uc3m | Universidad Carlos III de Madrid

Master Degree in Computational and Applied Mathematics

2024-2025

Master Thesis

Leveraging Physics-Informed Neural Networks for Option Pricing Problems

Jose Pedro Melo Olivares

Pedro Echeverria, Ph.D

Francisco Bernal, Ph.D

Madrid, 2025

SUMMARY

This master's thesis explores the usage of physics-informed neural networks (PINNs) in option pricing problems, as they allow for improved evaluation speeds compared to traditional methods. This document presents results that confirm that PINNs are capable of tackling this problem with satisfactory results.

Keywords:

DEDICATION

To my partner, Paula, for her support, to Francisco Gomez and Pedro Echeverria from BBVA for their valuable feedback and guidance and to the UC3M math faculty for their help.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

The fast advances in machine learning over the past decade have profoundly transformed fields such as speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics [1]. Among these fields, the financial industry stands out by continuously seeking efficient and accurate computational methods for pricing complex financial products. Derivatives, which are financial instruments whose value depends on underlying assets [2], are particularly significant due to their extensive use for risk management, speculation, and hedging strategies.

Derivative's underlyings can be a wide range of financial assets, such as stocks, bonds, commodities, or indices [3]. The accurate pricing of such products is crucial in finance, as it influences investment decisions and risk management strategies and traditionally it has been done using either Monte Carlo simulations [4] or solving Partial Differential Equations (PDEs) -like the renowned Black-Scholes equation [5]. While PDE-based methods provide valuable theoretical insights, their applicability has been restricted by the "curse of dimensionality" [6] and the speed of evaluation, particularly when dealing with derivatives dependent on multiple assets or path-dependent features. As a result, Monte Carlo simulations have historically been the preferred numerical approach in practice. However, the advent of different machine-learning-based techniques now presents an opportunity to overcome these constraints effectively [7].

The emergence of Physics-Informed Neural Networks (PINNs) [8], a modern machine learning methodology, provides an innovative approach that addresses these challenges. PINNs leverage neural networks not only as universal approximators of functions but also explicitly incorporate the governing PDEs [8] of the underlying financial model into their training. This integration enables PINNs to learn directly from the mathematical structure of the problem, ensuring accuracy and consistency with the financial theory while dramatically reducing computation times, at the expense of amortized training times. This advantage is particularly appealing to financial institutions that require real-time or near-real-time pricing for complex derivatives.

Additionally, PINNs can generate complete price surfaces fast, facilitating immediate sensitivity analyses and risk assessment across a broad range of market scenarios. Such capabilities are particularly valuable in context like valuation adjustments -also know as XVA-, and internal modeling frameworks.

This master's thesis aims to explore and validate the effectiveness of PINNs in solving derivative pricing problems. Guided by industry professionals from BBVA, the research presented herein assesses the capability of PINNs to efficiently and accurately price com-

plex derivatives through PDEs, demonstrating significant potential for practical adoption in financial institutions.

The subsequent chapters will present a comprehensive review of derivative pricing theory, introduce the mathematical foundations and computational framework of PINNs, describe the specific implementation details, and thoroughly evaluate their performance against traditional pricing methods.

1.2. Objectives

The primary objective of this thesis is to investigate the use of PINNs for the pricing of financial derivatives involving different dynamics and products and explore the potential this technique as a competitive alternative to traditional numerical procedure such as Monte Carlo simulations and finite difference methods.

To this end, the specific objectives are as follows:

- To review the theoretical foundations of derivative pricing and the associated PDEs for some representative selection of derivative products.
- To provide a concise overview of classical numerical methods for solving PDEs in the context of derivative pricing and highlighting their limitations.
- To assess the overall effectiveness and scalability of PINNs in pricing both single-asset and multi-asset derivatives.

This thesis also aims to provide practical insights relevant to the financial industry, supported by guidance from professionals at BBVA, and to contribute to the growing body of research that bridges deep learning and quantitative finance.

1.3. Thesis Structure

This thesis is structured into six main chapters, each addressing a different aspect of the research:

- **Chapter 1: Introduction**
  Presents the motivation behind the study, defines the objectives, and outlines the structure of the thesis.
- **Chapter 2: Theoretical Background**
  Introduces the fundamental concepts of derivative pricing, including the Black-Scholes model and its multi-asset extension. It also provides a brief overview of traditional numerical methods, such as finite difference methods and Monte Carlo simulations, and discusses their limitations in high-dimensional settings.

- **Chapter 3: Physics-Informed Neural Networks**
  Explains the concept of PINNs, starting with the theoretical underpinnings of neural networks. It then discusses the structure of PINNs, their components, and the training methodology used to enforce the underlying PDE constraints.

- **Chapter 4: Implementation and Results**
  Presents the results obtained from both single-asset and multi-asset derivative pricing experiments. The performance of the PINNs is analyzed and compared to expectations, highlighting strengths, challenges, and insights gained from the implementation.

- **Chapter 5: Conclusions and Future Work**
  Summarizes the main findings of the study, reflects on the limitations encountered, and proposes directions for future research in the application of PINNs in quantitative finance.

This structure is intended to provide a logical and progressive narrative that moves from theory to implementation and finally to critical evaluation and discussion.

2. THEORETICAL BACKGROUND

This chapter provides the foundational concepts necessary for understanding the pricing of financial derivatives, the mathematical tools traditionally used for their valuation, and the challenges associated with high-dimensional problems. It begins by introducing what derivatives are and the different types commonly traded in financial markets, before discussing the theoretical models used to price them and the limitations of classical numerical methods.

2.1. Derivative Pricing Fundamentals

2.1.1. Derivatives Overview

Financial derivatives are contracts whose cash flows are linked to the value of one or several *underlyings*—shares, commodities, foreign-exchange rates, or interest-rate curves [2], [3]. By re-packaging the risk of these underlyings, derivatives let market participants insure portfolios, express directional views with limited capital, and design bespoke payoffs that would be hard to replicate through spot trading alone.

Single-asset vanilla options. The starting point for almost every option textbook is the *European* call or put. A European call delivers, at a fixed future date T, the right to buy the underlying at the pre-agreed strike K; the put grants the right to sell. The payoffs

$$\max(S_T - K, 0), \qquad \max(K - S_T, 0)$$

are therefore functions of the future spot price S_T only, so valuing the contract amounts to forecasting the *distribution* of S_T under a risk-neutral measure. The celebrated Black-Scholes formula achieves this under a log-normal assumption for returns, but observed market prices exhibit persistent *volatility smiles*, forcing practitioners to enrich the model with time-varying or random volatility. Such extensions break the closed-form solution and motivate numerical solvers—finite differences, Fourier integrals, Monte-Carlo simulations, and, as this thesis explores, Physics-Informed Neural Networks (PINNs).

Early-exercise complexity: the American put. An American option can be exercised *any time* up to expiry. Because the holder may choose the most favourable moment, the pricing problem now involves deciding when to stop as well as what the discounted payoff might be. In mathematical terms the valuation becomes a *free-boundary* problem: one must find both the option price and the moving boundary separating the *hold and exercise* regions. Tree methods, finite-difference schemes, and regression-based Monte-Carlo algorithms cope with this added layer, but each suffers either slow convergence or high variance.

Going multi-asset: best-of basket calls. Retail structured notes often promise the performance of "whichever of these three tech stocks does best." Mathematically the payoff is

$$\max(\max_{1\le i\le d} S_{i,T} - K, 0),$$

where only the strongest performer among d assets matters. Pricing therefore depends on how the assets move *together*. Even if each asset were independent, traditional finite-difference grids would expand exponentially with d; when cross-correlation is introduced, the curse of dimensionality worsens. Monte-Carlo sampling remains feasible but noisy. These basket options hence provide an ideal test bed for high-dimensional PINN techniques.

Adding market factors: equity calls with stochastic volatility and rates. Long-dated equity or FX options are not only sensitive to the level of the underlying but also to the path followed by volatility and by interest rates used to discount future cash flows. In industry a popular approach is to combine a stochastic-volatility model (capturing the volatility smile) with a one-factor description of the short-term interest rate (capturing the term-structure of discount curves). Although each ingredient is tractable in isolation, their combination produces a three-dimensional valuation problem with no analytic shortcut.

Interest-rate derivatives. Unlike equity options, which depend on a traded asset, interest-rate contracts are tied to an entire curve of future borrowing costs. A *payer swaption* gives its owner the right, at time T, to lock in a fixed rate K on a standard interest-rate swap running from T to some final date T_n. If, at expiry, the prevailing market swap rate $L(T)$ exceeds K, the option is in the money and the holder enters the swap; otherwise it is left to lapse. One can think of the payoff as the present value of a series of future fixed-vs-floating exchanges, multiplied by $\max(L(T) - K, 0)$.

To compute today's price, a model must therefore (i) fit the current discount curve so that the valuation of the floating leg is consistent with observable bond prices, and (ii) describe how the whole curve might move between now and T. The one-factor Hull-White model remains a workhorse because it achieves task (i) exactly while keeping task (ii) low-dimensional. However, even with this simplification closed-form solutions exist only for the cleanest European style; callability clauses, collateral requirements, or cash-settlement conventions re-introduce numerical integration. PINNs offer a flexible alternative: once the governing partial differential equation is embedded into the neural network, curve fitting and option valuation are solved simultaneously.

Connecting the dots. From the plain European call to the swaption, each contract adds a new layer of reality—early-exercise privilege, multiple underlyings, extra risk factors, or dependence on an entire yield curve. These layers defeat the neat closed forms of classical option theory and stretch grid-based solvers beyond practicality. PINNs, by contrast, are mesh-free, dimension-agnostic, and naturally incorporate boundary or inequality constraints in their loss functions. The remainder of the thesis will quantify how these at-

tributes translate into speed and accuracy across our five representative derivatives [9], [10].

2.2. Modelling Framework

Financial derivatives are priced using tools derived from stochastic calculus. Throughout this section we fix a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{Q})$ satisfying the usual conditions and interpret $\mathbb{Q}$ as a *risk-neutral* measure. All stochastic processes introduced below are $(\mathcal{F}_t)$-adapted and, unless stated otherwise, driven by one or several standard $\mathbb{Q}$-Brownian motions with the specified instantaneous correlation.

2.2.1. The Black-Scholes Model

We begin by explaining option price under the assumptions of the Black-Scholes model. This framework, proposed by [11], assumes option prices are a consecuence of *hedging*, a procedure used to eliminate risk associated with the fluctuation of the underlying asset. The argument used is as follows.

Let's assume that a price process behaves as a random variable $(S_t)_{t\geq 0}$ that follows a geometric Brownian motion give by the stochastic differential process:

$$\mathrm{d}S_t = \mu S_t \,\mathrm{d}t + \sigma S_t \,\mathrm{d}W_t^{\mathbb{P}}, \qquad S_0 > 0, \tag{2.1}$$

where μ is the mean of the process and σ its annualized volatility. Now let us propose a portfolio composed of a sum of the derivative $V(S,t) := V_t$, and some units of the underlying asset. This portfolio is also a stochastic process given by

$$\Pi_t = V_t - \Delta S_t. \tag{2.2}$$

A well know fact from economics **<empty citation>** is that riskless porfolios are required to yield the same rate of return as risk-free investments. This implies that Π_t evolves as

$$\Pi_t = e^{rt} \implies \mathrm{d}\Pi_t = r\Pi_t \mathrm{d}t = r(V_t - \Delta S_t)\mathrm{d}t. \tag{2.3}$$

Applying Ito's lemma to the left side we get

$$\mathrm{d}\Pi_t = \mathrm{d}V_t - \Delta S_t, \tag{2.4}$$

where $\mathrm{d}V$ is given by the following PDE

$$\mathrm{d}V_t = \Big(\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial t} + \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\Big)\mathrm{d}t + \sigma \frac{\partial V}{\partial S}\mathrm{d}W_t^{\mathbb{P}}. \tag{2.5}$$

Choosing

$$\Delta_t = \frac{\partial V}{\partial S}(t, S_t)$$

eliminates the Brownian increment in the hedged portfolio, because the $\sigma\,\partial_S V\,\mathrm{d}W_t^{\mathbb{P}}$ term in $\mathrm{d}V_t$ exactly cancels the corresponding term in $\Delta_t\,\mathrm{d}S_t$. The remaining drift of Π_t must equal $r\Pi_t\,\mathrm{d}t$ by (2.3), so

$$r(V_t - \Delta_t S_t) \;=\; \frac{\partial V}{\partial t} \;+\; \mu S_t \frac{\partial V}{\partial S} \;+\; \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 V}{\partial S^2}.$$

Substituting Δ_t and rearranging yields the *Black–Scholes partial differential equation*

$$\boxed{\frac{\partial V}{\partial t} \;+\; \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \;+\; rS\,\frac{\partial V}{\partial S} \;-\; rV \;=\; 0,} \tag{2.6}$$

to be solved for $V(t,S)$ with terminal condition $V(T,S) = \Phi(S)$, where Φ denotes the contract's payoff. For the sake of notation, we introduce the operator $\mathcal{L}$, which is defined for the Black-Scholes model as

$$\mathcal{L}^{\mathrm{BS}} := \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS\,\frac{\partial}{\partial S}, \tag{2.7}$$

and rewrite (2.6) as

$$\frac{\partial V}{\partial t} + \mathcal{L}^{\mathrm{BS}} V - rV = 0. \tag{2.8}$$

One key result of the Black-Scholes model is that the risk-neutral dynamics of the underlying asset must has a drift equal to the risk-free rate r in the risk-neutral measure $\mathbb{Q}$. This fact is implied by the Feyman-Kac theorem, which states that the solution to the PDE (2.8) can be expressed as

$$V(t,S) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[\Phi(S_T)|S_t = S], \tag{2.9}$$

where $\mathbb{E}^{\mathbb{Q}}[\cdot]$ denotes the expectation under the risk-neutral measure $\mathbb{Q}$ and S_t follows the geometric Brownian motion (2.1) with drift $\mu = r$.

2.2.2. Multi-Asset Extension of the Black-Scholes Model

Now we proceed to extend the model for multiple underlying assets. From the previous section we know that the underlying assets must have a drift equal to the risk-free rate r in the risk-neutral measure $\mathbb{Q}$. For $d \geq 2$ underlyings the vector $\mathbf{S}_t = (S_t^1, \ldots, S_t^d)^\top$ satisfies

$$\mathrm{d}S_t^i = r\,S_t^i\,\mathrm{d}t + \sigma_i S_t^i\,\mathrm{d}W_t^{\mathbb{Q},i}, \qquad \mathrm{d}W_t^{\mathbb{Q},i}\,\mathrm{d}W_t^{\mathbb{Q},j} = \rho_{ij}\,\mathrm{d}t, \quad 1 \leq i,j \leq d. \tag{2.10}$$

Let $V(t,\mathbf{S})$ denote the option value. The same replication argument gives

$$\mathcal{L}^{\mathrm{BS},d} := \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sigma_i\sigma_j\rho_{ij}S_iS_j\,\frac{\partial}{\partial S_iS_j} + r\sum_{i=1}^{d} S_i\partial_{S_i} - r, \tag{2.11}$$

with the same PDE to solve

$$\frac{\partial V}{\partial t} + \mathcal{L}^{\mathrm{BS},d} V - rV = 0. \tag{2.12}$$

As the goal is to price using PINNs, is important to mention this technique benefits from input variables that are centred and have comparable magnitudes. Its possible to achieve this by expressing prices in *log-moneyness* $x_i := \ln(S_i/K)$ and by running the problem backward in time. Formally, for maturity T define

$$x_i = \ln \frac{S_i}{K}, \qquad \tau = T - t, \qquad u(\tau, \mathbf{x}) = \frac{V(t, \mathbf{S})}{K},$$

where $\mathbf{S} = (S_1, \ldots, S_d)^\top$ and K is the strike. Starting from the multi-asset Black-Scholes PDE and applying the above change of variables, the chain rule yields the *dimensionless backward* equation

$$\frac{\partial u}{\partial \tau} = \frac{1}{2} \sum_{i=1}^{d} \sigma_i^2 \Big(\frac{\partial^2 u}{\partial x_i^2} - \frac{\partial u}{\partial x_i} \Big) + \frac{1}{2} \sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} \frac{\partial^2 u}{\partial x_i \, \partial x_j} + r \sum_{i=1}^{d} \frac{\partial u}{\partial x_i} - ru. \tag{2.13}$$

to be solved for $\tau \in (0, T]$ and $\mathbf{x} \in \mathbb{R}^d$ with terminal condition $u(0, \mathbf{x}) = \Phi(\mathbf{x})/K$. Again, it is convenient to rewrite (2.13) in operator form

$$-\frac{\partial u}{\partial \tau} + \widetilde{\mathcal{L}}^{\mathrm{BS},d} u - ru = 0, \qquad u(0, \mathbf{x}) = \frac{\Phi(\mathbf{x})}{K} \, ,$$

where the *dimensionless generator*

$$\widetilde{\mathcal{L}}^{\mathrm{BS},d} := \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sigma_i \sigma_j \rho_{ij} \frac{\partial}{\partial x_i x_j} - \frac{1}{2} \sum_{i=1}^{d} \sigma_i^2 \frac{\partial}{\partial x_i} \tag{2.14}$$

acts on twice differentiable test functions and is elliptic with constant coefficients. Equation (**??**) will serve as the reference problem for the mesh-free PINN solver in Section **??**.

2.2.3. Extensions Beyond Black-Scholes

Stochastic Volatility

Its a common observation in the market that option prices do not follow the log-normal distribution assumed by the Black-Scholes model. In general, the implied volatilities vary with strike and maturity, forming the well-known *volatility smile*. A single constant parameter σ is therefore insufficient to reproduce observed prices [3]. Moreover, for maturities beyond one year the assumption of a deterministic discount rate becomes untenable; the entire yield curve moves randomly in response to macroeconomic news and monetary policy [3]. This is key when pricing long-dated equity options, which are sensitive to equity, volatility, and interest rate dynamics. These considerations motivate models that combine *stochastic volatility* with *stochastic interest rates*.

Among the many specifications proposed, the square-root process of [12] offers a parsimonious and analytically tractable alternative to Black-Scholes. Under the risk-neutral measure $\mathbb{Q}$ the state vector (S_t, v_t) satisfies

$$\begin{aligned} \mathrm{d}S_t &= r\, S_t \, \mathrm{d}t + \sqrt{v_t}\, S_t \, \mathrm{d}W_t^{\mathbb{Q},S}, \\ \mathrm{d}v_t &= \kappa(\theta - v_t)\, \mathrm{d}t + \sigma_v \sqrt{v_t}\, \mathrm{d}W_t^{\mathbb{Q},v}, \\ \mathrm{d}W_t^{\mathbb{Q},S} \, \mathrm{d}W_t^{\mathbb{Q},v} &= \rho \, \mathrm{d}t, \end{aligned} \tag{2.15}$$

where $\kappa > 0$ is the mean-reversion speed, $\theta > 0$ the long-run variance, $\sigma_v > 0$ the volatility of volatility, and $\rho \in [-1, 1]$ the instantaneous correlation. The Feller condition $2\kappa\theta > \sigma_v^2$ guarantees $v_t > 0$ almost surely.

In order to get the pricing PDE, we set $V(t, S, v)$ to denote the price, at time t, of a derivative maturing at T with payoff $\Phi(S_T)$. Assuming V is twice continuously differentiable in S and v and once in t, Itô's lemma applied to $V(t, S_t, v_t)$ gives

$$\begin{aligned} \mathrm{d}V = &\left(\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \kappa(\theta - v)\frac{\partial V}{\partial v} + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2 V}{\partial v^2} + \rho\,\sigma_v vS\,\frac{\partial^2 V}{\partial S\,\partial v}\right)\mathrm{d}t \\ &+ \sqrt{v}\,S\,\frac{\partial V}{\partial S}\,\mathrm{d}W_t^{\mathbb{Q},S} + \sigma_v\sqrt{v}\,\frac{\partial V}{\partial v}\,\mathrm{d}W_t^{\mathbb{Q},v}. \end{aligned}$$

From the risk-neutral valuation theory [13], we know that all assets, including the derivative V, must have a drift equalt to the risk-free rate r in the risk-neutral measure $\mathbb{Q}$. This implies that the drift of the process $\mathrm{d}V$ must be equal to

$$\mathrm{d}V_t = rV_t\mathrm{d}t. \tag{2.16}$$

Using this fact and the Feynman-Kac theorem, which states that given the terminal condition $V(T, S, v) = \Phi(S)$, the equation

$$V(t, S, v) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[\Phi(S_T)|S_t = S, v_t = v], \tag{2.17}$$

is solution to the PDE

$$\frac{\partial V}{\partial t} + \tfrac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\sigma_v vS\,\frac{\partial^2 V}{\partial S\,\partial v} + \tfrac{1}{2}\sigma_v^2 v\frac{\partial^2 V}{\partial v^2} + \kappa(\theta - v)\frac{\partial V}{\partial v} + rS\,\frac{\partial V}{\partial S} - rV = 0, \tag{2.18}$$

which is the *Heston PDE*. Again, we can rewrite this PDE in operator form

$$\mathcal{L}^{\text{Heston}} := \frac{1}{2}vS^2\frac{\partial^2}{\partial S^2} + \rho\sigma_v vS\,\frac{\partial^2}{\partial S\,\partial v} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2}{\partial v^2} + \kappa(\theta - v)\frac{\partial}{\partial v} + rS\,\frac{\partial}{\partial S}, \tag{2.19}$$

and rewrite the PDE as

$$\frac{\partial V}{\partial t} + \mathcal{L}^{\text{Heston}}V - rV = 0. \tag{2.20}$$

Stochastic Interest Rates

For maturities beyond a few months or for product where the underlying is a interest rate, modelling the short rate as a constant parameter is inadequate [14] as it fails to capture the time-varying nature of interest rates. A more realistic approach is to assume that the short rate r_t follows a stochastic process, which allows it to evolve in response to macroeconomic factors, monetary policy, and other market dynamics. There is a plethora of models that capture this behaviour, but two of the most popular are the *Vasicek* model [15] and the *Hull-White* model [16]. Both models assume that the short rate follows a mean-reverting process, which is a common feature in interest rate dynamics. The key

difference is that the Hull-White model allows for a time-dependent mean reversion level, making it more flexible in fitting the current yield curve.

We can define both models in terms of the following stochastic differential equations (SDEs):

$$\mathrm{d}r_t = \kappa(\theta_t - r_t)\,\mathrm{d}t + \sigma_r\,\mathrm{d}W_t^{\mathbb{Q},r}. \tag{2.21}$$

where in both κ is the mean-reversion speed and θ_t is the time-dependent long-run mean. In the Vasicek model, θ_t is a constant, while in the Hull-White model it can be a deterministic function of time. A derivative whose value depends on the short rate , $V(t, r)$, has a pricing PDE

$$\frac{\partial V}{\partial t} + \kappa(\theta_t - r)\frac{\partial V}{\partial r} + \frac{1}{2}\sigma_r^2\frac{\partial^2 V}{\partial r^2} - rV = 0. \tag{2.22}$$

This PDE is derived in the same fashion as the ones from the previous sections. In operator form, we can write

$$\mathcal{L}^{\mathrm{SR}} := \frac{1}{2}\sigma_r^2\frac{\partial}{\partial r^2} + \kappa(\theta_t - r)\frac{\partial}{\partial r}, \tag{2.23}$$

and rewrite the PDE as

$$\frac{\partial V}{\partial t} + \mathcal{L}^{\mathrm{SR}}V - rV = 0. \tag{2.24}$$

What sets aside short-rate models from the previous sections is that they are *affine* in the state variable r_t. Affine models are those whose dynamics can be expressed as a linear combination of the state variable and its exponential function. This property allows for closed-form solutions for the prices of certain derivatives, such as zero-coupon bonds. See anex **??** for a more detailed discussion on affine models.

2.2.4. Joint Stochastic Volatility and Stochastic Rates

Finally, we drop both the assumption of constant volatility and the assumption of constant interest rates, and consider a model that combines stochastic volatility and stochastic interest rates. As already mentioned, this is desirable for pricing long-dated products.

2.3. Traditional Numerical Approaches

Brief Overview of Finite Difference Methods

Finite Difference Methods (FDM) are among the most widely used techniques for numerically solving partial differential equations such as (**??**) or its multi-dimensional counterpart (**??**). The core idea behind FDM is to discretize the time and asset domains into a structured grid and approximate the derivatives in the PDE using finite differences.

For example, in the single-asset case, the time derivative can be approximated by a backward difference:

$$\frac{\partial V}{\partial t} \approx \frac{V^n - V^{n-1}}{\Delta t}, \tag{2.25}$$

and spatial derivatives using central differences:

$$\frac{\partial V}{\partial S} \approx \frac{V_{i+1}^n - V_{i-1}^n}{2\Delta S} \qquad \frac{\partial^2 V}{\partial S^2} \approx \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{\Delta S^2} \tag{2.26}$$

This transforms the PDE into a system of algebraic equations, which can be solved using standard linear algebra techniques such as LU decomposition or iterative solvers.

Despite their simplicity and interpretability, FDMs suffer from the CoD. In the d-dimensional case, the number of grid points grows exponentially with d, making them impractical for high-dimensional problems. As a result, their application is limited to cases with small d or where symmetry or decomposition strategies can be employed to reduce complexity.

Monte Carlo Methods

Monte Carlo (MC) methods are stochastic techniques that simulate multiple sample paths of the underlying assets under the risk-neutral measure $\mathbb{Q}$ and average the discounted payoff [4]. In the context of the multi-asset Black-Scholes model (**??**), this involves simulating correlated geometric Brownian motions. An example of a MC pricing algorithm is shown below:

Algorithm 1 MC Pricing of a General Payoff

1: **Input:** Number of simulations N, time horizon T, risk-free rate r, strike K, asset parameters $(\mu_i, \sigma_i, \rho_{ij})$
2: **for** $n = 1$ to N **do**
3: Simulate one path for each asset S_T^i using correlated Brownian motion
4: Compute the path-dependent payoff: $\text{Payoff}^{(n)} = \max\left(\max_i S_T^i - K, 0\right)$
5: **end for**
6: Compute the option price:

$$V(0, \mathbf{S}_0) \approx e^{-rT} \cdot \frac{1}{N} \sum_{n=1}^{N} \text{Payoff}^{(n)}$$

MC methods are particularly attractive for high-dimensional problems, as their convergence rate is independent of the number of dimensions. However, they converge slowly — with a rate of $O(1/\sqrt{N})$ — so often require variance reduction techniques (e.g., control variates, antithetic variables) to be efficient.

Another drawback, where PDE methods outperform MC methods, is that they are not well-suited for pricing American options or other path-dependent derivatives, as they require the ability to exercise at multiple points in time. While there are techniques to adapt MC methods for American options, such as the Longstaff-Schwartz method [17], they are often less efficient than PDE-based methods. This is where PINNs can provide a

significant advantage, as they can naturally handle path-dependent payoffs and American-style options, among others.

3. PHYSICS-INFORMED MACHINE LEARNING

3.1. Introduction to Physics-Informed Neural Networks (PINNs)

As previously discussed, traditional numerical methods for solving PDEs in derivative pricing, such as FDM and MC simulations, often struggle with computational complexity, particularly when dealing with high-dimensional financial instruments. PINNs, introduced by Raissi et al. [8], provide an innovative framework that explicitly incorporates governing PDEs directly into neural network training. Rather than relying solely on large amounts of data, PINNs leverage physical laws and constraints embedded within the problem formulation to enhance predictive accuracy, efficiency, and reliability of the solutions. This methodology is especially promising for high-dimensional derivative pricing problems, as it effectively mitigates the CoD while maintaining consistency with the underlying financial theory.

Consider the PDE problem defined by:

$$\begin{aligned} \mathcal{N}_I[u(t,x)] &= 0, \quad x \in \Omega, \quad t \in [0,T], \\ \mathcal{N}_B[u(t,x)] &= 0, \quad x \in \partial\Omega, \quad t \in [0,T], \\ \mathcal{N}_0[u(t^*,x)] &= 0, \quad x \in \Omega, \quad t^* = 0, \end{aligned} \tag{3.1}$$

where $\mathcal{N}_I$ denotes the PDE operator in the interior of the spatial domain Ω, $\mathcal{N}_B$ represents the boundary condition operator on $\partial\Omega$, and $\mathcal{N}_0$ enforces initial conditions at $t = 0$.

PINNs approximate the PDE solution $u(t,x)$ with a neural network $u_\theta(t,x)$, parameterized by θ. The optimal parameters θ are found by minimizing the composite loss function:

$$\mathcal{L}(\theta) = \lambda_1 \mathcal{L}_{\text{PDE}}(\theta) + \lambda_2 \mathcal{L}_{\text{boundary}}(\theta) + \lambda_3 \mathcal{L}_{\text{initial}}(\theta), \tag{3.2}$$

with each component explicitly defined as:

$$\begin{aligned} \mathcal{L}_{\text{PDE}}(\theta) &= \int_0^T \int_\Omega |\mathcal{N}_I[u_\theta(t,x)]|^2 \, \mathrm{d}x \, \mathrm{d}t, \\ \mathcal{L}_{\text{boundary}}(\theta) &= \int_0^T \int_{\partial\Omega} |\mathcal{N}_B[u_\theta(t,x)]|^2 \, \mathrm{d}s \, \mathrm{d}t, \\ \mathcal{L}_{\text{initial}}(\theta) &= \int_\Omega |\mathcal{N}_0[u_\theta(0,x)]|^2 \, \mathrm{d}x. \end{aligned} \tag{3.3}$$

These integrals are typically approximated by Monte Carlo sampling, choosing discrete collocation points within the respective domains.

3.2. Challenges in Training PINNs

Training PINNs involves addressing several critical challenges:

- **Loss imbalance:** Significant differences in the magnitude of PDE and boundary losses can lead to convergence issues, addressed through adaptive weighting strategies [18].
- **Spectral bias:** Neural networks prioritize low-frequency solutions, complicating representation of high-frequency or multi-scale phenomena [19].
- **Causality violations:** Training PINNs over entire temporal domains simultaneously may violate physical causality, prompting the use of time-adaptive training schemes [20].
- **Computational complexity:** Extensive computational resources are needed, particularly for high-dimensional PDEs, due to expensive gradient computations [21].

3.3. Developments and Applications

PINN's field of investigation has witnessed substantial advancements aimed at addressing their inherent limitations, such as training inefficiencies, scalability challenges, and difficulties in capturing complex solution behaviors. This section delineates notable developments that enhance the capability and applicability of PINNs.

Extended Physics-Informed Neural Networks

Extended Physics-Informed Neural Networks (XPINNs) introduce a generalized space-time domain decomposition strategy, partitioning the problem domain into multiple subdomains. Each subdomain is assigned a distinct neural network, facilitating parallel training and improved scalability. This approach is particularly effective for complex geometries and high-dimensional problems, as it allows for localized learning and reduces computational overhead [22].

Variational and Augmented PINNs

Variational PINNs (VPINNs) reformulate the loss function using variational principles, integrating test functions and quadrature rules to enhance solution accuracy and convergence rates. This method is beneficial for problems where traditional PINNs struggle with stability and precision [23]. **Revisar** Augmented PINNs (APINNs) incorporate additional physical constraints or auxiliary variables into the neural network architecture, improving the network's ability to capture complex solution features and ensuring adherence to underlying physical laws [24].

Neural Operators: DeepONet and Fourier Neural Operators

Neural operators, such as Deep Operator Networks (DeepONet) and Fourier Neural Operators (FNO), represent a paradigm shift by learning mappings between function spaces rather than individual function evaluations. DeepONet employs a branch and trunk network to approximate nonlinear operators, enabling rapid predictions across varying input functions [25]. FNO leverages the Fourier transform to learn global solution operators efficiently, offering advantages in handling complex, high-dimensional PDEs with reduced computational costs [26].

Applications of PINNs

In terms of applications, PINNs have been successfully employed in various fields, including:

- **Fluid dynamics:** Modeling complex flows including turbulent and laminar regimes, and inferring hidden physical fields from sparse observational data [8], [27].
- **Biomedicine:** Improving medical imaging reconstruction and personalized patient modeling via inverse problems involving PDE constraints [28].
- **Materials science and climate modeling:** Inferring material properties and assimilating observational data into physics-based climate models for improved predictions [21].

Overall, the PINN methodology represents a robust integration of deep learning and physical modeling, actively evolving through continuous theoretical improvements and expanding applications.

Training PINNs

As already said, training PINNs involves minimizing the total loss $\mathcal{L}(\theta)$ with respect to the neural network parameters θ. This is typically done using gradient-based optimization algorithms, such as stochastic gradient descent (SGD) or its variants, which iteratively update the parameters based on the computed gradients of the loss function with respect to the parameters. The gradients are computed using automatic differentiation, which allows for efficient and accurate calculation of the gradients of the loss function with respect to the parameters. The optimization process continues until a stopping criterion is met, such as a maximum number of iterations or convergence of the loss function.

As usually data of the true solution is not available, the training process relies on sampling collocation points from the domain of interest, which are used to compute the loss terms. The collocation points are synthetically generated using a predefined procedure, for example, by sampling from a uniform distribution over the domain of interest.

A typical training procedure for PINNs is as follows:

Algorithm 2 Training procedure of PINNs

1: **Input:** Neural network architecture, PDE operator, boundary and initial conditions.
2: Initialize neural network parameters θ.
3: Sample total collocation points from Ω, $\partial\Omega$ and Ω_0.
4: **while** not converged **do**
5: Obtain a subset of collocation points from the total set.
6: Compute PDE residuals at subset of collocation points.
7: Compute total loss $\mathcal{L}(\theta)$ using (3.2) at the collocation points.
8: Compute gradients of $\mathcal{L}(\theta)$ with respect to θ.
9: Update parameters θ using an optimization algorithm.
10: **end while**

Given the multitude of options and techniques available for training PINNs, it is crucial to carefully select the most appropriate methods for each specific problem. In the following sections, we will discuss the various techniques and strategies that can be employed to enhance the training process and improve the performance of PINNs. We will explore different neural network architectures, sampling strategies, loss balancing techniques, and optimization methods that can be used to train PINNs effectively. By understanding the strengths and weaknesses of each approach, we can make informed decisions about which techniques to apply in our specific use case, ultimately leading to more accurate and efficient solutions for pricing multi-asset derivatives.

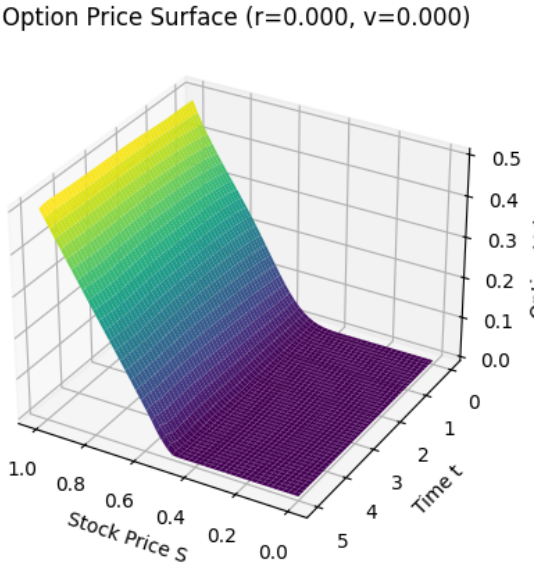
4. IMPLEMENTATIONS AND RESULTS

4.1. Implementation Details

4.2. Call Option

We begin by attempting to price a call option using the Black-Scholes model. The price surface can be seen in Figure 4.1.

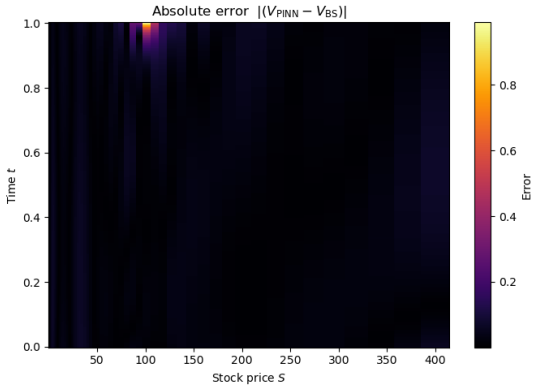
Figure 4.1
Call option pricing using the Black-Scholes model. The neural network is able to produce a solution with a relative error of around 10^{-4}.



The relative error of the solution is shown in Figure 4.2, where we can see that the error is concentrated around the maturity, close to the strike price, which is expected as the payoff of a call option is not differentiable near this region.

Figure 4.2

Relative error of the call option pricing using the Black-Scholes model. The error is concentrated around the maturity, close to the strike price.



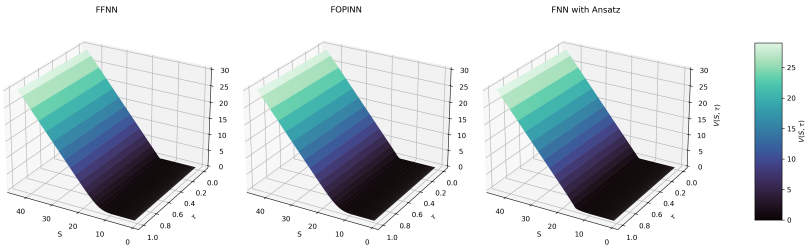
This results were obtained using a small feedforward neural network with just one hidden layer with 10 neurons, and a total of 1000 collocation points sampled uniformly across the domain. The training process took around 1000 iterations to converge, and the total loss was computed using the PDE residuals and the boundary conditions. collocation points, drawn from a uniform distribution across the domain. Training took around 1000 iterations to converge, with the total time of 2 minutes in total.

4.2.1. Multi-Asset Option

As metioned in previous sections, we can extend the previous case to many assets in order to assess PINNs capabilities to tackle PDEs with multiple state variables. We train again a small neural networks but now using the dimensionless form of the Black-Scholes PDE defined in (**??**). Figure 4.3 shows the price surface for 2, 3, 5, and 7 assets.

Figure 4.3

Call option pricing using the Black-Scholes model for multiple assets. The neural network is able to produce a solution with a relative error of around 10^{-4}.

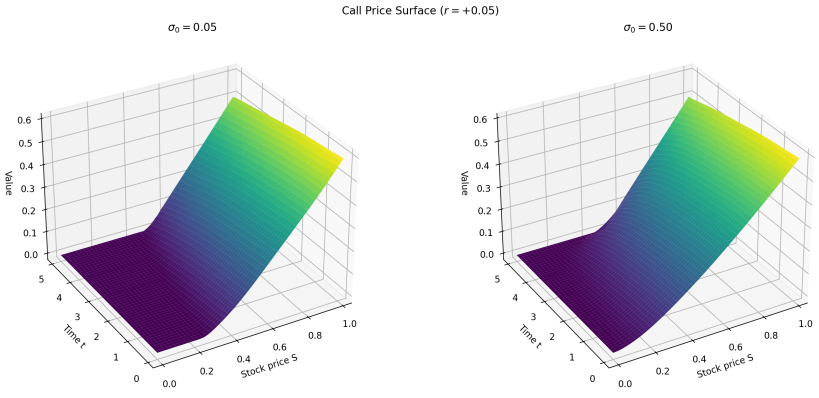


As expected, the problems becomes more challenging as the number of assets (dimensions) increases. To asses this increased complexity, we train each model for the same number of iterations and number of collocation points.

4.2.2. Call Option with Stochastic Volatility and Interest Rates

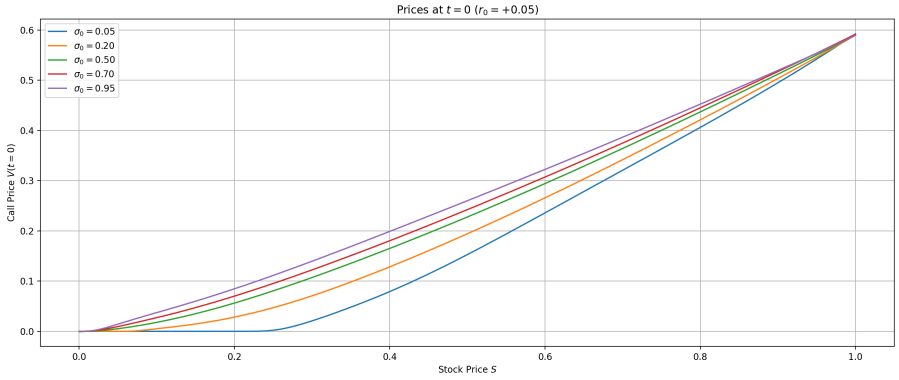
Finally, we can extend the previous case to include stochastic volatility and interest rates. In Figure 4.5, we shown the behaviour of the price surface for different values of the volatility of volatility.

Figure 4.4
Call option pricing using the Heston-Hull-White model. The neural network is able to produce a solution with a relative error of around 10^{-4}.



As expected, higher values make the surface flatter as the uncertainty of the underlying increases, as seen in Figure **??**.

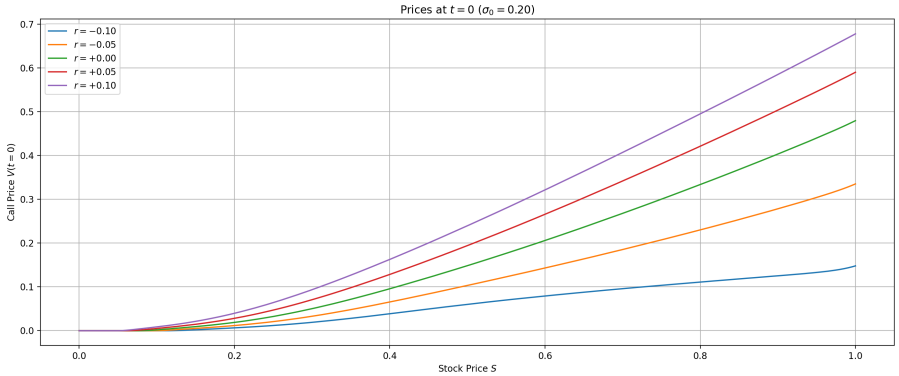
Figure 4.5
Call option pricing using the Heston-Hull-White model. The neural network is able to produce a solution with a relative error of around 10^{-4}.



Another key feature as already mentioned is that for longer maturities, interest rate curves have a significant impact on the price of the option, making this feature a key component of the model.

Figure 4.6

Call option pricing using the Heston-Hull-White model. The neural network is able to produce a solution with a relative error of around 10^{-4}.

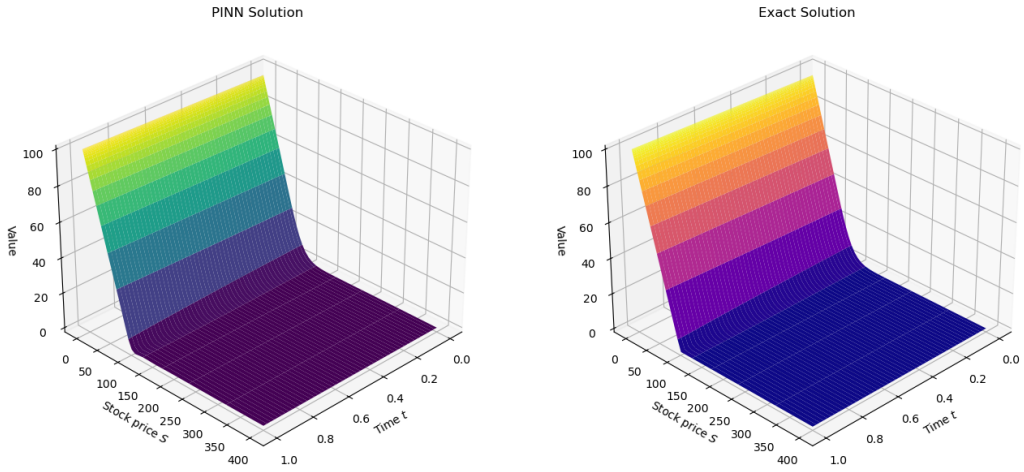


4.3. Put Option

The next product is the European Put Option, which is similar to the call option but with a different payoff function. The price surface can be seen in Figure 4.7. Notice that in this case, the stock axis is reversed, as the payoff of a put option is the opposite of a call option.

Figure 4.7

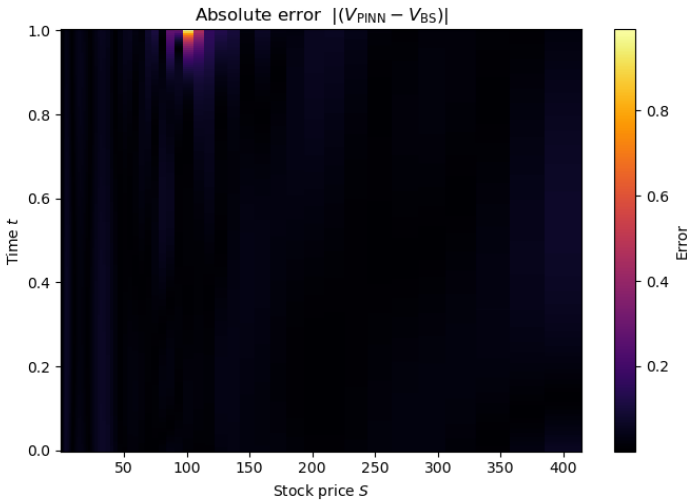
European put prices. At the left, the price surface using a PINN and at the right, the price surface using the exact solution.



The absolute error of the solution is shown in Figure 4.8, where we can see that the error is concentrated around the maturity, close to the strike price, which is expected as the payoff of a put option is not differentiable near this region.

Figure 4.8

Absolute error of the put option pricing using the Black-Scholes model. The error is concentrated around the maturity, close to the strike price.



4.3.1. American Put Option

A interesting variant of the put option is the American Put Option, which allows the holder to exercise the option at any time before maturity. This variantion makes the problem more complex, as we need to take into account the early exercise feature of the option.

Figure 4.9

American put prices. At the left, the price surface using a PINN and at the right, the price surface using the exact solution.

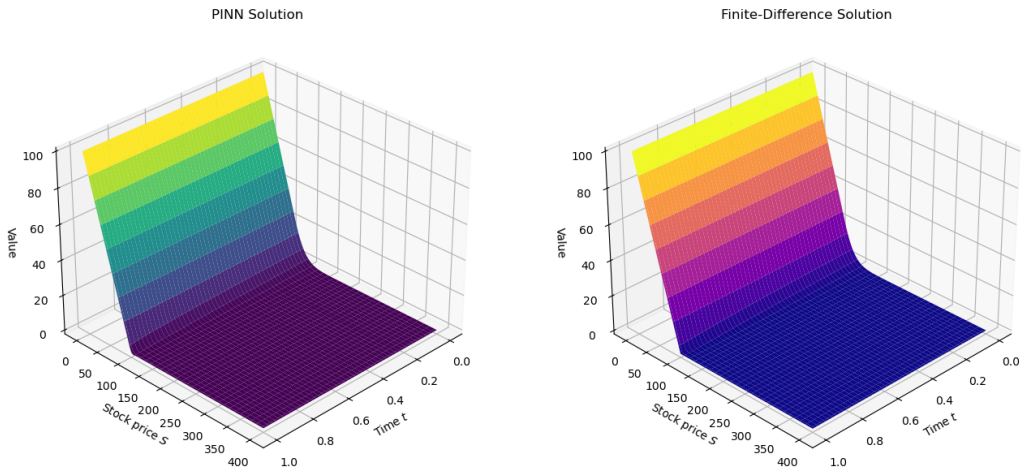


Figure 4.11

European swaption prices. At the left, the price surface using a PINN and at the right, the price surface using the exact solution.

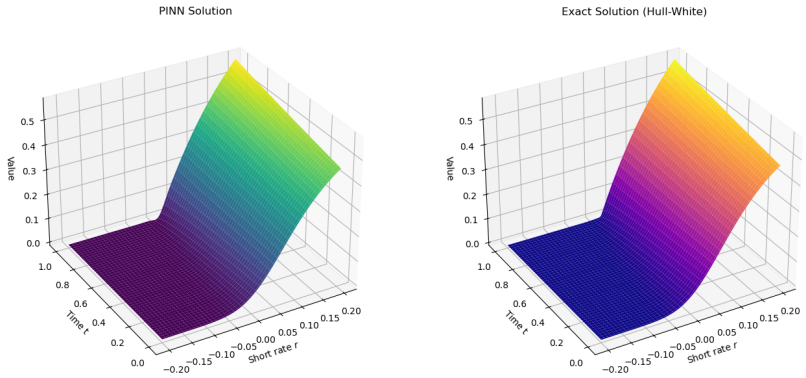
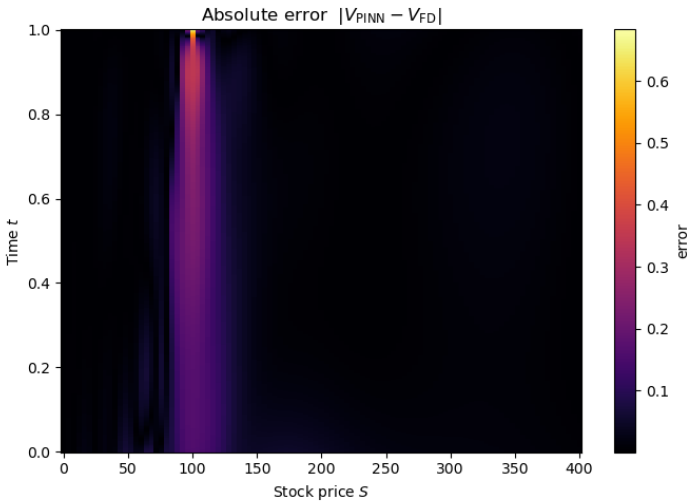


Figure 4.10

Absolute error of the American put option pricing using the Black-Scholes model. The error is concentrated around the maturity, close to the strike price.

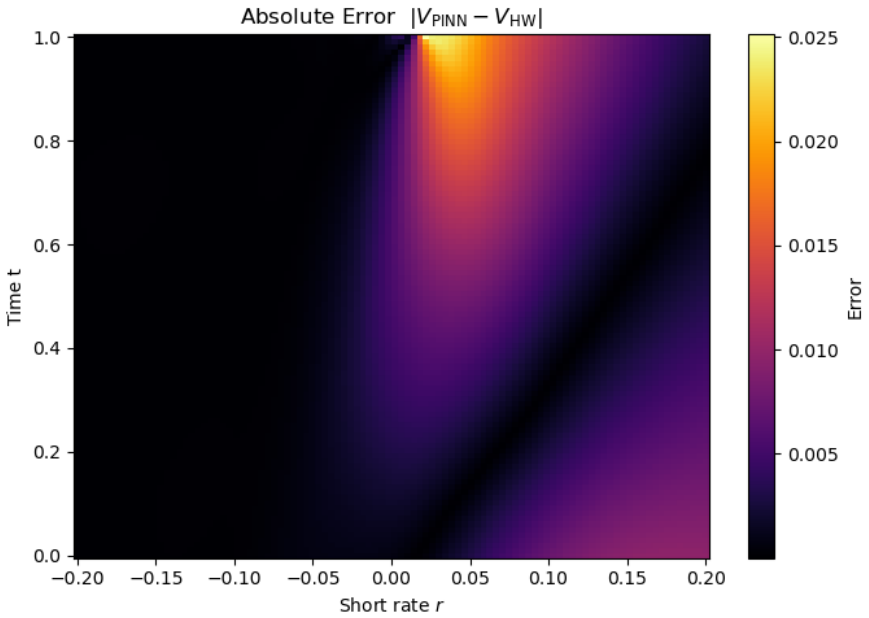


4.4. Swaption

Finally, our last product to evaluate is the European Swaption, for which we use the Hull-White model to price it.

Figure 4.12

European swaption prices. At the left, the price surface using a PINN and at the right, the price surface using the exact solution.



5. CONCLUSIONS AND FUTURE WORK

5.1. Discussion

5.2. Summary of Findings

5.3. Limitations and Future Research Directions

BIBLIOGRAPHY

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, May 2015. DOI: 10.1038/nature14539.

[2] J. Hull, *Options, futures, and other derivatives*, eng, Ninth edition, Global edition. Boston: Pearson Education Limited, 2018 - 2018, ISBN: 1292212896.

[3] P. Wilmott, "Paul wilmott on quantitative finance," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:153668090.

[4] P. Glasserman, *Monte Carlo methods in financial engineering*. New York: Springer, 2004, ISBN: 0387004513 9780387004518 1441918221 9781441918222. [Online]. Available: http://www.amazon.com/Financial-Engineering-Stochastic-Modelling-Probability/dp/0387004513/ref=pd_sim_b_68?ie=UTF8&refRID=1AN8JXSDGMEV2RPHFC2A.

[5] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973, ISSN: 00223808, 1537534X. [Online]. Available: http://www.jstor.org/stable/1831029 (visited on 04/25/2025).

[6] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[7] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, Aug. 2018, ISSN: 1091-6490. DOI: 10.1073/pnas.1718942115. [Online]. Available: http://dx.doi.org/10.1073/pnas.1718942115.

[8] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[9] B. Huge and A. Savine, *Differential machine learning*, 2020. arXiv: 2005.02347 [q-fin.CP]. [Online]. Available: https://arxiv.org/abs/2005.02347.

[10] J. B. Heaton, N. G. Polson, and J. H. Witte, *Deep learning in finance*, 2018. arXiv: 1602.06561 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1602.06561.

[11] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973, ISSN: 00223808, 1537534X. [Online]. Available: http://www.jstor.org/stable/1831029 (visited on 04/23/2025).

[12] S. L. Heston, "A closed-form solution for options with stochastic volatility with applications to bond and currency options," *The Review of Financial Studies*, vol. 6, no. 2, pp. 327–343, 1993, ISSN: 08939454, 14657368. [Online]. Available: http://www.jstor.org/stable/2962057 (visited on 07/06/2025).

[13] T. Björk, *Arbitrage Theory in Continuous Time* (Oxford Finance Series). Oxford University Press, Incorporated, 2004, ISBN: 9780191533846. [Online]. Available: https://books.google.es/books?id=TJgjgJATeVIC.

[14] D. Brigo and F. Mercurio, *Interest Rate Models Theory and Practice* (Springer Finance). Springer Berlin Heidelberg, 2013, ISBN: 9783662045534. [Online]. Available: https://books.google.es/books?id=USvrCAAAQBAJ.

[15] O. Vasicek, "An equilibrium characterization of the term structure," *Journal of Financial Economics*, vol. 5, no. 2, pp. 177–188, 1977, ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(77)90016-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304405X77900162.

[16] J. Hull and A. White, "Pricing interest-rate-derivative securities," *The Review of Financial Studies*, vol. 3, no. 4, pp. 573–592, Apr. 2015, ISSN: 0893-9454. DOI: 10.1093/rfs/3.4.573. eprint: https://academic.oup.com/rfs/article-pdf/3/4/573/24416170/030573.pdf. [Online]. Available: https://doi.org/10.1093/rfs/3.4.573.

[17] S. Longstaff, "Valuing american options by simulation: A simple least squares approach," *Review of Finance Studies*, vol. 14, pp. 113–147, Jan. 2001.

[18] S. Wang, Y. Teng, and P. Perdikaris, *Understanding and mitigating gradient pathologies in physics-informed neural networks*, 2020. arXiv: 2001.04536 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2001.04536.

[19] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, *An expert's guide to training physics-informed neural networks*, 2023. arXiv: 2308.08468 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2308.08468.

[20] S. Wang, S. Sankaran, and P. Perdikaris, *Respecting causality is all you need for training physics-informed neural networks*, 2022. arXiv: 2203.07404 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2203.07404.

[21] G. E. e. a. Karniadakis, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021.

[22] Z. Hu, A. D. Jagtap, G. E. Karniadakis, and K. Kawaguchi, "When do extended physics-informed neural networks (xpinns) improve generalization?" *SIAM Journal on Scientific Computing*, vol. 44, no. 5, A3158–A3182, Sep. 2022, ISSN: 1095-7197. DOI: 10.1137/21m1447039. [Online]. Available: http://dx.doi.org/10.1137/21M1447039.

[23] E. Kharazmi, Z. Zhang, and G. E. Karniadakis, *Variational physics-informed neural networks for solving partial differential equations*, 2019. arXiv: 1912.00873 [cs.NE]. [Online]. Available: https://arxiv.org/abs/1912.00873.

[24] Z. Hu, A. D. Jagtap, G. E. Karniadakis, and K. Kawaguchi, "Augmented physics-informed neural networks (apinns): A gating network-based soft domain decomposition methodology," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107 183, Nov. 2023, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.107183. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2023.107183.

[25] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, Mar. 2021, ISSN: 2522-5839. DOI: 10.1038/s42256-021-00302-5. [Online]. Available: http://dx.doi.org/10.1038/s42256-021-00302-5.

[26] Z. Li, N. Kovachki, K. Azizzadenesheli, *et al.*, *Fourier neural operator for parametric partial differential equations*, 2021. arXiv: 2010.08895 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2010.08895.

[27] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Computational Mechanics*, vol. 64, no. 2, pp. 525–545, Jun. 2019, ISSN: 1432-0924. DOI: 10.1007/s00466-019-01740-0. [Online]. Available: http://dx.doi.org/10.1007/s00466-019-01740-0.

[28] C. Banerjee, K. Nguyen, O. Salvado, T. Tran, and C. Fookes, *Pinns for medical image analysis: A survey*, 2024. arXiv: 2408.01026 [eess.IV]. [Online]. Available: https://arxiv.org/abs/2408.01026.