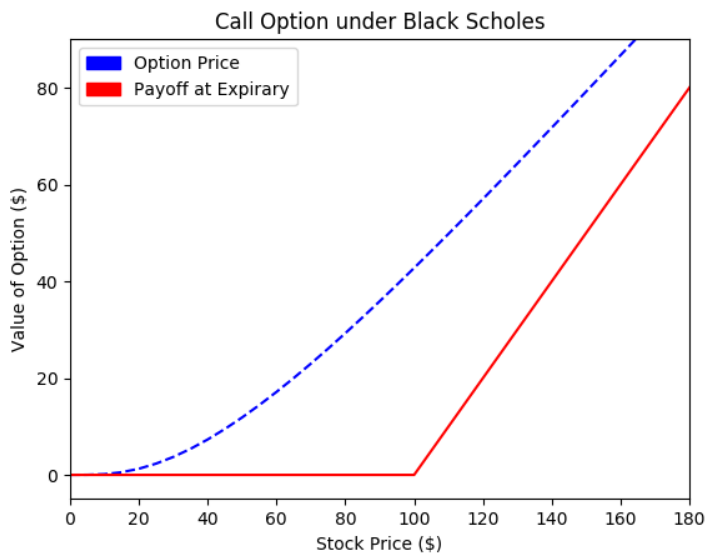


Figure 8: $T = 10, E = 100, r = .03, \sigma = .3, NAS = 45$

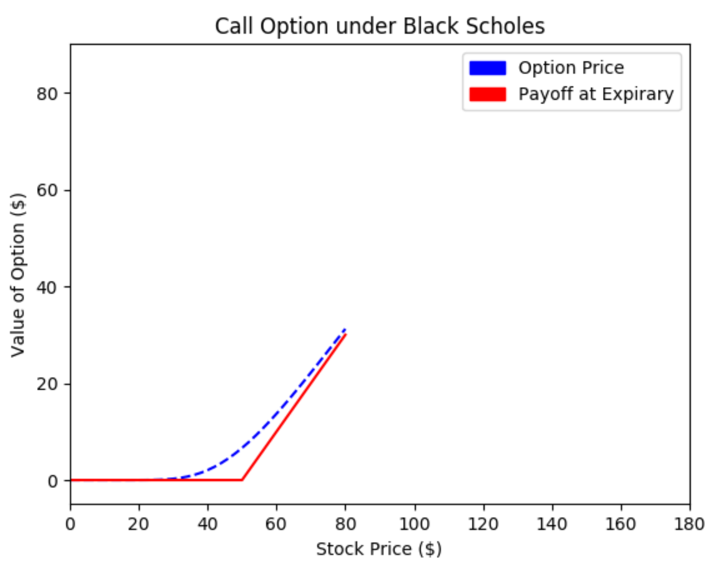


Figure 9: $T = 1, E = 50, r = .03, \sigma = .3, NAS = 45$

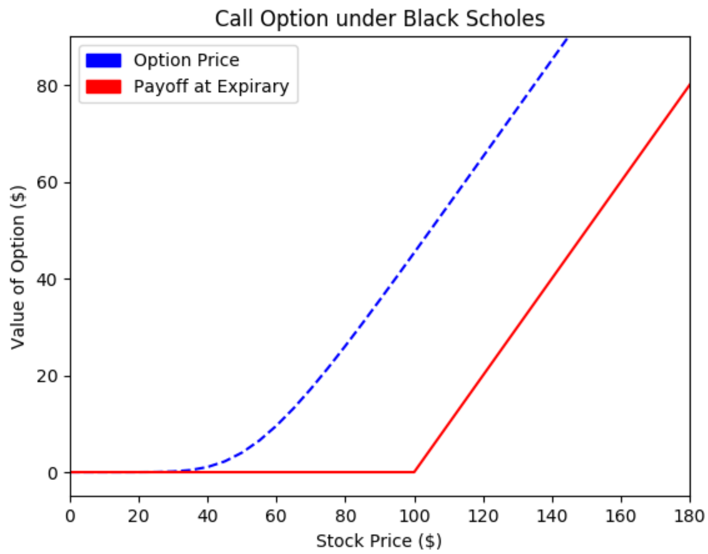


Figure 10: $T = 1, E = 100, r = .60, \sigma = .3, NAS = 45$

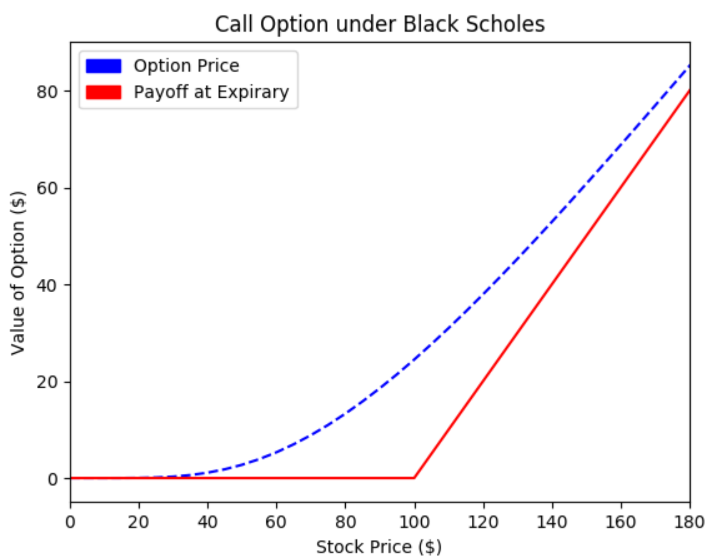


Figure 11: $T = 1, E = 100, r = .03, \sigma = .6, NAS = 45$

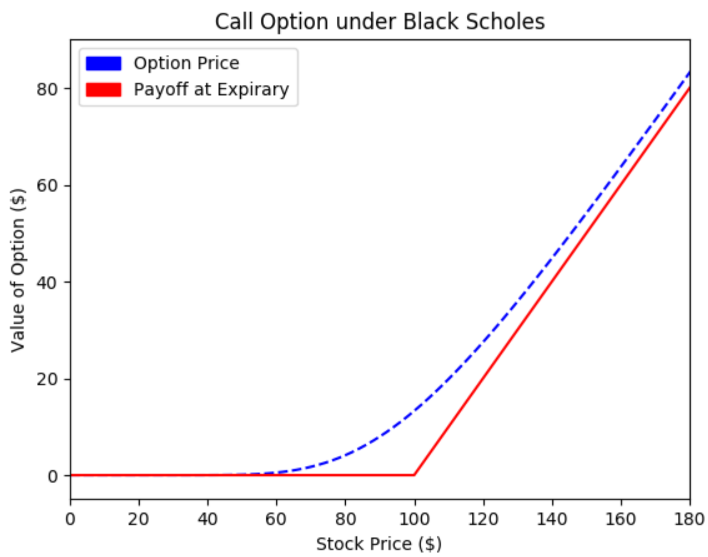


Figure 12: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 100$

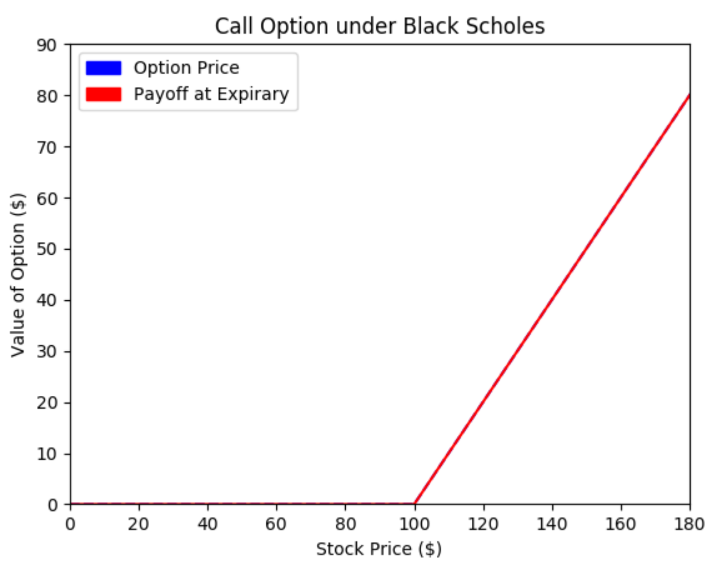


Figure 13: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 100, TimeStep = 100000$

9.2 Implicit Results

If one is to compare the exact solution to the implicit solution one would see that the implicit solution did rather well under certain conditions. From the data gathered a few observations can be made; first and for most the change in the asset step greatly changes the results of the solution, as the asset step grew larger in size the more accurate the implicit solution got. It was also seen that volatility and risk free rate were the most sensitive factors in this method, thus changes in volatility and risk free rate effect the outcome of the solution (see Figure 17, 20).

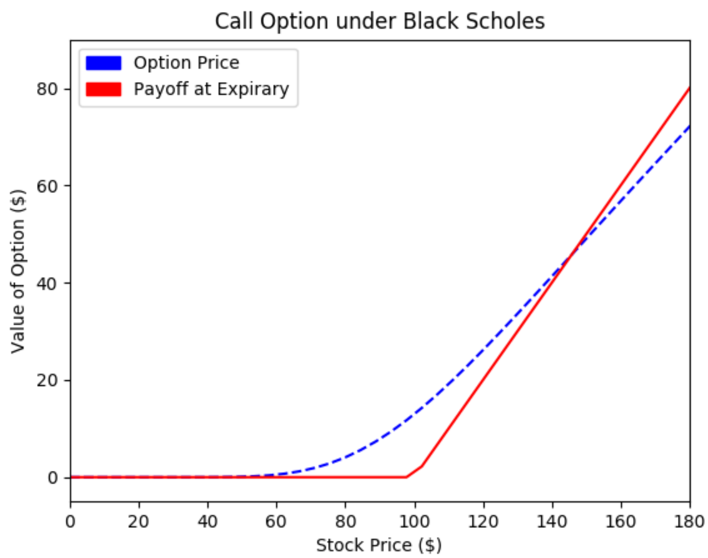


Figure 14: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 45$

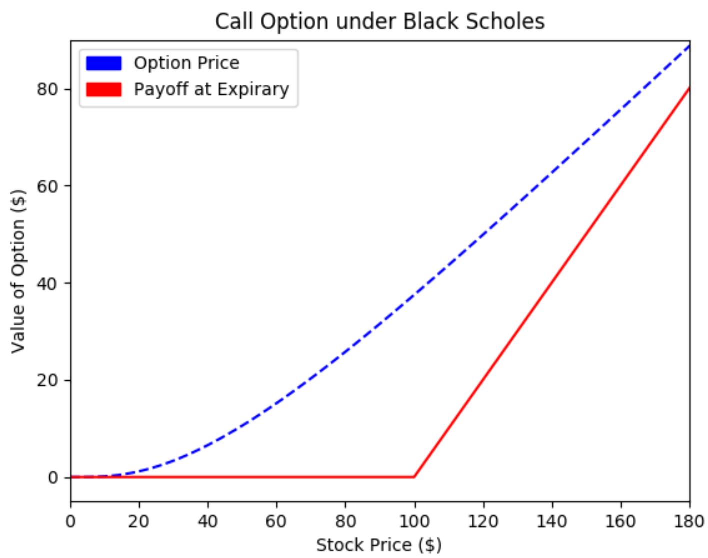


Figure 15: $T = 10, E = 100, r = .03, \sigma = .3, NAS = 100$

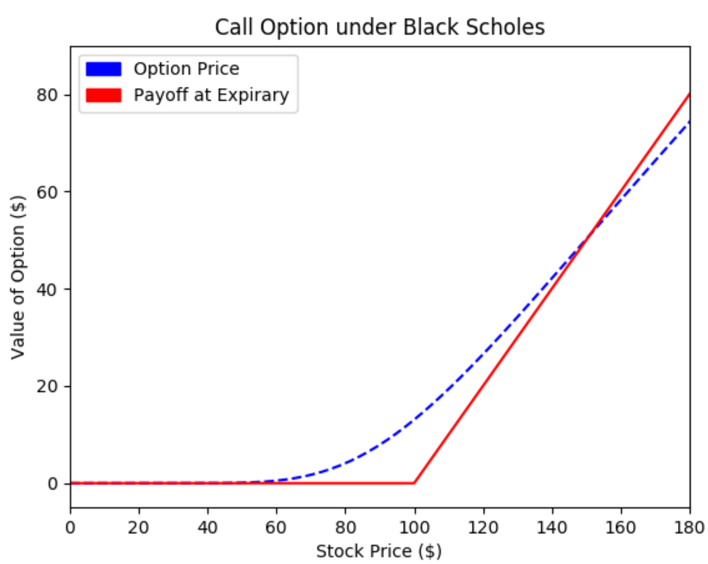


Figure 16: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 150$

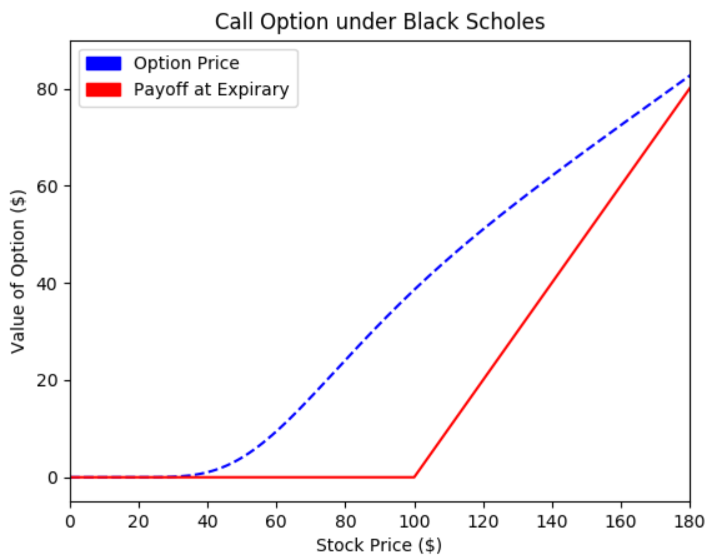


Figure 17: $T = 1, E = 100, r = .60, \sigma = .3, NAS = 150$

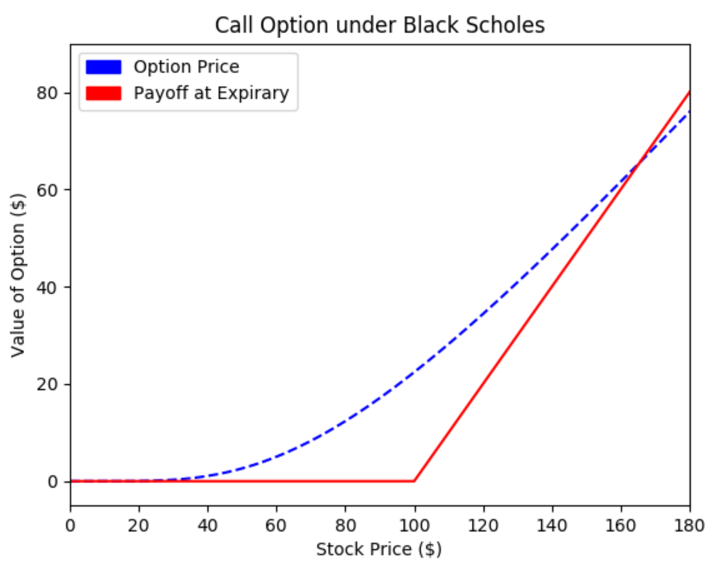


Figure 18: $T = 1, E = 100, r = .03, \sigma = .6, NAS = 150$

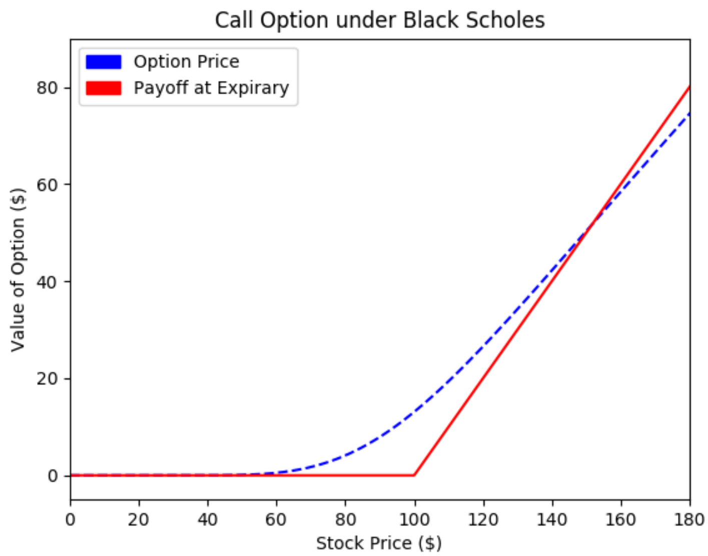


Figure 19: $T = 1, E = 100, r = .03, \sigma = .3, NAS = 200$

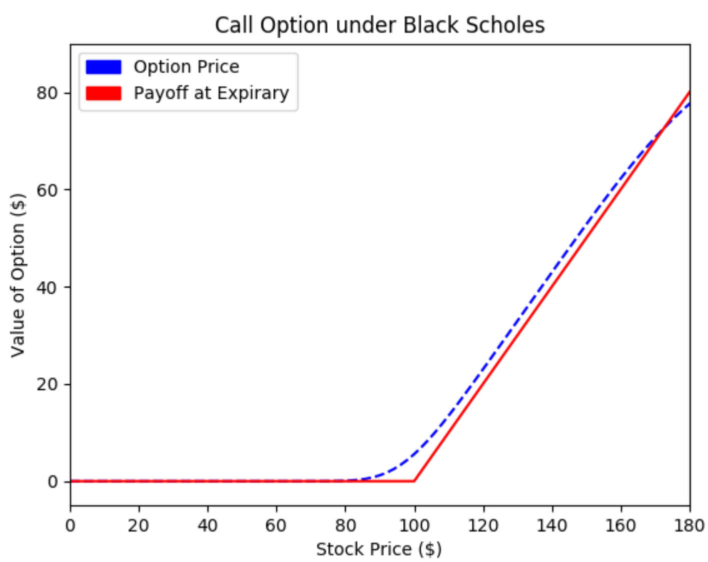


Figure 20: $T = 1, E = 100, r = .03, \sigma = .1, NAS = 100$

10 Summary of the project

As shown by the table below, our explicit results were much more accurate than implicit. This could be due to a number of reasons, the main one being implicit methods are generally more complex to implement.

Comparison of Numerical Methods - Option Prices			
Strike Price ($)	Exact Solution ($)	Explicit Solution ($)	Implicit Solution ($)
50	51.53	51.46	49.00
100	103.06	102.91	98.01
150	154.60	154.37	147.01
200	206.13	205.82	196.01
250	257.66	257.73	245.00
300	309.19	308.73	294.00
350	360.73	360.19	343.44

Table 1: Computed with the following parameters :
Time = 1 year, Stock Price = 2 * Strike Price , Risk free rate = 3 %,
Volatility = 30%

Perhaps for future research a better model for the implicit method can be developed.

11 Python Code

```
#Explicit Method
import math
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib
from scipy.stats import norm
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

T = 1 #Time to Expiry in Years
E = 300#Strike
r = .03 #Risk Free Rate
sigma = .3 #Volatility
NAS = 100 #Number of Asset Steps - Higher is more accurate, but more
    time consuming
ds = 2 * E / NAS #Asset Value Step Size
dt = (0.9/NAS/NAS/sigma/sigma) #Time Step Size
NTS = int(T / dt) + 1 #Number of time steps
value = np.zeros((int(NAS+1), int(NTS)))
price = np.arange(NAS*ds,-1,-ds)
value[:,-1]= np.maximum(price - E,0)
for x in range(1,NTS):
    value[-1,-x-1] = value[-1,-x]* math.exp(-r*dt)
for x in range(1,int(NTS)):
    for y in range(1,int(NAS)):
        Delta = (value[y-1,-x] - value[y+1,-x]) / 2 / ds
        value[y+1,-x]
        Gamma = (value[y-1,-x] - (2 * value[y,-x]) + value[y+1,-x]) / ds
            / ds
        Theta = (-.5 * sigma**2 * price[y]**2 * Gamma) - (r * price[y] *
            Delta) + (r * value[y,-x])
        value[y,-x-1] = value[y,-x] - Theta * dt
        value[0,-x-1] = 2 * value[1,-x-1] - value[2,-x-1]
BSV = pd.DataFrame(value);BSV = BSV.set_index(price)
BSVPlot = BSV.sort_index(ascending=True)
BSVPlot[0].plot(color = 'blue', linestyle = 'dashed')
BSVPlot[NTS-1].plot(color = 'red')
blue_patch = mpatches.Patch(color='blue', label='Option
    Price');red_patch = mpatches.Patch(color='red', label='Payoff at
    Expirary')
plt.legend(handles=[blue_patch,red_patch])
plt.axis([0,180, -5,90])
plt.title('Call Option under Black Scholes')
plt.ylabel('Value of Option (\$)');plt.xlabel('Stock Price (\$)')
plt.show()
print(BSVPlot[0])
```

```
#Implicit Method
import math
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib
from scipy.stats import norm
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

T = 1 #Time to Expiry in Years
E = 100 #Strike
r = .03 #Risk Free Rate
sigma = .6 #Volatility
NAS = 150 #Number of Asset Steps - Higher is more accurate, but more
    time consuming
ds = 2 * E / NAS #Asset Value Step Size
dt = (0.9/NAS/NAS/sigma/sigma) #Time Step Size
NTS = int(T / dt) + 1 #Number of time steps
value = np.zeros((int(NAS+1), int(NTS)))
price = np.arange(NAS*ds,-1,-ds)
value[:,-1]= np.maximum(price - E,0)
for x in range(1,NTS):
    value[1,-x+1] = value[1,-x]* math.exp(-r*dt)
for x in range(1,int(NTS)):
    for y in range(1,int(NAS)):
        Delta = (value[y-1,-x] - value[y+1,-x]) / 2 / ds
        value[y+1,x]
        Gamma = (value[y-1,-x] - (2 * value[y,-x]) + value[y+1,-x]) / ds
            / ds
        Theta = (-.5 * sigma**2 * price[y]**2 * Gamma) - (r * price[y] *
            Delta) + (r * value[y,-x])
        value[y,-x-1] = value[y,-x] - Theta * dt
        value[0,-x-1] = 2 * value[1,-x-1] - value[2,-x-1]
BSV = pd.DataFrame(value); BSV = BSV.set_index(price)
BSVPlot = BSV.sort_index(ascending=True)
BSVPlot[0].plot(color = 'blue', linestyle = 'dashed')
BSVPlot[NTS-1].plot(color = 'red')
blue_patch = mpatches.Patch(color='blue', label='Option Price')
red_patch = mpatches.Patch(color='red', label='Payoff at Expirary')
plt.legend(handles=[blue_patch,red_patch])
plt.axis([0,180, -5,90])
plt.title('Call Option under Black Scholes')
plt.ylabel('Value of Option ($)')
plt.xlabel('Stock Price ($)')
plt.show()
```

```python
#exact solution

import math
import numpy as np
from scipy import stats
from scipy.stats import norm
def exactSolution(r,sigma,price,T,E):
    d1 = (np.log(price/E) + T * (r+sigma**2/2)) / sigma*np.sqrt(T)
    d2 = d1 - sigma*np.sqrt(T)
    return price*np.exp(-0 * T)*norm.cdf(d1)-E*np.exp(-r*T)*norm.cdf(d2)
```

12 Course Evaluation

Student A:

Overall, I though this course was a pretty interesting one. I enjoyed learning new methods to solve equations and I liked how coding was included. There are certainly a lot of real world applications I could think of to apply to this course. I would have liked more practice with the concepts, and more time for projects to be developed. Since I do have a computer science background I felt this course was a great blend of mathematics and coding. However, for those who have no coding background I can't see how they would be successful in this class. I really enjoyed the poster presentations, it was a great experience and enjoyable to see all of the other student's work. At this point, I am unable to comment on the teamwork part of the course.

Student B:

To put it simply I have mixed feeling on the course, mostly good though. I found the course challenging, interesting, and enjoyable. The material was covered in great detail, giving me a better understanding of the this field of mathematics. That said I do feel that the project should be started earlier, and at the same time I feel that the prerequisites should be greater, or that the student should be more aware of what is required from them in this course, before hand. Other than that I would say this course was well put together. I had also enjoyed the poster presentations, I felt that it was great experience and it was great to see other student's projects.

13 Bibliography

References

[1] *Derivation of the Black Scholes Equation.*
`www.sjsu.edu/faculty/watkins/blacksch2.htm`

[2] Gina Dura, Ana-Maria Mosneagu. *Numerical Approximation of Black Scholes Equation*. 2010.

[3] John Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, 2003.

[4] *Computational Finance*. 2011.
`kodu.ut.ee/ rkangro/computational`$_{f}inance/finmat11/finmat_{e}ng_{1}1.pdf$

[5] *The Black Scholes Equation.*
`people.math.gatech.edu/ meyer/MA6635/chap4.pdf`