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 and flexibility to traditional methods. This document evaluates PINNs as a numerical method for pricing financial derivatives, focusing on different types of financial instruments.

Keywords: Physics-Informed Neural Networks, quantitaive finance, option pricing, derivatives.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

The rapid progress in machine learning over the past decade has significantly transformed various fields, including speech recognition, object recognition, and detection, among others [1]. Notably, the financial industry has emerged as a key sector continually seeking innovative and accurate computational methods to price complex financial products. Derivatives, whose value depends on underlying assets [2], are particularly prominent in this context, given their widespread application in risk management, speculative trading, and hedging strategies.

Derivatives can be based on a wide range of financial assets, including stocks, bonds, commodities, and indices [3]. Accurate pricing of these products is vital in finance, as it informs investment decisions and risk management strategies. Traditionally, pricing has been achieved using either Monte Carlo simulations [4] or by solving partial differential equations (PDEs), such as the well-known Black-Scholes equation [5]. While PDE-based methods offer valuable theoretical insights, their application has been limited by the "curse of dimensionality" [6].

The mergence of Physics-Informed Neural Networks (PINNs) [7] has introduced a groundbreaking approach to tackling these challenges. By harnessing the power of neural networks as universal function approximators, PINNs uniquely incorporate the underlying financial model's governing PDEs [7] into their training process. This synergy enables PINNs to learn directly from the mathematical framework of the problem, ensuring accuracy and consistency with financial theory while significantly reducing computational times. Although this comes at the cost of increased training times, the benefit is particularly attractive 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 (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 complex 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 extensions. It also provides a brief overview of traditional numerical methods, such as finite difference methods and Monte Carlo simulations, and discusses their limitations for pricing contexts.
- **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 for the differents PDEs and instruments shown in the previous sections. The performance of the PINNs is analyzed, from precision and speed perspectives.

- **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 pricing problems.

2. THEORETICAL BACKGROUND

This chapter provides the foundational concepts necessary for understanding the pricing of financial products, the mathematical tools traditionally used for their valuation. 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, or payments, are linked to the value of one or several *underlyings* —shares, commodities, foreign-exchange rates, or interest rates [2], [3]. This instruments let market participants insure portfolios, manage exposure to financial risks or express directional views.

The starting point for almost every derivatives textbook is the European call or put option. At one hand, the European call delivers, at a fixed future date T, the right to buy the underlying at the pre-agreed strike K price. At the other, the European put grants the right to sell the underlying asset. The payoffs functions

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

describe the optionality and depend only on the future spot price S_T. Valuing the contract amounts to forecasting the *distribution* of S_T under a special probability under wich the counterpaties in the transacction are able to eliminate, or *hedge*, the risks associated with movements of the underlying asset. The Black-Scholes formula achieves this under a set of assumptions, for example that the underlying follows a geometric Brownian motion with constant drift and volatility. Under these assumptions, the call price is given by the closed-form expression

$$C(S_t, K, T) = S_t N(d_1) - K e^{-r(T-t)} N(d_2),$$

where N is the cumulative distribution function of the standard normal distribution, r is the risk-free interest rate, and

$$d_1 = \frac{\ln(S_t/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, \qquad d_2 = d_1 - \sigma\sqrt{T-t}.$$

The Black-Scholes model is a cornerstone of modern finance, but its assumption of constant volatility is often too simplistic for real-world markets. Observations show that volatility frequently varies over time, prompting practitioners to enhance the basic model

by allowing volatility itself to be random or dynamic [8]. Although these modifications make the pricing equations more realistic, they no longer yield simple closed-form solutions, thus requiring numerical methods—such as finite differences, Monte Carlo simulations, Fourier integrals, or, as this thesis investigates, Physics-Informed Neural Networks (PINNs).

However, not all financial derivatives fit neatly into this framework. For instance, American options differ from European ones by allowing the holder to exercise at any time up to expiry, adding complexity as the optimal timing of exercise becomes part of the pricing problem. Technically, valuation transforms into a free-boundary problem, where one needs to determine both the option price and the dynamic boundary dividing hold and exercise regions. Traditional numerical approaches—tree methods, finite-difference schemes, or regression-based Monte Carlo—tend to struggle with either slow convergence or high variance.

Similarly, basket options —which derive value from multiple underlying assets— introduce another layer of complexity. Such options, commonly used in structured products, depend heavily on the joint behavior of the underlyings. The payoff may, for instance, be based only on the best-performing asset in a basket, making valuation challenging due to the exponential growth in complexity as more assets are included. This "curse of dimensionality" makes traditional finite-difference methods impractical and Monte Carlo simulations noisy, further motivating advanced numerical techniques such as high-dimensional PINNs.

Complexity continues to increase with *long-dated equity or FX options*, which are sensitive not only to the underlying asset price but also to volatility dynamics and varying interest rates. To tackle this, one possible way is to combine stochastic-volatility models, capturing realistic volatility dynamics, with interest-rate models to discount future cash flows accurately. Individually manageable, these factors collectively create intricate, multidimensional valuation problems lacking straightforward analytic solutions.

Moreover, unlike equity or FX derivatives tied to a single traded asset, *interest-rate derivatives* such as *payer swaptions* depend on the evolution of an entire yield curve. A payer swaption grants its holder the right at future date T to lock in a fixed rate K on an interest-rate swap spanning from T to some final maturity T_n. At expiry, if the prevailing market swap rate $L(T)$ exceeds K, the option is exercised; otherwise, it expires worthless. The payoff essentially represents the present value of future fixed-versus-floating payments, scaled by $\max(L(T) - K, 0)$. Pricing thus necessitates a model that simultaneously fits current discount curves and realistically forecasts their evolution. Simplified models such as the one-factor Hull-White approach partly achieve this balance, yet closed-form solutions quickly vanish once real-world constraints (e.g., callability or collateral requirements) are introduced, compelling numerical integration or flexible numerical methods. PINNs emerge as attractive alternatives, seamlessly integrating curve-fitting and pricing within a single computational framework.

From the plain European call to complex structures such as American options, basket contracts, and swaptions, each derivative type introduces incremental complexity: multiple underlyings, additional risk factors, early-exercise features, and yield curve dependencies, among others. Collectively, these complexities erode the clean closed-form solutions of classical theory and push traditional numerical methods beyond practicality. By contrast, PINNs present themselves as a mesh-free, dimension-agnostic alternative that naturally incorporates boundary or inequality constraints through their loss functions. The remainder of this thesis systematically quantifies how these advantageous properties translate into gains in computational speed, flexibility, and accuracy across five representative derivative contracts [9], [10].

2.2. Modelling Framework

Financial derivatives can be valued from two complementary angles. On the one hand, the price is the discounted expectation of the payoff under a risk-neutral probability measure $\mathbb{Q}$; on the other, it is the unique classical solution of a partial differential equation (PDE). The multidimensional Feynman-Kac formula establishes the bridge between the stochastic and the analytic representations: once the Markov dynamics of the state vector have been specified, it converts a conditional expectation into a PDE—and vice-versa—at virtually no cost.

This equivalence is the backbone of the chapter. We will first record the general Feynman-Kac statement and then use it as a template to derive the PDEs associated with increasingly rich models: the Black-Scholes geometric Brownian motion, its correlated multi-asset version, and several extensions that incorporate stochastic volatility and stochastic interest rates.

2.2.1. Feynman-Kac Representation

A fundamental result connecting PDEs to stochastic processes is the (multidimensional) Feynman-Kac theorem, which is presented here in a financial context. Under standard regularity assumptions, the value function $u : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ that solves

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i(t,\mathbf{x})\sigma_j(t,\mathbf{x})x_i x_j\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d}\mu_i(t,\mathbf{x})x_i\frac{\partial u}{\partial x_i} - r_t u = 0, \\ u(T,\mathbf{x}) = \Phi(\mathbf{x}), \end{aligned} \tag{2.1}$$

admits the probabilistic representation

$$u(t,\mathbf{x}) = \mathbb{E}^{\mathbb{Q}}\Big[e^{-\int_t^T r_s \mathrm{d}s}\Phi(\mathbf{X}_T)\Big|\mathbf{X}_t = \mathbf{x}\Big], \tag{2.2}$$

where the state process $\mathbf{X}_t = (X_t^{(1)}, \ldots, X_t^{(d)})$ follows the correlated geometric Itô diffusion

$$\mathrm{d}X_t^{(i)} = \mu_i(t,\mathbf{X}_t)X_t^{(i)}\mathrm{d}t + \sigma_i(t,\mathbf{X}_t)X_t^{(i)}\mathrm{d}W_t^{(i)}, \qquad i = 1,\ldots,d, \tag{2.3}$$

with quadratic covariations $\mathrm{d}W_t^{(i)}\mathrm{d}W_t^{(j)} = \rho_{ij}\mathrm{d}t$. Here μ_i and σ_i are (possibly state- and time-dependent) drift and volatility functions, r_t is the short-rate process, ρ is the constant correlation matrix, and Φ specifies the terminal payoff at maturity T.

2.2.2. The Black-Scholes Model

We begin by briefly reviewing the framework proposed by Black and Scholes [11]. They assumed the underlying asset price $(S_t)_{t\geq 0}$ evolves according to a geometric Brownian motion described by the stochastic differential equation (SDE):

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t^{\mathbb{P}}, \quad S_0 > 0, \tag{2.4}$$

where μ denotes the expected return under the real-world measure $\mathbb{P}$, and σ is the asset volatility.

The key insight behind derivative pricing is the construction of a hedged portfolio:

$$\Pi_t = V(t, S_t) - \Delta_t S_t,$$

where $V(t, S_t)$ is the derivative price, and Δ_t denotes the quantity of the underlying asset held. By applying Itô's lemma, it can be shown that choosing

$$\Delta_t = \frac{\partial V}{\partial S}$$

makes the portfolio instantaneously risk-free. Consequently, it must earn the risk-free rate r, leading to the well-known *Black-Scholes PDE*:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{2.5}$$

with terminal condition $V(T, S) = \Phi(S)$ and all partial derivatives are evaluated at (t, S_t). The Feyman Kac theorem in the context of the Black-Scholes setting reduces to:

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

where the risk-neutral dynamics simplify to

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}W_t^{\mathbb{Q}}.$$

Boundary Conditions

To solve the PDE for specific derivatives, appropriate boundary conditions must be defined. For example, a European call option payoff, $\Phi(S) = \max(S - K, 0)$, implies the following conditions:

- **Terminal condition:** At maturity T,

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

- **Lower boundary** ($S \to 0$)**:** As the asset price approaches zero, is safe to assume that for a call option
$$V(t, 0) = 0,$$
but a more general approach is to set the value of the option to be independent of the underlying, resulting in:
$$\frac{\partial V}{\partial t} - rV = 0.$$
- **Upper boundary** ($S \to \infty$)**:** For large underlying asset prices, the option price grows approximately linearly with the asset price, leading to:
$$\frac{\partial^2 V}{\partial S^2} \to 0, \quad S \to \infty.$$

Boundary conditions for other derivatives can be found in standard references such as [3].

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

Now we proceed to extend the model for multiple underlying assets. Following the same steps from the previous section, we derive that the underlying assets under the risk neutral mesuare 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

$$\mathrm{d}S_t^i = rS_t^i \mathrm{d}t + \sigma_i S_t^i \mathrm{d}W_t^{\mathbb{Q},i}, \qquad \mathrm{d}W_t^{\mathbb{Q},i} \mathrm{d}W_t^{\mathbb{Q},j} = \rho_{ij} \mathrm{d}t, \quad 1 \leq i, j \leq d. \tag{2.6}$$

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

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial}{\partial S_i S_j} + r \sum_{i=1}^{d} S_i \partial_{S_i} - rV = 0. \tag{2.7}$$

Dimensionless Backward Equation

As the goal is to price using PINNs, is that the inital weights of the network are proportional to the input variables to avoid numerical instabilities. Its possible to achieve this by changing variables to dimensionless quantities. Formally, for maturity T define

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

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

$$\frac{\partial u}{\partial \tau} = \frac{1}{2} \sum_{i=1}^{d} \sigma_i^2 \Big(\frac{\partial^2 u}{\partial x_i^2} - \frac{\partial u}{\partial x_i} \Big) + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sigma_i \sigma_j \rho_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + r \sum_{i=1}^{d} \frac{\partial u}{\partial x_i} - ru. \tag{2.8}$$

to be solved for $\tau \in (0, T]$ and $\mathbf{x} \in \mathbb{R}^d$ with terminal condition $u(0, \mathbf{x}) = \Phi(\mathbf{x})/K$. Equation (2.8) will serve as the reference problem for PINN solver in Section 2.2.3.

Boundary Conditions

To solve the multi-asset case described adove via the PDE (2.8) is a significant challenge due to the complexity of the boundary conditions. This conditions appear as an extension of the single-asset case and are explained below.

- **Terminal condition:** At maturity T, the option value is given by the payoff function of the dimensionless problem:

$$u(0, \mathbf{x}) = \Phi(\mathbf{x})/K,$$

where Φ is the payoff function of the option. In this thesis we will focus on the case of the best-of-basket option, which has the payoff

$$\Phi(\mathbf{x}) = \max(\max_{1\leq k\leq d} e^{x_k} - 1, 0).$$

- **Lower boundaries ($S_k \to 0$):** As the underlying asset price S_k approaches zero, the option behaves as the $d-1$-asset case. This is equivalent to solving (2.8) but removing the asset S_k from the problem. Mathematically, the boundary condition is given by:

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

- **Upper boundaries ($S_k \to \infty$):** In the limit where the underlying asset price S_k becomes very large, we also want the option to behave linearly with respect to the underlying asset price S_k. As now we are working with dimensionless variables, we need to ensure that $\frac{\partial^2 V}{\partial S_k^2} = 0$ is appropiately scaled. The resulting condition is

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sum_{\substack{i=1\\ i\neq k}}^{d} \sigma_i^2\left(\frac{\partial^2 u}{\partial x_i^2} - \frac{\partial u}{\partial x_i}\right) + \frac{1}{2}\sum_{\substack{i=1\\ i\neq k}}^{d}\sum_{\substack{j=1\\ j\neq k}}^{d} \sigma_i\sigma_j\rho_{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + r\sum_{i=1}^{d}\frac{\partial u}{\partial x_i} - ru.$$

2.2.4. Stochastic Volatility Extension

As previously discussed, the Black-Scholes assumption of constant volatility and deterministic interest rates is often insufficient for capturing market realities, especially for longer-dated equity and foreign-exchange options. Market-implied volatilities typically vary by strike price and maturity, and the yield curve itself fluctuates in response to economic factors and monetary policy. To realistically address these dynamics, practitioners extend the standard framework by introducing *stochastic volatility* and *stochastic interest rates*. These enhancements allow models to better match observed market prices and effectively capture the joint sensitivity of derivatives to underlying asset levels, volatility changes, and interest-rate movements.

There are several approaches to derive the PDE we are interested in. Here we will proceed with the hedging argument so in later sections we can use the Feynman-Kac theorem to obtain the probabilistic representation of the solution.

Starting in the physical measure $\mathbb{P}$, assume

$$\begin{aligned} \mathrm{d}S_t &= \mu S_t \mathrm{d}t + \sqrt{v_t} S_t \mathrm{d}W_t^{\mathbb{P},S}, \\ \mathrm{d}v_t &= \kappa(\theta - v_t)\mathrm{d}t + \sigma_v \sqrt{v_t}\mathrm{d}W_t^{\mathbb{P},v}, \\ \mathrm{d}W_t^{\mathbb{P},S}\mathrm{d}W_t^{\mathbb{P},v} &= \rho \mathrm{d}t, \end{aligned} \tag{2.9}$$

with μ the physical drift of the stock, κ the mean-reversion speed of variance and θ its long-run level. Let $V(t,S,v)$ be the value of the contract we wish to price and $\widehat{V}(t,S,v)$ the value of a second traded option chosen for its sensitivity to v. Construct the self-financing portfolio

$$\Pi_t = V_t - \Delta_t S_t - \Phi_t \widehat{V}_t, \tag{2.10}$$

where Δ_t and Φ_t denote (short) positions in the underlying and in the auxiliary option, respectively. The diffusion of the portfolio is given by

$$\mathrm{d}\Pi_t = \mathrm{d}V_t - \Delta_t \mathrm{d}S_t - \Phi_t \mathrm{d}\widehat{V}_t, \tag{2.11}$$

and we proceed by analizing each part. First, applying Itô's lemma to V and $\widehat{V}$ and substituting (2.9) yields

$$\begin{aligned} \mathrm{d}V &= \Big(\frac{\partial V}{\partial t} + \mathcal{L}^{\mathbb{P}} V\Big)\mathrm{d}t + \frac{\partial V}{\partial S}\sqrt{v}S\,\mathrm{d}W_t^{\mathbb{P},S} + \frac{\partial V}{\partial v}\sigma_v \sqrt{v}\mathrm{d}W_t^{\mathbb{P},v}, \\ \mathrm{d}\widehat{V} &= \Big(\frac{\partial \widehat{V}}{\partial t} + \mathcal{L}^{\mathbb{P}} \widehat{V}\Big)\mathrm{d}t + \frac{\partial \widehat{V}}{\partial S}\sqrt{v}S\,\mathrm{d}W_t^{\mathbb{P},S} + \frac{\partial \widehat{V}}{\partial v}\sigma_v \sqrt{v}\mathrm{d}W_t^{\mathbb{P},v}, \end{aligned} \tag{2.12}$$

where

$$\mathcal{L}^{\mathbb{P}} = \mu S \frac{\partial}{\partial S} + \kappa(\theta - v)\frac{\partial}{\partial v} + \frac{1}{2}vS^2\frac{\partial^2}{\partial S^2} + \rho\sigma_v v S \frac{\partial^2}{\partial S\,\partial v} + \frac{1}{2}\sigma_v^2 v \frac{\partial^2}{\partial v^2}.$$

Replacing (2.12) into (2.11) and rearanging, we get the stochastic part of $\mathrm{d}\Pi_t$ is

$$\Big[\frac{\partial V}{\partial S} - \Delta_t - \Phi_t \frac{\partial \widehat{V}}{\partial S}\Big]\sqrt{v}S\,\mathrm{d}W_t^{\mathbb{P},S} + \Big[\frac{\partial V}{\partial v} - \Phi_t \frac{\partial \widehat{V}}{\partial v}\Big]\sigma_v \sqrt{v}\mathrm{d}W_t^{\mathbb{P},v}.$$

Setting each coefficient to zero eliminates both Brownian terms. From the $\mathrm{d}W_t^{\mathbb{P},v}$ term we obtain

$$\Phi_t = \frac{\partial V/\partial v}{\partial \widehat{V}/\partial v}, \tag{2.13}$$

and from the $\mathrm{d}W_t^{\mathbb{P},S}$ term

$$\Delta_t = \frac{\partial V}{\partial S} - \Phi_t \frac{\partial \widehat{V}}{\partial S}. \tag{2.14}$$

With (2.13)-(2.14) the diffusion in Π_t is zero, so the drift of $\mathrm{d}\Pi_t$ must equal the risk-free rate:

$$\mathrm{d}\Pi_t = r(V_t - \Delta_t S_t - \Phi_t \widehat{V}_t)\mathrm{d}t. \tag{2.15}$$

After substitution of (2.13), (2.14), (2.12) and (2.15) into (2.11) and after some algebra we obtain the relation:

$$\frac{\frac{\partial V}{\partial t} + \tilde{\mathcal{L}}V - rV}{\frac{\partial V}{\partial v}} = \frac{\frac{\partial \widehat{V}}{\partial t} + \tilde{\mathcal{L}}\widehat{V} - r\widehat{V}}{\frac{\partial \widehat{V}}{\partial v}}, \tag{2.16}$$

where $\tilde{\mathcal{L}}$ is the generator

$$\tilde{\mathcal{L}} = rS\frac{\partial}{\partial S} + \kappa(\theta - v)\frac{\partial}{\partial v} + \frac{1}{2}vS^2\frac{\partial^2}{\partial S^2} + \rho\sigma_v vS\frac{\partial^2}{\partial S\,\partial v} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2}{\partial v^2}.$$

Notice that the main difference between $\mathcal{L}^{\mathbb{P}}$ and $\tilde{\mathcal{L}}$ is that the drift of the underlying asset has been replaced by the risk-free rate r.

Relation (2.16) must hold for *every* pair of sufficiently smooth pay-offs $(V, \widehat{V})$, so the common value of the two quotients can depend *only* on the state variables (t, S, v)—not on the particular contract. We therefore introduce the *(instantaneous) market price of volatility risk*

$$\lambda(t, S, v) := \frac{\frac{\partial \widehat{V}}{\partial t} + \tilde{\mathcal{L}}\widehat{V} - r\widehat{V}}{\frac{\partial \widehat{V}}{\partial v}}$$

Substituting this definition back into the numerator of (2.16) yields

$$\frac{\partial V}{\partial t} + \tilde{\mathcal{L}}V - rV = \lambda(t, S, v)\frac{\partial V}{\partial v}.$$

Collecting terms, we obtain the *risk-neutral* generator

$$\mathcal{L}^{\mathbb{Q}} = rS\frac{\partial}{\partial S} + \big[\kappa(\theta - v) - \lambda(t, S, v)\big]\frac{\partial}{\partial v} + \frac{1}{2}vS^2\frac{\partial^2}{\partial S^2} + \rho\sigma_v vS\frac{\partial^2}{\partial S\,\partial v} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2}{\partial v^2}.$$

Hence *every* European-style payoff $V(t, S, v)$ satisfies the pricing PDE

$$\frac{\partial V}{\partial t} + \mathcal{L}^{\mathbb{Q}}V - rV = 0, \qquad V(T, S, v) = \Phi(S) \tag{2.17}$$

under the risk-neutral measure $\mathbb{Q}$. It is custumary to assume that investors are *not* compensated for bearing volatility risk, i.e. $\lambda(t, S, v) \equiv 0$. Then, the risk-neutral PDE is given by

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

Using the Feynman-Kac theorem, the unique solution to (2.18) is given by the expectation

$$V(t, S, v) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}\Big[\Phi(S_T) \;\Big|\; S_t = S, v_t = v\Big], \tag{2.19}$$

where $(S_t, v_t)_{t\in[0,T]}$ evolves according to the SDE

$$\begin{aligned} \mathrm{d}S_t &= rS_t\mathrm{d}t + \sqrt{v_t}S_t\mathrm{d}W_t^{\mathbb{Q},S}, \\ \mathrm{d}v_t &= \kappa(\theta - v_t)\mathrm{d}t + \sigma_v\sqrt{v_t}\mathrm{d}W_t^{\mathbb{Q},v}, \\ \mathrm{d}W_t^{\mathbb{Q},S}\,\mathrm{d}W_t^{\mathbb{Q},v} &= \rho\mathrm{d}t, \end{aligned} \tag{2.20}$$

where each parameter is defined as in Section 2.2.4.

Because (S_t, v_t) is a two-dimensional Itô diffusion under $\mathbb{Q}$, the Feynman-Kac formula applies without further hypotheses: it guarantees that the unique bounded solution to (2.18) can be written as the discounted expectation in (2.19).

In other words, once the risk-neutral SDEs (2.20) have been fixed, the PDE follows automatically—no replication or delta- hedging argument is needed. This shortcut will prove invaluable when other extensions are introduced in the next sections.

Boundary Conditions

When including the assumption of stochastic volatility, we need to specify the boundary conditions for small and large values of the volatility. In general, practitioners specify [12]:

- **Lower boundary** $v \to 0$**:** As the volatility v approaches zero, there is no uncertainty in the underlying, therefore, we can set

$$V(t, S, 0) = e^{-(T-t)r}\Phi(S_T), \quad S_T = S_t e^{(T-t)r}.$$

- **Upper boundary** $v \to \infty$**:** As the volatility v becomes very large, the option price can move at most proportial to the underlying asset price S. This translates into the condition

$$\frac{\partial V}{\partial v} = S, \quad v \to \infty.$$

2.2.5. Stochastic Interest Rates Extension

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 [13] 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 [14] and the *Hull-White* model [15]. Both models assume that the short rate follows a Ornstein-Uhlenbeck, 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, directly in the risk-neutral measure $\mathbb{Q}$, 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 $\theta(t)$ is the long-run mean. In the Vasicek model, $\theta(t)$ 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

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

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 [13] for a more detailed discussion on affine models.

Boundary Conditions

As before, now we need to include boundary conditions for the short rate dimension. Again using as reference [12], we have:

- $r \to -\infty$ **and** $r \to \infty$ For very high and low values of the short rate, the option becomes insesitive to the short rate, and we can set

$$\frac{\partial V}{\partial r} = 0, \quad r \to \pm\infty.$$

2.2.6. 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 both factors as source of randomness. In this case, the joint dynamics of the state vector (S_t, v_t, r_t) under the risk-neutral measure $\mathbb{Q}$ are given by the SDEs

$$\begin{aligned}
\mathrm{d}S_t &= r_t S_t \mathrm{d}t + \sqrt{v_t} S_t \mathrm{d}W_t^{\mathbb{Q},S}, \\
\mathrm{d}v_t &= \kappa_v(\theta_v - v_t)\mathrm{d}t + \sigma_v \sqrt{v_t}\mathrm{d}W_t^{\mathbb{Q},v}, \\
\mathrm{d}r_t &= \kappa_r(\theta_r(t) - r_t)\mathrm{d}t + \sigma_r \mathrm{d}W_t^{\mathbb{Q},r}, \\
\mathrm{d}W_t^{\mathbb{Q},S}\mathrm{d}W_t^{\mathbb{Q},v} &= \rho_{SV}\mathrm{d}t, \\
\mathrm{d}W_t^{\mathbb{Q},S}\mathrm{d}W_t^{\mathbb{Q},r} &= \rho_{SR}\mathrm{d}t, \\
\mathrm{d}W_t^{\mathbb{Q},v}\mathrm{d}W_t^{\mathbb{Q},r} &= \rho_{VR}\mathrm{d}t,
\end{aligned} \tag{2.23}$$

where $\rho_{SV}, \rho_{SR}, \rho_{VR} \in [-1, 1]$ are the instantaneous correlations between the Brownian motions driving the dynamics of the underlying asset, the volatility, and the short rate, respectively. The corresponding pricing PDE for a derivative whose value depends on the

state vector (S_t, v_t, r_t) is

$$\begin{aligned}
&\frac{\partial V}{\partial t} + \kappa_r(\theta_r(t) - r)\frac{\partial V}{\partial r} + \kappa_v(\theta_v - v)\frac{\partial V}{\partial v} + rS\frac{\partial V}{\partial S} - rV \\
&\quad + \rho_{SV}\sigma_v vS\,\frac{\partial^2 V}{\partial S\,\partial v} + \rho_{SR}\sigma_r S\,\sqrt{v}\frac{\partial^2 V}{\partial S\,\partial r} + \rho_{VR}\sigma_v\sigma_r\,\sqrt{v}\frac{\partial^2 V}{\partial v\partial r} \\
&\quad + \frac{1}{2}\sigma_r^2\frac{\partial^2 V}{\partial r^2} + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2 V}{\partial v^2} = 0.
\end{aligned} \tag{2.24}$$

As no new dimensions are added compared to the previous sections, the boundary conditions are the same as in Section 2.2.4 and 2.2.5.

2.3. Traditional Numerical Approaches

Overview of Finite Difference Methods

Finite Difference Methods (FDM) are among the most widely used techniques for numerically solving partial differential equations such as (2.5) or its multi-dimensional counterpart (2.6). 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 the curse of dimensionality. 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]. When pricing derivatives involving multiple risk factors or assets, 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

Input: Number of simulations N, time horizon T, risk-free rate r, strike K, asset parameters $(\mu_i, \sigma_i, \rho_{ij})$
for $n = 1$ to N **do**
    Simulate one path for each asset S_T^i using correlated Brownian motion
    Compute the path-dependent payoff: $\text{Payoff}^{(n)} = \max\left(\max_i S_T^i - K, 0\right)$
end for
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 early excersice or path-dependent derivatives, as they require the ability to exercise at multiple points in time or have information of the path realization of the underlying asset. While there are techniques to adapt MC methods for American options, such as the Longstaff-Schwartz method [16], they are often less efficient and precise 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 arising from the derivative characteristics or the dimensionality of the problem. PINNs, introduced by Raissi et al. [7], provide an innovative framework that explicitly incorporates governing PDEs directly into neural network training. Rather than relying solely on solution data, PINNs leverage physical laws and constraints embedded within the problem formulation to enhance predictive accuracy, efficiency, and reliability of the solutions.

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:

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

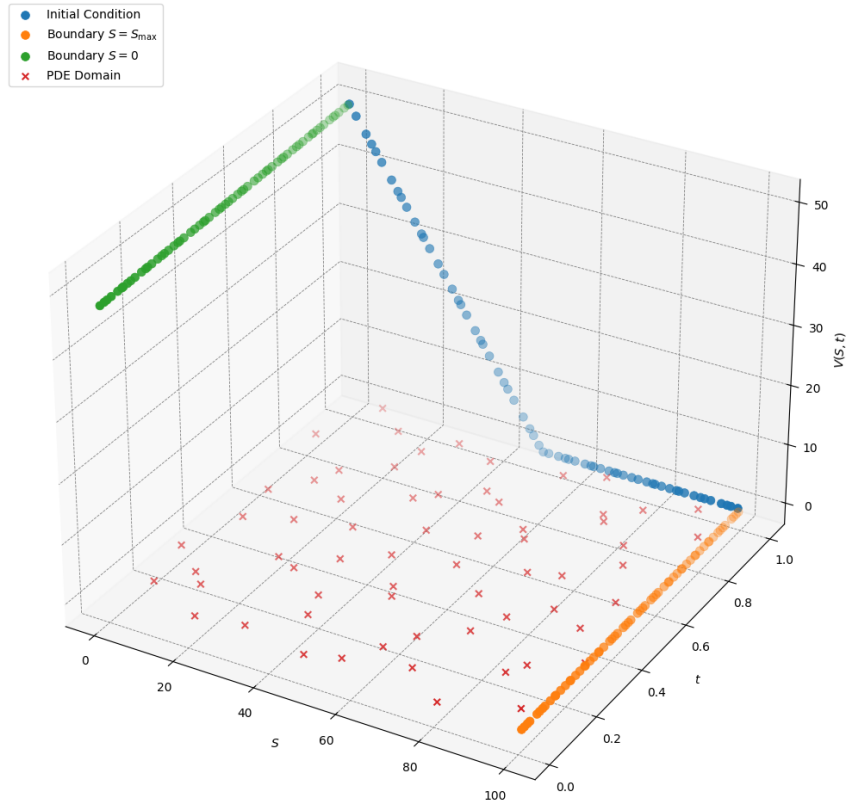
These integrals are typically approximated by Monte Carlo sampling, choosing discrete collocation points within the respective domains.

3.2. Training PINNs

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

Figure 3.1

Collocation points for a put option. The collocation points are sampled from the interior of the domain Ω, the boundary $\partial\Omega$ and the initial condition Ω_0.



based on the computed gradients of the loss function with respect to the parameters. Another alternative is to use quasi-Newton methods, such as the L-BFGS algorithm, which provide an approximation of the Hessian matrix. The problem with this optimizers is that they are very suceptible to conditioning of the Hessian matrix, which can lead to slow convergence or even divergence [17].

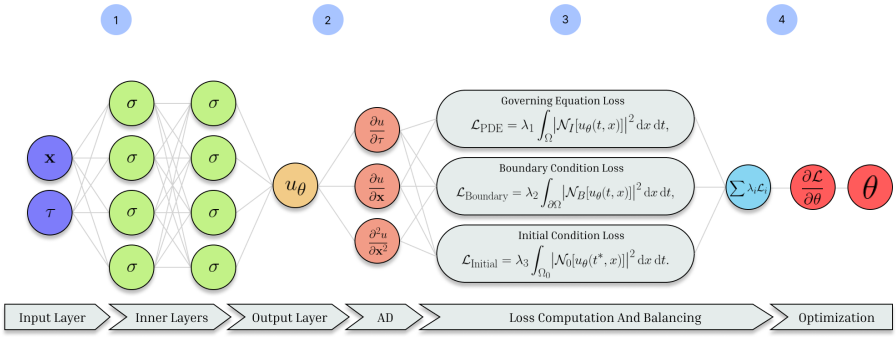
In general, the gradients used by this optimization methods are computed using automatic differentiation, which allows for efficient and accurate calculation of the gradients of the loss function with respect to the parameters. The optimization process continues until a stopping criterion is met, such 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.

Before training, it is common to initialize the neural network parameters θ randomly or using a pre-trained model. This step is key as it can significantly affect the convergence speed and the final solution obtained by the PINN. If no special initialization is used, the

Figure 3.2

Schema of the training procedure of PINNs. (1) Collocation points are feed into the neural, then (2) gradients of the solution are computed using automatic differentiation, (3) the PDE residuals are computed at the collocation points, and finally (4) the parameters of the neural network are updated using an optimization algorithm. This resulting losses are added together to obtain the total loss $\mathcal{L}(\theta)$, which is then used to compute the gradients of the parameters θ. Finally, the parameters are updated using an optimization algorithm. This process is repeated until convergence.



parameters are usually initialized using a random normal distribution with mean zero and a small standard deviation, this is why dimensionless formulation of the PDEs is so important, as it allows for a more stable training process [18].

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**

Fine-tuning the training process is no simple task, as there are multiple parts that can be adjusted to improve the performance of the PINN. In the next section a breaf discussion of the selected setup for this thesis is presented, which will be further detailed in the implementation chapter.

4. IMPLEMENTATION AND RESULTS

4.1. Implementation Details

Most of the results presented in this thesis use relatively small feedforward neural networks. These smaller architectures provided an optimal balance between computational efficiency and approximation accuracy. Regarding activation functions, we primarily employed the hyperbolic tangent (*tanh*), given its smooth and differentiable nature. Although we also experimented with the Softplus activation—which closely resembles typical payoff functions—we observed only marginal improvements in convergence speed, insufficient to justify its use over *tanh*.

In terms of optimization methods, quasi-Newton algorithms, particularly the L-BFGS solver implemented in PyTorch, consistently delivered the best outcomes. Other solvers such as BFGS and SSBroyden were also implemented and tested, and in some cases yielded better results. Gradient-based optimizers, notably Adam, were evaluated but generally failed to match the accuracy or convergence speed achieved by quasi-Newton approaches.

Sampling collocation points was predominantly carried out using a uniform distribution over the domain of interest to evaluate PDE residuals and boundary conditions. Alternative sampling strategies, including Sobol and Halton sequences, were tested but did not lead to significant improvements in accuracy or computational efficiency.

All models were implemented in Python using the PyTorch framework. The complete implementation is publicly available on GitHub [19]. Computations were performed on a personal computer equipped with an Intel i7-12700K CPU and an NVIDIA RTX 2070 GPU, facilitating efficient training of neural network models.

4.2. Call Option

As an initial test case for the methodology, we priced a standard European call option under the classical Black-Scholes model, whose PDE in Eq. (2.5) admits a closed-form solution that serves as a convenient benchmark. The numerical experiment used the parameters in Table 4.1.

Parameter	Value
Strike price, K	\$100
Maturity, T	1.0 year
Risk-free rate, r	0.05
Volatility, σ	0.20

Table 4.1
Call Option Parameters

The physics-informed neural network (PINN) consisted of two hidden layers with 20 neurons each and was trained with the BFGS optimizer on 75000 collocation points distributed across the interior domain Ω, the boundary $\partial\Omega$, and the initial hypersurface Ω_0. Despite its simple architecture, the network reproduced the analytical solution accurately, achieving an L_2 error of 1.5054×10^{-5}.

Figure 4.1
Call option price under the Black-Scholes PDE. The PINN solution closely matches the analytical benchmark.

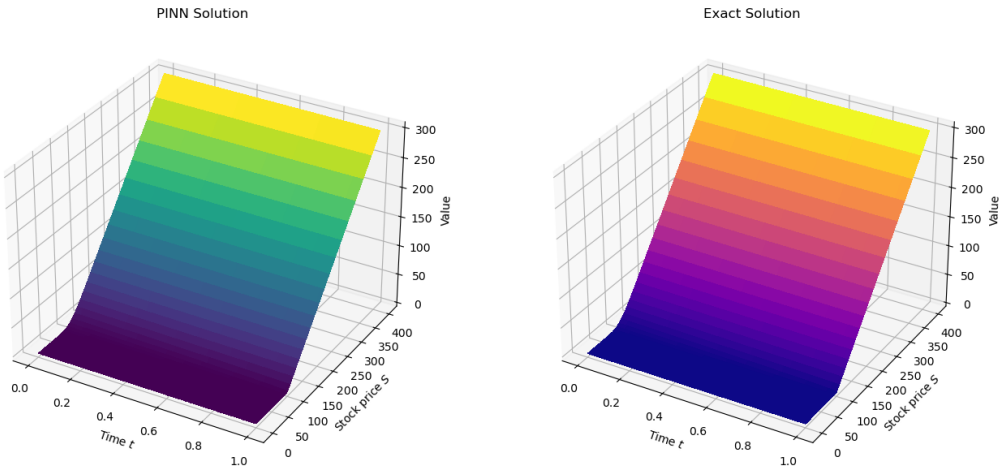
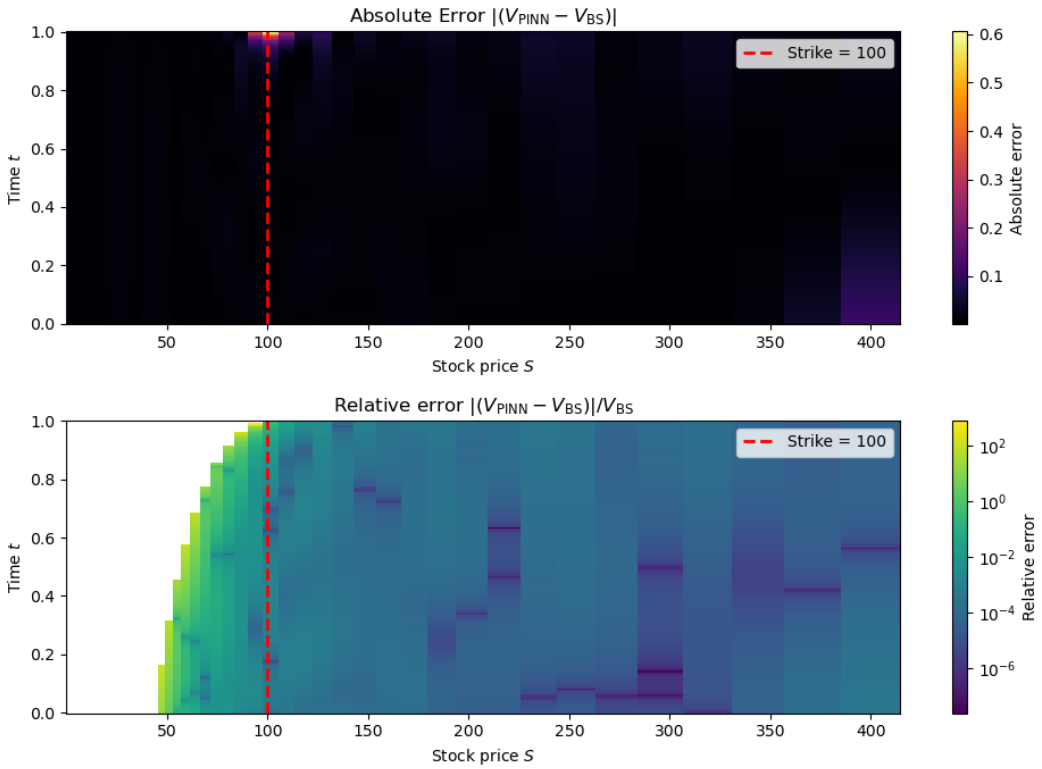


Figure 4.1 shows that the learned pricing surface is visually indistinguishable from the Black-Scholes benchmark, while Figure 4.2 depicts the absolute error, which is largest near maturity and the strike.

Figure 4.2

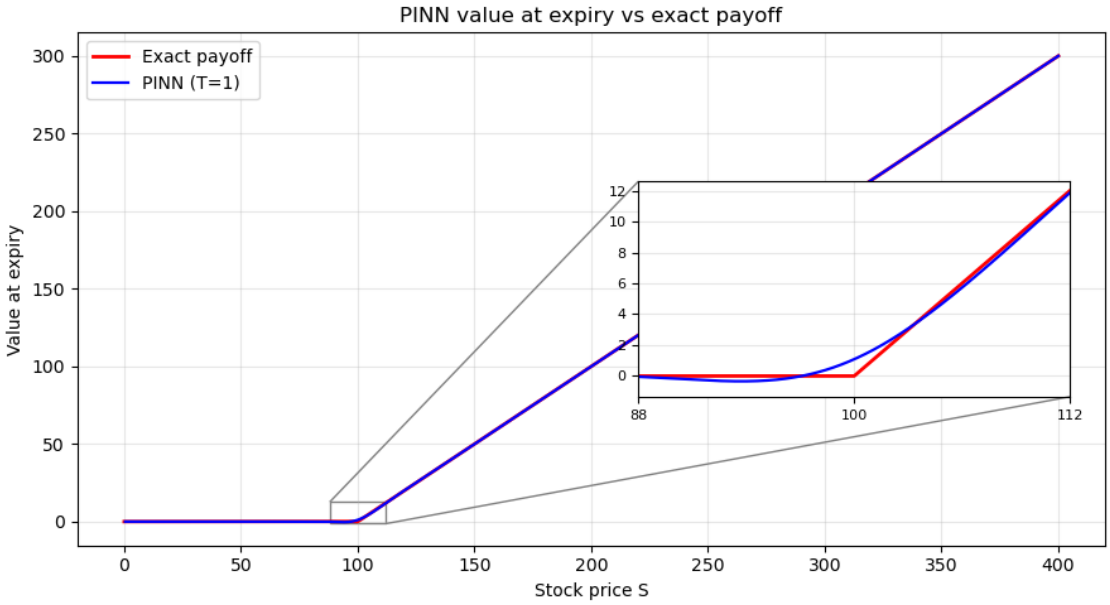
Absolute error of the PINN solution. Errors peak close to maturity and the strike price.



The pronounced spike in error around the strike at expiry originates from the kink in the payoff function. Approximating a function with a discontinuous first derivative by smooth basis functions (including neural networks) typically produces oscillatory overshoots near the non-smooth point; this behaviour is a manifestation of the *Gibbs phenomenon*.

Figure 4.3

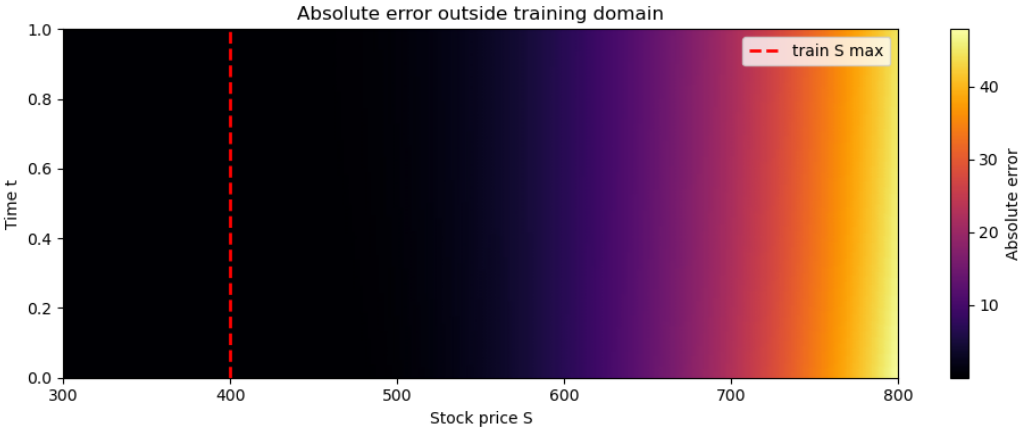
Solution at expiry. The PINN solution closely matches the analytical benchmark, except around the strike price where the absolute error peaks.



The relative error in Figure 4.4 highlights a limitation common to data-driven methods: reliability falls off rapidly when the model is evaluated outside the region covered by training points. Within the training domain, however, the accuracy is sufficient for professional use.

Figure 4.4

Relative error of the PINN solution. The error is concentrated around the strike price and maturity.



Even for this straightforward example, training required about two minutes, underscoring the computational burden that PINNs impose. Once trained, evaluation is fast—in our test, roughly 1.45× faster than the closed-form Black-Scholes formula—but the initial training cost remains a practical consideration.

4.2.1. Multi-Asset Option

To test the ability of PINNs to solve higher-dimensional problems, we extend the single-asset experiment to multiple underlying assets. A compact network with three hidden layers of ten neurons each is first trained on the dimensionless form of the multi-asset Black-Scholes PDE in Eq. (2.8).

Because training costs increase significantly with dimensionality—due both to the complexity of the boundary conditions and the exponential growth in the volume of the domain—we employ a deliberately small set of collocation points: 500 in the interior domain Ω, 500 on the boundary $\partial\Omega$, and 500 on the initial hypersurface Ω_0. It is important to note that the number of required points on the boundary increases linearly with the number of assets, which further contributes to the computational burden as dimensionality grows. The specific parameters used in this test case are summarized in Table 4.2.

Parameter	**Value**
Strike price, K	\$100
Maturity, T	1.0 year
Risk-free rate, r	0.05
Volatilities, σ_i	0.20
Correlation, ρ_{ij}	0.25

Table 4.2
Multi-Asset Call Option Parameters

The surfaces obtained from the trained PINN, shown in Figures 4.5 and 4.6, correspond to the first two asset dimensions in the dimensionless and original coordinate spaces, respectively.

As in the single-asset case, the solution displays the expected features: values near zero when asset prices are far below the strike, approximately linear growth as asset prices increase well above the strike, and a ridge or kink near the strike region. These qualitative features are preserved as dimensionality increases, but the quality of the approximation deteriorates. The surfaces reveal increased irregularity and loss of precision, particularly in regions near the domain boundaries.

Figure 4.5

Dimensionless multi-asset call option price (first two asset axes).

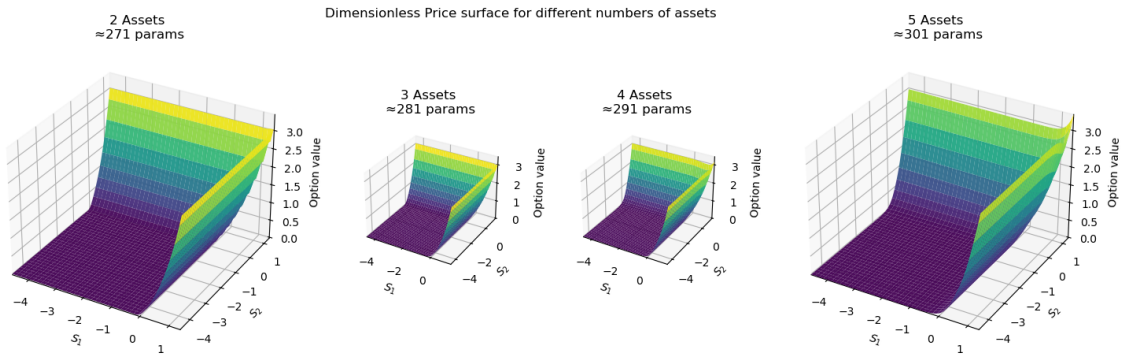
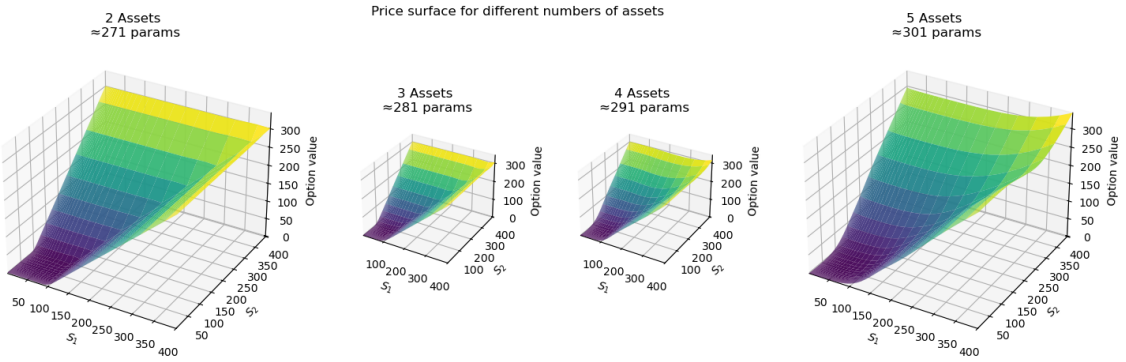


Figure 4.6

Multi-asset call option price in original coordinates (first two asset axes).

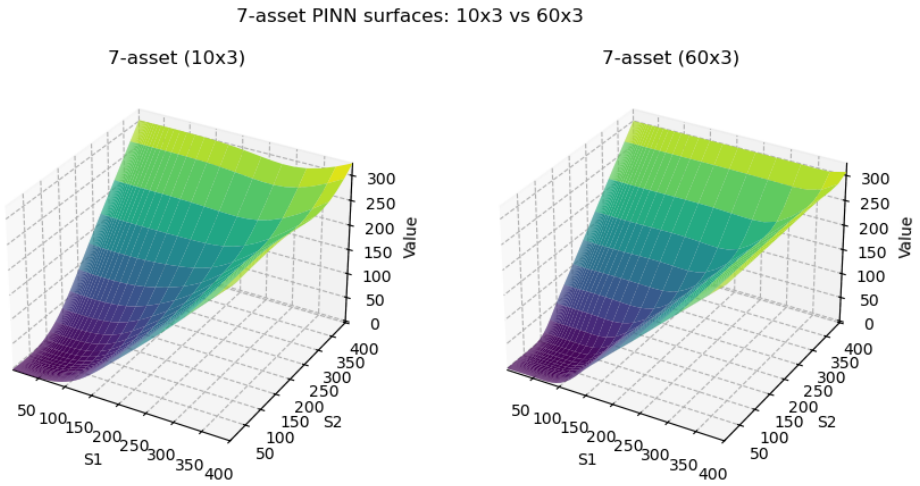


At the boundaries of the domain, especially in cases with more than three assets, the solution begins to exhibit an undesired curvature because the small network struggles to capture accurately the solution. In our experiments, increasing the number of boundary collocation points did not lead to significant improvements. Instead, better accuracy was achieved by increasing the expressiveness of the network.

By retaining the same number of layers but increasing the width to 60 neurons per layer, the network was able to represent the solution more faithfully, as demonstrated in Figure 4.7, which compares both architectures in a seven-asset configuration.

Figure 4.7

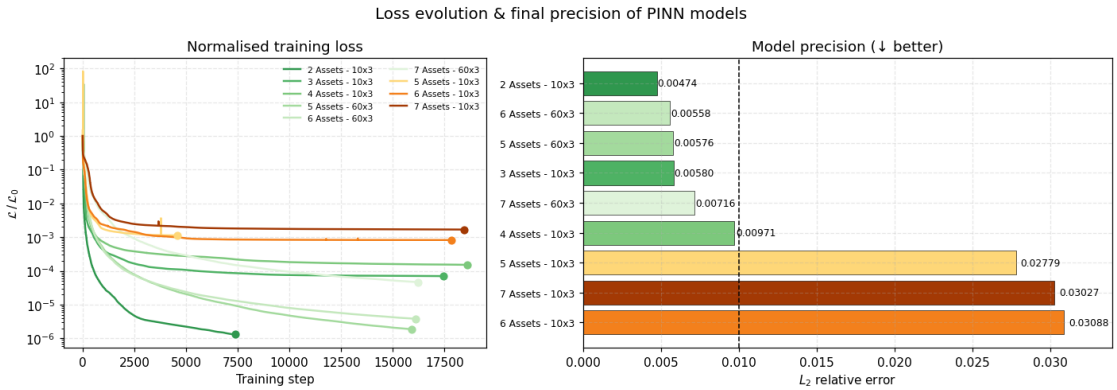
Seven-asset call option surfaces for two architectures. The network with 60 neurons per layer captures the solution more faithfully than the smaller 10-neuron model.



With the larger network and careful tuning of hyperparameters, it was possible to achieve a relative L_2 error below 10^{-3} (or better, depending on the case), which is generally considered acceptable for most practical applications. However, it is worth noting that such accuracy was not guaranteed across all dimensions and required repeated experimentation to achieve. Moreover, although the final error levels are satisfactory, there remains a lack of a comprehensive theoretical framework to rigorously bound the approximation error in higher dimensions. Figure 4.8 illustrates the convergence behaviour, where it is evident that larger networks converge more slowly but more reliably. In contrast, smaller networks occasionally terminate early due to reaching capacity limits, which may lead to misleadingly short training times but significantly worse solutions.

Figure 4.8

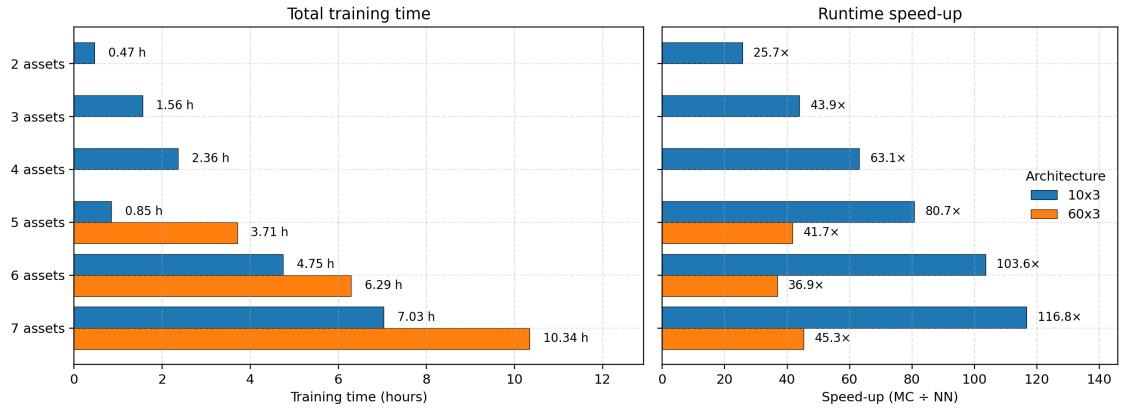
Training loss for multi-asset PINNs. Higher dimensions slow convergence and flatten the loss curve. The smaller network sometimes terminates early, yielding a shorter training time but a noticeably larger error.



Once trained, PINNs offer excellent performance in terms of computational speed. As shown in Figure 4.9, the inference time of a trained PINN is significantly faster than traditional Monte Carlo simulation methods, achieving substantial speedups even on a standard CPU. This computational advantage makes PINNs particularly attractive for real-time evaluations and repeated queries.

Figure 4.9

CPU timing comparison between PINN inference and Monte Carlo simulation.



Despite the impressive inference speed, the long training times required for PINNs present a major limitation in professional financial settings. In practice, model parameters such as volatilities, correlations, and interest rates are frequently recalibrated to reflect changing market conditions.

Additionally, option-specific inputs such as strike prices and maturities may vary across instruments or need to be adjusted in scenario analyses. In such environments, it is common to perform rapid recalculations or sensitivity analyses, which demand flexible and responsive models. The requirement to retrain a PINN from scratch for each change in input parameters makes the method impractical for many real-time applications. To be viable in professional workflows, a PINN-based solution would either need to retrain extremely quickly, generalize across a wide range of parameter configurations, or be integrated selectively into systems that can tolerate delayed response times.

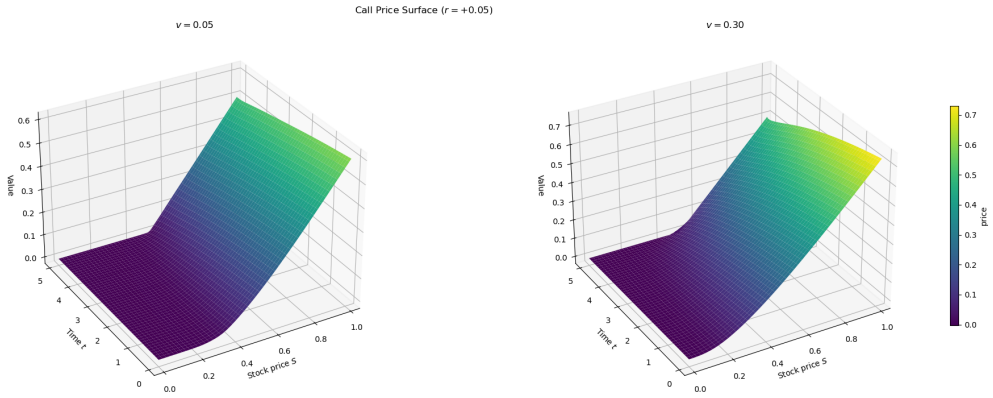
4.2.2. Call Option with Stochastic Volatility and Interest Rates

We now extend our previous examples by introducing stochastic volatility and stochastic interest rates using the Heston-Hull-White model. This combined framework allows us to capture realistic market features, including volatility smiles and interest rate fluctuations. Figure 4.10 illustrates the price surfaces produced by the PINN under various volatility (v) levels. The complete set of parameters used for this scenario is provided in Table 4.3.

Parameter	Value
Strike price, K	\$0.5
Maturity, T	5.0 years
Initial risk-free short rate, r_0	0.05
Hull-White mean reversion speed, a_r	0.1
Hull-White mean reversion level, θ_r	0.05
Hull-White interest rate volatility, σ_r	0.02
Heston initial volatility, v_0	0.04
Heston mean reversion speed, κ_v	1.5
Heston long-term volatility, θ_v	0.04
Heston volatility of volatility, σ_v	0.2
Correlation between asset and volatility, ρ_{Sv}	-0.7
Correlation between asset and interest rate, ρ_{Sr}	0.0
Correlation between volatility and interest rate, ρ_{vr}	0.0

Table 4.3
Parameters for the Call Option under the Heston-Hull-White Model

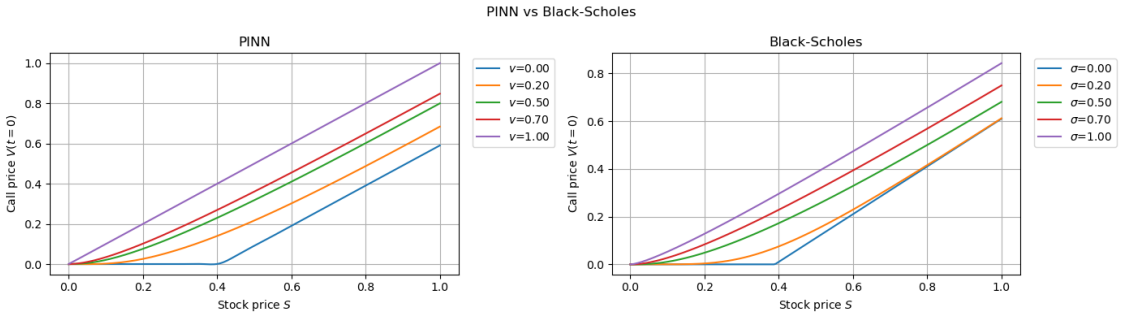
Figure 4.10
Call option pricing surfaces using the Heston-Hull-White model for different volatility levels.



Since no analytical solution exists for the combined Heston-Hull-White model, we qualitatively compare the PINN results to those obtained using the simpler Black-Scholes model (see Figure 4.11). It is possible to observe that for high underlying asset values, option prices behave nearly linearly with the underlying; at lower values, the price closely resembles the discounted payoff.

Figure 4.11

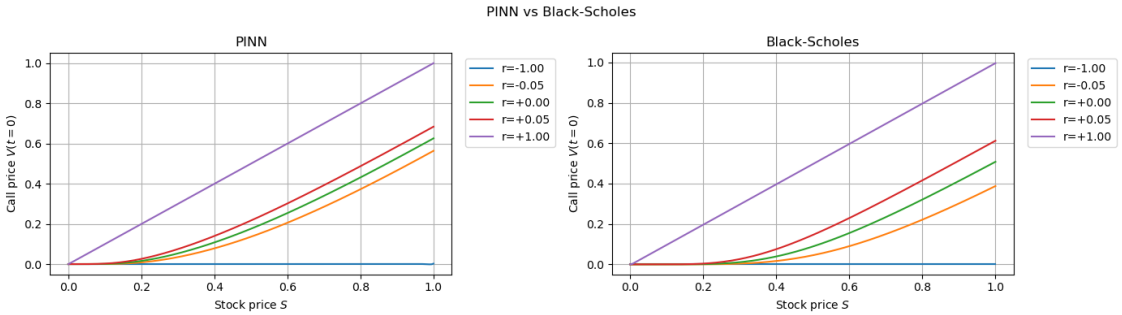
Price sensitivity to changes in volatility (v). Higher values decrease curvature in the price surface.



The effect of interest-rate dynamics on option pricing is shown in Figure 4.12. As short-term interest rates increase, option prices generally rise, reflecting the increased cost associated with hedging positions and financing derivatives contracts over longer horizons. This aligns with standard market expectations.

Figure 4.12

Price sensitivity to short rates in the Heston-Hull-White model. Higher interest rates raise the option's hedging cost, increasing the option price.



As the model incorporates stochastic volatility and interest rates, the resulting price surfaces in each dimension exhibit an increaded price, comparing to the Black-Scholes solution, as more uncertainty is introduced. This can be seen in Figure 4.13 and Figure 4.14, where the price surfaces are shown at $t = 0$ for the stock and volatility dimensions, and the stock and interest rate dimensions, respectively.

Figure 4.13
Price surface at $t = 0$ for the stock and volatility dimension.

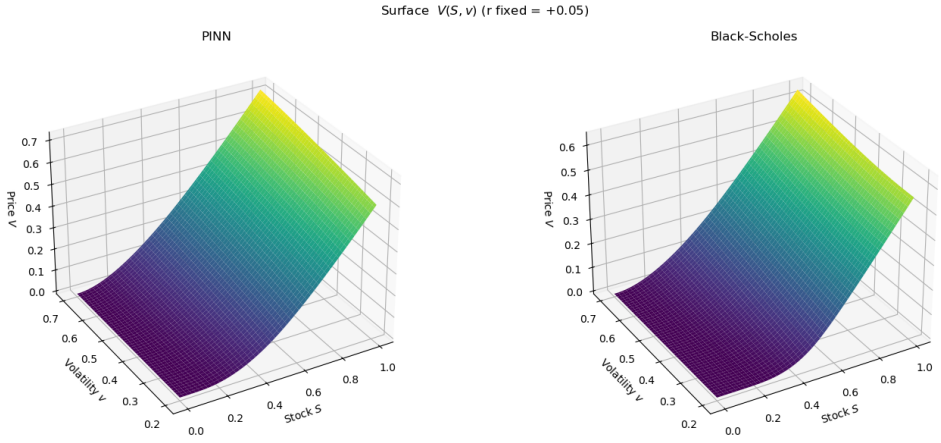
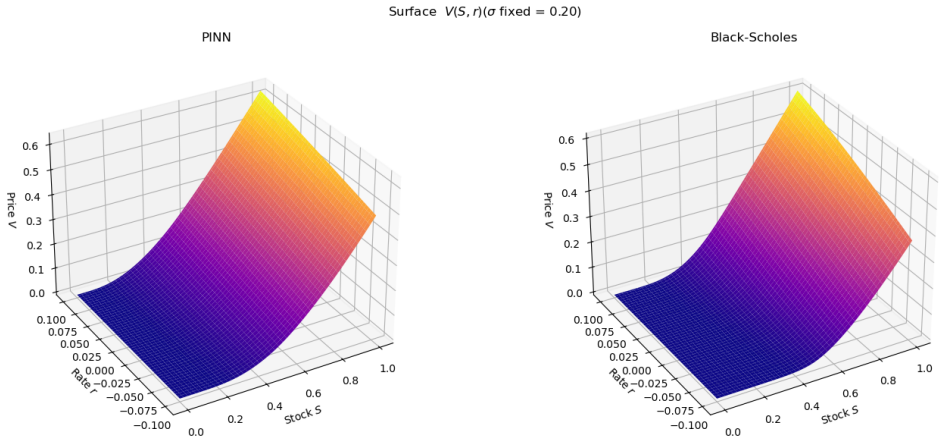


Figure 4.14
Price surface at $t = 0$ for the stock and volatility dimension.



4.3. Put Option Pricing

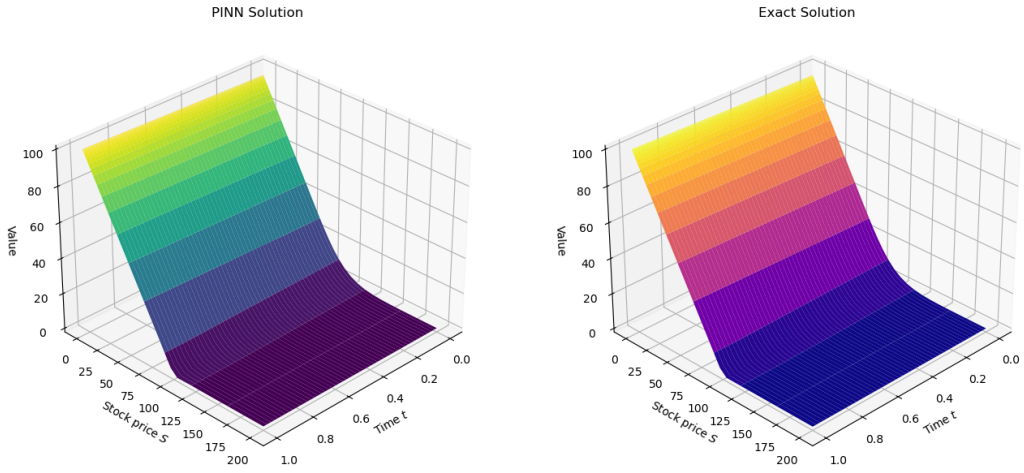
In this section, we aim to analyze early excersice features, where the most common example is the American put option, but we will also consider European put options for comparison. Puts are characterized by their distinct payoff function

$$\Phi(S) = \max(K - S, 0).$$

The pricing surface obtained from the PINN compared to the exact Black-Scholes analytical solution is displayed in Figure 4.15. Unlike call options, put option prices increase as the underlying asset value decreases, reflecting their payoff structure.

Figure 4.15

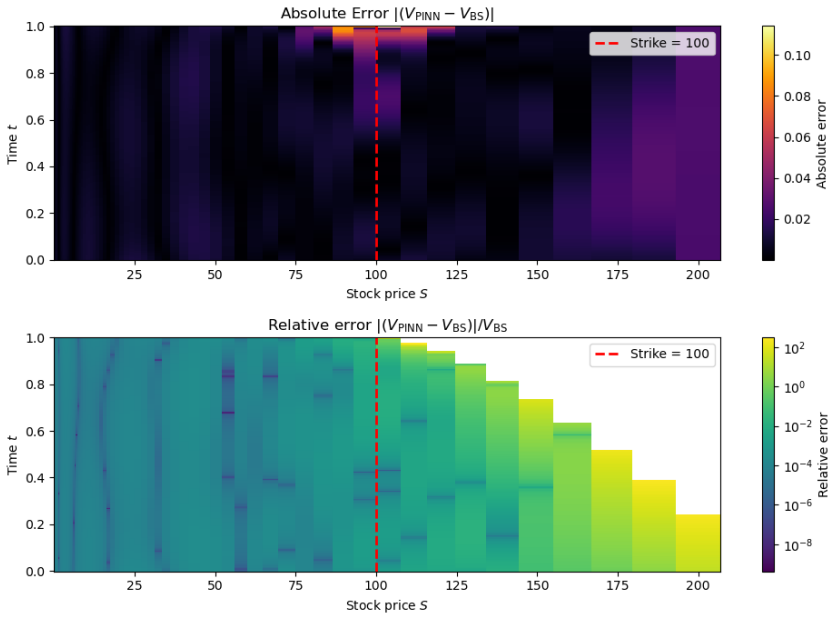
European put option pricing: comparison between PINN (left) and analytical Black-Scholes solution (right).



The absolute error in the put option pricing is shown in Figure 4.16, revealing error concentrations near maturity and around the strike price, consistent with the call option.

Figure 4.16

Absolute error distribution for the European put option. Errors peak around maturity near the strike price.



In this particular case, we limit our selves to state the L_2 error obtained by the PINN, 5.85×10^{-5}, as the case is very similar to the call option.

4.3.1. American Put Option Pricing

An American put option introduces complexity due to its early-exercise feature, allowing the holder to exercise anytime before maturity. Consequently, pricing requires solving a free-boundary problem.

There are various ways to incorporate into the modeling framework free-boundary restrictions. In this thesis we explore the formulation proposed by **wilmot2**, where the task is to setup a linear complementary problem. Mathematically, it is like this:

Some mathematical formulation

The resulting PINN solution, compared against a benchmark numerical solution (FDM), is shown in Figure 4.17.

Figure 4.17
American put option prices: PINN solution (left) compared to a numerical benchmark solution (right).

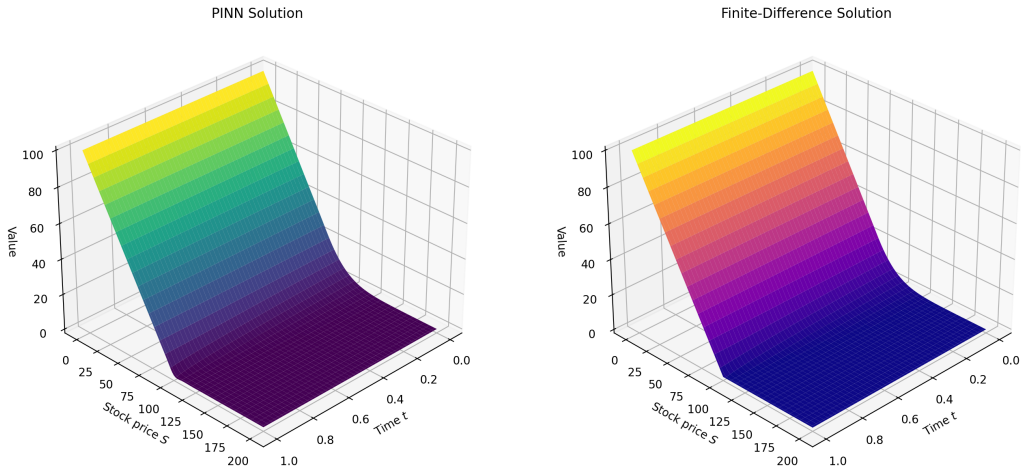
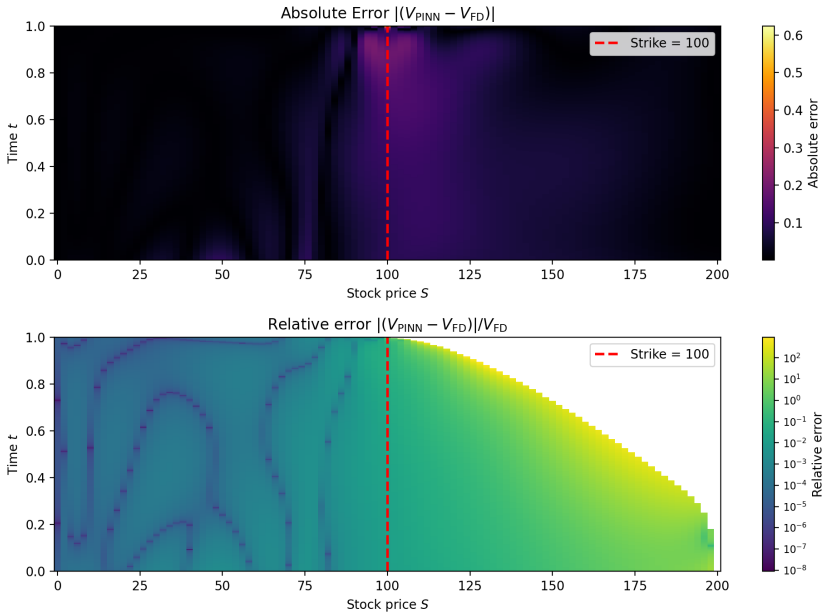


Figure 4.18 shows the absolute error in pricing the American put. The error is again concentrated around the strike price, though errors now extend across a wider region due to the early-exercise boundary's dynamic nature.

Figure 4.18

Absolute error for American put option pricing. Errors primarily arise around the dynamic early-exercise boundary.



4.4. Swaption Pricing

The final derivative instrument we evaluate is a European *payer swaption*, which gives the holder the right, but not the obligation, to enter into a fixed-for-floating interest rate swap at a specified future date. The underlying model used for pricing is the one-factor Hull-White model, as described in Section 2.2.5.

In this framework, the short rate r_t evolves according to the stochastic differential equation defined in 2.22. The swaption price depends on the future evolution of the short rate and on the value of the underlying swap, which itself is the present value of fixed versus floating rate payments over the swap's tenor.

Figure 4.19

European payer swaption pricing surface. Left: PINN solution. Right: reference solution computed with an analytic approximation or semi-analytical method.

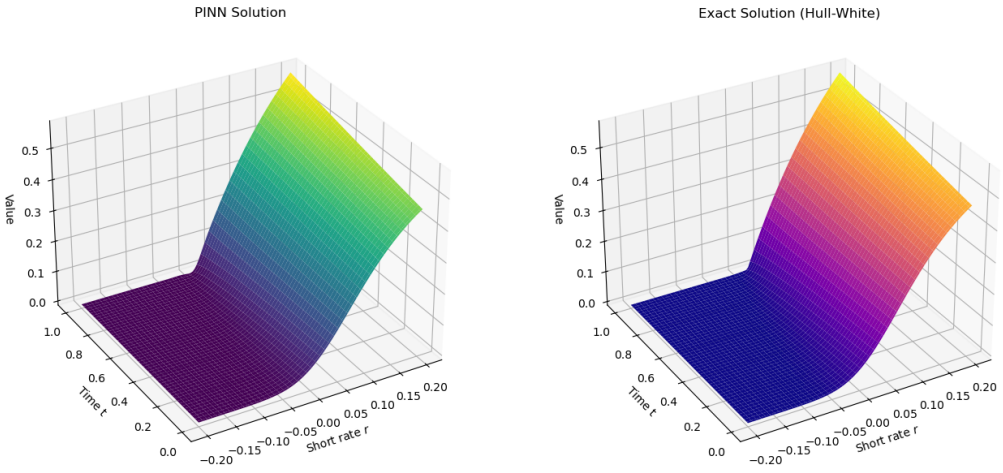
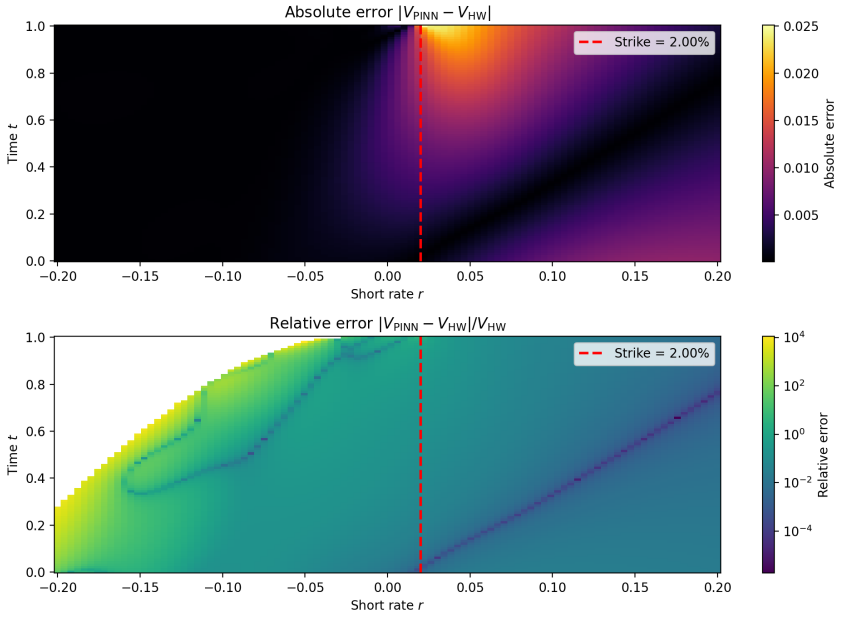


Figure 4.19 shows the pricing surface of a European swaption obtained using a PINN, compared against a benchmark solution from a conventional method. The neural network accurately captures the swaption price profile, particularly across key regions of the short-rate and strike axes.

The absolute error of the PINN solution is shown in Figure 4.20. Most of the error is concentrated, again, near the strike region and close to expiry—areas where option payoffs exhibit low smoothness or where the valuation is highly sensitive to changes in rates.

Figure 4.20

Absolute error in pricing the European payer swaption. The error is primarily concentrated near expiry and around the strike.



In general, the PINN demonstrates strong agreement with analytical benchmarks. This confirms its ability to handle the pricing of interest-rate derivatives with path-dependent structures and stochastic dynamics.

5. DISCUSSION, LIMITATIONS, AND OUTLOOK

5.1. Discussion of the Main Findings

This thesis has demonstrated that PINNs can recover option-pricing surfaces for a variety of pay-off structures and stochastic dynamics with relative errors below 10^{-3} in one- and two-dimensional settings, while keeping the evaluation cost per query essentially constant once training is completed. In low-dimensional cases the networks converge reliably and, after training, produce prices as fast as closed-form formulae—an advantage over finite-difference and Monte-Carlo solvers that scale at least linearly in the number of grid points or simulated paths. However, convergence becomes noticeably slower and less stable as soon as (1) the state dimension grows beyond three, (2) boundary conditions become more complex, making training harder, (3) model parameters/contract terms change and a fresh training run is needed.

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] 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.

[8] 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).

[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] D. Duffy, *Numerical Methods in Computational Finance: A Partial Differential Equation (PDE/FDM) Approach* (Wiley Finance). Wiley, 2022, ISBN: 9781119719670. [Online]. Available: https://books.google.es/books?id=6cZ6EAAAQBAJ.

[13] 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.

[14] 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.

[15] 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.

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

[17] J. F. Urbán, P. Stefanou, and J. A. Pons, "Unveiling the optimization process of physics informed neural networks: How accurate and competitive can pinns be?" *Journal of Computational Physics*, vol. 523, p. 113 656, Feb. 2025, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2024.113656. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2024.113656.

[18] N. Malekjani, R. Kharaghani, and E. Tsotsas, "A comparative study of dimensional and non-dimensional inputs in physics-informed and data-driven neural networks for single-droplet evaporation," *Chemical Engineering Science*, vol. 306, p. 121 214, Jan. 2025. DOI: 10.1016/j.ces.2025.121214.

[19] J. Melo, *Leveraging physics-informed neural networks for option pricing problems*, https://github.com/jmelo11/pricing_pinns, GitHub repository, accessed July 2025, 2025.