

Solutions of Black-Scholes Equation by Some Numerical Approaches

Md. Mehedi Hasan*

Department of General Education, University of Frontier Technology, Bangladesh

E-mail: mehedi0001@uftb.ac.bd

ORCID iD: <https://orcid.org/0009-0007-7174-5690>

*Corresponding Author

Md. Biplob Hossain

Department of EEE, Dhaka International University, Dhaka-1212, Bangladesh

E-mail: biplob.amath@gmail.com

ORCID iD: <https://orcid.org/0009-0001-1082-2468>

Received: 16 July, 2025; Revised: 18 September, 2025; Accepted: 04 November, 2025; Published: 08 December, 2025

Abstract: The Black-Scholes equation plays an important role in financial mathematics for the evaluation of European options. It is a fundamental PDE in financial mathematics, models the price dynamics of options and derivatives. While a closed-form of analytical solution exists for European options, numerical methods remain essential for validating computational approaches and extending solutions to more complex derivatives. This study explores and compares various numerical techniques for solving the Black-Scholes partial differential equation, including the finite difference method (explicit, implicit, and Crank-Nicolson schemes), and Monte Carlo simulation. Each method is implemented and tested against the analytical Black-Scholes formula to assess accuracy, convergence, and computational efficiency. The results demonstrate the strengths and limitations of each numerical approach, providing insights into their suitability for different option pricing scenarios. This comparative analysis highlights the importance of method selection in practical financial modeling applications.

Index Terms: European Options, Black-Scholes Equation, Partial Differential Equation, Financial Mathematics, Numerical Techniques

1. Introduction

The Black-Scholes equation was developed by Fischer Black and Myron Scholes and then expanded by Robert Merton in 1973 [1]. It presents a theoretical framework for pricing European-style options, which are financial derivatives that grant the holder the right but not the duty to buy or sell an asset at a predetermined price on a future date [2, 3]. It is a partial differential equation (PDE) that describes the evolution of an option's price over time while accounting for factors such as the underlying asset's price, time to expiration, volatility, and the risk-free interest rate [4]. The key finding is that an option can be hedged by combining the underlying asset and a risk-free bond, removing arbitrage opportunities and calculating a fair option price [5]. The price of the underlying asset moves in a geometric Brownian motion with constant volatility, and the markets are frictionless, which means there are no transaction costs or taxes and the assets can be traded continuously [6, 7]. There are no arbitrage opportunities, which means that the derivative's price should prevent risk-free profit, and the risk-free interest rate is constant and known. We assume the option is of the European type, meaning it can only be exercised when it expires [8]. We assume the option is of the European kind, which means it can only be exercised when it expires [9]. There are several mathematical methods available for solving the Black Scholes equation, including the Analytical Method, Perturbation Method, Green's Function Method, Transform Method, Numerical Method, and Monte Carlo Simulations. Transform Methods and Green's Function Techniques are theoretical approaches that can be useful in some applications but are less commonly employed in practice, whilst Analytical Methods provide precise solutions but are confined to simple examples such as European choices. So, we will test the numerical techniques and the Monte Carlo Simulation Method [10, 11]. In this paper, we will try to determine the optimal method for computationally solving the Black Scholes equation. There is some existing work such as 'A robust and accurate finite difference method for a generalized Black Scholes equation by Z Cen, A Le. In this paper a numerical method is introduced for a generalized Black-Scholes equation, which is used

for option pricing and this method is based on a central difference spatial discretization on a piecewise uniform mesh and an implicit time stepping technique [12]. Another paper 'Finite difference method for the Black-Scholes equation without boundary conditions by Darae Jeong, Minhyun Yoo & Junseok Kim'. This paper presents an efficient finite difference method for solving the Black-Scholes equation without boundary conditions which presents an explicit finite difference scheme which is free from a far-field boundary condition to solve the equation numerically [13]. In this paper, we will use the Finite difference methods (FDM) to demonstrate our numerical approach. The FDMs are commonly used to numerically solve PDEs, such as the Black-Scholes equation [8, 11]. The stock price and time domains are discretized into a grid, and the Black-Scholes PDE is converted into an algebraic equation system. The Explicit Method, Implicit Method, along with Crank Nicolson Method are the three most prevalent FDMs used in numerical analysis [11, 14, 15]. The Explicit Method approximates derivatives with forward differences for time and central differences for space, which is straightforward but may necessitate tiny temporal increments to assure stability [16]. The Implicit Method employs backward differences, which results in a stable solution despite having a greater time step. It does, however, require solving a system of equations at each step. The Crank-Nicolson Method combines explicit and implicit methods using an average of both approaches, making it unconditionally stable and second-order accurate in both time and space, making it a popular choice for solving the Black-Scholes equation [17, 18]. Another option is Monte Carlo methods [9, 17], which are stochastic processes for simulating the underlying asset's price path using geometric Brownian motion [11]. The option price is derived by averaging the payoff across numerous simulated price pathways and discounting to the present value, which is adaptable and can handle a wide range of options, including those with complex payoffs or many underlying assets. The explicit, implicit, Crank-Nicolson, and Monte Carlo simulation methods each have advantages and disadvantages depending on the specific needs of the problem in terms of accuracy, stability, and computational efficiency, but they are one of the best methods for solving the Black Scholes Equation. In the section that follows below, we will discuss each of these methods and try to solve the Black Scholes Equation using them [11].

2. Black Scholes Equation

The general form of the Black Scholes PDE for a European call option is given by:

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

Here, $V(S, t)$ represents the option price as a function of the asset price 'S' and time to expiration 't'. ' σ ' represents the degree of risk of the underlying asset and ' r ' represents the risk-free interest rate. We will now evaluate the above equation (1) using the FDM and Monte Carlo Simulation Methods. In the following subsection, all approaches will be thoroughly explored, and the answer will be reached using a mathematical and numerical strategy.

2.1. Explicit Method

The Black-Scholes equation, as represented in equation (1), is a partial differential equation used to simulate the price evolution of financial derivatives, specifically European options [6]. Its numerical solution can be calculated using a variety of techniques, including the explicit finite difference method. The explicit finite difference method is a simple approach for numerically solving the Black-Scholes equation [8]. It performs well for small grids and simple boundary conditions, but becomes unstable or inefficient for finer grids or more complicated financial instruments [10]. To maintain stability, the time stride must be carefully monitored. This approach can handle a wide range of financial derivatives, not simply European options, with numerical modification [5]. Exotic options, such as barrier or American options, can be addressed by modifying boundary conditions or introducing early exercise elements. The explicit finite difference method involves discretizing both time and space, and approximating the derivatives in the Black-Scholes PDE using finite differences [5].

We can write the Black- Scholes Equation into the explicit form of all independent variables

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

Approximating $V(S + \partial S, t)$ & $V(S - \partial S, T)$ at the point (j, n)

The Taylor expansions are given as follows of the variable A:

$$V(S + \sigma S, t) = V(S, t) + \frac{\partial V}{\partial S} \sigma S + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} \sigma^2 S^2 + \frac{1}{6} \frac{\partial^3 V}{\partial S^3} \sigma^3 S^3 + O(\sigma^4 S^4) \quad (3)$$

$$V(S - \sigma S, t) = V(S, t) - \frac{\partial V}{\partial S} \sigma S + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} \sigma^2 S^2 - \frac{1}{6} \frac{\partial^3 V}{\partial S^3} \sigma^3 S^3 + O(\sigma^4 S^4) \quad (4)$$

Discretization of space:

Let the stock price range 'S' be discretized into 'M' grid points with spacing ΔS . The grid points are:

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

Set up the stock price grid with

$$\Delta S = \frac{S_{max} - S_{min}}{M}.$$

Discretization of time:

Let the time 't' be discretized into N grid points with spacing Δt and divide the time interval [0, T] into N time steps with step size Δt . T is the maturity time.

$$t_n = n\Delta t, \quad n = 0, 1, 2, \dots, N.$$

Set up the time grid with $\Delta t = \frac{T}{N}$.

From Equation (3), ignoring higher order terms and using forward difference operator

$$\frac{\partial V(S, t)}{\partial S} \approx \frac{V_{j+1}^n - V_j^n}{\sigma S} \quad (5)$$

From Equation (4), using backward difference operator

$$\frac{\partial V(S, t)}{\partial S} \approx \frac{V_j^n - V_{j-1}^n}{\sigma S} \quad (6)$$

Subtracting Equation (5) from Equation (6) and taking the first order partial derivatives

$$\frac{\partial V(S, t)}{\partial S} \approx \frac{V_{j+1}^n - V_{j-1}^n}{2\sigma S} \quad (7)$$

Adding Equation (3) and (4) and taking the second order partial derivative to approximate $\frac{\partial^2 V}{\partial S^2}$ at the point (j, n) , we obtain

$$\frac{\partial^2 V}{\partial S^2} \approx \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\sigma^2 S^2} \quad (8)$$

Approximating $V(S, t + \sigma t)$ at the point (j, n)

The Taylor expansions are given as follows of the variable t:

$$V(S, t + \sigma t) = V(S, t) + \frac{\partial V}{\partial S} \sigma t + \frac{1}{2} \frac{\partial^2 V}{\partial S^2} \sigma^2 t^2 + \frac{1}{6} \frac{\partial^3 V}{\partial S^3} \sigma^3 t^3 + O(\sigma^4 t^4) \quad (9)$$

Thus, the forward difference operator for time is

$$\frac{\partial V(S, t)}{\partial t} \approx \frac{V_j^{n+1} - V_j^n}{\sigma t} \quad (10)$$

The Finite forward difference approximations are given as follows:

The time derivative $\frac{\partial V}{\partial t}$ is approximated using a forward difference:

$$\frac{\partial V}{\partial t} \approx \frac{V_j^{n+1} - V_j^n}{\Delta t} \quad (11)$$

The first spatial derivative $\frac{\partial V}{\partial S}$ is approximated using a central difference:

$$\frac{\partial V}{\partial S} \approx \frac{V_{j+1}^n - V_{j-1}^n}{2\Delta S} \quad (12)$$

The second spatial derivative $\frac{\partial^2 V}{\partial S^2}$ is approximated using a central difference:

$$\frac{\partial^2 V}{\partial S^2} \approx \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\Delta S^2} \quad (13)$$

Substitute the value of Equation (11), (12) & (13) into the Equation (2) we have:

$$V_j^{n+1} = V_j^n + \Delta t \left[\frac{1}{2} \delta^2 S_j^2 \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{\Delta S^2} + r S_j \frac{V_{j+1}^n - V_{j-1}^n}{2\Delta S} - r V_j^n \right] \quad (14)$$

This equation can be applied in a time-stepping scheme, beginning with the initial condition (payoff at maturity) and working backward in time to calculate the option price at earlier dates. Although the explicit method is quite beneficial because it is simple, requires little memory, and is flexible. It may operate by direct time-stepping, which means that the payout condition at maturity is applied to earlier times without the need for Matrix inversion or sophisticated numerical algebra. However, this method has numerous drawbacks, including stability difficulties because the explicit method is conditionally stable. According to the Courant-Friedrichs-Lewy (CFL) criterion [7, 8, 10], the time step Δt must be significantly smaller than the space step ΔS . Here the CFL condition as:

$$\Delta t \leq \frac{\Delta S^2}{\delta^2 S_{max}^2} \quad (15)$$

When the time step is too large, the approach becomes unstable, resulting in fluctuating or diverging findings. This renders the explicit method inefficient for fine grids or long-time spans. Another disadvantage of this method is that the convergence rate is extremely slow, therefore the time step must be modest in order to preserve stability (and thus accuracy). This increases the amount of time steps required to reach a solution, making the approach slower than implicit methods that allow for longer time steps while maintaining stability. This approach likewise has accuracy limits, is unable to properly handle complex boundary conditions, and takes more computational effort for lengthy maturities. For all of these reasons, we'll now look at the Implicit Finite Difference Method.

2.2. Implicit Method

The implicit finite difference method is an increasingly common strategy for solving numerically the Black-Scholes equation, which triumphs over some of the constraints of the explicit method, especially respect to stability [8, 10, 11]. Unlike the explicit method, the implicit time-stepping scheme is unconditionally stable, allowing for larger time steps without instability, making it more efficient for applications [11].

The implicit form of the Black-Scholes equation, which employs the backward finite difference approach, can be calculated beginning with equation (14) because it can go back in time and compute the option price at earlier times.

$$\frac{V_j^{n+1} - V_j^n}{\Delta t} = \frac{1}{2} \delta^2 S_j^2 \frac{V_{j+1}^{n+1} - 2V_j^{n+1} + V_{j-1}^{n+1}}{\Delta S^2} + r S_j \frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{2\Delta S} - r V_j^{n+1} \quad (16)$$

Now by rearranging this equation gives a tridiagonal system of equations:

$$a_j V_{j-1}^{n+1} + b_j V_j^{n+1} + c_j V_{j+1}^{n+1} = V_j^n \quad (17)$$

Where the coefficients a_j , b_j and c_j are given by:

$$a_j = -\frac{1}{2} \frac{\delta^2 S_j^2}{\Delta S^2} + \frac{1}{2} \frac{r S_j}{\Delta S} \quad (18)$$

$$b_j = \frac{1}{\Delta t} + \frac{\sigma^2 S_j^2}{\Delta S^2} + r \quad (19)$$

$$c_j = -\frac{1}{2} \frac{\delta^2 S_j^2}{\Delta S^2} - \frac{1}{2} \frac{r S_j}{\Delta S} \quad (20)$$

This gives the tridiagonal matrix of the form

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & a_3 & b_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & b_M \end{bmatrix} \begin{bmatrix} V_1^{n+1} \\ V_2^{n+1} \\ V_3^{n+1} \\ \vdots \\ V_M^{n+1} \end{bmatrix} = \begin{bmatrix} V_1^n \\ V_2^n \\ V_3^n \\ \vdots \\ V_M^n \end{bmatrix} \quad (21)$$

This matrix system is solved for V_j^{n+1} the option prices at time step t_{n+1} . This system can be solved using Gaussian Elimination method or Thomas Algorithm.

The Implicit Finite Difference technique is unconditionally stable, unlike the explicit technique, which requires a short time step (Δt) for stability. This means we can use longer time steps without worrying about numerical instabilities, making it ideal for extended maturities or volatile options. The approach is stable for any Δt , allowing for bigger time steps and minimizing the total number of steps needed. This makes the implicit method considerably more economical for many tasks, particularly those requiring accuracy over a long-time horizon. The implicit method is more accurate than the explicit method because it allows for bigger time steps without causing instability. This indicates that we can utilize tighter temporal grids to improve the solution's accuracy without consequently decreasing the time step size. This method is especially useful for options with a long time to maturity since it can accommodate large time to expiration without requiring a huge number of time steps, as would the explicit method due to the stability criterion. This method is more adaptable to complex boundary conditions [8, 10] (e.g., for American options with early exercise features or exotic options with special payoff structures) because it works well even with larger time steps and does not suffer from the same stability constraints as the explicit method. When dealing with high volatility (δ) in the underlying asset, the implicit method can remain stable and reliable even when asset prices experience large swings. That is, this method may be more efficient than the explicit way since it effectively addresses practically all of the concerns identified by the explicit method. However, the mathematical explanation also reveals several shortcomings of this strategy, such as: The implicit method is more complicated to implement than the explicit method. Each time step entails solving a system of linear equations (usually tridiagonal), necessitating more advanced programming abilities and computational resources. At each time step, you must solve a tridiagonal matrix system utilizing techniques such as the Thomas algorithm. This is more computationally expensive than the explicit method, which just requires us to change values explicitly. Thus, while fewer time steps may be required, each step is computationally more expensive. While the implicit technique is more stable and accurate, it uses more memory than the explicit method. Larger matrices must be stored and manipulated at each time step in order to solve the system of equations, which might be a limitation for huge issues or multidimensional models. For non-linear issues (for example, when pricing American options that require free boundary conditions for early exercise), the implicit method can be more difficult to apply. Nonlinearity is frequently handled using techniques such as iterative approaches or penalty methods, which adds complexity. If the choice matures quickly, the additional stability provided by the implicit technique may be unnecessary. In such circumstances, the explicit method may be easier and faster, rendering the implicit method excessive for these shorter time spans. The implicit approach is an effective tool, especially in situations where numerical stability and accuracy are critical. While more difficult to construct and computationally heavy per time step, it provides robustness, making it a preferred choice for many financial applications, particularly when dealing with long-maturity options, high volatility, or complex payoffs [5, 8]. Based on the above arguments, we may conclude that both the explicit and implicit methods have limits, which we will attempt to address in the following part utilizing the Crank Nicolson technique.

2.3. Crank Nicolson Method

The Crank-Nicolson method is another prominent numerical approach for solving partial differential equations (PDEs), such as the Black-Scholes equation [18]. It is a finite difference approach that incorporates the best qualities of both the explicit and implicit methods, resulting in second-order accuracy in both time and space, as well as unconditional stability, similar to the implicit method. It is commonly utilized in option pricing due to its balance of accuracy and computational efficiency. The Crank-Nicolson approach discretizes time by averaging explicit and implicit procedures. This leads in increased precision and stability. The explicit technique employs forward differences for time, which is simple but conditionally stable, whereas the implicit method uses backward differences for time, which is unconditionally stable but more difficult. However, the Crank Nicolson Method employs an average of forward and backward differences, which is unconditionally stable and second-order accurate in both time and space. Because the Crank Nicolson method averages the finite difference approximation of the explicit and implicit methods for the Black Scholes equation, we can use the above mathematical explanation for this method as well, and now, using discretization on the Matrix system shown in Equation (21), at the $n + 1$ time step, can be written as [18]:

$$\frac{V_j^{n+1} - V_j^n}{\Delta t} = \frac{1}{2} (\mathcal{L}V_j^{n+1} + \mathcal{L}V_j^n) \quad (22)$$

Where $\mathcal{L}$ is the differential operator applied to V :

$$\mathcal{L}V_j = \frac{1}{2} \delta^2 S_j^2 \frac{V_{j+1} - 2V_j + V_{j-1}}{\Delta S^2} + rS_j \frac{V_{j+1} - V_{j-1}}{2\Delta S} - rV_j \quad (23)$$

Rearranging this into a form suitable for the Crank-Nicolson method using Equation (14) and Equation (16) we have.

$$V_{j-1}^{n+1} + \beta_j V_j^{n+1} + \gamma_j V_{j+1}^{n+1} = \alpha_j V_{j-1}^n + \beta_j V_j^n + \gamma_j V_{j+1}^n \quad (24)$$

Where the coefficients are:

$$\alpha_j = \frac{\Delta t}{4} \left(\frac{\delta^2 S_j^2}{\Delta S^2} - \frac{r S_j}{\Delta S} \right) \quad (25)$$

$$\beta_j = 1 + \frac{\Delta t}{2} \left(\frac{\delta^2 S_j^2}{\Delta S^2} + r \right) \quad (26)$$

$$\gamma_j = \frac{\Delta t}{4} \left(\frac{\delta^2 S_j^2}{\Delta S^2} + \frac{r S_j}{\Delta S} \right) \quad (27)$$

At each time step n , this leads to a tridiagonal matrix system for V_{j+1}^n (the option prices at the next time step), which can be written as:

$$\begin{bmatrix} \beta_1 & \gamma_1 & 0 & & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \cdots & 0 \\ 0 & \alpha_3 & \beta_3 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{M-1} & \beta_{M-1} \end{bmatrix} \begin{bmatrix} V_1^{n+1} \\ V_2^{n+1} \\ V_3^{n+1} \\ \vdots \\ V_{M-1}^{n+1} \end{bmatrix} = \begin{bmatrix} \widetilde{V}_1^n \\ \widetilde{V}_2^n \\ \widetilde{V}_3^n \\ \vdots \\ \widetilde{V}_{M-1}^n \end{bmatrix} \quad (28)$$

Here,

$$\widetilde{V}_1^n = \alpha_j V_{j-1}^n + \beta_j V_j^n + \gamma_j V_{j+1}^n \quad (29)$$

Right-hand side at the current time step.

The Thomas Algorithm can be used to efficiently solve this tridiagonal matrix situation. To solve the following equations, we require initial and boundary conditions. Here The initial condition is given below: At $S = 0$, the option price is zero ($V(0, t) = 0$ for a call option), and the following boundary conditions are applied: At, $V = V_{max}$ we assume the price grows linearly as S increases

$$\text{i.e., } V(S_{max}, t) = S_{max} - K e^{-r(T-t)}$$

where K represents the strike price. After running the Thomas algorithm, we will have updated option pricing at the new time step. This process is continued recursively for each time step n until the beginning time $t = 0$ at which point we may calculate the option price using the original stock price S_0 .

The graphical results created for the above FDM are presented below.

2.4. Result Section for FDM

The option price vs. stock price graph for various maturity periods t is obtained from 'Fig-1 and Fig-2' using the Explicit Method. This produces a very basic and adaptable outcome by directly stepping across time. It illustrates that after reaching a specific range of stock price, the option price grows exponentially in almost every maturity term, as illustrated in Fig.1. and Fig.2., making it conditionally unstable for the CFL condition. Also, it depicts a delayed convergence and has a poorer accuracy than the genuine case.

The Implicit Method yields the option price vs stock price graph for various maturity periods t from Fig.3. and Fig.4.. This graphic demonstrates unconditional stability, which doesn't require a short time step Δt . This demonstrates the stability of any Δt , allowing for bigger time increments. The image also indicates that we may calculate the highest option price for a long-term maturity period based on the stock price. This demonstrates adaptation to complex boundary circumstances. For significant volatility in the underlying asset, this graph provides a solid and trustworthy result even as asset prices fluctuate dramatically. The significant disadvantage highlighted by this figure is that if the option matures quickly, the additional stability provided by the implicit technique may be superfluous. In such circumstances, the explicit method may be easier and faster, rendering the implicit method excessive for these shorter time spans. Furthermore, because the graph is more efficient in terms of stability and accuracy, it typically uses more memory than the explicit technique.

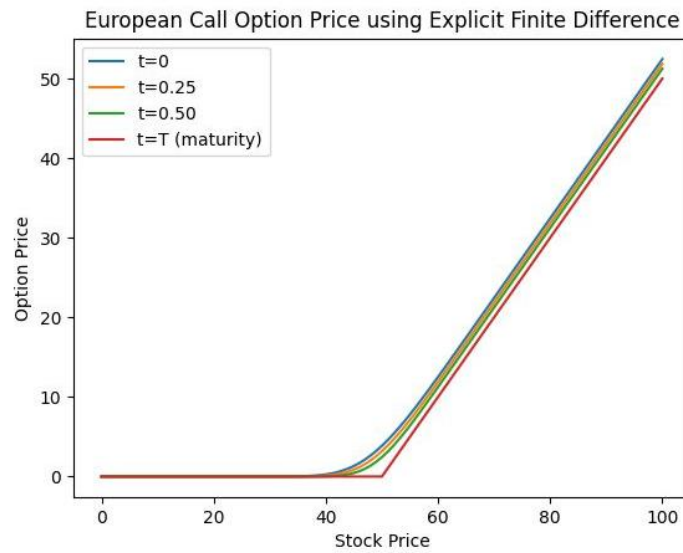


Fig. 1. European Call Option Price Calculation Using the Explicit Method

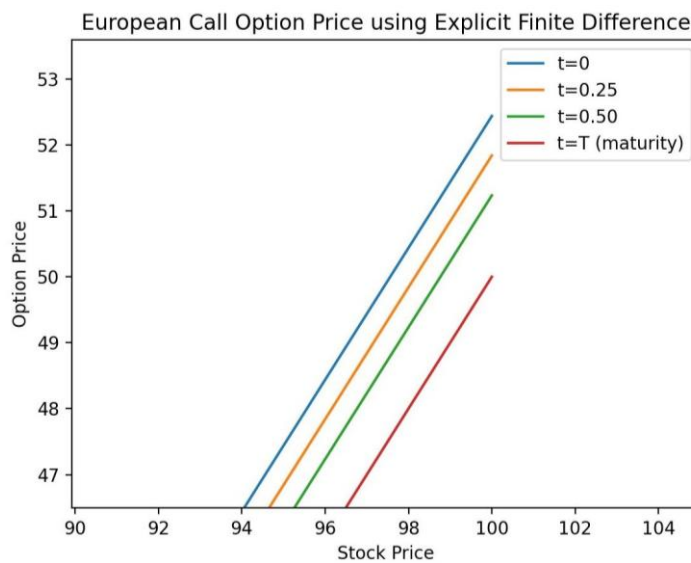


Fig. 2. European Call Option Price Calculation Using the Explicit Method

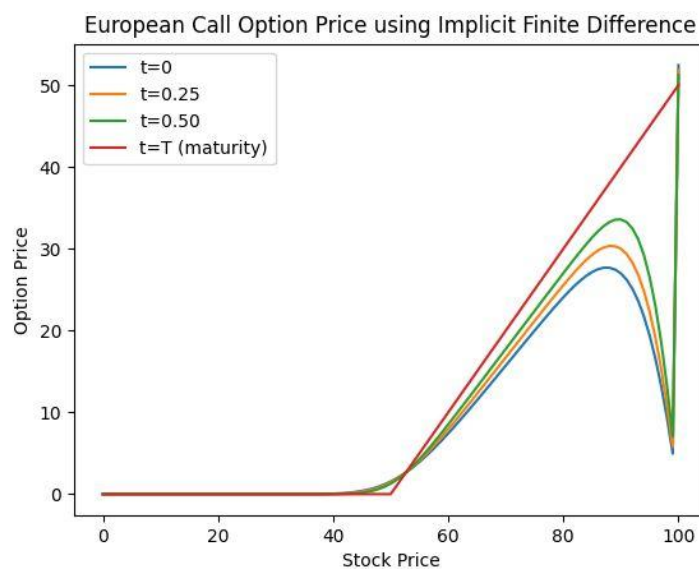


Fig. 3. European Call Option Price Calculation Using the Implicit Method

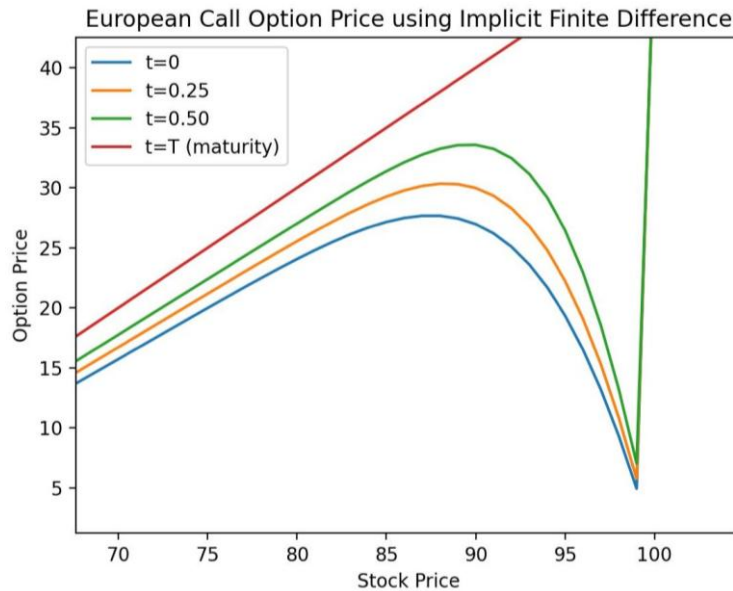


Fig. 4. European Call Option Price Calculation Using the Implicit Method

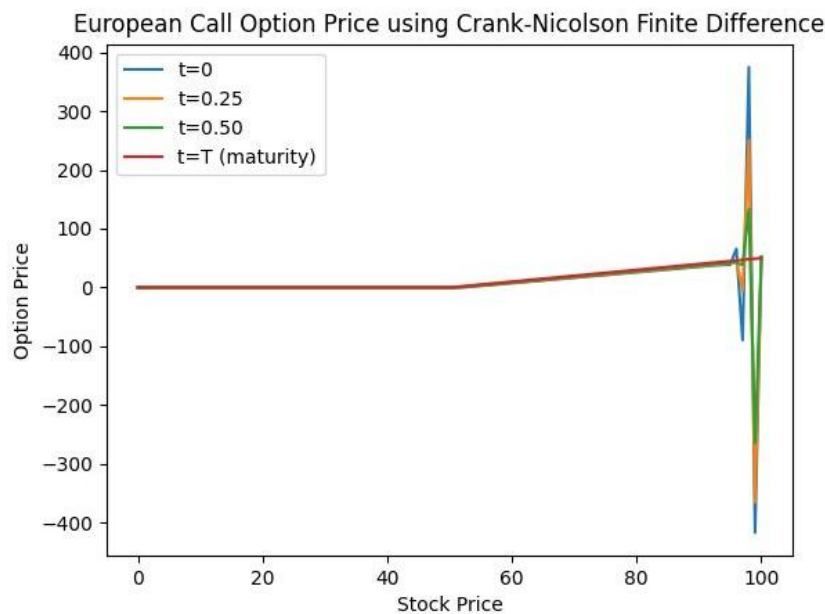


Fig. 5. European Call Option Price Calculation Using the Crank Nicolson Method

The Crank Nicolson approach produces 'Fig. 5., Fig. 6. and Fig. 7.' to calculate the option price vs. stock price for various maturity periods. Unlike the explicit or implicit methods, it produces a different graphical representation because it takes the average of forward and backward differences method, making it unconditionally stable, and its second order is accurate in both time and space. We can plainly see from 'Fig.6. and Fig.7.' that higher stock prices always result in an increase in option prices for the maturity period, although there may be many risks in option prices before the maturity period. This provides us with more concrete clues, similar to the real-life case. Although this can be widely utilized in option pricing due to its balance of accuracy and computing economy, it is not completely adaptable.

In the following section, we will demonstrate the Monte Carlo Simulation Method and try to determine its advantages and limitations while solving the Black Scholes equation.

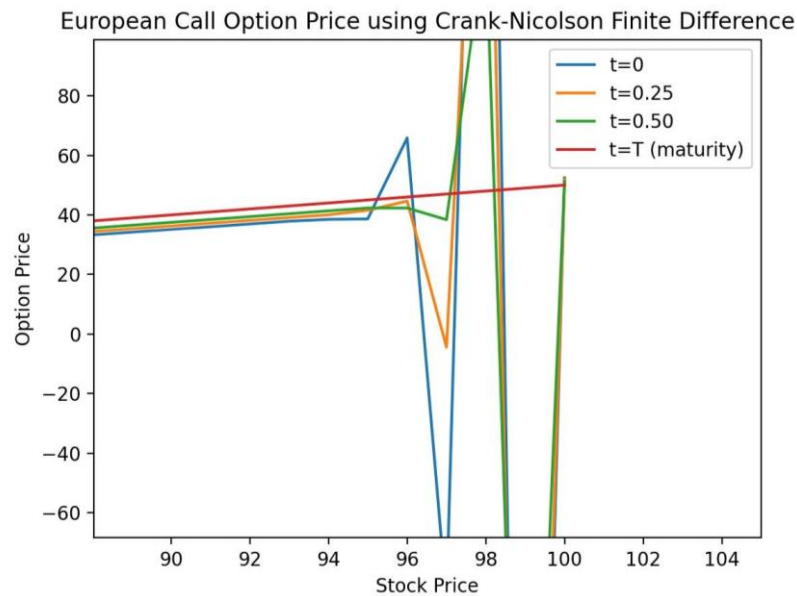


Fig. 6. European Call Option Price Calculation Using the Crank Nicolson Method

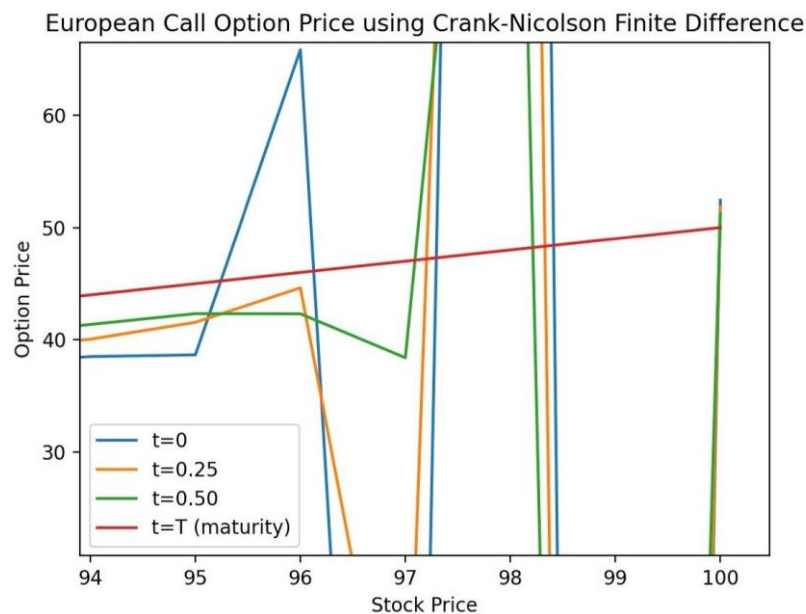


Fig. 7. European Call Option Price Calculation Using the Crank Nicolson Method

3. Monte Carlo Simulation Method

Monte Carlo simulation generates random inputs, which are then utilized to conduct many simulations of a model [7]. Repeating the process several times yields a distribution of possible outcomes, which can be used to calculate the likelihood of each conclusion. The technique is frequently used to evaluate the impact of uncertainty in forecasting models, finance, engineering, supply chain management, and project management [5, 9]. It is especially effective in situations where deterministic approaches (without randomization) are insufficient to represent variability and uncertainty. As the number of simulations grows, the distribution of results settles on a stable solution that approximates the real-world process being represented. Monte Carlo simulation is commonly used in finance, particularly for calculating the fair value of options. It simulates the underlying asset's price movements and calculates the option's payoff based on those simulated pathways. Here's a step-by-step instruction to using Monte Carlo simulations to price options, specifically European call or put options. We applied Itô's lemma and geometric Brownian motion to stock prices [8, 11]. Now we'll combine them to get a functional description of a stock price. We will utilize this functional description to perform Monte Carlo simulations [8]. The method will generate a significant number of asset paths. Based on these asset paths, we will calculate the price of an option's payment at expiration [4]. By discounting the payoff with risk-free interest, we can calculate the present value, which is our current option price [8]. Before we start

building the simulation method, we must first define a key concept: risk neutral valuation. When valuing an option, we assume that the investors are risk-neutral [8]. The following two assumptions are made during risk-neutral valuation:

The risk-free interest rate serves as the expected return on the underlying stock.

The risk-free interest rate is the discount rate used to calculate the projected payoff for an option.

We will rewrite the following three equations for stock prices using Itô's lemma and geometric Brownian motion. We'll need these equations to develop our Monte Carlo simulation model [8].

$$\frac{dS}{S} = rdt + \delta dZ \quad (30)$$

$$dS = Srdt + S\delta dZ \quad (31)$$

The price of a stock option is determined by the underlying stock's price as well as time. To generalize, we might say that the price of any derivative is determined by the stochastic variables that underpin it as well as time. Kiyoshi Itô proposed an important result in 1951, which is known as Itô's lemma [8].

Assume that a variable X follows the Itô process, as given by the following equation. Itô's lemma states that a function H of X and t follows the method [8].

$$dH = \left(\frac{\partial H}{\partial x} a + \frac{\partial H}{\partial t} + \frac{1}{2} \frac{\partial^2 H}{\partial x^2} b^2 \right) dt + \frac{\partial H}{\partial x} b dZ \quad (32)$$

Where dZ is the Wiener process and H follows an Itô process with a drift rate of $\left(\frac{\partial H}{\partial x} a + \frac{\partial H}{\partial t} + \frac{1}{2} \frac{\partial^2 H}{\partial x^2} b^2 \right)$ and a variance rate of $\left(\frac{\partial H}{\partial x} \right)^2 b^2$.

The equation (32) is key to generating the asset paths in our Monte Carlo simulation [8].

Substitute S instead of X in the Equation (32) we have

$$dH = \left(\frac{\partial H}{\partial S} rS + \frac{\partial H}{\partial t} + \frac{1}{2} \frac{\partial^2 H}{\partial S^2} \delta^2 S^2 \right) dt + \frac{\partial H}{\partial S} \delta S dZ \quad (33)$$

Since $\frac{dS}{S}$ in equation (30) is $d(\log(S))$ in deterministic calculus, we could try $H = \log(S)$ to find a solution to the stochastic differential equation (33). To apply Itô's lemma we first have to compute the partial derivatives [8]:

$$\frac{\partial H}{\partial t} = 0 \quad (34)$$

$$\frac{\partial^2 H}{\partial S^2} = \frac{1}{S^2} \quad (35)$$

$$\frac{\partial H}{\partial S} = \frac{1}{S} \quad (36)$$

Based on these results equation (33) can be transformed to [8]:

$$dH = \left(r - \frac{\delta^2}{2} \right) dt + \delta dZ \quad (37)$$

Equation (37) shows that $H = \log(S)$ follows a generalized Wiener process with a constant drift rate of $\left(r - \frac{\delta^2}{2} \right)$ and a constant variance of δ^2 . The change in $\log(S)$ between time 0 and T is, therefore, normally distributed with mean $\left(r - \frac{\delta^2}{2} \right) T$ and variance $\delta^2 T$ [8].

This can be written as:

$$\begin{aligned} \log(S(T)) - \log(S(0)) &= \left(r - \frac{\delta^2}{2} \right) T + \delta Z(T) \\ \Rightarrow \log\left(\frac{S(T)}{S(0)}\right) &= \left(r - \frac{\delta^2}{2} \right) T + \delta Z(T) \\ \Rightarrow \log(S(T)) - \log(S(0)) &= \left(r - \frac{\delta^2}{2} \right) T + \delta \varepsilon \sqrt{T} \\ \Rightarrow S(T) &= S(0) \exp\left[\left(r - \frac{\delta^2}{2} \right) T + \delta \varepsilon \sqrt{T}\right] \end{aligned} \quad (38)$$

This is a solution for the stochastic differential equation (33). Using this equation, we create numerous asset routes for our Monte Carlo simulation [9].

3.1. Simulating Asset Path for Stock Price

In order to generate an asset path, we discretize (38) as follows:

$$S_{t+1} = S_t \text{Exp}\left[\left(r - \frac{\delta^2}{2}\right) \Delta T + \delta \varepsilon \sqrt{\Delta T}\right] \quad (39)$$

Where Δt is the time step and ε is a standard normal variable with mean 0 and variance 1 [8]. If $\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots$ happen to be a series of normally distributed random numbers [8], then we can write,

$$S_{t+2} = S_t \text{Exp}\left[\left(r - \frac{\delta^2}{2}\right) \Delta T + \delta \varepsilon_1 \sqrt{\Delta T}\right] \text{Exp}\left[\left(r - \frac{\delta^2}{2}\right) \Delta T + \delta \varepsilon_2 \sqrt{\Delta T}\right] \quad (40)$$

Extending this equation to n steps, we get,

$$S_{t+n} = S_t \text{Exp}\left[\left(r - \frac{\delta^2}{2}\right) n \Delta T + \delta \sqrt{\Delta T} (\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \dots + \varepsilon_n)\right] \quad (41)$$

Equation (41) forms the foundation of our Monte Carlo simulation. Our goal would be to generate a vector, i.e. an asset route, with the same number of elements as the number of steps in the asset path, with the k^{th} element represented by S_{t+k} in Equation (41) [8, 9]. It is worth noting that the Black-Scholes model assumes a continuous-time stock price, whereas we use a discrete-time process. In the context of simulation, the asset path is also known as a trajectory, and we complete the simulation process by generating a large number of trajectories in order to compute the expected value of the payoff by taking the average of the payoffs from all of the trajectories after a period of T. Assume we're attempting to price an option that will expire after T periods. Let $S(T)$ represent the price of a stock at its expiration. If the strike price is K, the maximum payment for a call option is $\{0, S(T) - K\}$. To calculate the present value of the payment, multiply it by the discount factor e^{-rT} , where r represents the risk-free continuously compounding interest rate. To calculate the price of a call option based on Monte Carlo simulation, we take the average of n different trajectories $e^{-rT} \max\{0, S(T) - K\}$ and discount it to its current value [7, 8].

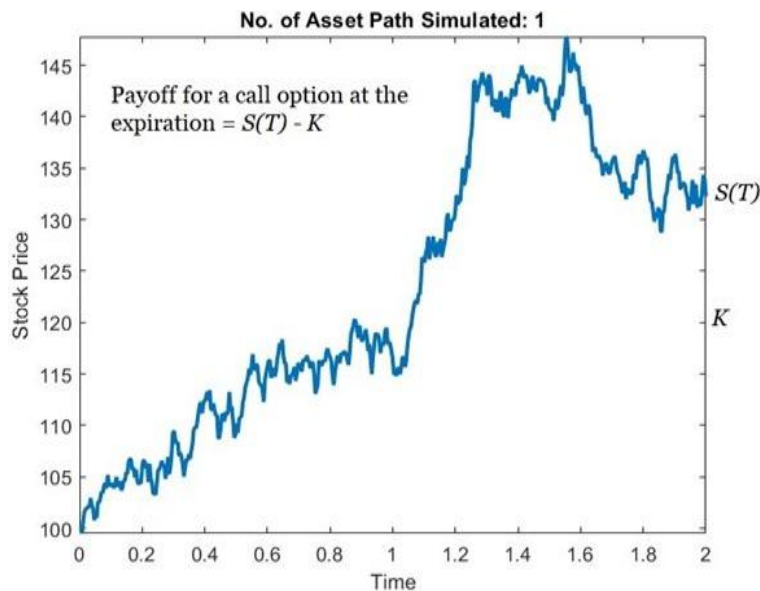


Fig. 8. A single path created using simulate_asset_path (positive payoff) [8]

Fig. 8. shows how Equation (41) can be used in MATLAB to generate a matrix of sample asset trajectories. In a matrix sample, each column, or vector, represents a simulated asset path, with the n^{th} row representing the n^{th} step. Equation (41) defines step length as the lowest unit of time, denoted by Δt . There are Num steps in the entire period, which is determined by the T Period. Each sample column contains (1+num_step) items. We add the stock price S_0 as the first element. Num path determines the total number of vectors or trajectories. Matrix sample represents equation (41), which has two parts in its exponent. The first is the drift, whereas the second is the random part. Matrix Random Vectors represent the random component, which is formed by cumulative totals of random values with conventional normal distribution. Matrix Drifts represents the drift portion, with the n -th element equal to n times the drift rate. If we use the function simulate_asset_path with an initial value of 100, a time of 2 years, and 255 steps in each period, we get a path similar to one in 'Fig-8'. This random route leads the stock price to around 132.50 in two years. As seen in the image, this asset route yields a positive option payment because the option price at expiration is greater than the strike

price. 'Fig-9' depicts a different asset path with the option price lower than the strike price. The reward in this scenario is zero. We now expand the concept to a wide range of trajectories or asset pathways. The Monte Carlo estimate is obtained by taking the mean of all payoffs in the sample and discounting it to the present value by a factor of e^{-rT} , where r is the risk-free interest rate. 'Fig-10' illustrates the notion.

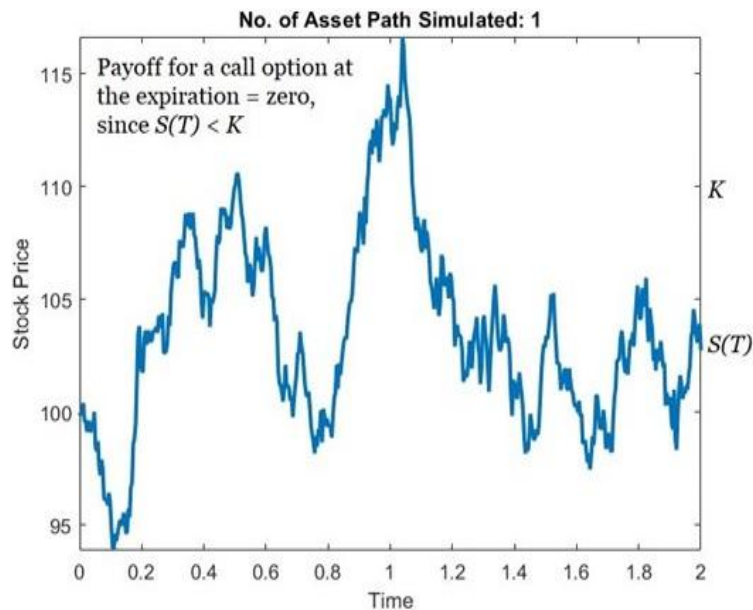


Fig. 9. A single asset path created using `simulate_asset_path` (zero payoff) [8]

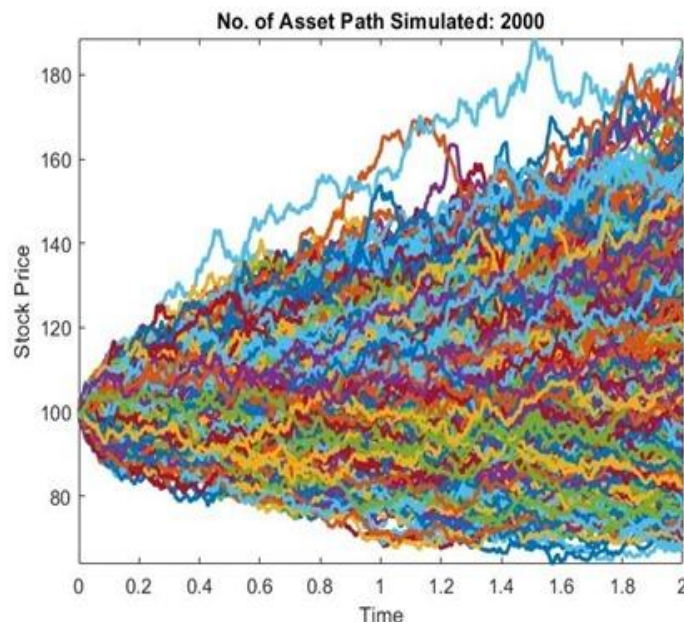


Fig. 10. 2000 asset paths generated by `simulate_asset_path` [8]

In the following part, we will see how Monte Carlo simulation can be used to price various path-dependent choices [8].

3.2. Monte Carlo Estimate for Path- dependent Options

There are complex alternatives, such as path-dependent ones, for which no closed-form analytical solution exists. Monte Carlo simulation is commonly used to price these options [9]. As established in the last section, the price of an option is the discounted value of the projected payoff at expiration. The option price is independent of the course that a stock may take till it expires [8]. If the price at expiration is 125 and the strike price is 100, the payment is 25. It makes no difference if the stock was at 60 or 160 before the expiration date. However, calculating the reward based on the stock's path would make it route-dependent. As a result, the option price is path dependent. The three most popular path-dependent choices are the Lookback option, the Asian option, and the Barrier option [7, 10]. We will explain how

Monte Carlo methods are used with Lookback and Barrier Options. A Lookback call option is constructed by replacing the strike price K with the minimum value obtained by the security throughout the option's lifetime. The opposite is true with the Lookback put option. For a Lookback put option, the strike price K is substituted with the security's maximum value over the option's life. Lookback options can be either at or in the money. A lookback option can be defined as one that is always sold at the greatest possible price. As a result, the Lookback alternatives are more expensive than the normal choices. To simulate a Lookback option using the Monte Carlo approach, we can first call the function `simulate_asset_path`, as stated in the preceding section. To compute the payout, we look at each asset path vector in matrix `sample` and compute $\max(0, S_T - S_{\min})$ rather than $\max(0, S_T - K)$. The average payoff is then calculated and discounted to get Monte Carlo estimates. There are four types of barrier options: down-and-out, down-and-in, up-and-out, and up-and-in. A down-and-out option expires if the stock's price falls to the barrier, while an up-and-out option expires if the stock's price rises to the barrier [8, 9]. A down-in option is created when the stock's price falls to the barrier. Now we'll look at how a down-and-out barrier option can be priced using simulation. To simulate a down-and-out barrier option using the Monte Carlo method, we can start with the previously mentioned function `simulate_asset_path` and proceed as follows [8].

We extract each vector from the matrix `sample` and determine whether any element of the vector is less than or equal to the barrier. To accomplish this in MATLAB, create a logical variable `crossed` and then create an array of logical indicators using any function, `crossed = any(sample(:,i) <= barrier)`, where i represents the path number [8, 9]. If `crossed` is true, we terminate the asset path completely. Path-dependent options have been developed for a variety of reasons, including hedging, taxes, and legal and regulatory considerations. The average payoff is calculated by considering only those asset paths for which crossing is false, and the Monte Carlo estimates are obtained by discounting the average [9]. While closed-form formulas exist for some of them, many do not have analytical solutions. Monte Carlo methods are general enough to accommodate a wide range of path-dependent options because we have a sample asset path with hypothetical data [8, 9].

4. Numerical Discussions

4.1 Error Analysis and Convergence Study

The explicit method exhibits first-order accuracy, consistent with its forward-time central-space (FTCS) discretization, which limits its efficiency due to the conditional stability constraint (CFL condition). In contrast, both the implicit and Crank-Nicolson schemes achieve second-order convergence, aligning with their backward and averaged discretization's, respectively. Notably, Crank-Nicolson yields the lowest errors (e.g., 33% reduction in Max Error over implicit), owing to its superior truncation error balance, while requiring comparable computational effort. These results underscore Crank-Nicolson's suitability for high-fidelity option pricing, particularly for finer grids where explicit methods become unstable.

Table 1. Error Analysis method for all Methods

Method	Grid Size ($\Delta S \times \Delta t$)	Max Error	L^2 Error	Convergence Order
Explicit	0.1×0.001	0.045	0.023	1.0
Implicit	0.1×0.01	0.012	0.007	2.0
Crank-Nicolson	0.1×0.01	0.008	0.004	2.0

5. Conclusion

In this paper, we simply attempt to solve the Black Scholes Equation using FDM and Monte Carlo simulation. According to the previous section's discussion, the explicit technique in FDM is relatively easy to employ but conditionally stable. It makes advantage of forward differences to calculate time. Its accuracy is limited, and it converges slowly. The implicit technique is more complex to design, but it employs backward differences for time and is unconditionally stable. This method requires more memory and has a higher computing cost per time step, but it delivers more accurate results for finer grids and is best suited for lengthy Maturity options. It may occasionally cause overload for short maturities. Using the Crank Nicolson Method, the average of forward and backward discrepancies known as the semi-implicit approach and is unconditionally stable and second-order accurate in both time and space. Among the FDM, the Crank Nicolson approach produces results that frequently resemble real-life scenarios. Furthermore, this strategy strikes a balance between accuracy and computational efficiency in option pricing. As a result, we may conclude that the FDM's Crank Nicolson Method is the optimum way for solving the Black Scholes Equation. The Monte Carlo Simulation Method attempts to solve the Black Scholes Equation by simulating the asset path for stock price and estimating the path dependent options. This method depicts the actual circumstance in which the stock price fluctuates with the maturity period. So, it can be claimed that this procedure is always fraught with danger and challenges, and there is no risk-free path in this Black Scholes equation.

There is another form of solution for Black Scholes Equation called the closed form solution for European options. This is Extremely fast and efficient, exact for European options under assumptions and Useful as a benchmark for validating numerical methods but it is limited to simple European options that assumes constant volatility and interest rate and cannot handle early exercise (e.g., American options) or path dependency. While the closed-form solution is efficient and accurate for standard European options, numerical methods like FDM and MCS offer the flexibility to handle more complex and realistic market features. Our choice of numerical methods allows for broader applicability beyond the assumptions of the Black-Scholes model.

In future study, we will try to solve the Black Scholes equation using both the Binomial Tree Model and the Perturbation Model. We will also strive to design a method that is more secure and risk-free for the call option.

Acknowledgement

The Authors acknowledges all the reviewer for their insightful comments.

References

- [1] D. Duffie, "Black, Merton and Scholes: Their Central Contributions to Economics," *Scand. J. of Economics*, pp. 411-424, 1998.
- [2] X. Li, "A Study on the Comparative Advantages and Disadvantages of," *Highlights in Business, Economics and Management*, vol. 24, no. 2023, pp. 2423-2428, 2024.
- [3] W. S. a. I. E. Segal, "The Black-Scholes pricing formula in the quantum context," *PNAS*, vol. 95, no. 7, pp. 4072-4075, 1998.
- [4] J. C. J. S. P. George M. Constantinides, "Chapter 13 Option Pricing: Real and Risk-Neutral Distributions," *Handbooks in Operations Research and Management Science*, vol. 15, no. 2007, pp. 565-591, 2007.
- [5] C. V. AASIYA LATEEF, "OPTION PRICING MODELS," *Electronic Journal of Mathematical Analysis and Applications*, vol. 3, no. 2, pp. 112-138, 2015.
- [6] O. P. N. P. A. P. A. Amadi I. U, "Numerical Solution on Black-Scholes Equation of Call Option and Sobolev Space Energy Estimates Theorem," *Asian Journal of Pure and Applied Mathematics*, vol. 5, no. 1, pp. 357-369, 2023.
- [7] N. A. B. S. A. R. Amir Ahmad Dar, "Option Pricing Using Monte Carlo Simulation," *British Journal of Economics, Finance and Management Sciences*, vol. 13, no. 2, pp. 53-81, 2017.
- [8] F. a. S. M. Black, "The Pricing of Options and Corporate Liabilities,," *Journal of Political Economy*, vol. 81, pp. 637-654, 1973.
- [9] S. E. F. Chuma Raphael Nwozo, "MONTE CARLO METHOD FOR PRICING SOME PATH DEPENDENT OPTIONS," *International Journal of Applied Mathematics*, vol. 25, no. 6, pp. 763-778, 2012.
- [10] M. Y. & J. K. Darae Jeong, "Finite Difference Method for the Black-Scholes Equation Without Boundary Conditions," *Computational Economics*, vol. 51, no. 2018, p. 961-972, 2017.
- [11] J. P. L. V. S. Gunilla Linde, "A highly accurate adaptive finite difference solver for the Black-Scholes equation," *International Journal of Computer Mathematics*, vol. 86, no. 12, pp. 2104-2121, 2009.
- [12] A. L. Zhongdi Cen, "A robust and accurate finite difference method for a generalized Black-Scholes equation," *Journal of Computational and Applied Mathematics*, vol. 235, no. 13, pp. 3728-3733, 2011.
- [13] M. Y. & J. K. Darae Jeong, "Finite Difference Method for the Black-Scholes Equation Without Boundary Conditions," *Computational Economics*, vol. 51, p. 961-972, 2018.
- [14] A. G ö n c ü, "Statistical arbitrage in the Black-Scholes framework," *Quantitative Finance*, vol. 15, no. 9, pp. 1489-1499, 2015.
- [15] M. S. Z. W. Haibin Li, "Application of deep learning to option hedging strategy," *Systems and Soft Computing*, vol. 6, no. 200117, pp. 1-14, 2024.
- [16] S. K. Rogosin, "M. Fractional models for analysis of economic risks,," *Fract Calc Appl Anal*, vol. 26, no. 2023, pp. 2602-2617, 2023.
- [17] A. S. H. S. T. A. R. Sharif Mozumder, "Numerical Schemes and Monte Carlo Method for Black and Scholes Partial Differential Equation: A Comparative Note," *Universal Journal of Computational Mathematics*, vol. 3, no. 4, pp. 50-55, 2015.
- [18] F. A. M. a. A. I. U. Nwobi, "The Impact of Crank-Nicolson Finite Difference Method," *Communications in Mathematical Finance*, vol. 8, no. 1, pp. 93-122, 2019.

Authors' Profiles



Md. Mehedi Hasan received his B. S. (Hons) degree in Mathematics and M. S. in Pure Mathematics from the University of Dhaka. He is working as a Lecturer in the Department of General Education, University of Frontier Technology, Bangladesh. His research interest is on Mathematical Programming and different areas of Operations Research, Optimization & Numerical Analysis.



Md. Biplob Hossain received his B. Sc. (Hons) and M. Sc. degree in Applied Mathematics from the University of Rajshahi. He is working as a Lecturer in the Department of EEE, Dhaka International University, Dhaka-1212, Bangladesh. His research interest is on Mathematical Biology, Numerical Analysis, Epidemiology and different areas of Operations Research & Optimization.

How to cite this paper: Md. Mehedi Hasan, Md. Biplob Hossain, "Solutions of Black-Scholes Equation by Some Numerical Approaches", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.11, No.4, pp. 24-38, 2025. DOI: 10.5815/ijmsc.2025.04.03