Eric Shi
1003062552
CIV1174
April 8, 2022

CIV1174 Project – Black-Scholes Equation

1.1Introduction and Background

To better understand this report, it is crucial to understand what an option contract is and what are its basic features. When one buys an option contract, they are purchasing a right but not an obligation to purchase (call option) or sell (put option) an asset at a certain price (strike price) by a certain date (maturity date). The purchaser of the option can choose what they prefer as the strike price and maturity date, and the price of the contract will adjust accordingly. When the maturity date of a call option hits, if the underlying asset's price is at or above the strike price (at-the-money (ATM), in-the-money (ITM)), the purchaser of the call option can choose to exercise the contract and purchase the underlying asset at the strike price but if the underlying asset's price is below the strike price (Out-of-the-money (OTM)), the contract expires worthless because it does not make sense to exercise. Similarly, when the maturity date of a put option hits, if the underlying asset's price is at or below the strike price (ATM, ITM), the purchaser of the put option can choose to exercise the contract and sell the underlying asset at the strike price, but the contract expires worthless if the asset's price is above the strike price (OTM).

The goal of this project was to create and compare the accuracy of multiple options pricing models and evaluate its accuracy based on comparisons of finite element models to a finite difference model and a theoretical solution. There were three models in total, two finite element models and one finite difference model.

The Black-Scholes partial differential equation (PDE) is an equation that is used to determine options contract pricing, which is often used in the stock market. Specifically, this PDE is most commonly used for European option contracts as they cannot be exercised prior to the maturity date, while American options can be exercised prior to maturity. Mathematical adjustments to the Black-Scholes PDE must be made to account for American options. This paper focuses on European options and neglects the existence of the option to exercise contracts early. Equation 1 below is the most common form of the Black-Scholes PDE, and it resembles a general non-steady state advection-diffusion PDE.

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

Where V represents the value of the option contract ($), t represents the time in years since option purchase. $0 \leq t \leq T$ where T is the time in years at contract maturity, S represents the stock price of the asset underlying the option contract ($), σ represents the implied volatility of the option contract, and r represents the risk-free interest rate, typically taken as the 10-year treasury bond yield as a yearly interest rate.

A few of the biggest weaknesses of the Black-Scholes PDE are that it does not account for early exercising of contracts, as explained above, and it assumes that implied volatility and risk-free interest rate are constant, which they are not. Typically, the interest rate does not fluctuate significantly, but it is not uncommon for implied volatility to greatly fluctuate from hour to hour. Therefore, in practice, the resulting finite difference or finite element model must be run periodically to update itself.

Other limitations of the Black-Scholes Model as outlined by Investopedia are that they assume continuous trading, no broker fees or bid/ask pricing spreads, it assumes that stock prices follow

a lognormal pattern (i.e. geometric Brownian motion) which ignores large price swings, it assumes no dividends are paid out, and it assumes the market is run by ideal banks, meaning there are no arbitrage opportunities due to pricing discrepancies (Seth, 2022).

The finite element (FE) method that was used was a 1D application of the Galerkin Method of Weighted Residuals (GMWR) in the price dimension, followed by a 1D application of the Crank-Nicolson Method (CNM) in the temporal dimension.

1.2Boundary Conditions

In order to set boundary conditions for this problem, a simple time transformation must be performed as the current form of equation 1 is a backward-looking equation. There are terminal boundary conditions that must be flipped to become initial conditions. The transformation is done using equations 2 to 4 below.

$$\tau = T - t \tag{2}$$

$$d\tau = -dt \tag{3}$$

$$-\frac{\partial V}{\partial \tau} - rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0 \tag{4}$$

Where τ represents the time in years until contract maturity, where $\tau = 0$ is the maturity date and $\tau = T$ is the date of option contract purchase.

The boundary conditions of the Black-Scholes PDE are specified below.

For call option contracts:

$$V(0,\ \tau) = 0$$

$$V(\infty, \tau) = S - Ke^{-r\,\tau}$$

$$V(S, 0) = \max\,(S - K, 0)$$

For put option contracts:

$$V(0,\ \tau) =\ Ke^{-r\,\tau}$$

$$V(\infty, \tau) = 0$$

$$V(S, 0) = \max\,(K - S, 0)$$

Where K is defined as the strike price of the option contract. The $e^{-r\,\tau}$ is the application of the time value of money and it represents the discounting of money in time from the maturity date to any date of interest to determine what money is worth as it moves throughout time. Discounting is the concept of decreasing future money back to the present at a given interest rate, compounding is the concept of increasing present money to the future at a given interest rate. In this case, compound interest is applied continuously, meaning interest is applied at every infinitesimal time increment.

Note that an upper bound asset price of infinity is impossible, so the upper bound was set to a small multiple of the current asset price (anywhere from 2x to 5x). This results in a mesh that provides sufficient information as it is quite rare to more than double the value of an asset over the timeframes that stock option contracts are typically offered at. Additionally, having lower

upper bounds on asset price creates a mesh that does not require as many elements to converge because each element is smaller using the same number of elements and a uniform mesh.

This project focuses only on call option contracts for simplicity and to avoid redundancy. Put option contracts can be found with changes in boundary conditions, and no changes in any other part of the FE model.

1.3Global to Local Transformation

Similar to most 1D linear FE approximations, a global to local coordinate transformation simplifies the problem by homogenizing the integrations. Table 1 below summarizes the 1D linear iso-parametric transformations required to complete the FE model.

Table 1 – Summary of Iso-Parametric Global to Local Transformations

Variable	Global Coordinates	Local Coordinates
N_1^e	$\frac{(S_2^e - S)}{\Delta S}$	$\frac{1}{2}(1-\eta)$
N_2^e	$\frac{(S - S_1^e)}{\Delta S}$	$\frac{1}{2}(1+\eta)$
$\frac{dN_1^e}{d\eta}$	-	$-\frac{1}{2}$
$\frac{dN_2^e}{d\eta}$	-	$\frac{1}{2}$
$\frac{dN_1^e}{dS}$	$-\frac{1}{\Delta S}$	-
$\frac{dN_2^e}{dS}$	$\frac{1}{\Delta S}$	-

Limits of Integration	$\int_{S_1^e}^{S_2^e}$	$\int_{-1}^{1}$
$\frac{dS}{d\eta}$	$\frac{\Delta S}{2}$	$\frac{\Delta S}{2}$

For greater detail on the global to local transformation, refer to Appendix B page 1 for the hand calculations.

2.1Finite Element Derivation – Advection-Diffusion PDE

Applying GMWR to equation 4 results in equation 5 below.

$$\int_{S_1^e}^{S_2^e} \left(-\frac{\partial V}{\partial \tau} - rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV \right) N_i dS = 0 \tag{5}$$

Where N_i is a row vector of the element shape functions. V can be approximated by equation 6 below.

$$V^{\sim}(S,\tau) = \sum_{i=1}^{2} N_i(S) * V_i(\tau) \tag{6}$$

Using integration by parts and global to local transformations, equation 5 simplifies into the form of equation 7 below.

$$[K] * \frac{\partial V_j}{\partial \tau} + [C] * V_j = [\alpha] \tag{7}$$

Applying the Crank-Nicolson method to equation 7 results in equation 8 below.

$$[K] * \frac{V_j^{\tau+\Delta\tau} - V_j^{\tau}}{\Delta\tau} = \theta\left([\alpha] - [C] * V_j^{\tau+\Delta\tau}\right) + (1-\theta)\left([\alpha] - [C] * V_j^{\tau}\right) \tag{8}$$

Where $\theta = \frac{1}{2}$ and $[\alpha]$ is a by-product from using integration by parts and was canceled out in the derivation. The matrices in equation 8 are defined below as:

$[K] = \frac{\Delta S}{2}\int_{-1}^{1} N_j N_i d\eta$, where N_j is a column vector of the element shape functions.

$[C] = \frac{\Delta S}{2}\left((\sigma^2 - r)\int_{-1}^{1} S\frac{\partial N_j}{\partial S}\, N_i d\eta + \frac{1}{2}\sigma^2\int_{-1}^{1} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}d\eta + r\int_{-1}^{1} N_j N_i d\eta\right)$.

$[\alpha] = \frac{1}{2}\sigma^2 S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_1}^{S_2}$.

See Appendix B pages 1 to 3 for the hand calculations of the full derivation using GMWR and Crank-Nicolson on the classical Black-Scholes PDE.

2.2Transformation to Simplified Diffusion/Heat Equation

Using principles from Stochastic calculus, such as Itô's Lemma, and assuming Geometric Brownian Motion, the Black-Scholes PDE was transformed into a simplified diffusion equation shown in equation 9 below (quantpie, 2019).

$$\frac{\partial F}{\partial \tau} = \frac{1}{2}\sigma^2\frac{\partial^2 F}{\partial x^2} \tag{9}$$

Where $F = Ve^{r\,\tau}$, $x = \ln(S) + (r - \frac{1}{2}\sigma^2)\tau$, and the boundary conditions of the call option become:

$F(0,\ \tau) = 0$

$F(\infty, \tau) = (e^{x - \left(r - \frac{1}{2}\sigma^2\right)\tau} - Ke^{-r\,\tau}) * e^{r\,\tau}$

$F(x, 0) = \max\,(e^x - K, 0)$

See Appendix B page 5 for the full derivation of the transformation to this simplified PDE form.

2.3Finite Element Derivation – Heat Equation

Applying GMWR to equation 9 results in equation 10 below.

$$\int_{S_1^e}^{S_2^e} \left(\frac{\partial F}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial x^2}\right) N_i dx = 0 \tag{10}$$

Using the same methodology as equations 6, 7, and 8, equation 10 simplifies to equation 11 below after applying GMWR and Crank-Nicolson method.

$$[K] * \frac{F_j^{\tau+\Delta\tau} - F_j^{\tau}}{\Delta\tau} = \theta\left([\alpha] - [C] * F_j^{\tau+\Delta\tau}\right) + (1-\theta)\left([\alpha] - [C] * F_j^{\tau}\right) \tag{11}$$

Where $\theta = \frac{1}{2}$ and $[\alpha]$ is a by-product from using integration by parts and was canceled out in the derivation. The matrices in equation 11 are defined below as:

$[K] = \frac{\Delta x}{2}\int_{-1}^{1} N_j N_i d\eta$, where N_j is a column vector of the element shape functions.

$[C] = \frac{\Delta x}{2}\left(\frac{1}{2}\sigma^2 \int_{-1}^{1} \frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} d\eta\right)$.

$[\alpha] = \frac{1}{2}\sigma^2 N_i \left(\frac{\partial F^{\sim}}{\partial x}\right)\Big|_{x_1}^{x_2}$.

See Appendix B page 6 for the hand calculations of the derivation using GMWR and Crank-Nicolson on the simplified Black-Scholes PDE.

2.4Control Groups and Comparisons

To act as a control baseline comparison, finite difference approximation was applied to equation 4 using central differencing in the price dimension, and Crank-Nicolson Method in the temporal dimension. Since the derivation is near identical to Lecture 3, and the application is near identical to Assignment 3, the derivation steps will not be summarized in the body of this report. See Appendix B page 4 for the hand calculations of the finite difference derivation using the standard Black-Scholes PDE.

Additionally, there is a form of the Black-Scholes equation that simplifies into an equation form that can be solved and plotted for any given time. This was derived from the standard convolution method and is calculated using standard normal cumulative distribution functions as shown in equation 12 below (Shorif Hossan et al., 2020).

$$Call\ Option\ Price = S * N(d_1) - Ke^{-r\tau} * N(d_2) \tag{12}$$

Where $N(\,)$ is defined as the standard normal cumulative distribution function, and d_1 and d_2 are defined as followed.

$$d_1 = \frac{\ln\left(\frac{S}{K}\right)+\left(r+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}},\ d_2 = \frac{\ln\left(\frac{S}{K}\right)+\left(r-\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}$$

For this project, equation 12 was plotted at $\tau = T$ with every FE approximated model to evaluate the accuracy of each model. Additionally, equation 12 was solved for at the given stock price and $\tau = T$, to represent the current option price, which was then compared with the FE approximation of the current option price.

It is difficult to compare these results with real markets as the assumptions made in Black-Scholes equation are ideal. In reality, markets are not as price efficient, compounding styles vary,

brokerage fees exist, arbitrage opportunities and pricing spreads exist, etc. Therefore, accuracy was determined based on comparison with equation 12.

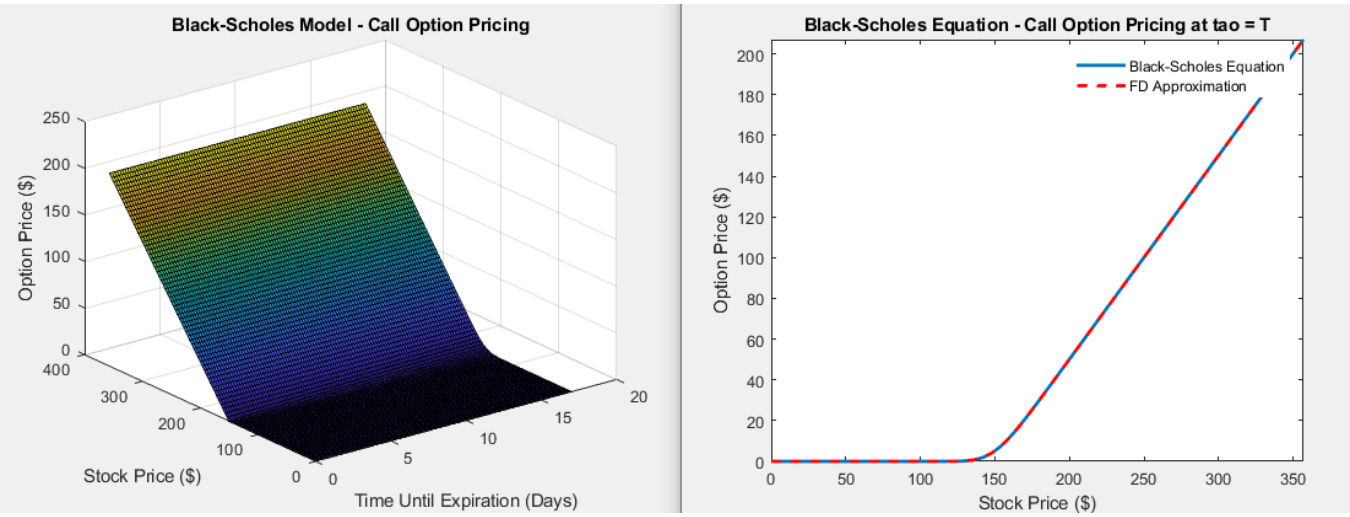
3.1 Results

Note that the blue line on the right-side graph for Figures 1 to 9 represents a plot of equation 12.

3.1.1 Example: Apple Stock

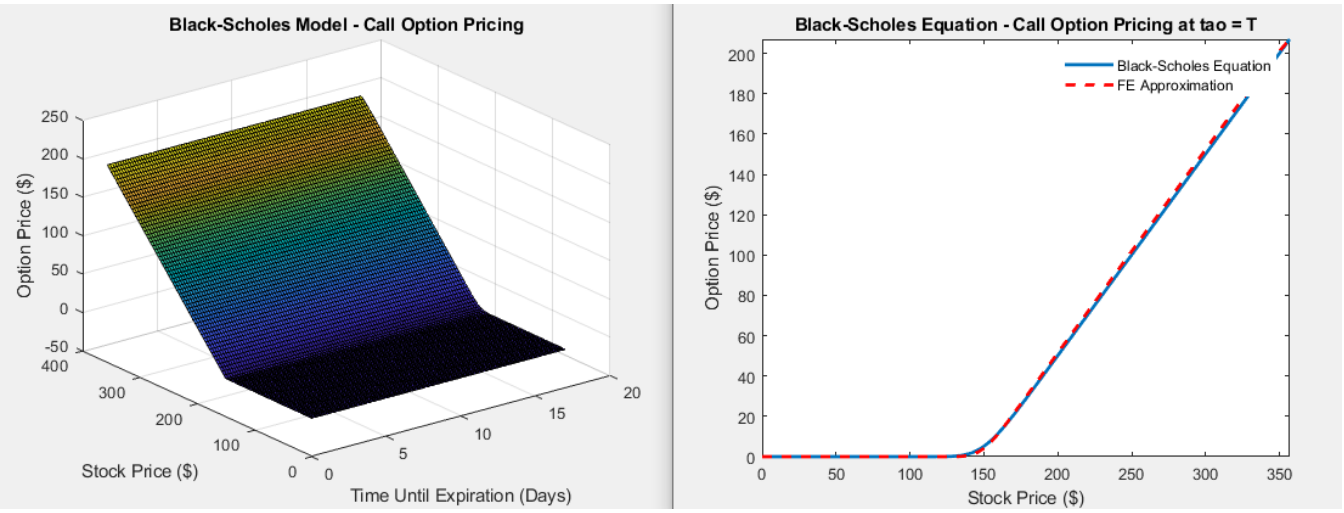
For the first iteration of results, all models attempted to solve the call option price of Apple stock with the following given information:

- Purchase date April 5, 2022. Maturity date April 22, 2022. 17 days until expiration.
- Strike price: K = $150
- Stock price: S = $178.44
- Implied volatility: σ = 39.43%
- Risk-free rate: r = 2.441%
- 0.17-day time-steps (100-time steps in total)
- 100-price steps in total
- Upper bound of 2*S = $356.88



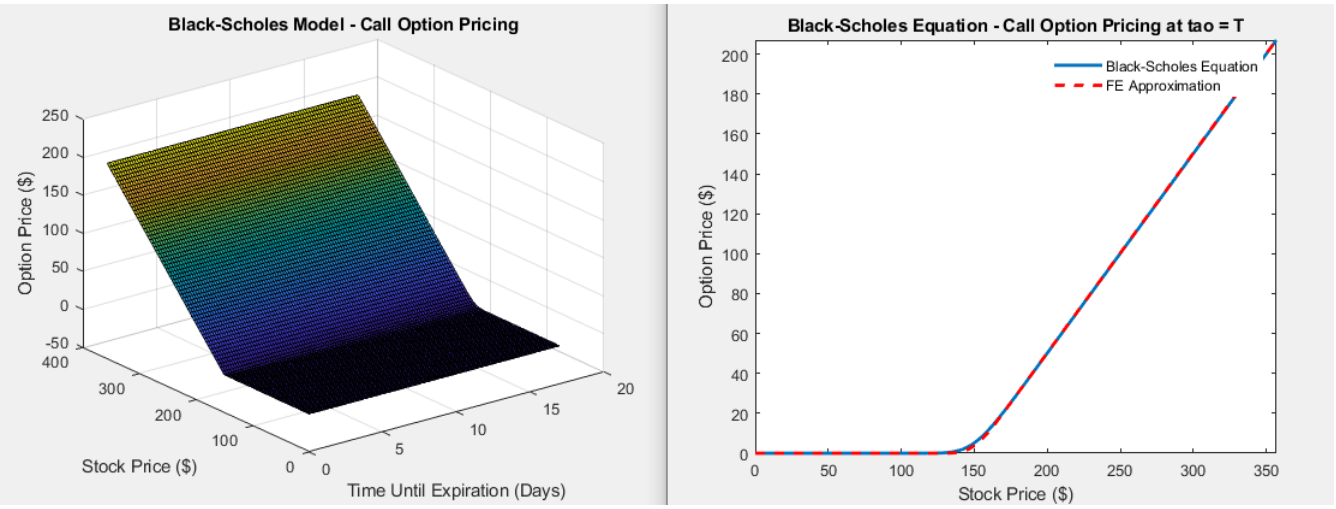

```
The current option price using Finite Difference Method is $28.7152
The current option price using Black-Scholes Equation is $28.7122>>
```

Figure 1 – Finite Difference Model – Apple Stock Example




```
The current option price using Finite Element Method is $29.7014
The current option price using Black-Scholes Equation is $28.7122>>
```

Figure 2 – Advection-Diffusion FE Model – Apple Stock Example



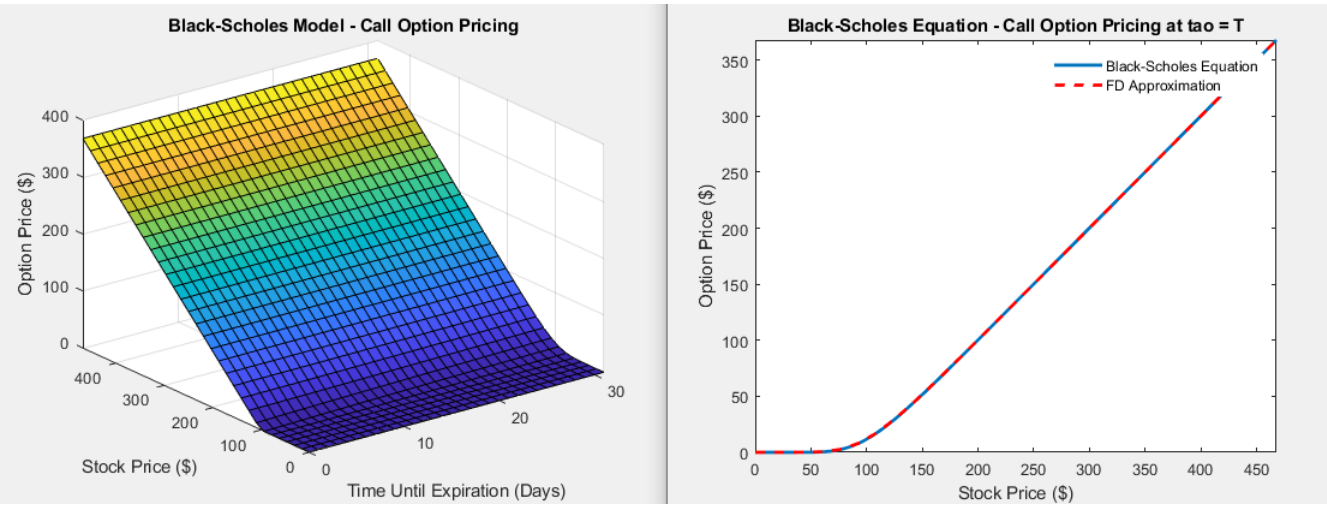

```
The current option price using Finite Element Method is $28.7261
The current option price using Black-Scholes Equation is $28.7122>>
```

Figure 3 – Heat FE Model – Apple Stock Example

3.1.2 Example: Facebook Stock

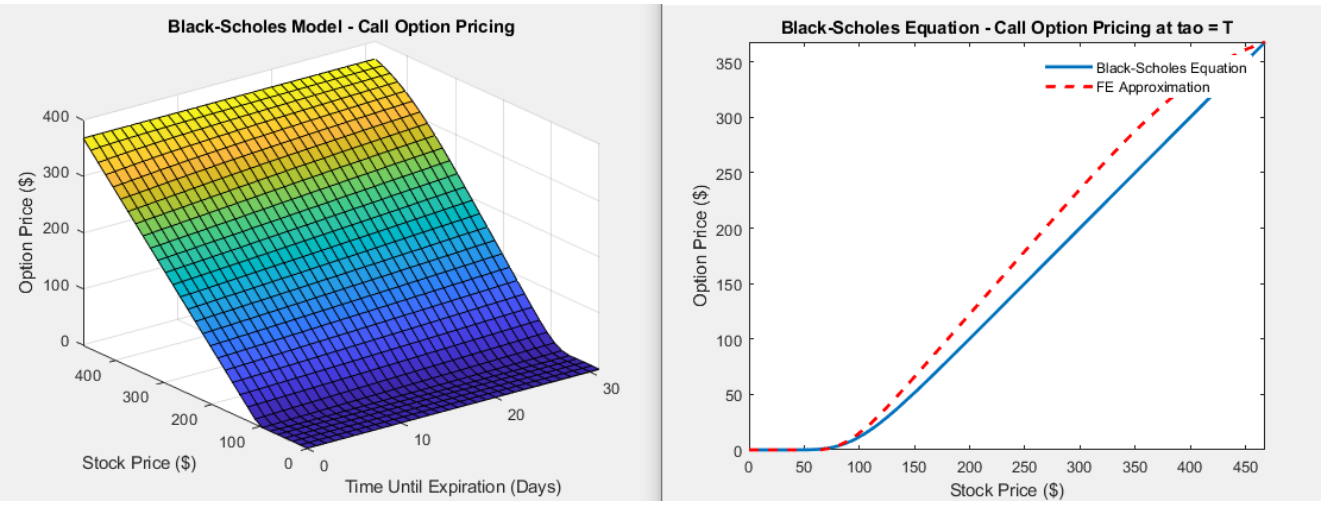
Test all models with Meta/Facebook stock with the following given information:

- Purchase date April 5, 2022. Maturity date May 6, 2022. 31 days until expiration.
- Strike price: K = $100
- Stock price: S = $233.89
- Implied volatility: σ = 100.49%
- Risk-free rate: r = 2.441%
- 1-day time-steps (31-time steps in total)
- 30-price steps in total
- Upper bound of 2*S = $467.78



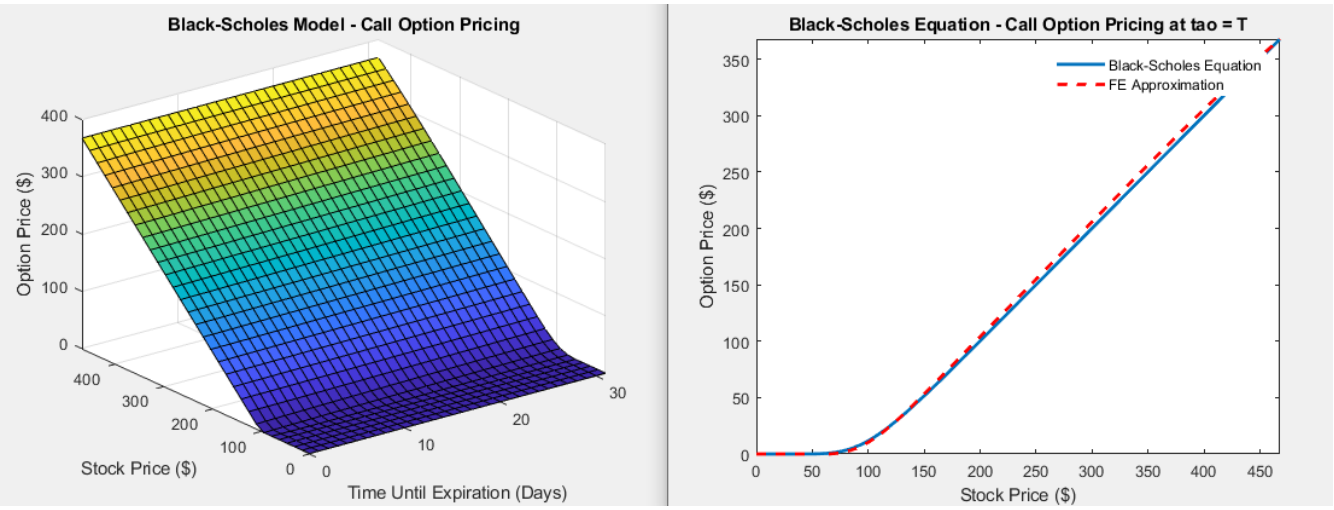

```
The current option price using Finite Difference Method is $134.1365
The current option price using Black-Scholes Equation is $134.1202>>
```

Figure 4 – Finite Difference Model – Facebook Stock Example




```
The current option price using Finite Element Method is $160.9869
The current option price using Black-Scholes Equation is $134.1202>>
```

Figure 5 – Advection-Diffusion FE Model – Facebook Stock Example



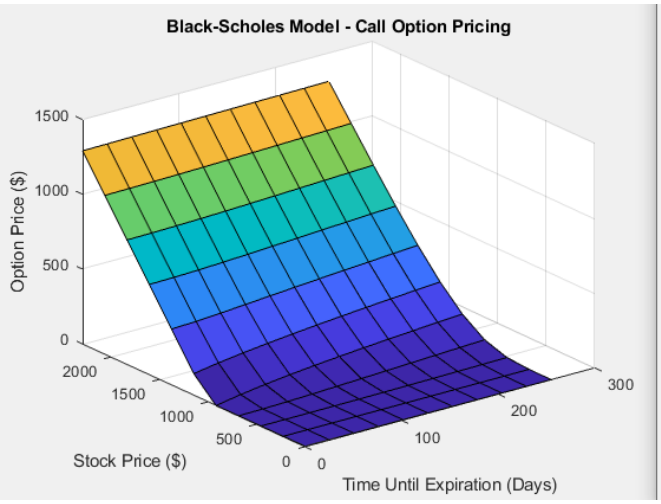

```
The current option price using Finite Element Method is $138.6688
The current option price using Black-Scholes Equation is $134.1202>>
```

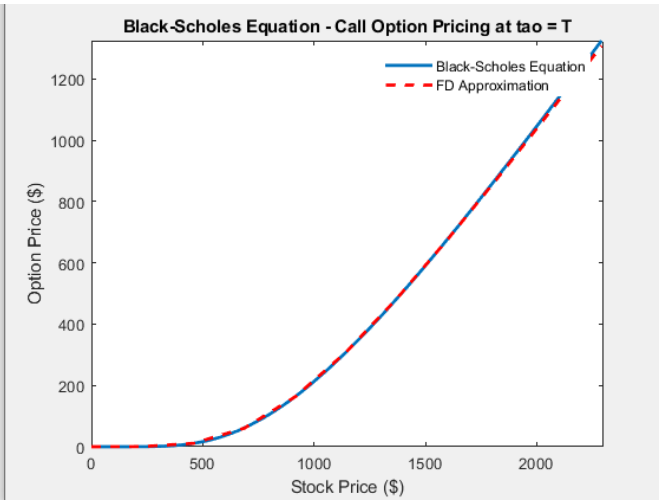
Figure 6 – Heat FE Model – Facebook Stock Example

3.1.3 Example: Tesla Stock

Test all models with Tesla stock with the following given information:

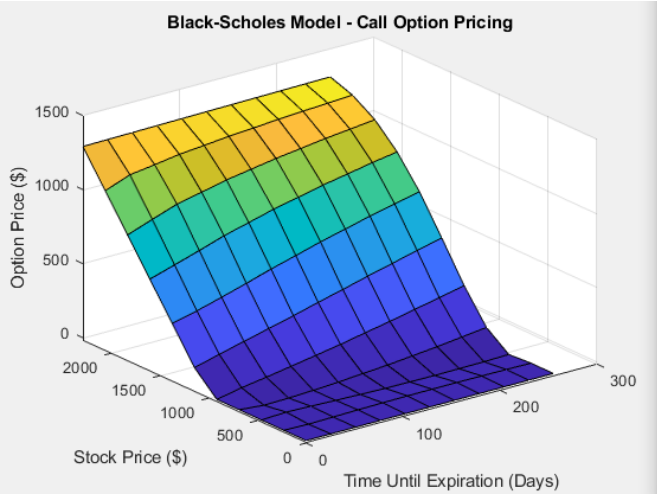
- Purchase date April 5, 2022. Maturity date Dec 16, 2022. 255 days until expiration.
- Strike price: K = $1000
- Stock price: S = $1145.45
- Implied volatility: σ = 62.51%
- Risk-free rate: r = 2.441%
- 25.5-day time-steps (10-time steps in total)
- 10-price steps in total
- Upper bound of 2*S = $2290.90

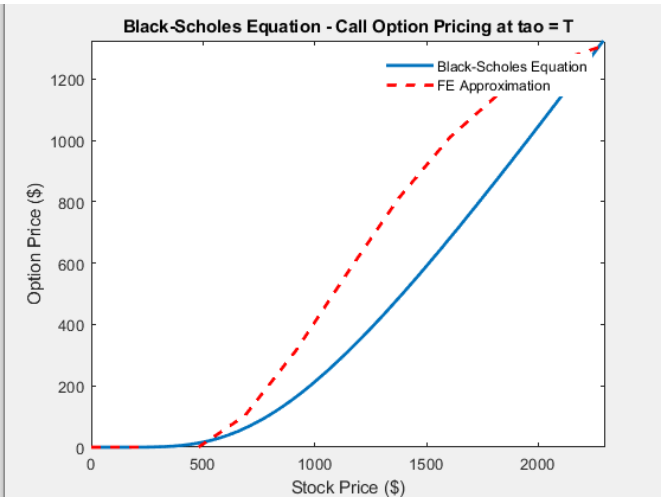





```
The current option price using Finite Difference Method is $309.7665
The current option price using Black-Scholes Equation is $309.5309>>
```

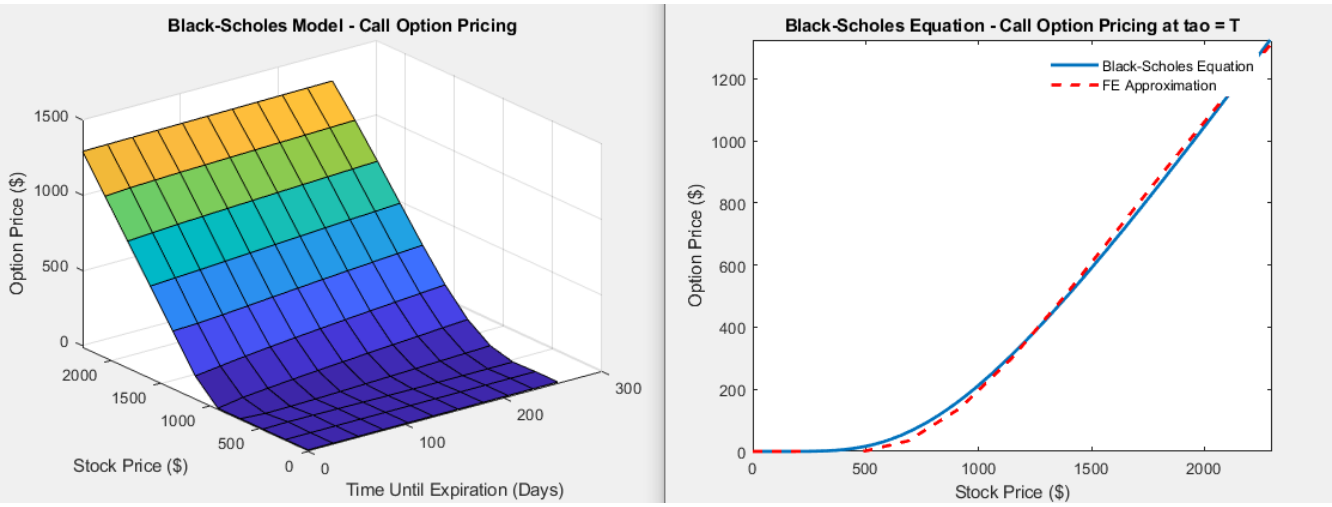
Figure 7 – Finite Difference Model – Tesla Stock Example






```
The current option price using Finite Element Method is $567.5594
The current option price using Black-Scholes Equation is $309.5309>>
```

Figure 8 – Advection-Diffusion FE Model – Tesla Stock Example




```
The current option price using Finite Element Method is $299.0810
The current option price using Black-Scholes Equation is $309.5309>>
```

Figure 9 – Heat FE Model – Tesla Stock Example

For a coarse mesh like the one used in this example, it is easier to verify that all boundary conditions match each other.

```
V_total =

   1.0e+03 *

         0         0         0         0         0         0         0         0         0         0         0
         0    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001    0.0002    0.0002    0.0003    0.0005
         0    0.0000    0.0002    0.0007    0.0014    0.0024    0.0037    0.0052    0.0070    0.0089    0.0111
         0    0.0016    0.0058    0.0115    0.0181    0.0251    0.0323    0.0395    0.0467    0.0537    0.0606
         0    0.0290    0.0525    0.0722    0.0892    0.1042    0.1179    0.1305    0.1421    0.1531    0.1635
    0.1455    0.1721    0.1940    0.2131    0.2304    0.2461    0.2606    0.2741    0.2867    0.2986    0.3098
    0.3745    0.3805    0.3916    0.4041    0.4168    0.4293    0.4415    0.4532    0.4645    0.4753    0.4856
    0.6036    0.6062    0.6110    0.6180    0.6263    0.6351    0.6442    0.6534    0.6624    0.6712    0.6797
    0.8327    0.8346    0.8374    0.8414    0.8465    0.8524    0.8587    0.8652    0.8717    0.8782    0.8845
    1.0618    1.0636    1.0656    1.0681    1.0713    1.0749    1.0788    1.0828    1.0869    1.0910    1.0950
    1.2909    1.2926    1.2943    1.2960    1.2977    1.2994    1.3011    1.3028    1.3044    1.3061    1.3078
```

Figure 10 – Finite Difference Pricing – Tesla Stock (Rows: Stock Price, Columns: τ)

V_total =

1.0e+03 *

0	0	0	0	0	0	0	0	0	0	0
0	-0.0038	-0.0007	0.0022	0.0045	0.0059	0.0059	0.0058	0.0043	0.0029	0.0003
0	0.0080	0.0026	-0.0032	-0.0086	-0.0129	-0.0147	-0.0165	-0.0155	-0.0144	-0.0106
0	-0.0151	-0.0118	-0.0049	0.0050	0.0173	0.0298	0.0462	0.0626	0.0827	0.1028
0	0.0335	0.0597	0.0874	0.1161	0.1451	0.1781	0.2099	0.2454	0.2791	0.3156
0.1455	0.1897	0.2319	0.2730	0.3144	0.3592	0.4009	0.4456	0.4861	0.5288	0.5676
0.3745	0.4137	0.4604	0.5089	0.5604	0.6058	0.6531	0.6940	0.7363	0.7729	0.8106
0.6036	0.6547	0.7063	0.7610	0.8042	0.8488	0.8846	0.9215	0.9518	0.9830	1.0092
0.8327	0.8922	0.9500	0.9856	1.0229	1.0495	1.0775	1.0988	1.1213	1.1391	1.1578
1.0618	1.1240	1.1469	1.1716	1.1863	1.2026	1.2139	1.2263	1.2356	1.2458	1.2538
1.2909	1.2926	1.2943	1.2960	1.2977	1.2994	1.3011	1.3028	1.3044	1.3061	1.3078

Figure 11 – Advection-Diffusion FE Pricing – Tesla Stock (Rows: Stock Price, Columns: τ)

V_total =

1.0e+03 *

0	0	0	0	0	0	0	0	0	0	0
0	-0.0001	-0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
0	0.0033	0.0019	0.0000	-0.0018	-0.0034	-0.0046	-0.0054	-0.0058	-0.0058	-0.0055
0	-0.0099	-0.0086	-0.0049	-0.0001	0.0053	0.0108	0.0165	0.0222	0.0280	0.0338
0	0.0270	0.0429	0.0572	0.0704	0.0827	0.0947	0.1060	0.1172	0.1278	0.1383
0.1455	0.1662	0.1863	0.2035	0.2192	0.2343	0.2482	0.2619	0.2746	0.2873	0.2991
0.3745	0.3804	0.3927	0.4061	0.4202	0.4332	0.4466	0.4588	0.4712	0.4826	0.4941
0.6036	0.6140	0.6220	0.6331	0.6431	0.6542	0.6640	0.6746	0.6840	0.6939	0.7028
0.8327	0.8438	0.8550	0.8620	0.8705	0.8773	0.8851	0.8919	0.8993	0.9059	0.9129
1.0618	1.0735	1.0793	1.0855	1.0896	1.0946	1.0986	1.1032	1.1073	1.1117	1.1157
1.2909	1.2926	1.2943	1.2960	1.2977	1.2994	1.3011	1.3028	1.3044	1.3061	1.3078

Figure 12 – Heat FE Pricing – Tesla Stock (Rows: Stock Price, Columns: τ)

All three models have explicitly defined boundary conditions, which is generally easy to implement. However, the heat equation FE model requires boundary condition variable transformations back to stock price (S) and time (τ) from F, and x. Since the entire first column, first row, and last row of the V_total matrices match each other for all three models, it was concluded that the heat equation FE model had a successful transformation of boundary condition variables back to stock price and time.

See Appendix B for the detailed derivations underlying each of these code outputs and see Appendix A for the full MATLAB scripts used to attain each graph and code output.

3.2 Analysis

From observation of the examples in section 3.1, it can be immediately recognized that the FE model using equation 4 (Advection-Diffusion PDE) probably has an implementation flaw either in its derivation or its MATLAB script as its pricing behavior is different than the other models and its pricing accuracy is far worse than the other models. However, the derivation of the model follows the same logic and sequences as the FE model using equation 9 (Diffusion/Heat PDE). Assuming that the implementation of the FE model using the Heat PDE is correct, the errors of the FE model using the Advection-Diffusion PDE should not be a fundamental flaw in logic and is likely caused by human errors or typos in the derivation or code. Further analysis from this point will neglect the FE model using the Advection-Diffusion PDE as it is incorrect.

One unexpected trend found in section 3.1 is that the finite difference model is the most accurate of the three models for all examples, when traditionally, the finite element method should be more accurate than the finite difference method.

3.3 Sanity Checks

It was considered unusual that the Heat FE model was less accurate than the finite difference model, as FE models are usually more accurate that finite difference models at the cost of higher computation time. Two procedures were completed to act as sanity checks to ensure that the Heat FE model was implemented correctly.

The first sanity check was to apply the same GMWR plus CNM to the concrete chloride concentration problem in assignment 3 to see if the resulting model would converge to the exact solution. Since the governing PDE for the Heat FE model heavily resembled the governing PDE for the concrete chloride concentration problem, the derivation and code implementation of the concrete chloride concentration problem was near identical to the Heat FE model. The only differences between the two problems were that the boundary conditions were different from each other, and the coefficient in front of the double derivative with respect to the space dimension were different. See Appendix B page 7 for the detailed derivation of the concrete chloride concentration FE model using GMWR and CNM, and Appendix A for the full MATLAB script that creates the concrete chloride concentration FE model and note its high resemblance to the derivation and MATLAB script of the Heat FE model.

Using 100 time-steps, and 100 elements, the resulting solution to assignment 3 using GMWR plus CNM is shown in the graph below.

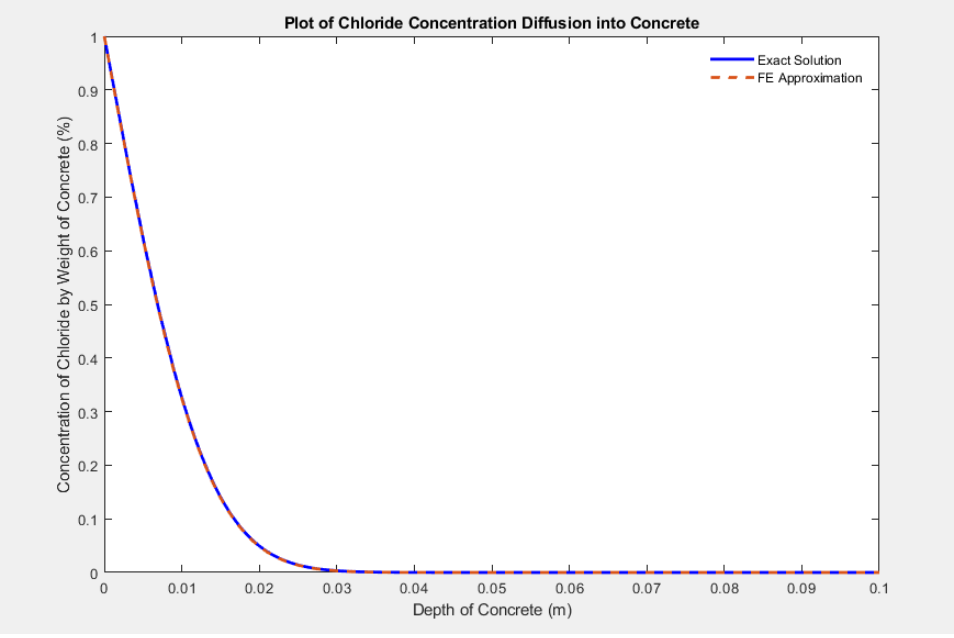


Figure 13 – Finite Element Implementation Sanity Check – Concrete Chloride Problem

Since the FE model for the concrete chloride concentration problem converged to the exact theoretical solution using GMWR and CNM, it was concluded that this model was implemented correctly, and by extension, the Heat FE model for Black-Scholes PDE was also implemented correctly.

The second sanity check was to compare my results with other scholarly articles that also applied GMWR and CNM to the Black-Scholes equation. Most other papers that created options pricing models only used finite difference methods because the granularity of options pricing models are coarse enough that discrete solutions are usually sufficient, so it was difficult to find comparable articles. One particular article from the Department of Applied Mathematics at the University of Dhaka, Bangladesh decided to perform the same comparison with one finite difference model and one finite element model on the Black-Scholes equation. For their finite difference model, they applied the Du Fort-Frankel finite difference method, which is a slightly different finite difference scheme than the CNM. For their finite element model, they applied GMWR plus CNM to the advection-diffusion version of the Black-Scholes PDE (Shorif Hossan et al., 2020). As a result, the article used the very similar methods compared to this project, different paths to the same destination. The four figures below highlight the results of their paper (Shorif Hossan et al., 2020).

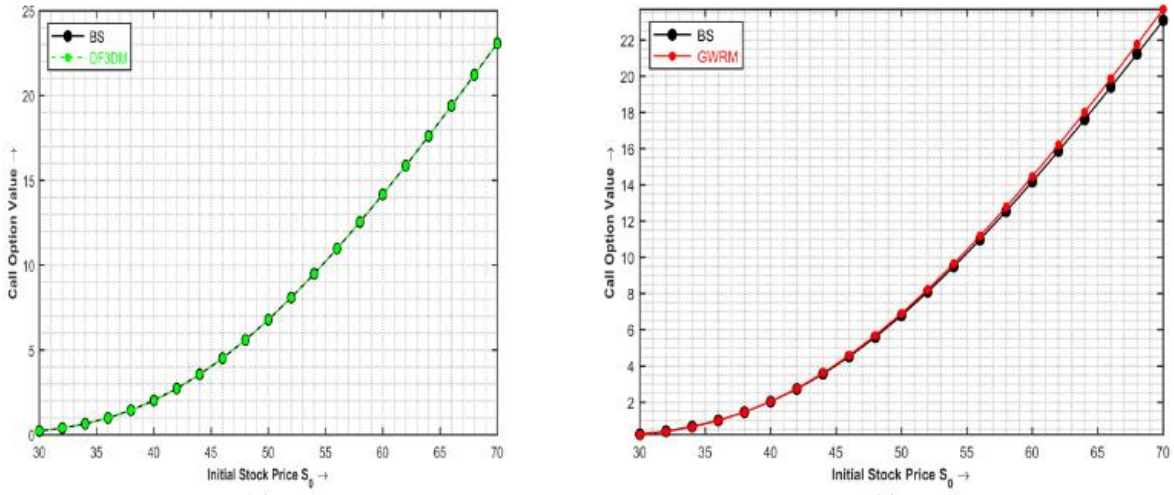


Figure-2. Comparing approximate call option value (by using (a) DF3DM, (b) GWRM) with exact (BS) value for data set 1.

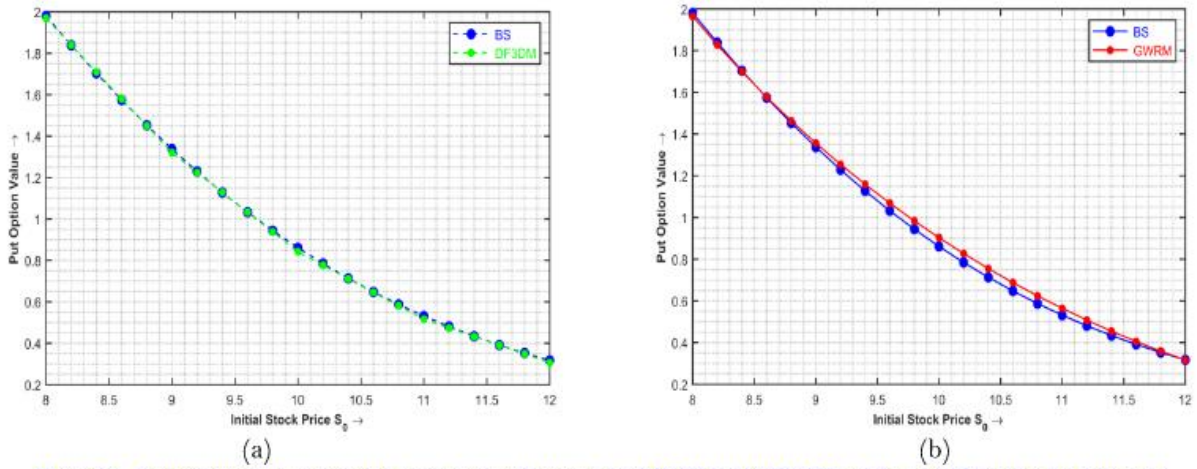


Figure-3. Comparing approximate put option value (by using (a) DF3DM, (b) GWRM) with exact (BS) value for data set 2.

Figure 14 – Finite Difference Method vs Finite Element Method (Shorif Hossan et al., 2020)

Where the paper also concluded that the results obtained from the finite difference model were better than the results obtained by the finite element model, and that the finite element model tends to deviate from the solution found using equation 12 at certain locations (Shorif Hossan et al., 2020). However, this paper did not suggest any reasons as to why this might have occurred.

The second sanity check supported the Heat FE model as they both concluded that the finite difference model was superior to the finite element model, but it also reaffirms that the Advection-Diffusion FE model was flawed as their paper implemented their FE model using the Advection-Diffusion version of the Black-Scholes PDE with different results than this project.

4.1 Conclusion

To determine European option pricing, numerical models for the Black-Scholes PDE were created using finite difference and finite element methods. The resulting models account for both call option pricing and put option pricing because they are distinguished only by their boundary conditions. This report focused mainly on call option pricing. However, Black-Scholes PDE does not account for American options as it does not consider the effects of having the ability to exercise in-the-money options prior to maturity.

Three options pricing models were created and compared with each other and compared with a theoretical Black-Scholes solution as shown in equation 12. One model using strictly finite difference methods with central differencing in the stock price dimension, and Crank-Nicolson Method in the time dimension. The other two models were created using a combination of finite element and finite difference methods, where Galerkin Method of Weighted Residuals was applied in the stock price dimension and Crank-Nicolson Method was applied in the time dimension. The difference between the two finite element models were that one was created using the classical Black-Scholes PDE, resembling an Advection-Diffusion PDE, and the other was created using a transformed Black-Scholes PDE with transformed boundary conditions, resembling a Heat/Diffusion PDE.

It was concluded that the finite element model created using the Advection-Diffusion PDE was not flawed in its derivation logic but flawed with potential human errors or typos either in the code or the derivation, and the flaw(s) have yet to be found. For some unknown reason, the finite difference model was more accurate than the Heat PDE finite element model when compared to equation 12. This apparent relationship was consistent with another article from the University of Dhaka that also attempted a very similar comparison. As a result, it was concluded that the finite difference model and the finite element model created using the Heat PDE were implemented correctly despite the oddity in the finite element model being less accurate than the finite difference model.

Since the finite difference model is simpler to derive, simpler to implement onto a program, requires less computation time to run, and is more accurate than the finite element model using GMWR and Crank-Nicolson Method, a finite difference Black-Scholes model is recommended over a finite element Black-Scholes model in practical applications.

References

quantpie. (2019). *Transformation of Black Scholes PDE to Heat Equation*. *YouTube*. Retrieved April 6, 2022, from https://www.youtube.com/watch?v=IgMoOcO095U.

Seth, S. (2022, March 14). *Circumventing the limitations of Black-Scholes*. Investopedia. Retrieved April 6, 2022, from https://www.investopedia.com/articles/active-trading/041015/how-circumvent-limitations-blackscholes-model.asp

Shorif Hossan, M., Hossain, A. B., & Shafiqul Islam, M. (2020). Numerical Solutions of Black-Scholes model by Du Fort-Frankel FDM and Galerkin WRM. *International Journal of Mathematical Research*, *9*(1), 1–10. https://doi.org/10.18488/journal.24.2020.91.1.10

Appendix A – MATLAB Scripts

Black-Scholes PDE Finite Difference Model:

```
clc, clear, close all
%% Given Data
K = 1000; % Strike Price
Stock_Price = 1145.45; % Current Stock Price
sigma = 62.51/100; % Implied Volatility
r = 0.02441; % Assume 2 Percent. Close to US 10 Year Treasury Yield

T = 255/365.25; % Years Until Expiration
dtao = 25.5/365.25; % 1 Day Time Steps

L = 2*Stock_Price; % Maximum Stock Price
n_blocks = 10; % Number of Elements
dS = L/n_blocks; % Price Step

n_nodes = n_blocks + 1; % Mesh Centered Grid
n_free = n_nodes - 2; % Number of Free Nodes

S = (0:dS:L)'; % Stock Price Vector from $0 to Maximum Stock Price
tao = (0:dtao:T)'; % Tao Vector, where tao = T - t
n_time = length(tao); % Number of Time Steps

V_1 = 0; % Option Price at Stock Price = $0 - Boundary Condition ($)
V_n = L - K*exp(-r*tao); % Option Price at Maximum Stock Price - Boundary
Condition ($)
V = max(S-K, 0); % Free Global Option Price Vector at tao = 0 - Initial
Condition ($)

V_total = zeros(n_nodes, n_time); % Initialize Option Price Storage Matrix
for All Time Steps
V_total(:, 1) = V; % Initial Condition at tao = 0
V_total(end, :) = V_n; % Boundary Condition at Manually Set Maximum Stock
Price
V = V(2:end-1); % Remove the fixed DOFs from V for FDM

syms SP
dplus_T = (log(SP/K)+(r+sigma^2/2)*(T))/(sigma*(T)^(1/2));
dminus_T = (log(SP/K)+(r-sigma^2/2)*(T))/(sigma*(T)^(1/2));
Exact_Solution = SP*normcdf(dplus_T) - K*exp(-r*T)*normcdf(dminus_T);

%% Matrix Initializations
A = zeros(n_free, n_free); % Free Global 'Stiffness' Matrix
B = zeros(n_free, 1); % Forcing Vector

for t = 2:n_time
    %% Global Matrix Assembly
    theta = 1/2; % Central difference method
    for i = 1:n_free
        %% Options Price Constants (Theta Differencing)
        a1 = theta*S(i+1)/(2*dS)*(r-sigma^2*S(i+1)/dS);         % Constant
for V(i-1, t+dt)
        a2 = 1/dtao + theta*((sigma*S(i+1)/dS)^2+r);            % Constant
for V(i, t+dt)
```

```
        a3 = -theta*S(i+1)/(2*dS)*(r+sigma^2*S(i+1)/dS);        % Constant
for V(i+1, t+dt)
        b1 = (1-theta)*S(i+1)/(2*dS)*(-r+sigma^2*S(i+1)/dS);    % Constant
for V(i-1, t)
        b2 = 1/dtao - (1-theta)*((sigma*S(i+1)/dS)^2+r);        % Constant
for V(i, t)
        b3 = (1-theta)*S(i+1)/(2*dS)*(r+sigma^2*S(i+1)/dS);     % Constant
for V(i+1, t)

        %% Free DOF A Matrix Construction
        if i == n_free
            A(i, i-1) = a1;
            A(i, i) = a2;
        elseif i == 1
            A(i, i) = a2;
            A(i, i+1) = a3;
        else
            A(i, i-1) = a1;
            A(i, i) = a2;
            A(i, i+1) = a3;
        end

        %% Free DOF B Matrix Construction
        if i == n_free
            B(i) = b1*V(i-1) + b2*V(i) + b3*V_n(t-1) - a3*V_n(t);
        elseif i == 1
            B(i) = (b1-a1)*V_1 + b2*V(i) + b3*V(i+1);
        else
            B(i) = b1*V(i-1) + b2*V(i) + b3*V(i+1);
        end
    end

    V = A\B;
    V_total(2:end-1, t) = V;
end
V_total

%% Post Processing
% Find the FDM Approximated Option Price at Current Stock Price and tao = T
current_index = find(S == max(S(S <= Stock_Price)));
% Linear Interpolation to Find Option Price In-Between Nodes
Option_Price = (V_total(current_index+1,end)-V_total(current_index,end))...
    /dS*(Stock_Price-S(current_index)) + V_total(current_index,end);

%% Plotting
figure(1)
tao_days = tao*365.25; % Change time scale to days for graphing
surf(tao_days, S, V_total)
xlabel('Time Until Expiration (Days)')
ylabel('Stock Price ($)')
zlabel('Option Price ($)')
title('Black-Scholes Model - Call Option Pricing')

figure(2)
fplot(SP, Exact_Solution, [0, L], 'LineWidth', 2)
hold on
```

```
plot(S, V_total(:, end), 'r--', 'LineWidth', 2)
hold off
xlabel('Stock Price ($)')
ylabel('Option Price ($)')
title('Black-Scholes Equation - Call Option Pricing at tao = T')
legend('Black-Scholes Equation', 'FD Approximation')
ylim([0 inf])

%% Print Results
fprintf('The current option price using Finite Difference Method is $%.4f\n',
Option_Price)
fprintf('The current option price using Black-Scholes Equation is $%.4f\n',
subs(Exact_Solution, Stock_Price))
```

Black-Scholes PDE in the Classical Advection-Diffusion Form Finite Element Model:

```
clc, clear, close all
%% Given Data
K = 1000; % Strike Price
Stock_Price = 1145.45; % Current Stock Price
sigma = 62.51/100; % Implied Volatility
r = 0.02441; % Assume 2 Percent. Close to US 10 Year Treasury Yield

T = 255/365.25; % Years Until Expiration
dtao = 25.5/365.25; % 1 Day Time Steps

L = 2*Stock_Price; % Maximum Stock Price
n_blocks = 10; % Number of Elements
dS = L/n_blocks; % Price Step

n_nodes = n_blocks + 1; % Linear Approximation
fixed_dofs = [1, n_nodes]; % Node numbers of fixed DOFs
free_dofs = setxor(1:n_nodes,fixed_dofs); % Node numbers of free DOFs

S = (0:dS:L)'; % Stock Price Vector from $0 to Maximum Stock Price
tao = (0:dtao:T)'; % Tao Vector, where tao = T - t
n_time = length(tao); % Number of Time Nodes

V_1 = 0; % Option Price at Stock Price = $0 - Boundary Condition ($)
V_n = L - K*exp(-r*tao); % Option Price at Maximum Stock Price - Boundary
Condition ($)
V = max(S-K, 0); % Free Global Option Price Vector at tao = 0 - Initial
Condition ($)

V_total = zeros(n_nodes, n_time); % Initialize Option Price Storage Matrix
for All Time Steps
V_total(:, 1) = V; % Initial Condition at tao = 0
V_total(end, :) = V_n; % Boundary Condition at Manually Set Maximum Stock
Price

syms SP
dplus_T = (log(SP/K)+(r+sigma^2/2)*(T))/(sigma*(T)^(1/2));
dminus_T = (log(SP/K)+(r-sigma^2/2)*(T))/(sigma*(T)^(1/2));
Exact_Solution = SP*normcdf(dplus_T) - K*exp(-r*T)*normcdf(dminus_T);

%% Define Connectivity of Global Free DOFs
conn = zeros(n_blocks, 2); % Initialize element to node connectivity
for i = 1:n_blocks
    conn(i, 1) = i; % Local node 1
    conn(i, 2) = i+1; % Local node 2
end

%% Define Shape Functions
syms n
N1n = 1/2*(1-n); N2n = 1/2*(1+n); % Linear shape functions in local
coordinates
dndS = 2/dS; % Relationship between dS and dn in the form of dn/dS
dN1dn = diff(N1n); dN2dn = diff(N2n); % Derivative of shape functions wrt n
dN1dS = dN1dn*dndS; dN2dS = dN2dn*dndS; % Derivative of shape functions wrt S
```

```
%% Derive Parts of 'Stiffness' Matrix using Method of Weighted Residuals
k11eB1 = int((1-n)*dN1dS*N1n, [-1 1]); k12eB1 = int((1-n)*dN1dS*N2n, [-1 1]);
k21eB1 = int((1-n)*dN2dS*N1n, [-1 1]); k22eB1 = int((1-n)*dN2dS*N2n, [-1 1]);
kB1 = [k11eB1, k12eB1; k21eB1, k22eB1];

k11eB2 = int((1+n)*dN1dS*N1n, [-1 1]); k12eB2 = int((1+n)*dN1dS*N2n, [-1 1]);
k21eB2 = int((1+n)*dN2dS*N1n, [-1 1]); k22eB2 = int((1+n)*dN2dS*N2n, [-1 1]);
kB2 = [k11eB2, k12eB2; k21eB2, k22eB2];

k11eD = int(N1n^2, [-1 1]); k12eD = int(N1n*N2n, [-1 1]);
k21eD = int(N2n*N1n, [-1 1]); k22eD = int(N2n^2, [-1 1]);
kD = [k11eD, k12eD; k21eD, k22eD];

k11eE1 = int((1-n)^2*dN1dS^2, [-1 1]); k12eE1 = int((1-n)^2*dN1dS*dN2dS, [-1
1]);
k21eE1 = int((1-n)^2*dN2dS*dN1dS, [-1 1]); k22eE1 = int((1-n)^2*dN2dS^2, [-1
1]);
kE1 = [k11eE1, k12eE1; k21eE1, k22eE1];

k11eE2 = int((1-n)*(1+n)*dN1dS^2, [-1 1]); k12eE2 = int((1-
n)*(1+n)*dN1dS*dN2dS, [-1 1]);
k21eE2 = int((1-n)*(1+n)*dN2dS*dN1dS, [-1 1]); k22eE2 = int((1-
n)*(1+n)*dN2dS^2, [-1 1]);
kE2 = [k11eE2, k12eE2; k21eE2, k22eE2];

k11eE3 = int((1+n)^2*dN1dS^2, [-1 1]); k12eE3 = int((1+n)^2*dN1dS*dN2dS, [-1
1]);
k21eE3 = int((1+n)^2*dN2dS*dN1dS, [-1 1]); k22eE3 = int((1+n)^2*dN2dS^2, [-1
1]);
kE3 = [k11eE3, k12eE3; k21eE3, k22eE3];

alpha_coeff = sigma^2/dS*[2, -1; -1, 2];

%% Global 'Stiffness' Matrix Assembly (Apply Theta Differencing)
A = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
next time step
B = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
current time step
theta = 1/2; % Central difference method
for i = 1:n_blocks
    S1 = S(i); S2 = S(i+1); % Local node stock price coordinates
    enodes = conn(i, :); % Element to node connectivity for local to global
mapping
    beta = kD\((sigma^2-r)*(1/2*S1*kB1 + 1/2*S2*kB2) + 1/8*...
        sigma^2*(S1^2*kE1 + 2*S1*S2*kE2 + S2^2*kE3) + r*kD); % [beta] in the
hand calculation
    ket1 = alpha_coeff\(1/dtao - beta*(1-theta)); % Local 'Stiffness' Matrix
for the current time step
    ket2 = alpha_coeff\(1/dtao + beta*theta); % Local 'Stiffness' Matrix for
the next time step
    A(enodes, enodes) = A(enodes, enodes) + ket2; % Assembly of the global
'stiffness' matrix
    B(enodes, enodes) = B(enodes, enodes) + ket1; % Assembly of the global
forcing vector
```

```
end

%% Global 'Stiffness' Matrix Partitioning
A_E = A(fixed_dofs, fixed_dofs);
A_EF = A(fixed_dofs, free_dofs);
A_FE = A(free_dofs, fixed_dofs);
A_F = A(free_dofs, free_dofs);

%% Solving for Option Pricing for all Stock Price Increments and Time Steps
for t = 2:n_time
    V_E2 = V_total(fixed_dofs, t);
    RS = B*V;
    V_F2 = A_F\(RS(2:end-1) - A_FE*V_E2); % Free DOF Option pricing at next
time step
    V_total(2:end-1, t) = V_F2;
    V = V_total(:, t);
end
V_total

%% Post Processing
% Find the FEM Approximated Option Price at Current Stock Price and tao = T
current_index = find(S == max(S(S <= Stock_Price)));
% Linear Interpolation to Find Option Price In-Between Nodes
Option_Price = (V_total(current_index+1,end)-V_total(current_index,end))...
    /dS*(Stock_Price-S(current_index)) + V_total(current_index,end);

%% Plotting
figure(1)
tao_days = tao*365.25; % Change time scale to days for graphing
surf(tao_days, S, V_total)
xlabel('Time Until Expiration (Days)')
ylabel('Stock Price ($)')
zlabel('Option Price ($)')
title('Black-Scholes Model - Call Option Pricing')

figure(2)
fplot(SP, Exact_Solution, [0, L], 'LineWidth', 2)
hold on
plot(S, V_total(:, end), 'r--', 'LineWidth', 2)
hold off
xlabel('Stock Price ($)')
ylabel('Option Price ($)')
title('Black-Scholes Equation - Call Option Pricing at tao = T')
legend('Black-Scholes Equation', 'FE Approximation')
ylim([0 inf])

%% Print Results
fprintf('The current option price using Finite Element Method is $%.4f\n',
Option_Price)
fprintf('The current option price using Black-Scholes Equation is $%.4f\n',
subs(Exact_Solution, Stock_Price))
```

Black-Scholes PDE in the Transformed Heat Form Finite Element Model:

```
clc, clear, close all
%% Given Data
K = 1000; % Strike Price
Stock_Price = 1145.45; % Current Stock Price
sigma = 62.51/100; % Implied Volatility
r = 0.02441; % Assume 2 Percent. Close to US 10 Year Treasury Yield

T = 255/365.25; % Years Until Expiration
dtao = 25.5/365.25; % 1 Day Time Steps

tao = (0:dtao:T)'; % Tao Vector, where tao = T - t
n_time = length(tao); % Number of Time Nodes
n_time_steps = n_time - 1; % Number of Time Steps

n_blocks = 10; % Number of Elements
n_nodes = n_blocks + 1; % Linear Approximation
fixed_dofs = [1, n_nodes]; % Node numbers of fixed DOFs
free_dofs = setxor(1:n_nodes,fixed_dofs); % Node numbers of free DOFs

L = 2*Stock_Price; % Maximum Stock Price
dS = (L-0.0000000001)/n_blocks; % Price Step
S = (0.0000000001:dS:L)'; % Stock Price Vector from $0 to Maximum Stock
Price, log(0) is undefined, so must use a very small number
x = zeros(n_nodes, n_time);
for i = 1:n_time
    x(:, i) = log(S(:)) + (r-1/2*sigma^2)*tao(i); % Space Vector, where x is
a function of stock price and tao
end

dx = zeros(n_blocks, 1);
for i = 1:n_blocks
    dx(i) = x(i+1, 1) - x(i, 1);
end

F_1 = 0; % Option Price at $0 stock price - Boundary Condition
F_n = (exp(x(end, :)-(r-1/2*sigma^2)*tao') - K*exp(-r*tao')).*exp(r*tao'); %
Option Price at max stock price - Boundary Condition
F = max(exp(max(x(:, 1), 0))-K, 0); % Option Price at tao = 0 - Initial
Condition

F_total = zeros(n_nodes, n_time); % Initialize Option Price Storage Matrix
for All Time Steps
F_total(:, 1) = F; % Initial Condition at tao = 0
F_total(end, :) = F_n; % Boundary Condition at Manually Set Maximum Stock
Price

syms SP
dplus_T = (log(SP/K)+(r+sigma^2/2)*(T))/(sigma*(T)^(1/2));
dminus_T = (log(SP/K)+(r-sigma^2/2)*(T))/(sigma*(T)^(1/2));
Exact_Solution = SP*normcdf(dplus_T) - K*exp(-r*T)*normcdf(dminus_T);

%% Define Connectivity of Global Free DOFs
```

```
conn = zeros(n_blocks, 2); % Initialize element to node connectivity
for i = 1:n_blocks
    conn(i, 1) = i; % Local node 1
    conn(i, 2) = i+1; % Local node 2
end

%% Define Shape Functions
syms n
N1n = 1/2*(1-n); N2n = 1/2*(1+n); % Linear shape functions in local
coordinates
dndx = 2./dx; % Relationship between dx and dn in the form of dn/dx
dN1dn = diff(N1n); dN2dn = diff(N2n); % Derivative of shape functions wrt n
dN1dx = dN1dn*dndx; dN2dx = dN2dn*dndx; % Derivative of shape functions wrt x

%% Derive Parts of 'Stiffness' Matrix using Method of Weighted Residuals
k11eA = int(N1n^2, [-1 1]); k12eA = int(N1n*N2n, [-1 1]);
k21eA = int(N2n*N1n, [-1 1]); k22eA = int(N2n^2, [-1 1]);
kA = [k11eA, k12eA; k21eA, k22eA];

k11eC = double(int(dN1dx.^2, [-1 1])); k12eC = double(int(dN1dx.*dN2dx, [-1
1]));
k21eC = double(int(dN2dx.*dN1dx, [-1 1])); k22eC = double(int(dN2dx.^2, [-1
1]));

alpha_coeff = sigma^2*[2, -1; -1, 2]; % Coefficient to rearrange the alpha
term in the derivation so that it cancels out

%% Global 'Stiffness' Matrix Assembly (Apply Theta Differencing)
A = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
next time step
B = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
current time step
theta = 1/2; % Central difference method
for i = 1:n_blocks
    kC = [k11eC(i), k12eC(i); k21eC(i), k22eC(i)];
    enodes = conn(i, :); % Element to node connectivity for local to global
mapping
    beta = kA\(1/2*sigma^2*kC); % [beta] in the hand calculation
    ket1 = alpha_coeff\(1/dtao - (1-theta)*beta)*dx(i); % Local 'Stiffness'
Matrix for the current time step
    ket2 = alpha_coeff\(1/dtao + theta*beta)*dx(i); % Local 'Stiffness'
Matrix for the next time step
    A(enodes, enodes) = A(enodes, enodes) + ket2; % Assembly of the global
'stiffness' matrix for the next time step
    B(enodes, enodes) = B(enodes, enodes) + ket1; % Assembly of the global
'stiffness' matrix for the current time step
end

% for i = 1:n_blocks
%     kC = [k11eC(i), k12eC(i); k21eC(i), k22eC(i)];
%     enodes = conn(i, :); % Element to node connectivity for local to global
mapping
%     ket1 = dx(i)/sigma^2*(1/3/dtao*[2, 1; 1, 2] - (1-
theta)*(sigma/dx(i))^2*[1, -1; -1, 1]); % Local 'Stiffness' Matrix for the
current time step
```

```
%     ket2 = dx(i)/sigma^2*(1/3/dtao*[2, 1; 1, 2] +
(theta)*(sigma/dx(i))^2*[1, -1; -1, 1]); % Local 'Stiffness' Matrix for the
next time step
%     A(enodes, enodes) = A(enodes, enodes) + ket2; % Assembly of the global
'stiffness' matrix for the next time step
%     B(enodes, enodes) = B(enodes, enodes) + ket1; % Assembly of the global
'stiffness' matrix for the current time step
% end

%% Global 'Stiffness' Matrix Partitioning
A_E = A(fixed_dofs, fixed_dofs);
A_EF = A(fixed_dofs, free_dofs);
A_FE = A(free_dofs, fixed_dofs);
A_F = A(free_dofs, free_dofs);

for j = 1:n_time
    %% Solving for Option Pricing for all Stock Price Increments and Time
Steps
    if j >= 2
        F_E2 = F_total(fixed_dofs, j); % Boundary Conditions at next time
step
        RS = B*F;
        F_F2 = A_F\(RS(2:end-1) - A_FE*F_E2); % Free DOF Option pricing at
next time step
        F_total(2:end-1, j) = F_F2;
        F = F_total(:, j);
    end
end
F_total;

V_total = zeros(n_nodes, n_time);
for i = 1:n_time
    V_total(:, i) = F_total(:, i)*exp(-r*tao(i));
end
V_total

%% Post Processing
% Find the FEM Approximated Option Price at Current Stock Price and tao = T
current_index = find(S == max(S(S <= Stock_Price)));
% Linear Interpolation to Find Option Price In-Between Nodes
dS_current = S(current_index+1)-S(current_index);
Option_Price = (V_total(current_index+1,end)-V_total(current_index,end))...
    /dS_current*(Stock_Price-S(current_index)) + V_total(current_index,end);

%% Plotting
figure(1)
tao_days = tao*365.25; % Change time scale to days for graphing
surf(tao_days, S, V_total)
xlabel('Time Until Expiration (Days)')
ylabel('Stock Price ($)')
zlabel('Option Price ($)')
title('Black-Scholes Model - Call Option Pricing')

figure(2)
fplot(SP, Exact_Solution, [0, L], 'LineWidth', 2)
hold on
```

```
plot(S, V_total(:, end), 'r--', 'LineWidth', 2)
hold off
xlabel('Stock Price ($)')
ylabel('Option Price ($)')
title('Black-Scholes Equation - Call Option Pricing at tao = T')
legend('Black-Scholes Equation', 'FE Approximation')
ylim([0 inf])

%% Print Results
fprintf('The current option price using Finite Element Method is $%.4f\n',
Option_Price)
fprintf('The current option price using Black-Scholes Equation is $%.4f\n',
subs(Exact_Solution, Stock_Price))
```

Sanity Check No. 1 Concrete Chloride Problem Finite Element Model:

```
clc, clear, close all
%% Sanity Check with Assignment 3 Problem
%% Given Data
D = 6*10^-12; % Apparent Diffusion Constant (m^2/s)
n_time = 100;
t_total = n_time*24*60*60; % Total time (seconds)
L = 0.1; % Total depth (m)
n_blocks = 100; % Number of grid blocks
n_nodes = n_blocks+1;
dx = L/n_blocks; % Space step
dt = t_total/n_time; % 1 Day time step (seconds)
V = 0; % Advection Constant
phi_1 = 1; % Concentration at the surface - Boundary Condition (%)
F = zeros(n_nodes, 1);
F(1) = 1;

x_coord = zeros(n_nodes, 1);
for i = 1:n_nodes
    x_coord(i) = (i-1)*dx;
end

fixed_dofs = 1; % Node numbers of fixed DOFs
free_dofs = setxor(1:n_nodes,fixed_dofs); % Node numbers of free DOFs

%% Define Connectivity of Global Free DOFs
conn = zeros(n_blocks, 2); % Initialize element to node connectivity
for i = 1:n_blocks
    conn(i, 1) = i; % Local node 1
    conn(i, 2) = i+1; % Local node 2
end

%% Define Shape Functions
syms n
N1n = 1/2*(1-n); N2n = 1/2*(1+n); % Linear shape functions in local
coordinates
dndx = 2/dx; % Relationship between dx and dn in the form of dn/dx
dN1dn = diff(N1n); dN2dn = diff(N2n); % Derivative of shape functions wrt n
dN1dx = dN1dn*dndx; dN2dx = dN2dn*dndx; % Derivative of shape functions wrt x

%% Derive Parts of 'Stiffness' Matrix using Method of Weighted Residuals
k11eA = int(N1n^2, [-1 1]); k12eA = int(N1n*N2n, [-1 1]);
k21eA = int(N2n*N1n, [-1 1]); k22eA = int(N2n^2, [-1 1]);
kA = [k11eA, k12eA; k21eA, k22eA];

k11eC = double(int(dN1dx.^2, [-1 1])); k12eC = double(int(dN1dx.*dN2dx, [-1
1]));
k21eC = double(int(dN2dx.*dN1dx, [-1 1])); k22eC = double(int(dN2dx.^2, [-1
1]));
kC = [k11eC, k12eC; k21eC, k22eC];

%% Global 'Stiffness' Matrix Assembly (Apply Theta Differencing)
A = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
next time step
```

```
B = zeros(n_nodes, n_nodes); % Initialize Global 'Stiffness' Matrix for the
current time step
theta = 1/2; % Central difference method
beta = double(kA\(D*kC)); % [beta] in the hand calculation
for i = 1:n_blocks
    enodes = conn(i, :); % Element to node connectivity for local to global
mapping
    ket1 = (1/dt*[2/3, 1/3; 1/3, 2/3] - (1-theta)*beta*[2/3, 1/3; 1/3, 2/3]);
% Local 'Stiffness' Matrix for the current time step
    ket2 = (1/dt*[2/3, 1/3; 1/3, 2/3] + theta*beta*[2/3, 1/3; 1/3, 2/3]); %
Local 'Stiffness' Matrix for the next time step
    A(enodes, enodes) = A(enodes, enodes) + ket2; % Assembly of the global
'stiffness' matrix for the next time step
    B(enodes, enodes) = B(enodes, enodes) + ket1; % Assembly of the global
'stiffness' matrix for the current time step
end

%% Global 'Stiffness' Matrix Partitioning
A_E = A(fixed_dofs, fixed_dofs);
A_EF = A(fixed_dofs, free_dofs);
A_FE = A(free_dofs, fixed_dofs);
A_F = A(free_dofs, free_dofs);

F_E = F(fixed_dofs); % Boundary Conditions at current time step

for j = 1:n_time
    %% Solving for Option Pricing for all Stock Price Increments and Time
Steps
    if j >= 2
        F_F = F(free_dofs); % Free DOF Option pricing at current time step
        RS = B*F;
        F(2:end) = A_F\(RS(2:end) - A_FE*F_E);
    end
end

%% Solve for Exact Phi and Plot vs the Final Time Step Numerical Solution
% Exact Solutiion in Symbolic Form
syms x
C = phi_1*(1 - erf(x/(2*(D*t_total)^(1/2))));

figure(1)
fplot(x, C, [0, 0.1], 'b', 'LineWidth', 2)     % Exact Solution Plot
hold on
plot(x_coord, F, '--', 'LineWidth', 2)   % Approximate Solution Plot
hold off
title('Plot of Chloride Concentration Diffusion into Concrete')
xlabel('Depth of Concrete (m)')
ylabel('Concentration of Chloride by Weight of Concrete (%)')
legend('Exact Solution', 'FE Approximation')
```

Appendix B – Derivations and Hand Calculations

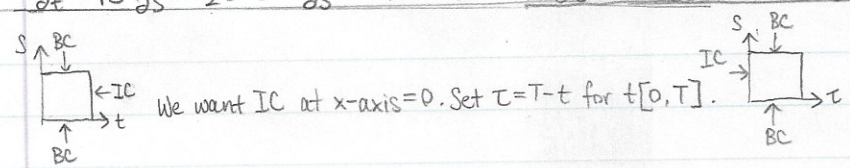
(1)

General Advection-Diffusion Partial Differential Equation.

$\frac{\partial c}{\partial t} = D\nabla^2 C - \nabla \cdot vc + R$, $\nabla = \frac{\partial}{\partial x}$. C = call option price (concentration), S = stock price (displacement), r = risk-free rate, σ = implied volatility

$\frac{\partial c}{\partial t} + V\frac{\partial c}{\partial S} - D\frac{\partial^2 C}{\partial S^2} - R = 0$. → Convert to Black Scholes equation. $R = rC$, $D = -\frac{1}{2}\sigma^2 S^2$, $V = rS$.

$\frac{\partial c}{\partial t} + rS\frac{\partial c}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0$. → $\boxed{\frac{-\partial c}{\partial \tau} + rS\frac{\partial c}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0}$



We want IC at x-axis = 0. Set $\tau = T - t$ for $t \in [0, T]$.

$d\tau = -dt \Rightarrow \boxed{\frac{dt}{d\tau} = -1}$

$\frac{\partial c}{\partial t} \times \frac{\partial t}{\partial \tau} = \frac{\partial c}{\partial \tau} = \frac{-\partial c}{\partial t}$.

Boundary Conditions

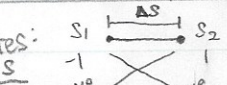
European Call: BC → $C(0,\tau) = 0$, $C(\infty,\tau) = BIG = S - Ke^{-r\tau}$ → intrinsic + TVM. IC → $c(S,0) = \max(S-K, 0)$.

European Put: BC → $P(\infty,\tau) = 0$, $P(0,\tau) = BIG = Ke^{-r\tau} - S$ ↑ IC → $P(S,0) = \max(K-S, 0)$.

Try a 1D FEM solution with stock price and a Crank-Nicolson scheme for time steps.

$\tilde{C} = N_1^e C_1 + N_2^e C_2$ (1D Linear Approximation). → $N_i^e(S)$, $C_i(\tau)$, $\tilde{C}(S,\tau)$.

In global coordinates:



$N_1^e = \frac{S_2 - S}{S_2 - S_1} = \frac{S_2 - S}{\Delta S}$

$N_2^e = \frac{S - S_1}{S_2 - S_1} = \frac{S - S_1}{\Delta S}$

$\frac{dN}{dS} = \frac{dN}{d\eta}\cdot\frac{d\eta}{dS}$: $\boxed{\frac{dN_1^e}{dS} = \left(\frac{-1}{2}\right)\left(\frac{2}{\Delta S}\right) = \frac{-1}{\Delta S}}$, $\boxed{\frac{dN_2^e}{dS} = \left(\frac{1}{2}\right)\left(\frac{2}{\Delta S}\right) = \frac{1}{\Delta S}}$

In local coordinates:

$N_1^e(\eta) = \frac{1}{2}(1-\eta)$

$N_2^e(\eta) = \frac{1}{2}(1+\eta)$

$\boxed{\partial S = \frac{\Delta S}{2}\partial\eta}$ $\int_{S1}^{S2} = \int_{\eta=-1}^{\eta=1}$

Transformation: $S(\eta) = N_1^e(\eta)S_1 + N_2^e(\eta)S_2$

$= \frac{1}{2}(1-\eta)S_1 + \frac{1}{2}(1+\eta)S_2$

$\boxed{\frac{\partial S}{\partial \eta} = \frac{S_2 - S_1}{2} = \frac{\Delta S}{2}}$

Apply MWR

$\int_{S1}^{S2}\left(-\frac{\partial V}{\partial \tau} + \frac{\partial V}{\partial S}rS + \frac{1}{2}\sigma^2 S^2\left(\frac{\partial^2 V}{\partial S^2}\right) - rV\right)N_i\,dS = 0$, $N_j = \{N\}$, $N_i = \langle N\rangle$, j = row # (col vector), i = col # (row vector)

Use V as general Φ value, works for both C/P. C = Call, P = Put

$\int_{S1}^{S2}\left(\frac{-\partial(\Sigma N_j V_j)}{\partial \tau} + rS\frac{\partial(\Sigma N_j V_j)}{\partial S} + \frac{1}{2}\sigma^2 S^2\left(\frac{\partial^2(\Sigma N_j V_j)}{\partial S^2}\right) - r(\Sigma N_j V_j)\right)N_i\,dS = 0$, $\tilde{V}(\tau,S) = \Sigma V_j(\tau)\cdot N_j(S)$, $V_j(\tau) = \{V\} = V_j$, $V_i = \langle V\rangle$

$-\frac{\partial V_j}{\partial \tau}\int_{S1}^{S2} N_j N_i\,dS + \int_{S1}^{S2}\left(rS\frac{\partial N_j}{\partial S} + \frac{1}{2}\sigma^2 S^2\left(\frac{\partial^2 (N_j)}{\partial S^2}\right) - rN_j\right)N_i\,dS\,V_j = 0$ $S = \frac{1}{2}(1-\eta)S_1 + \frac{1}{2}(1+\eta)S_2$.

Where S_1, S_2 = constant, known.

(*) $-\frac{\partial V_j}{\partial \tau}\int_{S1}^{S2} N_j N_i\,dS + V_j\left(r\int_{S1}^{S2} S\frac{\partial N_j}{\partial S}N_i\,dS + \frac{1}{2}\sigma^2\int_{S1}^{S2} S^2\left(\frac{\partial^2 N_j}{\partial S^2}\right)N_i\,dS - r\int_{S1}^{S2} N_j N_i\,dS\right) = 0$ (D) (B) (C) (D)

For (C): IBP: $u = S^2 N_i$, $du = \left(2SN_i + S^2\frac{\partial N_i}{\partial S}\right)dS$, $dv = \frac{\partial^2 N_j}{\partial S^2}dS$, $v = \frac{\partial N_j}{\partial S}$

$(C) = \frac{1}{2}\sigma^2\int_{S1}^{S2} S^2\left(\frac{\partial^2 N_j}{\partial S^2}\right)N_i\,dS = \frac{1}{2}\sigma^2\left(S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}^{S2} - \int_{S1}^{S2}\frac{\partial N_j}{\partial S}\left(2SN_i + S^2\frac{\partial N_i}{\partial S}\right)dS\right)$

$= \frac{1}{2}\sigma^2\left(S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}^{S2} - \int_{S1}^{S2} 2S\frac{\partial N_j}{\partial S}N_i\,dS - \int_{S1}^{S2} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}dS\right)$

Sub (C) in (*)

$-\frac{\partial V_j}{\partial \tau}\int_{S1}^{S2} N_j N_i\,dS + V_j\left(r\int_{S1}^{S2} S\frac{\partial N_j}{\partial S}N_i\,dS - \sigma^2\int_{S1}^{S2} S\frac{\partial N_j}{\partial S}N_i\,dS - \frac{1}{2}\sigma^2\int_{S1}^{S2} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}dS - r\int_{S1}^{S2} N_j N_i\,dS\right) = -\frac{1}{2}\sigma^2 S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}^{S2}$

(*)(2) $\frac{\partial V_j}{\partial \tau}\int_{S1}^{S2} N_j N_i\,dS + V_j\left(-r\int_{S1}^{S2} S\frac{\partial N_j}{\partial S}N_i\,dS + \sigma^2\int_{S1}^{S2} S\frac{\partial N_j}{\partial S}N_i\,dS + \frac{1}{2}\sigma^2\int_{S1}^{S2} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}dS + r\int_{S1}^{S2} N_j N_i\,dS\right) = \frac{1}{2}\sigma^2 S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}^{S2}$ (D) (B) (B) (E) (D)

Look at RHS of (*)(2): $\frac{1}{2}\sigma^2\left(S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S2} - S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}\right)$, $i = 1, 2$. $\left(\frac{\partial \tilde{V}}{\partial S}\right)$ = estimate of ΔV for every \$1 change in S.

When $i=1$: $\frac{1}{2}\sigma^2\left(S_2^2 N_1(S_2)\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S2} - S_1^2 N_1(S_1)\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}\right) = -\frac{1}{2}\sigma^2 S_1^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}$

When $i=2$: $\frac{1}{2}\sigma^2\left(S_2^2 N_2(S_2)\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S2} - S_1^2 N_2(S_1)\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1}\right) = \frac{1}{2}\sigma^2 S_2^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S2}$

$RHS = \frac{1}{2}\sigma^2\begin{bmatrix} -S_1^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S1} \\ S_2^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}$

(2)

$$\frac{\partial V_j}{\partial \tau}\int_{S_1}^{S_2} N_j N_i\,dS + V_j\left((\sigma^2-r)\int_{S_1}^{S_2} S\frac{\partial N_j}{\partial S}N_i\,dS + \frac{1}{2}\sigma^2\int_{S_1}^{S_2} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}\,dS + r\int_{S_1}^{S_2} N_j N_i\,dS\right) = \frac{1}{2}\sigma^2 S^2 N_i\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_1}^{S_2}$$

(D) (B) (E) (D)

From here on, the letters (B), (D), (E) only correspond to the integrals.

Convert to local coordinates:

(3) (*)

$$\frac{\Delta S}{2}\left(\frac{\partial V_j}{\partial \tau}\int_{-1}^{1} N_j N_i\,d\eta + V_j\left((\sigma^2-r)\int_{-1}^{1} S\frac{\partial N_j}{\partial S}N_i\,d\eta + \frac{1}{2}\sigma^2\int_{-1}^{1} S^2\frac{\partial N_j}{\partial S}\frac{\partial N_i}{\partial S}\,d\eta + r\int_{-1}^{1} N_j N_i\,d\eta\right)\right) = \frac{1}{2}\sigma^2\begin{bmatrix} -S_1^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_1} \\ S_2^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_2}\end{bmatrix}$$

(D) (B) (E) (D)

For (B): $S = \frac{1}{2}(1-\eta)S_1 + \frac{1}{2}(1+\eta)S_2$

$$\frac{1}{2}S_1\int_{-1}^{1}(1-\eta)\begin{bmatrix}\frac{\partial N_1^e}{\partial S}\\ \frac{\partial N_2^e}{\partial S}\end{bmatrix}\begin{bmatrix}N_1^e & N_2^e\end{bmatrix}d\eta + \frac{1}{2}S_2\int_{-1}^{1}(1+\eta)\begin{bmatrix}\frac{\partial N_1^e}{\partial S}\\ \frac{\partial N_2^e}{\partial S}\end{bmatrix}\begin{bmatrix}N_1^e & N_2^e\end{bmatrix}d\eta$$

(B1) (B2)

$$= \frac{1}{2}S_1\int_{-1}^{1}(1-\eta)\begin{bmatrix}\frac{-1}{\Delta S}\left(\frac{1}{2}(1-\eta)\right) & \frac{-1}{\Delta S}\left(\frac{1}{2}(1+\eta)\right)\\ \frac{1}{\Delta S}\left(\frac{1}{2}(1-\eta)\right) & \frac{1}{\Delta S}\left(\frac{1}{2}(1+\eta)\right)\end{bmatrix}d\eta + \frac{1}{2}S_2\int_{-1}^{1}(1+\eta)\begin{bmatrix}\frac{-1}{\Delta S}\left(\frac{1}{2}(1-\eta)\right) & \frac{-1}{\Delta S}\left(\frac{1}{2}(1+\eta)\right)\\ \frac{1}{\Delta S}\left(\frac{1}{2}(1-\eta)\right) & \frac{1}{\Delta S}\left(\frac{1}{2}(1+\eta)\right)\end{bmatrix}d\eta$$

$$= \frac{1}{2}S_1\int_{-1}^{1}\begin{bmatrix}\frac{-1}{\Delta S}\left(\frac{1}{2}(1-\eta)^2\right) & \frac{-1}{\Delta S}\left(\frac{1}{2}(1+\eta)(1-\eta)\right)\\ \frac{1}{\Delta S}\left(\frac{1}{2}(1-\eta)^2\right) & \frac{1}{\Delta S}\left(\frac{1}{2}(1+\eta)(1-\eta)\right)\end{bmatrix}d\eta + \frac{1}{2}S_2\int_{-1}^{1}\begin{bmatrix}\frac{-1}{\Delta S}\left(\frac{1}{2}(1+\eta)(1-\eta)\right) & \frac{-1}{\Delta S}\left(\frac{1}{2}(1+\eta)^2\right)\\ \frac{1}{\Delta S}\left(\frac{1}{2}(1+\eta)(1-\eta)\right) & \frac{1}{\Delta S}\left(\frac{1}{2}(1+\eta)^2\right)\end{bmatrix}d\eta$$

$$= \frac{S_1}{4\Delta S}\begin{bmatrix}\frac{-(1-\eta)^3}{3}(-1) & -\left(\eta-\frac{\eta^3}{3}\right)\\ \frac{(1-\eta)^3}{3(-1)} & \left(\eta-\frac{\eta^3}{3}\right)\end{bmatrix}\Bigg|_{-1}^{1} + \frac{S_2}{4\Delta S}\begin{bmatrix}-\left(\eta-\frac{\eta^3}{3}\right) & -\frac{(1+\eta)^3}{3}\\ \left(\eta-\frac{\eta^3}{3}\right) & \frac{(1+\eta)^3}{3}\end{bmatrix}\Bigg|_{-1}^{1}$$

$$= \frac{S_1}{4\Delta S}\begin{bmatrix}0-\frac{8}{3} & -\frac{2}{3}+\left(-1+\frac{1}{3}\right)\\ 0+\frac{8}{3} & \frac{2}{3}-\left(-1+\frac{1}{3}\right)\end{bmatrix} + \frac{S_2}{4\Delta S}\begin{bmatrix}-\frac{4}{3} & -\frac{8}{3}+0\\ \frac{4}{3} & \frac{8}{3}-0\end{bmatrix}$$

$$= \frac{1}{4\Delta S}\left(S_1\begin{bmatrix}-\frac{8}{3} & -\frac{4}{3}\\ \frac{8}{3} & \frac{4}{3}\end{bmatrix} + S_2\begin{bmatrix}-\frac{4}{3} & -\frac{8}{3}\\ \frac{4}{3} & \frac{8}{3}\end{bmatrix}\right) = \frac{1}{3\Delta S}\left(S_1\begin{bmatrix}-2 & -1\\ 2 & 1\end{bmatrix} + S_2\begin{bmatrix}-1 & -2\\ 1 & 2\end{bmatrix}\right)$$

For (D):

$$\int_{-1}^{1}\begin{bmatrix}N_1^eN_1^e & N_1^eN_2^e\\ N_2^eN_1^e & N_2^eN_2^e\end{bmatrix}d\eta = \int_{-1}^{1}\begin{bmatrix}\frac{1}{4}(1-\eta)^2 & \frac{1}{4}(1-\eta)(1+\eta)\\ \frac{1}{4}(1-\eta)(1+\eta) & \frac{1}{4}(1+\eta)^2\end{bmatrix}d\eta = \frac{1}{4}\begin{bmatrix}\frac{(1-\eta)^3}{-3} & \left(\eta-\frac{\eta^3}{3}\right)\\ \left(\eta-\frac{\eta^3}{3}\right) & \frac{(1+\eta)^3}{3}\end{bmatrix}\Bigg|_{-1}^{1}$$

$$= \frac{1}{4}\begin{bmatrix}0+\frac{8}{3} & \frac{4}{3}\\ \frac{4}{3} & \frac{8}{3}-0\end{bmatrix} = \frac{1}{3}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$$

For (E): $S^2 = \frac{1}{4}(1-\eta)^2S_1^2 + \frac{1}{2}(1-\eta)(1+\eta)S_1S_2 + \frac{1}{4}(1+\eta)^2S_2^2$

$$\frac{1}{4}\left(S_1^2\int_{-1}^{1}(1-\eta)^2\begin{bmatrix}\frac{\partial N_1^e}{\partial S}\frac{\partial N_1^e}{\partial S} & \frac{\partial N_1^e}{\partial S}\frac{\partial N_2^e}{\partial S}\\ \frac{\partial N_2^e}{\partial S}\frac{\partial N_1^e}{\partial S} & \left(\frac{\partial N_2^e}{\partial S}\right)^2\end{bmatrix}d\eta + 2S_1S_2\int_{-1}^{1}(1-\eta)(1+\eta)\begin{bmatrix}\frac{1}{(\Delta S)^2} & -\left(\frac{1}{\Delta S}\right)^2\\ -\left(\frac{1}{\Delta S}\right)^2 & \left(\frac{1}{\Delta S}\right)^2\end{bmatrix}d\eta + S_2^2\int_{-1}^{1}(1+\eta)^2\begin{bmatrix}\left(\frac{1}{\Delta S}\right)^2 & -\left(\frac{1}{\Delta S}\right)^2\\ -\left(\frac{1}{\Delta S}\right)^2 & \left(\frac{1}{\Delta S}\right)^2\end{bmatrix}d\eta\right)$$

(E1) (E2) (E3)

$$= \left(\frac{1}{2\Delta S}\right)^2\left(S_1^2\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\frac{(1-\eta)^3}{-3}\Bigg|_{-1}^{1} + 2S_1S_2\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\left(\eta-\frac{\eta^3}{3}\right)\Bigg|_{-1}^{1} + S_2^2\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\frac{(1+\eta)^3}{3}\Bigg|_{-1}^{1}\right)$$

$$= \frac{1}{4\Delta S^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\left(S_1^2\left(\frac{8}{3}\right) + 2S_1S_2\left(\frac{4}{3}\right) + S_2^2\left(\frac{8}{3}\right)\right) = \frac{\left(2S_1^2+2S_1S_2+2S_2^2\right)}{3\Delta S^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$$

$$\frac{\Delta S}{2}\left(\frac{\partial V_j}{\partial \tau}\left(\frac{1}{3}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\right) + V_j\left((\sigma^2-r)\left(\frac{1}{3\Delta S}\left(S_1\begin{bmatrix}-2 & -1\\ 2 & 1\end{bmatrix} + S_2\begin{bmatrix}-1 & -2\\ 1 & 2\end{bmatrix}\right)\right) + \frac{1}{2}\sigma^2\left(\frac{\left(2S_1^2+2S_1S_2+2S_2^2\right)}{3\Delta S^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\right) + \frac{r}{3}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\right)\right)$$

$$= \frac{1}{2}\sigma^2\begin{bmatrix}-S_1^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_1}\\ S_2^2\left(\frac{\partial \tilde{V}}{\partial S}\right)\Big|_{S_2}\end{bmatrix}$$

(3)

$$\left(\frac{\Delta S}{6}\right)\frac{\partial V_j}{\partial \tau}\begin{bmatrix}2&1\\1&2\end{bmatrix}+\frac{V_j\Delta S}{2}\left(\frac{\sigma^2-r}{3\Delta S}\left(S_1\begin{bmatrix}-2&-1\\2&1\end{bmatrix}+S_2\begin{bmatrix}-1&-2\\1&2\end{bmatrix}\right)+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)=\frac{1}{2}\sigma^2\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}$$

$$\frac{\partial V_j}{\partial \tau}\begin{bmatrix}2&1\\1&2\end{bmatrix}=\left(\frac{6}{\Delta S}\right)\left(\frac{1}{2}\sigma^2\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-\frac{V_j\Delta S}{2}\left(\frac{\sigma^2-r}{3\Delta S}\left(S_1\begin{bmatrix}-2&-1\\2&1\end{bmatrix}+S_2\begin{bmatrix}-1&-2\\1&2\end{bmatrix}\right)+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)\right)$$

$$\frac{\partial V_j}{\partial \tau}=\frac{1}{4-1}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\frac{6}{\Delta S}\right)\left(\frac{1}{2}\sigma^2\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-\frac{V_j\Delta S}{2}\left(\frac{\sigma^2-r}{3\Delta S}\left(S_1\begin{bmatrix}-2&-1\\2&1\end{bmatrix}+S_2\begin{bmatrix}-1&-2\\1&2\end{bmatrix}\right)+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)\right)$$

$$\frac{\partial V_j}{\partial \tau}=\frac{1}{\Delta S}\begin{bmatrix}4&-2\\-2&4\end{bmatrix}\left(\frac{1}{2}\sigma^2\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-\frac{V_j\Delta S}{2}\left(\frac{\sigma^2-r}{3\Delta S}\left(S_1\begin{bmatrix}-2&-1\\2&1\end{bmatrix}+S_2\begin{bmatrix}-1&-2\\1&2\end{bmatrix}\right)+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)\right)$$

$$\frac{\partial V_j}{\partial \tau}=\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-V_j\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\frac{\sigma^2-r}{3\Delta S}\begin{bmatrix}-2S_1-S_2&-S_1-2S_2\\2S_1+S_2&S_1+2S_2\end{bmatrix}+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)$$

(*)(4)

$$\boxed{\frac{\partial V_j}{\partial \tau}=\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\frac{\sigma^2-r}{3\Delta S}\begin{bmatrix}-(2S_1+S_2)&-(S_1+2S_2)\\(2S_1+S_2)&(S_1+2S_2)\end{bmatrix}+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)\begin{bmatrix}V_1^e\\V_2^e\end{bmatrix}}$$

Apply theta differencing to (*)(4)

$$\begin{bmatrix}\frac{V_1^{\tau+\Delta\tau}-V_1^{\tau}}{\Delta\tau}\\\frac{V_2^{\tau+\Delta\tau}-V_2^{\tau}}{\Delta\tau}\end{bmatrix}^e=\theta\left\{\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}-\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\frac{\sigma^2-r}{3\Delta S}\begin{bmatrix}-(2S_1+S_2)&-(S_1+2S_2)\\2S_1+S_2&S_1+2S_2\end{bmatrix}+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)\begin{bmatrix}V_1^{\tau+\Delta\tau}\\V_2^{\tau+\Delta\tau}\end{bmatrix}^e\right\}$$

(f) (*)

$$+(1-\theta)\left\{\underbrace{\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}}_{[\alpha]\text{ coeff.}}\underbrace{\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}}_{[\alpha]}-\underbrace{\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\frac{\sigma^2-r}{3\Delta S}\begin{bmatrix}-(2S_1+S_2)&-(S_1+2S_2)\\2S_1+S_2&S_1+2S_2\end{bmatrix}+\frac{(2S_1^2+2S_1S_2+2S_2^2)\sigma^2}{6\Delta S^2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix}+\frac{r}{3}\begin{bmatrix}2&1\\1&2\end{bmatrix}\right)}_{[\beta]}\begin{bmatrix}V_1^{\tau}\\V_2^{\tau}\end{bmatrix}^e\right\}$$

$\Delta S = S_2 - S_1$

$S_2 = S_1 + \Delta S$

$$\frac{V_j^{\tau+\Delta\tau}-V_j^{\tau}}{\Delta\tau}=\theta\left(\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau+\Delta\tau}-[\beta]V_j^{\tau+\Delta\tau}\right)+(1-\theta)\left(\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau}-[\beta]V_j^{\tau}\right)$$

$$\left(\frac{1}{\Delta\tau}+\theta[\beta]\right)V_j^{\tau+\Delta\tau}=\left(\frac{1}{\Delta\tau}-(1-\theta)[\beta]\right)V_j^{\tau}+\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\left(\theta\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau+\Delta\tau}+(1-\theta)\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau}\right)$$

$$\begin{bmatrix}2&-1\\-1&2\end{bmatrix}^{-1}\left(\frac{1}{\Delta\tau}+\theta[\beta]\right)V_j^{\tau+\Delta\tau}=\begin{bmatrix}2&-1\\-1&2\end{bmatrix}^{-1}\left(\frac{1}{\Delta\tau}-(1-\theta)[\beta]\right)V_j^{\tau}+\frac{\sigma^2}{\Delta S}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}^{-1}\begin{bmatrix}2&-1\\-1&2\end{bmatrix}^{-1}\left(\theta\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau+\Delta\tau}+(1-\theta)\begin{bmatrix}-S_1^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S1}\\S_2^2\left(\frac{\partial\tilde{V}}{\partial S}\right)\Big|_{S2}\end{bmatrix}^{\tau}\right)$$

All free degrees of freedom cancel out and fixed degrees of freedom are boundary conditions

Can't manipulate the matrices any further because the last term can only be ignored when there is an identity matrix as its coefficient (scalar multiples are fine, they cancel out).

→ Assemble this into global matrices, solve for $V^{\tau+\Delta\tau}$, set $V^{\tau+\Delta\tau}=V^{\tau}$ and the new $V^{\tau+\Delta\tau}$ is unknown (next time step), iterate over and over. (Ignore the fixed dofs those are boundary conditions)

Hilary

(4)

Black-Scholes Equation Using FDM. → Central Differencing + Crank-Nicolson Method.

$-\frac{\partial C}{\partial \tau} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0$. → Use V for value instead of C for call option.

$\frac{\partial V}{\partial \tau} = rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV$

$\frac{\partial V}{\partial S} = \frac{V_{i+1} - V_{i-1}}{2\Delta S}$ $\quad \frac{\partial^2 V}{\partial S^2} = \frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta S^2}$

$$\frac{V_i^{\tau+\Delta\tau} - V_i^{\tau}}{\Delta\tau} = \theta\left[rS\left(\frac{V_{i+1}^{\tau+\Delta\tau} - V_{i-1}^{\tau+\Delta\tau}}{2\Delta S}\right) + \frac{1}{2}\sigma^2 S^2\left(\frac{V_{i+1}^{\tau+\Delta\tau} - 2V_i^{\tau+\Delta\tau} + V_{i-1}^{\tau+\Delta\tau}}{\Delta S^2}\right) - rV_i^{\tau+\Delta\tau}\right]$$

$$+(1-\theta)\left[rS\left(\frac{V_{i+1}^{\tau} - V_{i-1}^{\tau}}{2\Delta S}\right) + \frac{1}{2}\sigma^2 S^2\left(\frac{V_{i+1}^{\tau} - 2V_i^{\tau} + V_{i-1}^{\tau}}{\Delta S^2}\right) - rV_i^{\tau}\right]$$

In the form: $a_1 V_{i-1}^{\tau+\Delta\tau} + a_2 V_i^{\tau+\Delta\tau} + a_3 V_{i+1}^{\tau+\Delta\tau} = b_1 V_{i-1}^{\tau} + b_2 V_i^{\tau} + b_3 V_{i+1}^{\tau}$

$$\frac{V_i^{\tau+\Delta\tau}}{\Delta\tau} - \frac{V_i^{\tau}}{\Delta\tau} - \frac{\theta rS}{2\Delta S}\left(V_{i+1}^{\tau+\Delta\tau} - V_{i-1}^{\tau+\Delta\tau}\right) - \frac{\theta\sigma^2 S^2}{2\Delta S^2}\left(V_{i+1}^{\tau+\Delta\tau} - 2V_i^{\tau+\Delta\tau} + V_{i-1}^{\tau+\Delta\tau}\right) + \theta r V_i^{\tau+\Delta\tau} - \frac{(1-\theta)rS}{2\Delta S}\left(V_{i+1}^{\tau} - V_{i-1}^{\tau}\right) - \frac{(1-\theta)\sigma^2 S^2}{2\Delta S^2}\left(V_{i+1}^{\tau} - 2V_i^{\tau} + V_{i-1}^{\tau}\right)$$

$$+(1-\theta)rV_i^{\tau} = 0$$

$$\left(\frac{\theta rS}{2\Delta S} - \frac{\theta\sigma^2 S^2}{2\Delta S^2}\right)V_{i-1}^{\tau+\Delta\tau} + \left(\frac{1}{\Delta\tau} + \frac{\theta\sigma^2 S^2}{\Delta S^2} + \theta r\right)V_i^{\tau+\Delta\tau} + \left(-\frac{\theta rS}{2\Delta S} - \frac{\theta\sigma^2 S^2}{2\Delta S^2}\right)V_{i+1}^{\tau+\Delta\tau}$$

$$= \left(-\frac{(1-\theta)rS}{2\Delta S} + \frac{(1-\theta)\sigma^2 S^2}{2\Delta S^2}\right)V_{i-1}^{\tau} + \left(\frac{1}{\Delta\tau} - \frac{(1-\theta)\sigma^2 S^2}{\Delta S^2} - (1-\theta)r\right)V_i^{\tau} + \left(\frac{(1-\theta)rS}{2\Delta S} + \frac{(1-\theta)\sigma^2 S^2}{2\Delta S^2}\right)V_{i+1}^{\tau}$$

Boundary Conditions for Call: $V(0)=0$, $V(5000) = 5000 - Ke^{-r\tau}$ → $V(0) = \hat{V}_1$, $V(5000) = \hat{V}_6$

Assume 5 elements with \$1000 steps. (Mesh centered)

Block 1: $V_1 = \hat{V}_1$, Block 6: $V_6 = \hat{V}_6$. Block 2: $a_1\hat{V}_1 + a_2 V_2^{\tau+\Delta\tau} + a_3 V_3^{\tau+\Delta\tau} = b_1\hat{V}_1 + b_2 V_2^{\tau} + b_3 V_3^{\tau}$.

Block 3: $a_1 V_2^{\tau+\Delta\tau} + a_2 V_3^{\tau+\Delta\tau} + a_3 V_4^{\tau+\Delta\tau} = b_1 V_2^{\tau} + b_2 V_3^{\tau} + b_3 V_4^{\tau}$ → block 4 similar.

Block 5: $a_1 V_4^{\tau+\Delta\tau} + a_2 V_5^{\tau+\Delta\tau} + a_3\hat{V}_6 = b_1 V_4^{\tau} + b_2 V_5^{\tau} + b_3\hat{V}_6$

$$\begin{bmatrix} a_2 & a_3 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 \\ 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_1 & a_2 \end{bmatrix}\begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}^{\tau+\Delta\tau} = \begin{bmatrix} (b_1 - a_1)\hat{V}_1 + b_2 V_2^{\tau} + b_3 V_3^{\tau} \\ b_1 V_2^{\tau} + b_2 V_3^{\tau} + b_3 V_4^{\tau} \\ b_1 V_3^{\tau} + b_2 V_4^{\tau} + b_3 V_5^{\tau} \\ b_1 V_4^{\tau} + b_2 V_5^{\tau} + b_3\hat{V}_6^{\tau} - a_3\hat{V}_6^{\tau+\Delta\tau} \end{bmatrix}$$

← start with initial conditions.

$a_1 = \frac{\theta S}{2\Delta S}\left(r - \frac{\sigma^2 S}{\Delta S}\right)$, $a_2 = \left(\frac{1}{\Delta\tau} + \frac{\theta\sigma^2 S^2}{\Delta S^2} + \theta r\right)$, $a_3 = \frac{-\theta S}{2\Delta S}\left(r + \frac{\sigma^2 S}{\Delta S}\right)$

$b_1 = \frac{(1-\theta)S}{2\Delta S}\left(-r + \frac{\sigma^2 S}{\Delta S}\right)$, $b_2 = \left(\frac{1}{\Delta\tau} - (1-\theta)\left(\frac{\sigma^2 S^2}{\Delta S^2} + r\right)\right)$, $b_3 = \frac{(1-\theta)S}{2\Delta S}\left(r + \frac{\sigma^2 S}{\Delta S}\right)$

S = Stock Price
V = Option Price
σ = implied volatility (constant)
K = Option Strike Price (constant)
r = risk-free interest rate (constant).

(5)

Transformation of Black-Scholes PDE to Heat Equation (1D Diffusion).

Source: quantpie youtube channel video titled ↑. (June 18, 2019).

Derivation uses Itô's lemma and random price action assumptions using Geometric Brownian Motion.

Start with the Black-Scholes PDE: $\boxed{\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV=0}$ → $V(s,t)$.

T = time at option price maturity.

For Call Option Pricing

$S(0)$ = known, $V(0)$ = ?

$S(T)$ = known, $V(T)=\max(S(T)-K,0)$

$S(t)$, $V(t)$, and we are solving for $V(t=0)$ with terminal BC's, so this is a backward looking equation.

Reverse the time to make the PDE a forward looking equation with initial BC's. $\boxed{\tau=T-t}$ = time until maturity.

time to maturity

$\ddot{S}(T)$ = known, $\ddot{V}(T)$ = ?

$\ddot{S}(0)$ = known, $\ddot{V}(0)=\max(\ddot{S}(0)-K,0)$.

Denote variable as functions of τ, i.e. $\ddot{S}(\tau)$, $\ddot{V}(s,\tau)$.

Stock price follows geometric Brownian motion: $S(T)=e^{\ln S(t)+(r-\frac{1}{2}\sigma^2)(T-t)+\sigma(W_T-W_t)}$

Ignore the stochastic term $\sigma(W_T-W_t)$ because the Black-Scholes PDE assumes full delta hedging of the portfolio (i.e. the random weighting variables W_T and W_t cancel out).

$\ddot{S}(0)=e^{\ln\ddot{S}(\tau)+(r-\frac{1}{2}\sigma^2)\tau}$, set $\boxed{x=\ln(\ddot{S}(0))=\ln\ddot{S}(\tau)+(r-\frac{1}{2}\sigma^2)\tau}$

Assuming a single stock price, $F(s,\tau)$ and $\ddot{V}(s,\tau)$

Forward Value of Option: $\boxed{F(\tau)=\ddot{V}(\tau)e^{r\tau}}$ → Continuous compounding until maturity date.

Start Transformation: ① $\tau=T-t$, ② $x=\ln\ddot{S}(\tau)+(r-\frac{1}{2}\sigma^2)\tau$, ③ $F(\ddot{S},\tau)=\ddot{V}(\ddot{S},\tau)e^{r\tau}\Rightarrow F_\tau=\ddot{V}_\tau e^{r\tau}$

① : $\frac{\partial V}{\partial S}=\frac{\partial\ddot{V}}{\partial\ddot{S}}$, $\frac{\partial V}{\partial t}=\frac{\partial\ddot{V}}{\partial\tau}\times\frac{\partial\tau}{\partial t}$, $\frac{\partial\tau}{\partial t}=-1\Rightarrow\partial\tau=-\partial t$. $\frac{\partial V}{\partial t}=\frac{-\partial\ddot{V}}{\partial\tau}$.

Sub into Black-Scholes PDE: $\boxed{\frac{-\partial\ddot{V}}{\partial\tau}+\frac{1}{2}\sigma^2\ddot{S}^2\frac{\partial^2\ddot{V}}{\partial\ddot{S}^2}+r\ddot{S}\frac{\partial\ddot{V}}{\partial\ddot{S}}-r\ddot{V}=0}$ (*)① → now moving forward in time.

② : $\ddot{V}(\ddot{S},\tau)\overset{\text{def}}{=}v(x,\tau)$. → Short form $\ddot{V}_\tau\overset{\text{def}}{=}v_\tau$.

$\frac{\partial\ddot{V}}{\partial\ddot{S}}=\frac{\partial v}{\partial x}\cdot\frac{\partial x}{\partial\ddot{S}}$, $\frac{\partial x}{\partial\ddot{S}}=\frac{1}{\ddot{S}}$. → $\frac{\partial\ddot{V}}{\partial\ddot{S}}=\frac{\partial v}{\partial x}\left(\frac{1}{\ddot{S}}\right)$. $\frac{\partial^2\ddot{V}}{\partial\ddot{S}^2}=\frac{\partial}{\partial\ddot{S}}\left(\frac{\partial v}{\partial x}\left(\frac{1}{\ddot{S}}\right)\right)=\frac{\partial}{\partial\ddot{S}}\left(\frac{\partial v}{\partial x}\right)\left(\frac{1}{\ddot{S}}\right)-\frac{\partial}{\partial x}\left(\frac{1}{\ddot{S}^2}\right)=\frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x}\right)\left(\frac{1}{\ddot{S}}\right)\left(\frac{\partial x}{\partial\ddot{S}}\right)-\frac{\partial v}{\partial x}\left(\frac{1}{\ddot{S}^2}\right)$

$=\frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x}\right)\left(\frac{1}{\ddot{S}^2}\right)-\frac{\partial v}{\partial x}\left(\frac{1}{\ddot{S}^2}\right)=\left(\frac{1}{\ddot{S}^2}\right)\left(\frac{\partial^2v}{\partial x^2}-\frac{\partial v}{\partial x}\right)$

$\frac{\partial\ddot{V}}{\partial\tau}=\frac{\partial v}{\partial\tau}+\frac{\partial v}{\partial x}\cdot\frac{\partial x}{\partial\tau}$ → When τ changes, x changes (Total derivative). $\frac{\partial x}{\partial\tau}=r-\frac{1}{2}\sigma^2$

$\frac{\partial\ddot{V}}{\partial\tau}=\frac{\partial v}{\partial\tau}+\frac{\partial v}{\partial x}\left(r-\frac{1}{2}\sigma^2\right)$

Sub into (*)①: $\frac{-\partial v}{\partial\tau}-\frac{\partial v}{\partial x}\left(r-\frac{1}{2}\sigma^2\right)+\frac{1}{2}\sigma^2\left(\frac{\partial^2v}{\partial x^2}-\frac{\partial v}{\partial x}\right)+r\frac{\partial v}{\partial x}-rv=0$.

$\boxed{\frac{-\partial v}{\partial\tau}+\frac{1}{2}\sigma^2\frac{\partial^2v}{\partial x^2}-rv=0}$ (*)②

③ : $\ddot{V}_\tau=F_\tau e^{-r\tau}=v_\tau$, $\frac{\partial v}{\partial x}=e^{-r\tau}\left(\frac{\partial F}{\partial x}\right)$, $\frac{\partial v}{\partial\tau}=e^{-r\tau}\left(\frac{\partial F}{\partial\tau}\right)-rFe^{-r\tau}$, $\frac{\partial^2v}{\partial x^2}=e^{-r\tau}\left(\frac{\partial^2F}{\partial x^2}\right)$.

$-e^{-r\tau}\left(\frac{\partial F}{\partial\tau}\right)+rFe^{-r\tau}+\frac{1}{2}\sigma^2e^{-r\tau}\left(\frac{\partial^2F}{\partial x^2}\right)-rFe^{-r\tau}=0$

$-\frac{\partial F}{\partial\tau}+\frac{1}{2}\sigma^2\left(\frac{\partial^2F}{\partial x^2}\right)=0$

$\boxed{\frac{\partial F}{\partial\tau}=\frac{1}{2}\sigma^2\frac{\partial^2F}{\partial x^2}}$ (*)③ — Resembling a heat equation (diffusion equation).

Boundary Conditions: $\ddot{S}_\tau=e^{x-(r-\frac{1}{2}\sigma^2)\tau}$, $v_\tau=F_\tau e^{-r\tau}$. $\ddot{S}(0)=e^x$, $v(x,0)=v_0=F_0=F(x,0)$, $\ddot{V}_0=v_0$.

$V(S,T)=\max(S(T)-K,0)$

$\ddot{V}(\ddot{S},0)=\max(\ddot{S}(0)-K,0)$

$v(x,0)=\max(e^x-K,0)$

$F(x,0)=\max(e^x-K,0)$. $F(\infty,\tau)=\left(e^{x-(r-\frac{1}{2}\sigma^2)\tau}-Ke^{-r\tau}\right)e^{r\tau}$, $F(0,\tau)=0$.

Hilroy

(6)

Apply MWR to the Black-Scholes Equation (1D Diffusion).

$\frac{\partial F}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial x^2} = 0.$

$\int_{x_1}^{x_2} \left(\frac{\partial F}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial x^2}\right) N_i \, dx = 0.$ $\quad \tilde{F} = N_1^e F_1 + N_2^e F_2$ (linear approximation).

$\int_{x_1}^{x_2} \left(\frac{\partial N_j F_j}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 N_j F_j}{\partial x^2}\right) N_i \, dx = 0.$ $\quad = \sum_{j=1}^{2} N_j^e F_j$ $\quad N_j = \{N\}$, $N_i = \langle N \rangle$

(*) $\int_{x_1}^{x_2} \frac{\partial F}{\partial \tau} N_i \, dx - \frac{1}{2}\sigma^2 \int_{x_1}^{x_2} \frac{\partial^2 N_j}{\partial x^2} N_i \, dx \, (F_j) = 0$

(A) (B)

For (B): IBP: $u = N_i$, $du = \frac{dN_i}{dx}dx$, $dv = \frac{\partial^2 N_j}{\partial x^2}dx$, $v = \frac{\partial \tilde{N}_j}{\partial x}$

$\int_{x_1}^{x_2} \frac{\partial^2 N_j}{\partial x^2} N_i \, dx = N_i \left(\frac{\partial \tilde{N}_j}{\partial x}\right)\Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} dx$

Sub (B) into (*)

$\int_{x_1}^{x_2} \frac{\partial F}{\partial \tau} N_i \, dx - \frac{1}{2}\sigma^2 \left(N_i \left(\frac{\partial \tilde{N}_j}{\partial x}\right)\Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} dx\right)(F_j) = 0$

$\int_{x_1}^{x_2} \frac{\partial F}{\partial \tau} N_i \, dx + \left(\frac{1}{2}\sigma^2 \int_{x_1}^{x_2} \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} dx\right)(F_j) = \frac{1}{2}\sigma^2 N_i \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}^{x_2}$

(*2) $\frac{\Delta x}{2}\left(\int_{-1}^{1} \frac{\partial F}{\partial \tau} N_i \, d\eta + \left(\frac{1}{2}\sigma^2 \int_{-1}^{1} \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} d\eta\right)(F_j)\right) = \frac{1}{2}\sigma^2 N_i \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}^{x_2}$

(A) (C)

Look at RHS $\quad \frac{1}{2}\sigma^2 \left(N_i \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2} - N_i \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}\right)$

When $i=1$: $\frac{1}{2}\sigma^2 \left(N_1(x_2)\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2} - N_1(x_1)\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}\right) = -\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}$

When $i=2$: $\frac{1}{2}\sigma^2 \left(N_2(x_2)\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2} - N_2(x_2)\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1}\right) = \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2}$

$RHS = \frac{1}{2}\sigma^2 \begin{bmatrix} -\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1} \\ \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2} \end{bmatrix}$

For (A): $\int_{-1}^{1} \frac{\partial F}{\partial \tau} N_i \, d\eta = \frac{\partial F_j}{\partial \tau} \int_{-1}^{1} N_j N_i \, d\eta = \frac{\partial F_j}{\partial \tau}\left(\frac{1}{3}\right)\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

For (C): $\int_{-1}^{1} \frac{\partial N_j}{\partial x} \frac{\partial N_i}{\partial x} d\eta = \left(\frac{1}{\Delta x}\right)^2 \int_{-1}^{1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} d\eta = \frac{2}{\Delta x^2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

$$\frac{\Delta x}{2}\left(\frac{\partial F_j}{\partial \tau}\left(\frac{1}{3}\right)\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} + \left(\frac{1}{2}\sigma^2\left(\frac{2}{\Delta x^2}\right)\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\right)\{F\}\right) = \frac{1}{2}\sigma^2 \begin{bmatrix} -\left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_1} \\ \left(\frac{\partial \tilde{F}}{\partial x}\right)\Big|_{x_2} \end{bmatrix}$$

$$\frac{\Delta x}{6}\frac{\partial F_j}{\partial \tau}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} + \frac{\sigma^2}{2\Delta x}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\{F\} = \frac{1}{2}\sigma^2 \begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}$$

$$\frac{\partial F_j}{\partial \tau} = \frac{6}{3\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\left(\frac{1}{2}\sigma^2 \begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix} - \frac{\sigma^2}{2\Delta x}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\{F\}\right)$$

$$\frac{\partial F_j}{\partial \tau} = \frac{3\sigma^2}{3\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix} - \frac{3\sigma^2}{3\Delta x^2}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\{F\} = \underbrace{\frac{\sigma^2}{\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}}_{[\alpha]} - \underbrace{\frac{3\sigma^2}{\Delta x^2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}_{[\beta]}\{F\}$$

Apply theta differencing

$$\frac{[F_j]^{\tau+\Delta\tau} - [F_j]^{\tau}}{\Delta\tau} = \theta\left\{\frac{\sigma^2}{\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}[\alpha]^{\tau+\Delta\tau} - [\beta][F_j]^{\tau+\Delta\tau}\right\} + (1-\theta)\left\{\frac{\sigma^2}{\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}[\alpha]^{\tau} - [\beta][F_j]^{\tau}\right\}$$

$$\left(\frac{1}{\Delta\tau} + \theta[\beta]\right)[F_j]^{\tau+\Delta\tau} = \left(\frac{1}{\Delta\tau} - (1-\theta)[\beta]\right)[F_j]^{\tau} + \frac{\sigma^2}{\Delta x}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\left(\theta\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}^{\tau+\Delta\tau} + (1-\theta)\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}^{\tau}\right)$$

$$\frac{\Delta x}{\sigma^2}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}^{-1}\left(\frac{1}{\Delta\tau} + \theta[\beta]\right)[F_j]^{\tau+\Delta\tau} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}^{-1}\left(\frac{1}{\Delta\tau} - (1-\theta)[\beta]\right)[F_j]^{\tau}\frac{\Delta x}{\sigma^2} + \cancel{\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}^{-1}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}}\left(\theta\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}^{\tau+\Delta\tau} + (1-\theta)\begin{bmatrix} -\frac{\partial \tilde{F}}{\partial x}\Big|_{x_1} \\ \frac{\partial \tilde{F}}{\partial x}\Big|_{x_2} \end{bmatrix}^{\tau}\right)$$

↳ Same logic as bottom of page (3). Note Δx is not constant, unlike ΔS on pages (1) to (3)

(7)

Sanity Check with Assignment 3 – Chloride Concrete Problem.

$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial \phi}{\partial x}\right) - V\frac{\partial \phi}{\partial x}$, $V=0$

$\frac{\partial \phi}{\partial t} - D\frac{\partial^2 \phi}{\partial x^2} = 0.$

Apply GMWR

$\int_{x_1}^{x_2}\left(\frac{\partial \phi}{\partial t} - D\frac{\partial^2 \phi}{\partial x^2}\right)N_i\,dx = 0 \qquad \tilde{\phi} = \phi_1 N_1 + \phi_2 N_2 = \sum_{j=1}^{2}\phi_j N_j$

$\int_{x_1}^{x_2} N_j N_i\,dx\,\frac{\partial \phi_j}{\partial t} - D\int_{x_1}^{x_2}\frac{\partial^2 N_j}{\partial x^2}N_i\,dx\,\phi_j = 0$

$\int_{x_1}^{x_2} N_j N_i\,dx\,\frac{\partial \phi_j}{\partial t} - D\int_{x_1}^{x_2}\frac{\partial^2 N_j}{\partial x^2}N_i\,dx\,\phi_j\Big) = 0$

$\frac{\Delta x}{2}\left(\int_{-1}^{1} N_j N_i\,d\eta\,\frac{\partial \phi_j}{\partial t} + D\int_{-1}^{1}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}\,d\eta\,\phi_j\right) = DN_i\left(\frac{\partial \tilde{\phi}}{\partial x}\right)\Big|_{x_1}^{x_2}$

Apply Theta Differencing

$\int_{-1}^{1} N_j N_i\,d\eta\left(\frac{\phi_j^{t+\Delta t} - \phi_j^t}{\Delta t}\right) + D\int_{-1}^{1}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}\,d\eta\left(\theta\phi_j^{t+\Delta t} + (1-\theta)\phi_j^t\right) = \frac{2D}{\Delta x}N_i\left(\frac{\partial \tilde{\phi}}{\partial x}\right)\Big|_{x_1}^{x_2}$

$\left(\frac{\int_{-1}^{1} N_j N_i\,d\eta}{\Delta t} + \theta D\int_{-1}^{1}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}\,d\eta\right)\phi_j^{t+\Delta t} = \frac{2D}{\Delta x}N_i\left(\frac{\partial \tilde{\phi}}{\partial x}\right)\Big|_{x_1}^{x_2} + \left(\frac{\int_{-1}^{1} N_j N_i\,d\eta}{\Delta t} - (1-\theta)D\int_{-1}^{1}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}\,d\eta\right)\phi_j^t$

$\left(\frac{1}{3\Delta t}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} + \frac{2\theta D}{\Delta x^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\right)\phi_j^{t+\Delta t} = \cancel{\frac{2D}{\Delta x}N_i\left(\frac{\partial \tilde{\phi}}{\partial x}\right)\Big|_{x_1}^{x_2}} + \left(\frac{1}{3\Delta t}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} - \frac{2(1-\theta)D}{\Delta x^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\right)\phi_j^t$

↑ Cancels out as seen in pages ⑥ and ③