DELHI TECHNOLOGICAL UNIVERSITY

FINANCIAL ENGINEERING (MC-306)

Midterm Innovative Project

SUBMITTED BY:

Aiman Siddiqua - 2K18/MC/008

Apoorva - 2K18/MC/019

SUBMITTED TO:

Dr. R Srivastava

Numerical Solutions for Black Scholes Partial Differential Equation

Using finite difference method to solve the Black-Scholes PDE to calculate the price of European Call options in Python

Contents

1 Introduction **1**
- 1.1 Options . . . 1
  - 1.1.1 Call Option . . . 2
  - 1.1.2 Put Option . . . 2

2 The Black-Scholes Model **4**

3 Derivation of Black-Scholes Partial Differential Equation **5**
- 3.1 Solving the Black-Scholes Equation . . . 6

4 Numerical Solution of Black Scholes PDE **7**
- 4.1 Boundary Conditions . . . 7
- 4.2 Estimating Derivatives . . . 8

5 Solution in Python **10**
- 5.1 Importing Libraries . . . 10
- 5.2 Initializing the parameters . . . 10
- 5.3 Discretization . . . 11
- 5.4 Boundary Conditions . . . 11
- 5.5 Construction of the tri-diagonal matrix D . . . 11
- 5.6 Backward iteration . . . 11
- 5.7 Finding the option at S_0 . . . 12
- 5.8 Plotting Graphs . . . 12

6 Outputs **13**
- 6.1 Option at S_0 . . . 13
- 6.2 Graphs . . . 13

1 Introduction

The Black–Scholes model is a mathematical model simulating the dynamics of a financial market containing derivative financial instruments such as options, futures, forwards and swaps. The model is based on a partial differential equation (PDE), the so-called Black-Scholes equation, from which one can deduce the Black-Scholes formula, which gives a theoretical estimate of the correct price of European stock options.

In this project, we attempted to write a program in Python that solves a Black-Scholes Partial Differential equation numerically by the finite difference method to calculate the price of a European call option.

1.1 Options

A European option contract is an agreement between two parties that gives one party the right to buy or sell the underlying asset at some specified date in the future for a fixed price. This right has a value, and the party that provides the option will demand compensation at the inception of the option.

Further, once initiated, options can be traded on an exchange much like the underlying asset.

The options under consideration in this report fall into two classes:

- Call Option
- Put Option

1.1.1 Call Option

A European call option on a stock gives the buyer the right but not the obligation to buy a number of shares of stock (the underlying asset for stock options) for a specified price (E) at a specified date (T) in the future.

If the price of the stock (S) is below the exercise price, the call option is said to be out of the money whereas a call option with an exercise price below the stock price is classified as in the money. The value of the option is thus a function of S and time, and its payoff, which describes the option's value at date T, is given by

$$C(S,T) = max(S - E, 0)$$

Fig 1. illustrates the payoff for a European call option with exercise price $E = 25$.

1.1.2 Put Option

A European put option gives the buyer the right to sell a number of shares of stock for a specified price at a specified time. The payoff which describes a put option's value at time T is given by

$$P(S,T) = max(E - S, 0)$$

Fig 2. below illustrates the payoff for a European put option with exercise price E = 25.

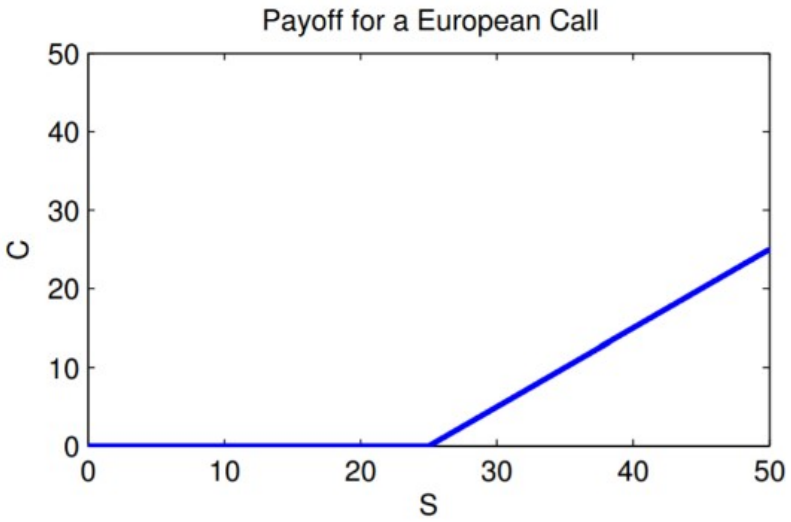

Figure 1: European Call Option

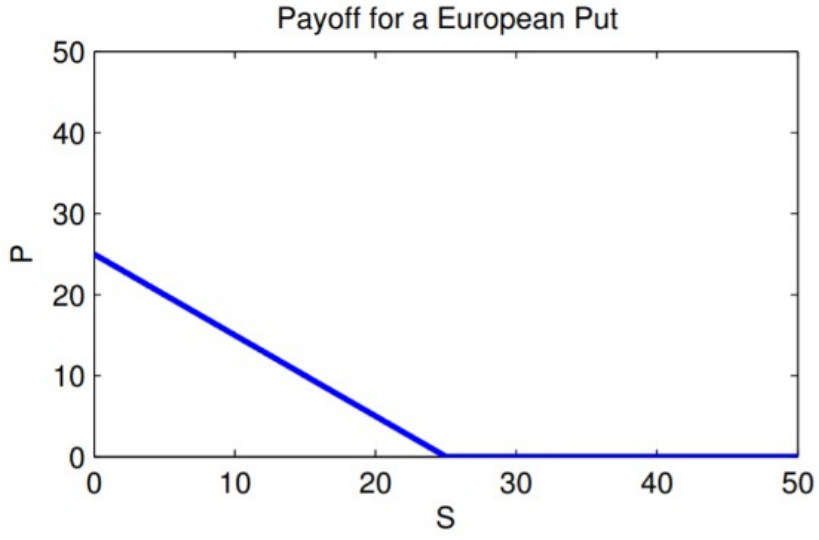

Figure 2: European Put Option

2 The Black-Scholes Model

The Black-Scholes model is a mathematical model for pricing an options contract. In particular, the model estimates the variation over time of financial instruments. It is a widely used model for option pricing.

The inputs for the Black-Scholes equation are volatility, the price of the underlying asset, the strike price of the option, the time until expiration of the option, and the risk-free interest rate. With these variables, it is theoretically possible for options sellers to set rational prices for the options that they are selling. Furthermore, the model predicts that the price of heavily traded assets follows a geometric Brownian motion with constant drift and volatility.

0.1. Assumptions for Black Scholes Model

The Black-Scholes model makes certain assumptions:

- The option is European and can only be exercised at expiration.
- No dividends are paid out during the life of the option.
- Markets are efficient (i.e., market movements cannot be predicted).
- There are no transaction costs in buying the option.
- The risk-free rate and volatility of the underlying are known and constant.
- The returns on the underlying asset are log-normally distributed.

3 Derivation of Black-Scholes Partial Differential Equation

Per the model's assumption, the stock price follows a geometric brownian motion. Let the stock price be $S(t)$ driven by the process.

$$\mathrm{d}S(t) = \mu \mathrm{d}t + \sigma S(t)\mathrm{d}W(t) \tag{1}$$

Let $V(S,t)$ be the value of the option on this stock.

Applying Ito's Lemma to the function $V(S,t)$ yields

$$\mathrm{d}V = \left(\frac{\partial V}{\partial S}\mu S + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2\right)\mathrm{d}t + \frac{\partial V}{\partial S}\sigma S\,\mathrm{d}Z \tag{2}$$

Suppose the derivative can be hedged. Let us replicate a portfolio consisting of shares of stock and derivatives that is instantaneously risk-less, and thus eliminates the stochastic term $\mathrm{d}Z$ in (2).

The instantaneously risk-less portfolio at time t consists of one long position in the derivative and a short position of exactly $\frac{\partial V}{\partial S}$ shares of the underlying stock. The value of this portfolio is given by

$$\Pi = V - \frac{\partial V}{\partial S}\mathrm{d}S \tag{3}$$

Combining (1), (2), (3) we get

$$\mathrm{d}\Pi = -\frac{\partial V}{\partial S}(\mu \mathrm{d}t + \sigma S(t)\mathrm{d}W(t)) + \left(\frac{\partial V}{\partial S}\mu S + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2\right)\mathrm{d}t + \frac{\partial V}{\partial S}\sigma S\,\mathrm{d}Z$$

$$\mathrm{d}\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2\right) \tag{4}$$

Since this portfolio is risk-less, the assumption that there are no arbitrage opportunities dictates that it must earn exactly the risk free rate. This

implies that the right hand side of (4) is equal to $r\Pi \mathrm{d}t$ where r represents the risk free rate of return. Setting the right hand side of (4) equal to $r\Pi \mathrm{d}t$ and using (3) one obtains the famous Black-Scholes partial differential equation:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 + \frac{\partial V}{\partial S} rS - rV = 0 \tag{5}$$

3.1 Solving the Black-Scholes Equation

Once the Black–Scholes PDE is derived for a derivative, the PDE can be solved numerically using standard methods of numerical analysis, such as a type of finite difference method. For certain cases like a European Call option, we have a solution for the equation using methods like the Feynman-Kac Theorem.

During the course, we studied such a solution for European Call and Put Options. However, in this project we shall explore the finite difference method for solving the Black-Scholes Partial Difference Equation.

4 Numerical Solution of Black Scholes PDE

We shall use the finite difference method to solve the PDE numerically and calculate the price of a European Call Option with strike price K.

The finite method involves setting up a grid of points. This grid will be used to simulate the PDE from point to point. The grid on the x-axis will represent the simulated times, which ranges from $[0, 1]$ and the y-axis will represent the possible stock prices, ranging from $[S_{min}, S_{max}]$.

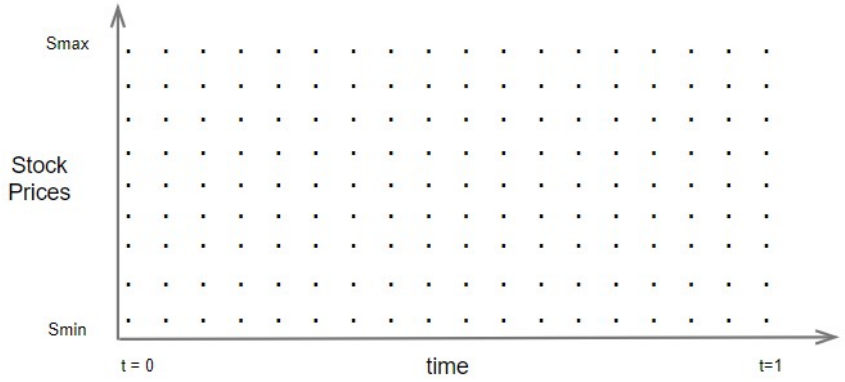

Figure 3: Grid of Points

4.1 Boundary Conditions

To simulate on the grid we need to determine 3 boundaries (edge) conditions of the grid, top, bottom and the last boundary of the grid.

- For the bottom, we set $S_{min} = 0$.
- For the top, the option value V at S_{max} can be deducted using a replication argument. If a derivative pays $S - K$ at time t, we can replicate

We need to ensure the option's value $V = max(S - K, 0)$ will always payout $S - K$. We can do this by setting S_{max} to be 8 standard deviations away from the mean, as the stock price is log-normally distributed. So, if we take $S_{max} = 8\sigma\sqrt{T - t}$, we have ensured that property.

it by purchasing 1 unit of stock and putting $Ke^{-r(1-t)}$ it into a risk-free bank account. So this makes the value of the option for large S: $V(t, S_{max}) = S_{max} - Ke^{-r(1-t)}$.

- The final boundary condition will be the European call option's payoff, as that will give the exact value for the option.

So the three boundary conditions are:

$$V(t, S_{min}) = 0$$

$$V(t, S_{max}) = S_{max} - Ke^{-r(1-t)}$$

$$V(1, S) = max(S - k, 0)$$

In the stock price direction (vertical) we introduce M points. To think about this, imagine that the possible range of S is $[0, 100]$ with $M = 100$. This will make 100 stock points with 1 at every integer. We do the same in the time direction (horizontal) with N steps. This will create a $N+1 \times M+1$ matrix, which we can use to create derivative estimates.

4.2 Estimating Derivatives

With the discretized space, we can use central difference estimates for the derivatives of the option value. Using the Greeks,

$$\Delta = \frac{\partial V(t, S_i)}{\partial S} = \frac{V(t, S_{i+1} - V(t, S_{i-1})}{2\delta S} \tag{6}$$

$$\Gamma = \frac{\partial^2 V(t, S_i)}{\partial S^2} = \frac{V(t, S_{i+1}) - 2V(t, S_i) + V(t, S_{i-1})}{(\delta S)^2} \tag{7}$$

Plugging these into the Black-Scholes PDE, we get

$$\frac{\partial V_i}{\partial t}+\frac{1}{2}\sigma^2 S_i^2\frac{V(t,S_{i+1})-2V(t,S_i)+V(t,S_{i-1})}{(\delta S)^2}+rS_i\frac{V(t,S_{i+1})-V(t,S_{i-1})}{2\delta S}-rV_i\approx 0$$

Using the standard method for numerically solving Stochastic Differential Equations we get the following equation:

$$V_{t+1}\approx V_t-\frac{1}{2}\sigma^2 S_i^2\frac{V(t,S_{i+1})-2V(t,S_i)+V(t,S_{i-1})}{(\delta S)^2}+rS_i\frac{V(t,S_{i+1})-V(t,S_{i-1})}{2\delta S}+rV_t$$

This is a system of ODEs, where V is the column vector of option prices at each timestep. Moreover, we need to add a vector to the above equation that contains the boundary conditions at time t : W_t, and then we can rewrite the equation in matrix notation, such that Λ contains the multipliers. This equation now contains all the information we have and is called the explicit method.

$$V_{t-1}\approx(1-\Lambda\delta t)V_t-\delta tW_t \tag{8}$$

5 Solution in Python

5.1 Importing Libraries

```
import numpy as np
import scipy as scp
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from matplotlib import cm
%matplotlib inline

from scipy import sparse
from scipy.sparse.linalg import splu
from scipy.sparse.linalg import spsolve

from IPython.display import display
import sympy; sympy.init_printing()

def display_matrix(m):
    display(sympy.Matrix(m))
```

5.2 Initializing the parameters

```
r = 0.1; sig = 0.2
S0 = 100; X0 = np.log(S0)
K = 100; Texpir = 1

Nspace = 3000   # M space steps
Ntime = 2000    # N time steps

S_max = 3*float(K)
S_min = float(K)/3

x_max = np.log(S_max)  # A2
x_min = np.log(S_min)  # A1
```

5.3 Discretization

```
x, dx = np.linspace(x_min, x_max, Nspace, retstep=True)   # space
    discretization
T, dt = np.linspace(0, Texpir, Ntime, retstep=True)       # time
    discretization
Payoff = np.maximum(np.exp(x)-K,0)          # Call payoff
```

5.4 Boundary Conditions

```
V = np.zeros((Nspace,Ntime))        # grid initialization
offset = np.zeros(Nspace-2)         # vector to be used for the
    boundary terms

V[:,-1] = Payoff                    # terminal conditions
V[-1,:] = np.exp(x_max) - K * np.exp(-r* T[::-1] )  # boundary
    condition
V[0,:] = 0                          # boundary condition
```

5.5 Construction of the tri-diagonal matrix D

```
sig2 = sig*sig; dxx = dx * dx

a = ( (dt/2) * ( (r-0.5*sig2)/dx - sig2/dxx ) )
b = ( 1 + dt * ( sig2/dxx + r ) )
c = (-(dt/2) * ( (r-0.5*sig2)/dx + sig2/dxx ) )

D = sparse.diags([a, b, c], [-1, 0, 1], shape=(Nspace-2,
    Nspace-2)).tocsc()
```

5.6 Backward iteration

```
for i in range(Ntime-2,-1,-1):
    offset[0] = a * V[0,i]
    offset[-1] = c * V[-1,i];
    V[1:-1,i] = spsolve( D, (V[1:-1,i+1] - offset) )
```

5.7 Finding the option at S_0

```
oPrice = np.interp(X0, x, V[:,0])
print(oPrice)
```

5.8 Plotting Graphs

```
S = np.exp(x)
fig = plt.figure(figsize=(15,6))
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122, projection='3d')

ax1.plot(S, Payoff, color='blue',label="Payoff")
ax1.plot(S, V[:,0], color='red',label="BS curve")
ax1.set_xlim(60,170); ax1.set_ylim(0,50)
ax1.set_xlabel("S"); ax1.set_ylabel("price")
ax1.legend(loc='upper left'); ax1.set_title("BS price at t=0")

X, Y = np.meshgrid(T, S)
ax2.plot_surface(Y, X, V, cmap=cm.ocean)
ax2.set_title("BS price surface")
ax2.set_xlabel("S"); ax2.set_ylabel("t"); ax2.set_zlabel("V")
ax2.view_init(30, -100) # this function rotates the 3d plot
plt.show()
```

6 Outputs

6.1 Option at S_0

```
In [12]: oPrice = np.interp(X0, x, V[:,0])
         print(oPrice)

         13.269144076030782
```

6.2 Graphs

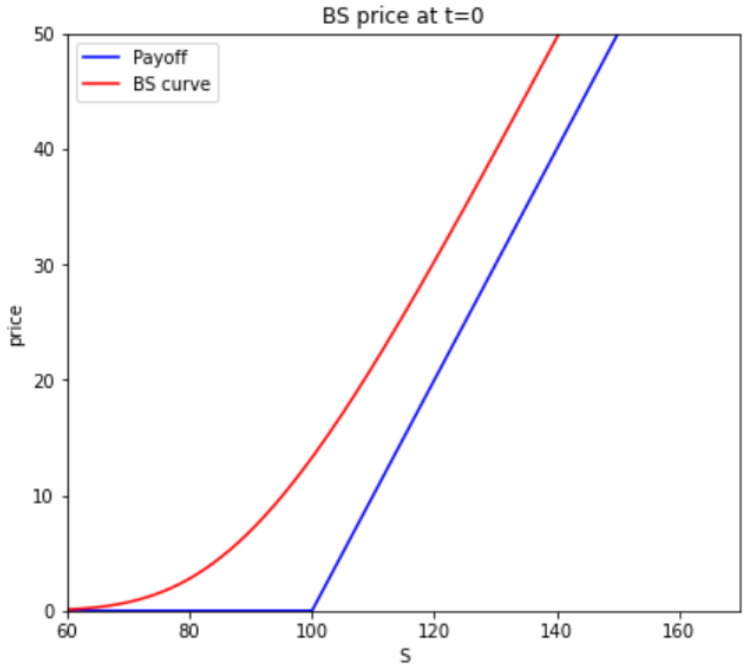

Figure 4: Black Scholes Price at t = 0

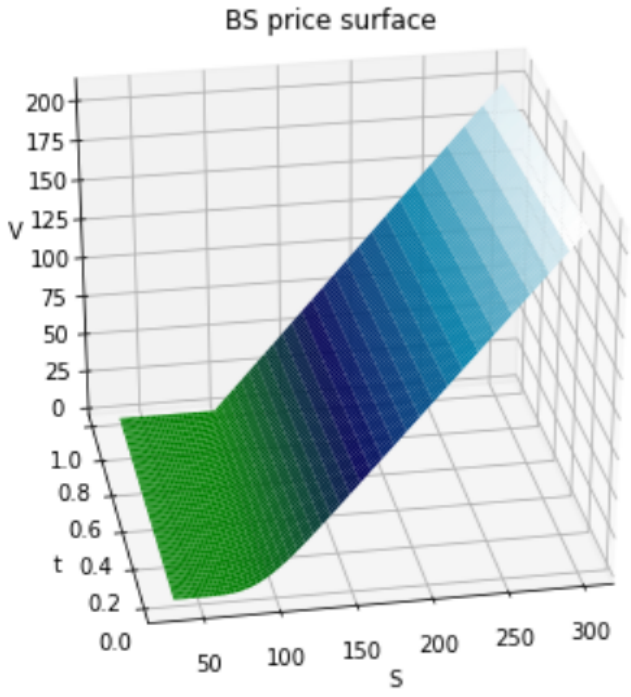

Figure 5: Black Scholes Price Surface

References

[1] "Option pricing using the Black-Scholes model, without the formula", Daniel Reti, *towardsdatascience.com/option-pricing-using-the-black-scholes-model-without-the-formula-e5c002771e2f*

[2] Hull, John, 1946-. Options, Futures, and Other Derivatives. Boston :Prentice Hall, 2012.

[3] "Solving the Black Scholes Equation using a Finite Difference Method", Daniel Hackmann, 2009

[4] "Finite Difference Schemes that Achieve Dynamical Consistency for Population Models" Thirteenth Virginia L. Chatelain Memorial Lecture presented by Talitha Washington at Kansas State University on November 9, 2017

[5] "The Pricing of Options and Corporate Liabilities." Fischer Black and Myron Scholes, Journal of Political Economy. Accessed Aug. 4, 2020.