SURGE 2025
End-Term Report

Ayush Raj
SURGE Application ID: *2530446*

Bachelor of Science, Mathematics and Scientific Computing
Indian Institute of Technology Kanpur

ayushr23@iitk.ac.in

Under the Guidance of

Prof. B.V. Rathish Kumar
Professor, Department of Mathematics and Statistics
Indian Institute of Technology Kanpur

bvrk@iitk.ac.in

July 12, 2025

Preface

This report is the outcome of the SURGE 2025 research project undertaken at IIT Kanpur. It focuses on the theoretical and computational aspects of solving partial differential equations in financial modeling, especially the Black-Scholes-Merton equation and its extensions. The motivation behind this project is to study and compare various advanced techniques—ranging from traditional numerical methods to modern machine learning frameworks—for modeling complex financial systems.

The project begins by establishing a strong foundation through the derivation and explanation of the Black-Scholes-Merton model and its role in option pricing. From there, the report explores the use of fractional calculus to introduce memory effects into the model. Subsequent chapters introduce Finite Difference Methods, High-Order Compact schemes, and finally, Physics-Informed Neural Networks, all used to solve deterministic and fractional PDEs.

This work is intended not only as a study of these individual methods but also as a comparative reference for researchers and practitioners looking to understand their relative strengths, applicability, and limitations in financial mathematics.

I am deeply grateful to my faculty mentor, Prof. BV Rathish Kumar, peers, and the SURGE team at IIT Kanpur for their support and guidance throughout this enriching research journey.

Abstract

In this project, I've been working on solving a more realistic version of the Black-Scholes-Merton (BSM) model by adding something called a time-fractional Caputo derivative. This helps account for the memory effects we often observe in real financial markets, something the classical model doesn't really capture well. The modified version, known as the time-fractional BSM (TFBS) model, gives us a better framework to price options when the market's behavior depends not just on the present but also on its past.

To tackle this equation, I've explored three different numerical approaches. The first is the traditional Finite Difference Method (FDM), then a more accurate version called High-Order Compact (HOC) FDM, and finally a modern deep learning-based technique called Physics-Informed Neural Networks (PINNs). Each method has its own strengths, and I've been comparing them based on how accurate, efficient, and flexible they are—especially when dealing with fractional derivatives.

We've also worked on simplifying the TFBS model through variable transformations and applied appropriate boundary conditions tailored to European call options. One of the core areas of our study is the PINN framework, where we train neural networks that naturally incorporate the physics of the equation using automatic differentiation to handle fractional derivatives.

Currently, we are fine-tuning the PINN architecture and training process to improve stability and performance, especially in higher dimensions or complex domains. Through this comparative study, we aim to figure out which approach works best for real-world option pricing when traditional assumptions no longer hold.

Keywords. Time-Fractional Black-Scholes-Merton Model, Caputo Derivative, Physics-Informed Neural Networks (PINNs), Finite Difference Methods, Option Pricing, Fractional PDEs, Computational Finance, Deep Learning

Contents

Chapter 1

The Black-Scholes-Merton Model: Theory and Applications

1.1 Introduction

The Black-Scholes-Merton (BSM) model is one of the most foundational frameworks in mathematical finance. Originally developed by Fischer Black, Myron Scholes, and later extended by Robert Merton, it revolutionized the pricing of options by introducing a partial differential equation governing the price of a derivative security. The model assumes that financial markets are frictionless and complete, and that price movements of the underlying assets can be described using a stochastic process known as geometric Brownian motion.

1.2 Geometric Brownian Motion and Asset Dynamics

Let $S(t)$ denote the price of an asset at time t. The assumption is that $S(t)$ evolves according to the following stochastic differential equation (SDE):

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t)$$

where μ is the drift (expected return), σ is the volatility of the asset, and $W(t)$ is a standard Brownian motion. This implies that the logarithmic returns of the asset are normally distributed.

The SDE has the analytical solution:

$$S(t) = S(0)\exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right)$$

which implies that $S(t)$ follows a log-normal distribution.

1.3 The Self-Financing Hedging Portfolio

To derive the BSM PDE, consider a portfolio consisting of Δ shares of the underlying asset and a short position in one option with price $V(S,t)$. The value of the portfolio is:

$$\Pi = \Delta S - V(S,t)$$

Using Itô's Lemma, the differential of the option price is:

$$dV = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}(dS)^2$$

Substitute the dynamics of dS and use the identity $(dW)^2 = dt$ to get:

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

To eliminate risk, we create a self-financing, riskless portfolio by choosing $\Delta = \frac{\partial V}{\partial S}$. The portfolio's value becomes:

$$\Pi = \frac{\partial V}{\partial S}S - V$$

Then the differential of the portfolio:

$$d\Pi = \frac{\partial V}{\partial S}dS - dV$$

Substitute all expressions and simplify:

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

Since this portfolio is riskless, it must earn the risk-free rate r:

$$d\Pi = r\Pi dt \Rightarrow -\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}\right) = r\left(\frac{\partial V}{\partial S}S - V\right)$$

Rearranging yields the Black-Scholes PDE:

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

1.4 Boundary and Initial Conditions for European Options

To solve the PDE, we must specify appropriate boundary and terminal conditions:

- **Terminal condition for European call:** $V(S,T) = \max(S-K, 0)$
- **Boundary conditions:**
  - $V(0,t) = 0$ (worthless if underlying asset is zero)
  - $V(S,t) \to S - Ke^{-r(T-t)}$ as $S \to \infty$ (mimics intrinsic value)

1.5 Closed-form Analytical Solution

Using a change of variables and solving the PDE using the Feynman-Kac theorem or similarity transforms, we get the closed-form solution:

$$V(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

where:

$$\begin{aligned} d_1 &= \frac{\ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= d_1 - \sigma\sqrt{T-t} \end{aligned}$$

$N(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

1.6 Model Assumptions and Implications

The derivation assumes:

- Frictionless markets with no transaction costs or taxes
- Short-selling is permitted
- Continuous rebalancing of portfolios is possible
- The risk-free rate and volatility are constant
- The underlying asset does not pay dividends
- No arbitrage opportunities exist

These assumptions facilitate mathematical tractability but may deviate from market realities.

1.7 Merton Extensions and Generalizations

1.7.1 Continuous Dividend Yield

For assets paying a continuous dividend yield q, the PDE becomes:

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

This affects the boundary condition and modifies the pricing formula:

$$V(S,t) = Se^{-q(T-t)}N(d_1) - Ke^{-r(T-t)}N(d_2)$$

with d_1 and d_2 adjusted accordingly.

1.7.2 Jump Diffusion Models

To incorporate sudden, discontinuous price changes, Merton proposed the jump diffusion model:

$$dS = \mu S dt + \sigma S dW + J S dq$$

where q is a Poisson process and J is the jump size. These models lead to integro-differential equations.

1.8 Model Limitations and Enhancements

- **Volatility Smile:** Implied volatility varies with strike/maturity, contradicting the constant σ assumption.
- **Stochastic Volatility Models:** Heston model introduces volatility as a stochastic process.
- **Transaction Costs:** Leland, Amster, and others modified BSM to account for discrete trading and transaction costs.
- **Memory Effects:** Fractional Black-Scholes models use Caputo derivatives to model long memory in volatility or returns.

1.9 Applications and Importance in Finance

- **Option Pricing:** Used for European vanilla options, currency derivatives, and convertible bonds.
- **Risk Management:** Basis for computing Greeks (Delta, Gamma, Vega) for hedging.
- **Benchmark Model:** All other models (e.g., stochastic volatility, jump diffusion) are viewed as generalizations of BSM.

1.10 Numerical Approaches and Modern Variants

Despite the availability of closed-form solutions for vanilla options, complex structures (barriers, early exercise, American options) require numerical techniques:

- **Finite Difference Methods (FDM):** Explicit, implicit, and Crank–Nicolson schemes
- **High-Order Compact FDM (HOCFDM):** Improved accuracy with small stencil
- **Monte Carlo Simulation:** Path-wise pricing under stochastic volatility
- **Machine Learning:** Neural networks and PINNs for high-dimensional PDEs

Chapter 2

Fractional Differential Equations and Caputo Derivative

2.1 Introduction to Fractional Calculus

Fractional calculus is a generalization of classical calculus that allows differentiation and integration to arbitrary (non-integer) orders. The concept dates back to discussions between L'Hôpital and Leibniz in the late 17th century. While standard calculus defines derivatives of integer order (e.g., first, second derivatives), fractional calculus studies operators such as D^α for $\alpha \in \mathbb{R}^+$.

2.2 Motivation and Applications

Fractional derivatives have gained traction due to their ability to model memory and hereditary properties inherent in many physical and biological systems. Applications include:

- Viscoelastic materials
- Anomalous diffusion processes
- Electrical circuits
- Image processing
- Finance: option pricing under market memory effects

2.3 Definitions of Fractional Derivatives

Several definitions of fractional derivatives exist. The most commonly used include:

2.3.1 Riemann–Liouville Derivative

Defined as:

$$D_a^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau$$

where $n = \lceil \alpha \rceil$ and $\Gamma(\cdot)$ is the gamma function.

2.3.2 Caputo Derivative

Defined as:

$$ {}^{C}D_a^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau $$

Unlike the Riemann–Liouville form, Caputo's derivative starts with classical derivatives $f^{(n)}(\tau)$, making it more suitable for initial value problems with physical meaning.

2.4 Comparison of Caputo and Riemann–Liouville

- **Initial Conditions:** Caputo allows for initial conditions in terms of standard derivatives.
- **Physical Interpretability:** Caputo is more interpretable for physical processes.
- **Laplace Transform:** The Laplace transform of the Caputo derivative:

$$ \mathcal{L}\{{}^{C}D^{\alpha} f(t)\} = s^{\alpha} F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} f^{(k)}(0) $$

2.5 Properties of the Caputo Derivative

- Linearity: $D^{\alpha}(af + bg) = aD^{\alpha} f + bD^{\alpha} g$
- Caputo derivative of a constant is zero
- Non-local operator: depends on entire history from a to t

2.6 Fractional Differential Equations (FDEs)

An FDE with a Caputo derivative is written as:

$$ {}^{C}D^{\alpha} y(t) + \lambda y(t) = f(t), \quad y(0) = y_0 $$

where $\alpha \in (0, 1)$, λ is a parameter, and $f(t)$ is a forcing function. These equations are generalizations of classical ODEs.

2.7 Solution Methods

2.7.1 Laplace Transform Method

Apply Laplace transform and use known identities of the Caputo derivative. Inversion of the transformed equation yields the solution in terms of the Mittag–Leffler function:

$$ y(t) = y_0 E_{\alpha}(-\lambda t^{\alpha}) $$

where

$$ E_{\alpha}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)} $$

2.7.2 Numerical Approaches

- L1 Scheme: Finite difference approximation of Caputo derivative
- Predictor–Corrector PECE method
- Grünwald–Letnikov approximations

2.8 Caputo in Financial Models

The Caputo derivative allows modeling of market behaviors with memory. A time-fractional Black-Scholes model includes:

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

Such models better capture volatility clustering and non-Markovian features.

Chapter 3

Finite Difference Method: Theory, Schemes, and Applications

3.1 Introduction

The Finite Difference Method (FDM) is a widely used numerical technique to approximate solutions of differential equations by replacing derivatives with finite difference quotients. It is particularly effective for problems involving partial differential equations (PDEs), including those in heat transfer, fluid flow, structural mechanics, and quantitative finance.

3.2 Discretization of Derivatives

To implement FDM, we discretize the continuous domain into grid points. Suppose $u(x)$ is a function defined on $[a, b]$. The domain is divided into N intervals of width $\Delta x = (b - a)/N$, with grid points $x_i = a + i\Delta x$.

3.2.1 Forward Difference Approximation

$$u'(x_i) \approx \frac{u(x_{i+1}) - u(x_i)}{\Delta x} \quad \text{(First order accurate)}$$

3.2.2 Backward Difference Approximation

$$u'(x_i) \approx \frac{u(x_i) - u(x_{i-1})}{\Delta x} \quad \text{(First order accurate)}$$

3.2.3 Central Difference Approximation

$$u'(x_i) \approx \frac{u(x_{i+1}) - u(x_{i-1})}{2\Delta x} \quad \text{(Second order accurate)}$$

3.2.4 Second Derivative

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{\Delta x^2} \quad \text{(Second order accurate)}$$

3.3 FDM for Time-Dependent Problems

Consider the 1D heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

We discretize both time and space:

- $x_i = i\Delta x$, for $i = 0, 1, ..., N$
- $t^n = n\Delta t$, for $n = 0, 1, ..., T$

Let u_i^n be the approximation to $u(x_i, t^n)$.

3.3.1 Explicit Scheme (Forward Euler)

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

Simple but conditionally stable.

3.3.2 Implicit Scheme (Backward Euler)

$$u_i^{n+1} - \frac{\alpha \Delta t}{\Delta x^2}(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}) = u_i^n$$

Unconditionally stable but requires solving linear systems.

3.3.3 Crank–Nicolson Scheme

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{2\Delta x^2}\left[(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}) + (u_{i+1}^n - 2u_i^n + u_{i-1}^n)\right]$$

Second-order accurate in both time and space, and unconditionally stable.

3.4 Stability and Consistency

3.4.1 Von Neumann Stability Analysis

Used to assess stability for linear PDEs. Assume $u_i^n = \xi^n e^{ikx_i}$ and substitute into the scheme to analyze $|\xi|$.

3.4.2 Lax Equivalence Theorem

If a consistent finite difference scheme is stable, then it is convergent.

3.5 Applications in Finance: Black–Scholes Equation

The Black–Scholes PDE is:

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

Discretizing using FDM (e.g., Crank–Nicolson) allows efficient valuation of options.

3.5.1 Transformation to Heat Equation Form

Define:

$$x = \ln\left(\frac{S}{K}\right), \qquad \tau = T - t,$$
$$V(S,t) = Ke^{-r\tau}u(x,\tau)$$

The transformed PDE becomes:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} + \text{lower order terms}$$

3.6 Two-Dimensional Problems

For 2D Laplace equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Use 5-point stencil:

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$$

This leads to a sparse linear system.

3.7 Handling Boundary and Initial Conditions

- **Dirichlet:** Fixed values at the boundaries
- **Neumann:** Specify derivatives; use one-sided differences
- **Robin:** Linear combination of value and derivative

3.8 Advantages of FDM

- Easy to implement
- Efficient on structured grids
- Conceptually intuitive
- Works well for rectangular domains

3.9 Limitations

- Hard to handle irregular geometries
- Less suited for unstructured grids
- Requires careful treatment for stability and convergence

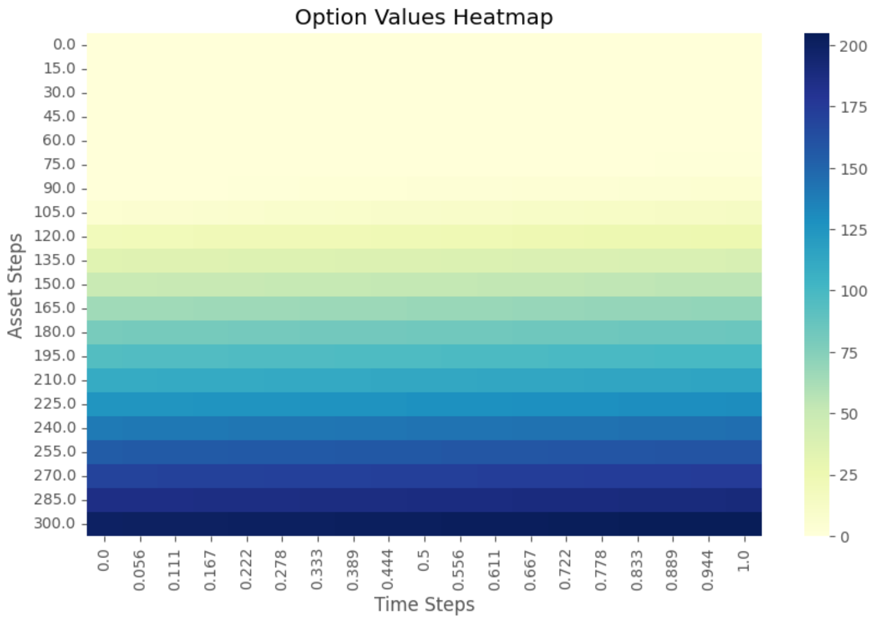

Figure 3.1: Heatmap of option values computed via the explicit finite-difference scheme. The horizontal axis shows equally spaced time steps from $t = 0$ (at maturity) to $t = T$, and the vertical axis shows discretized asset prices S. Notice how the value surface transitions smoothly from the payoff at maturity (bottom row) to the value at $t = 0$ (top row).

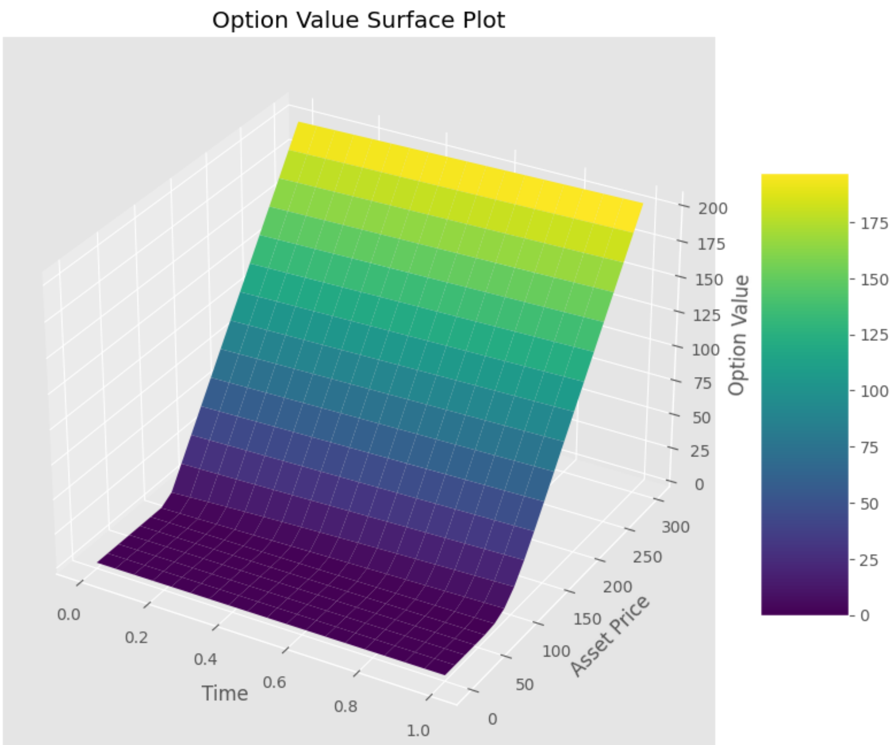

Figure 3.2: 3D surface plot of the same finite-difference solution. Along the time axis the option "flattens" toward zero prior to maturity, and along the asset axis it grows roughly linearly for high S.

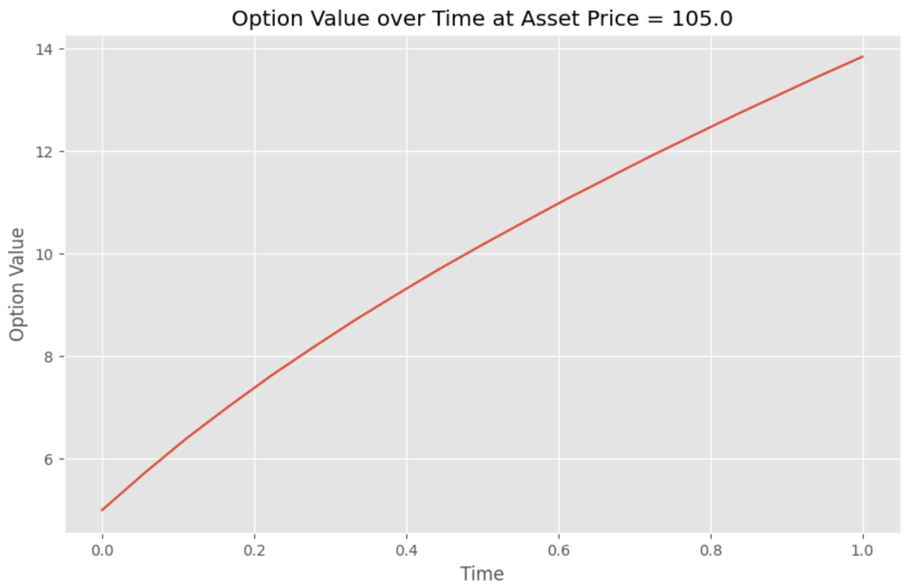

Figure 3.3: Time-evolution of the option value for a fixed asset price $S = 105$. The red curve shows how the option decays smoothly as time progresses towards maturity under the finite-difference approximation.

Chapter 4

High-Order Compact Finite Difference Method (HOCFDM)

4.1 Introduction and Motivation

The High-Order Compact Finite Difference Method (HOCFDM) enhances the classical Finite Difference Method (FDM) by achieving high-order accuracy using compact stencils. This is particularly useful in solving partial differential equations (PDEs) with high precision on coarse grids, preserving accuracy without significantly increasing computational complexity.

4.2 Basic Idea of Compact Schemes

Unlike traditional high-order schemes that use wider stencils, HOCFDM derives higher-order accuracy using information from fewer grid points. This results in:

- Reduced bandwidth of resulting matrices
- Higher accuracy per grid point
- Better resolution of steep gradients and wave phenomena

4.3 Fourth-Order Compact Scheme for Second Derivatives

Let $u''(x)$ be approximated using a compact stencil:

$$\alpha u''_{i-1} + u''_i + \alpha u''_{i+1} = \frac{a}{h^2}(u_{i+1} - 2u_i + u_{i-1})$$

Choosing $\alpha = 1/4$ and $a = 3/2$ yields a fourth-order approximation for u''_i.

4.3.1 Derivation Using Taylor Expansion

Expand $u_{i\pm1}$ around x_i:

$$u_{i\pm1} = u_i \pm hu'_i + \frac{h^2}{2}u''_i \pm \frac{h^3}{6}u'''_i + \frac{h^4}{24}u_i^{(4)} + \mathcal{O}(h^5)$$

Eliminating odd terms and solving leads to compact high-order formulas.

4.4 Compact Scheme for the Heat Equation

For the 1D heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Discretize space using the compact second derivative, and time using Crank–Nicolson:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \left[\frac{1}{2}(u_i'')^n + \frac{1}{2}(u_i'')^{n+1} \right]$$

Substitute u'' using the compact formula. Results in a tridiagonal system with higher accuracy.

4.5 Extension to Nonuniform Grids

HOC schemes can be extended to nonuniform grids using:

- Non-centered finite difference weights
- Weighted harmonic averages
- Coordinate transformations

However, derivation becomes more algebraically intensive.

4.6 Two-Dimensional HOC Scheme

For the 2D Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

Use the compact stencil in each direction:

$$\alpha u''_{i-1,j} + u''_{i,j} + \alpha u''_{i+1,j} = \text{RHS in x}$$
$$\alpha u''_{i,j-1} + u''_{i,j} + \alpha u''_{i,j+1} = \text{RHS in y}$$

This yields a block-tridiagonal system.

4.7 Applications in Financial PDEs

4.7.1 Black–Scholes Equation

The standard FDM may lack sufficient accuracy near discontinuities (e.g., option strike). HOCFDM provides better spatial resolution:

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

Discretize the second derivative term using the fourth-order compact formula, while using implicit or Crank–Nicolson in time.

4.7.2 Caputo Time-Fractional BSM

Recent works (e.g., Roul and Goura, 2021) combine HOC in space with L1 approximation in time for fractional models:

$$^{C}D_t^{\alpha} V + \mathcal{L}_S[V] = 0$$

- Space: HOC for $\frac{\partial^2 V}{\partial S^2}$ and $\frac{\partial V}{\partial S}$
- Time: L1 discretization of the Caputo derivative

Improves accuracy and efficiency in solving fractional option pricing problems.

4.8 Accuracy and Efficiency

- Fourth-order accuracy with same stencil width
- Sparse matrix systems (tridiagonal in 1D)
- Significantly better performance for long-time simulations and stiff PDEs

4.9 Challenges and Implementation

- Complicated derivation for nonuniform grids or nonlinear PDEs
- Requires careful boundary condition handling
- Algebraically more complex than FDM

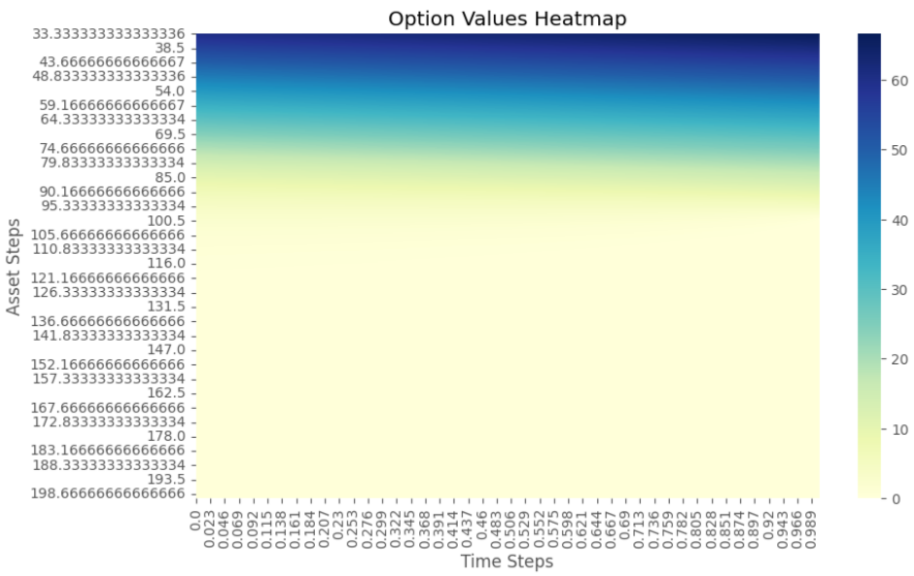

Figure 4.1: Heatmap of option values computed via the *fourth-order compact* finite-difference stencil. The compact scheme yields a smooth, high-resolution profile over time (horizontal axis) and asset price (vertical axis) while using only three-point stencils in space.

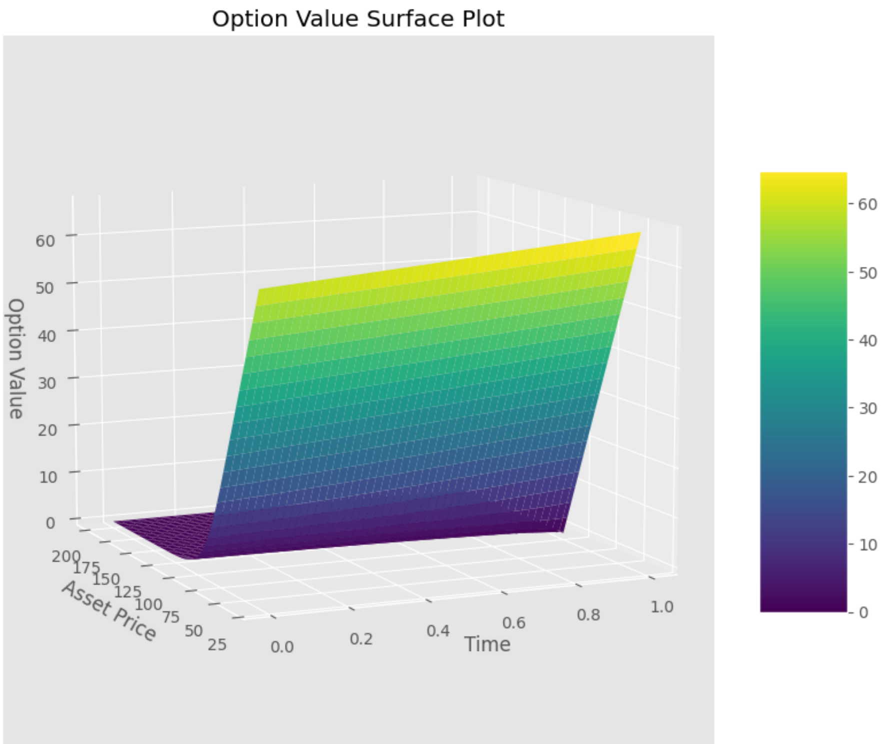

Figure 4.2: 3D surface plot of the HOCFDM solution. Notice the sharper capture of the strike kink and the smoother curvature compared with a standard second-order method, thanks to the compact fourth-order spatial discretization.

Chapter 5

Physics-Informed Neural Networks (PINNs)

5.1 Introduction

Physics-Informed Neural Networks (PINNs) represent a groundbreaking fusion between scientific computing and machine learning. In contrast to traditional numerical methods that rely on discretization, PINNs incorporate the governing physical laws of a system directly into the architecture and training of neural networks. This makes them well-suited for solving both forward and inverse problems in partial differential equations (PDEs), especially where data is sparse or domains are high-dimensional.

5.2 Mathematical Foundations of Neural Networks

A neural network is a parameterized function of the form:

$$u_\theta(x) = f_L(f_{L-1}(\dots f_1(x)))$$

where each layer f_l is defined as:

$$f_l(x) = \sigma(W_l x + b_l)$$

Here, W_l and b_l are the weight matrix and bias vector of layer l, and σ is a non-linear activation function. The parameters θ are optimized to minimize a task-specific loss function.

5.2.1 Activation Functions

Common choices include:

- ReLU: $\max(0, x)$
- Tanh: $\tanh(x)$
- Sigmoid: $\frac{1}{1+e^{-x}}$
- Swish: $x \cdot \sigma(x)$

5.2.2 Universal Approximation Theorem

A feedforward neural network with a single hidden layer can approximate any continuous function on a compact subset of $\mathbb{R}^n$ to arbitrary accuracy, given enough neurons.

5.3 PINNs Formulation: Embedding Physics into Learning

Let the general PDE be:

$$\mathcal{N}[u](x,t) = f(x,t), \quad (x,t) \in \Omega \times [0,T]$$

with suitable initial/boundary conditions. A PINN seeks a neural approximation $u_\theta(x,t)$ such that the total loss is minimized:

$$\mathcal{L}(\theta) = \lambda_f \mathcal{L}_f + \lambda_d \mathcal{L}_d + \lambda_{bc} \mathcal{L}_{bc}$$

where:

- $\mathcal{L}_f$: PDE residual loss (physics)
- $\mathcal{L}_d$: Data fitting loss
- $\mathcal{L}_{bc}$: Initial/boundary condition losses

5.3.1 PDE Residual Loss

Using automatic differentiation (AD), we compute $\mathcal{N}[u_\theta]$ and enforce:

$$\mathcal{L}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathcal{N}[u_\theta](x_f^i, t_f^i) - f(x_f^i, t_f^i) \right|^2$$

5.4 Training Process

1. Sample collocation points (x_f^i, t_f^i) from domain
2. Sample initial and boundary data points
3. Compute forward pass to obtain $u_\theta(x,t)$
4. Compute physics residuals using AD
5. Minimize total loss via optimizer (Adam, L-BFGS)

5.5 Worked Example: Solving a PDE with PINNs

Consider the 1D heat equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in [0,1],\ t \in [0,1]$$

With initial condition $u(x,0) = \sin(\pi x)$ and boundary conditions $u(0,t) = u(1,t) = 0$.

- Define $u_\theta(x,t)$ as a fully connected neural network
- Use boundary data loss:

$$\mathcal{L}_{BC} = \sum_t \left[|u_\theta(0,t)|^2 + |u_\theta(1,t)|^2\right]$$

- Initial condition loss:

$$\mathcal{L}_{IC} = \sum_x |u_\theta(x,0) - \sin(\pi x)|^2$$

- PDE residual loss calculated using autodiff

5.6 Application to Black–Scholes Equation

We model option price $V(S,t)$ using a neural network $V_\theta(S,t)$. The Black-Scholes equation is:

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

PINN architecture captures:

- PDE loss at interior points
- Boundary loss (Dirichlet/Neumann)
- Terminal payoff loss: $V(S,T) = \max(S-K, 0)$

5.6.1 Discretized Loss Expression

$$\mathcal{L} = \sum_{i=1}^{N_{int}} |\mathrm{BSM}(V_\theta(S_i,t_i))|^2 + \sum_{j=1}^{N_b} |V_\theta(S_j,t_j) - g(S_j,t_j)|^2$$

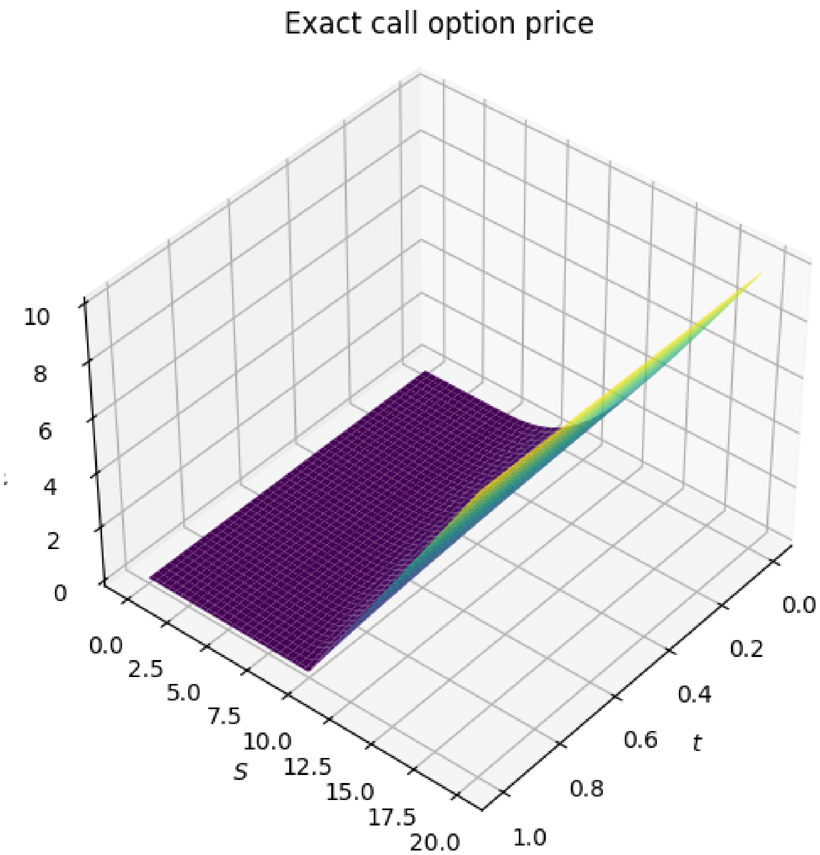

Figure 5.1: Exact solution surface of the European call option using the Black–Scholes analytical formula. The plot displays $V(S,t)$ over time $t \in [0,1]$ and asset price $S \in [0,20]$. The sharp kink at the strike price highlights the non-linearity PINNs must approximate.

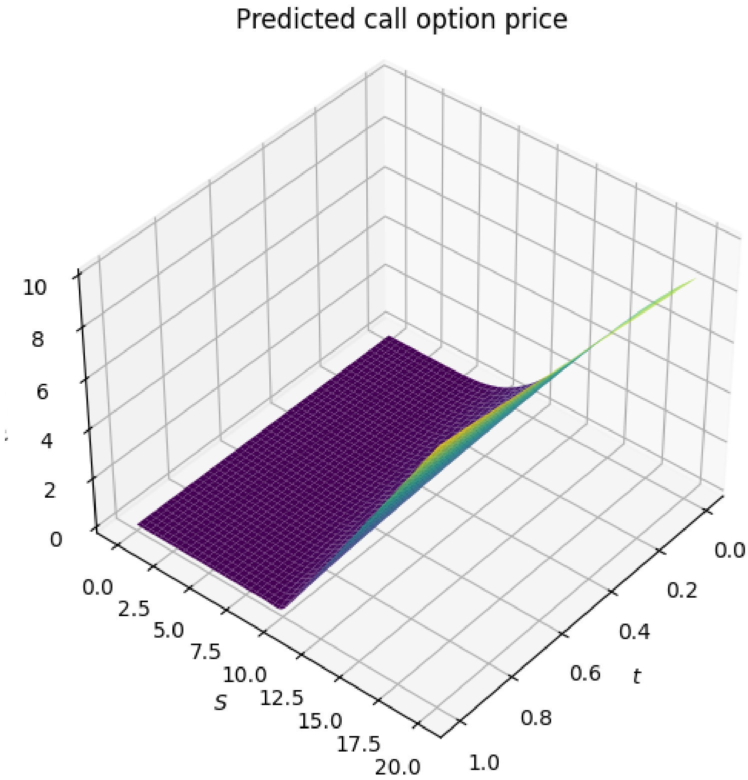

Figure 5.2: PINN-based predicted option surface after training. The PINN successfully captures the structure of the analytical solution, including the steep slope change near the strike price, reflecting accurate learning of the option payoff dynamics.

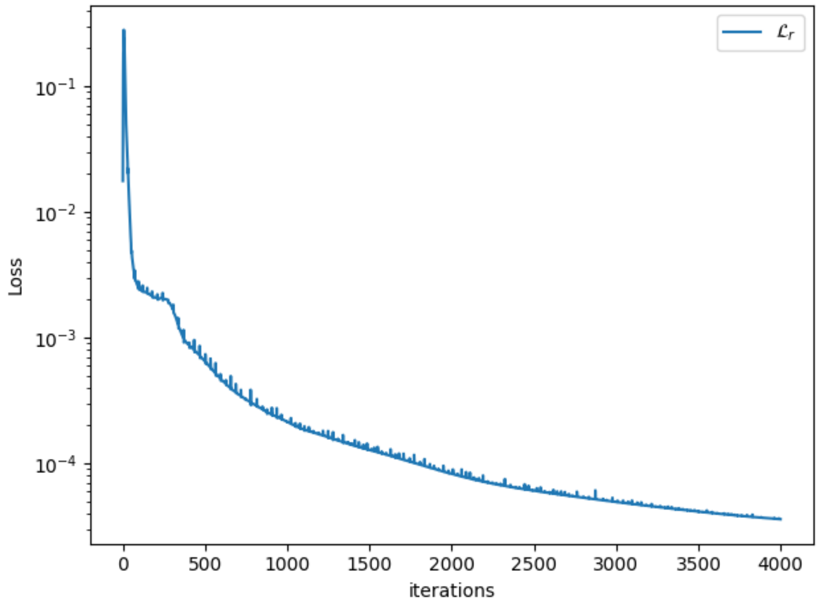

Figure 5.3: Log-scale residual loss curve $\mathcal{L}_f$ over 4000 iterations of PINN training. The steep initial drop followed by smooth convergence toward 10^{-4} validates effective learning and physics compliance across the domain.

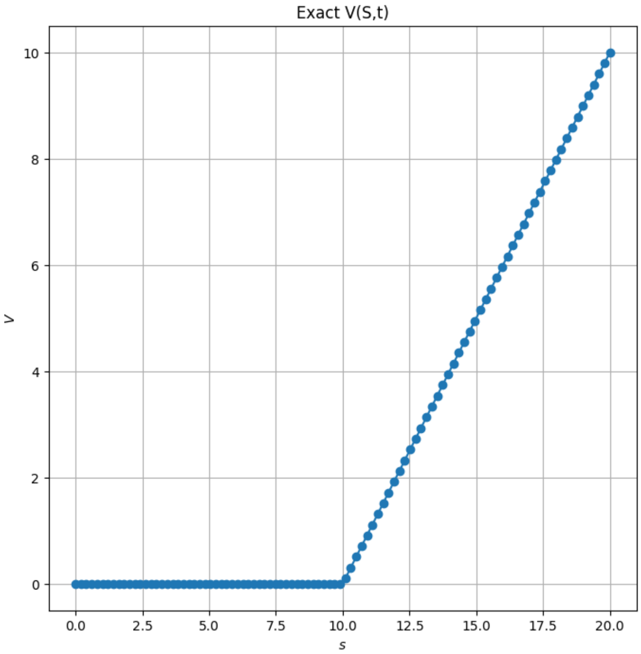

Figure 5.4: Exact Black–Scholes payoff curve $V(S,0)$ at maturity. Below the strike $K = 10$, the option value is zero; above it, the curve rises linearly with $S - K$, highlighting the non-smoothness PINNs must learn.

Chapter 6

Conclusion

This project explored various numerical and machine learning-based methods for solving differential equations, especially in the context of financial modeling through the Black-Scholes-Merton framework. Beginning with a rigorous treatment of the classical model, we extended our study to include fractional-order formulations using Caputo derivatives, offering a richer modeling tool for memory-dependent processes.

The project then moved through progressively more advanced numerical methods. Finite Difference Methods provided a solid baseline for discretizing and solving PDEs. High-Order Compact Finite Difference Methods enhanced this foundation with higher accuracy and better resolution using compact stencils, showing substantial improvements in capturing sharp transitions in solution behavior.

Finally, we explored Physics-Informed Neural Networks (PINNs), which fuse neural networks with physical law constraints. This emerging method offers tremendous potential in approximating complex differential equations in a mesh-free and data-driven manner. We presented comparative visual results, training behavior, and examples demonstrating how PINNs can serve as flexible and accurate solvers for option pricing problems.

Further Research

- Investigate hybrid models combining PINNs with traditional FDM or FEM techniques to leverage strengths from both methodologies.
- Extend the PINN framework to solve time-fractional differential equations using Caputo derivatives, particularly in financial contexts with memory effects.
- Analyze computational efficiency trade-offs between HOCFDM and PINNs in high-dimensional option pricing and multi-asset models.

Chapter 7

Bibliography

1. Finite Difference Methods in Financial Engineering – A Partial Differential Equation Approach – Daniel J. Duffy
2. Numerical solving for generalized Black-Scholes-Merton model with neural finite element method – Yinghao Chen
3. A compact finite difference scheme for fractional Black-Scholes option pricing model – Pradip Roul