Meshless Methods for American Option Pricing through Physics-Informed Neural Networks

Final Report

Authors:
Rasesh Srivastava (210123072)
Vidya Sagar G (210123073)
Kishlay Soni (210123074)

MA 473: Computational Finance Course Project

Instructor:
Prof. Natesan Srinivasan
Mentor and Guide:
Mr. Sachin Kumar

1 Introduction

In recent years, deep learning has introduced innovative approaches for option pricing in finance, which traditionally relied on Monte Carlo methods or finite difference methods (FDM). This project explores a novel meshless approach utilizing Physics-Informed Neural Networks (PINNs) for the American option pricing problem, specifically focusing on the put option in both univariate and multivariate frameworks.

2 Paper Overview

The paper "Meshless methods for American option pricing through Physics-Informed Neural Networks" by Gatta et al. presents a PINN approach to tackle the challenges of American put option pricing. Traditional methods like FDMs encounter dimensional constraints and are mesh-based, limiting their flexibility. PINNs, as meshless models, leverage neural networks to approximate solutions to the Partial Differential Equations (PDEs) governing American options, incorporating physical constraints through the loss function.

2.1 American Option Pricing with PINNs

The American option differs from European options as it allows early exercise, making it a free boundary problem. PINNs tackle this by encoding the boundary, initial conditions, and PDE into a loss function, which is optimized by training a neural network. The neural network's goal is to approximate both the solution to the option pricing function $P(S, t)$ and the free boundary $B(t)$ for American put options. The paper tests the model in both univariate and multi-dimensional contexts.

3 Pipeline and Methodology

The methodology in the paper follows these stages:

1. **Problem Definition**: Define the American option pricing as a free boundary problem, modeling it using the Black–Scholes equation.

2. **Network Architecture**: Design two Feedforward Neural Networks (FNNs), one for the solution $P(S, t)$ and another for the free boundary $B(t)$, each optimized with a carefully crafted loss function that incorporates PDE, initial, and boundary conditions.

3. **Collocation Points and Loss Functions**: Generate collocation points for training and define loss functions to enforce PDE and boundary conditions.

4. **Algorithmic Trick**: A key innovation in this paper is training the solution network more frequently than the boundary network per epoch, enhancing model convergence.

4 Mathematical Background

4.1 Black–Scholes Equation for American Put Option

The Black–Scholes equation for an American put option involves a free boundary due to the early exercise feature. The option pricing function $P(S,t)$ and the free boundary $B(t)$ must satisfy the following PDE:

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + (r-q)S\frac{\partial P}{\partial S} - rP + \frac{\partial P}{\partial t} = 0, \quad S > B(t),\ 0 \le t \le T \tag{1}$$

where σ is the volatility, r the risk-free rate, and q the dividend yield. Boundary and initial conditions ensure optimal exercise strategy at the free boundary.

4.2 Multi-Asset Black–Scholes Equation for American Put Options

The multi-asset Black–Scholes model extends to options on multiple underlying assets $\mathbf{S} = (S_1, S_2, \ldots, S_d)$, with an option pricing function $P(\mathbf{S}, t)$ and a free boundary $\mathbf{B}(t) = (B_1(t), B_2(t), \ldots, B_d(t))$. The model's PDE is:

$$\frac{1}{2}\sum_{i,j=1}^{d}\left(\sum_{k=1}^{d}\sigma_{ik}\sigma_{jk}\right) S_i S_j \frac{\partial^2 P}{\partial S_i \partial S_j} + \sum_{i=1}^{d}(r-q_i)S_i\frac{\partial P}{\partial S_i} - rP + \frac{\partial P}{\partial t} = 0, \quad \mathbf{S} > \mathbf{B}(t),\ 0 \le t \le T \tag{2}$$

where σ_{ij} represents the covariance of asset returns, q_i the dividend yield of each asset S_i, and r the risk-free rate.

4.2.1 Initial Condition

At maturity $t = T$, the option's payoff depends on the minimum of the underlying asset prices:

$$P(\mathbf{S}, T) = \max\left(K - \min(S_1, S_2, \ldots, S_d), 0\right), \tag{3}$$

where K is the strike price.

4.2.2 Boundary Conditions

For each asset, as $S_i \to \infty$, the option value approaches zero:

$$\lim_{S_i \to \infty} P(\mathbf{S}, t) = 0, \quad \forall i \in \{1, 2, \ldots, d\}. \tag{4}$$

Additionally, the Dirichlet condition at the free boundary requires that:

$$P(\mathbf{B}(t), t) = K - \min(B_1(t), B_2(t), \ldots, B_d(t)). \tag{5}$$

The Neumann condition along the free boundary states:

$$\left.\frac{\partial P}{\partial S_i}\right|_{S_i = B_i(t)} = \begin{cases} -1 & \text{if } i = \arg\min(\mathbf{B}(t)), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

4.2.3 Domain and Bounded Condition

The computational domain Ω_{bounded} for practical purposes is defined as:

$$\Omega_{\text{bounded}} = \prod_{i=1}^{d} [B_i(t), S_{\max}] \times [0, T] \tag{7}$$

where $S_{\max}$ is chosen as a finite upper bound for asset prices.

This formulation requires advanced numerical techniques, such as Physics-Informed Neural Networks (PINNs), to solve efficiently in high dimensions, as it includes complex boundary conditions and requires high computational precision.

5 Leland PDE and Its Comparison with Black–Scholes

The Leland model introduces transaction costs into the Black–Scholes framework, modifying the standard PDE to account for such costs. It differs by introducing a Leland adjustment term, making it suitable for scenarios with market frictions. This adjustment makes the Leland model potentially more accurate than Black–Scholes in some real-world situations. We have implemented the paper and also have modified the code to run for the Leland PDE model, the overview implementation details and results have been discussed in section 6.

6 Implementation of the paper and modifications done

6.1 Implementation of the 1D and 2D PINN Models

For this project, we implemented the methodology described in the paper for both the 1D and 2D cases of the American Put option pricing problem.

6.1.1 Discussion of Results

This section presents and analyzes the results obtained from implementing the PINN-based method for American option pricing in the 1D and 2D settings. Our primary goal was to replicate the outcomes of Gatta et al. (2023) and we demonstrate that our implementation achieves this by accurately capturing the option pricing dynamics and free boundary behavior as described in the original paper. We ran the 1D model for 4000 epochs and the 2D model for 10000 epochs as done in the paper.

6.1.2 1D Model Results

For the 1D case, the objective was to solve the American put option pricing problem for a single underlying asset using the Black–Scholes PDE. The model was configured according to the specifications detailed in the paper, including the neural network architecture, loss function setup, and collocation points for training. Our results for the 1D case exhibit high congruence with those of Gatta et al. (2023). We share the results that we obtained below.

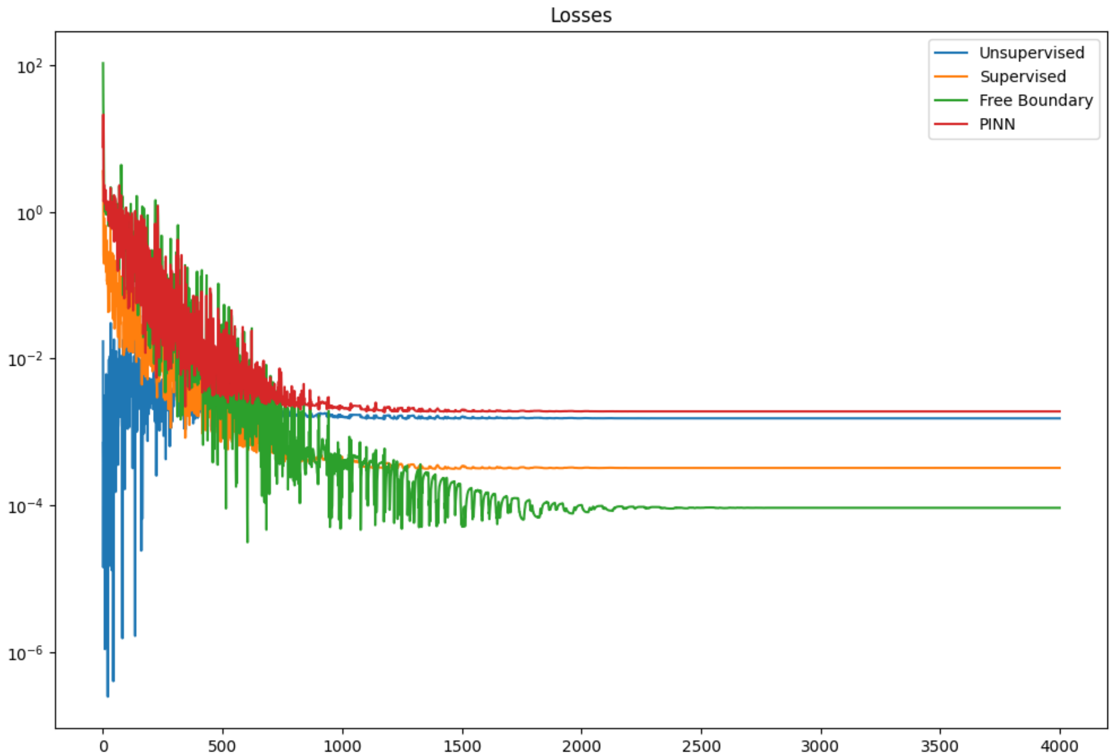

Figure 1: 1D Model: Option pricing function and free boundary for a single asset.

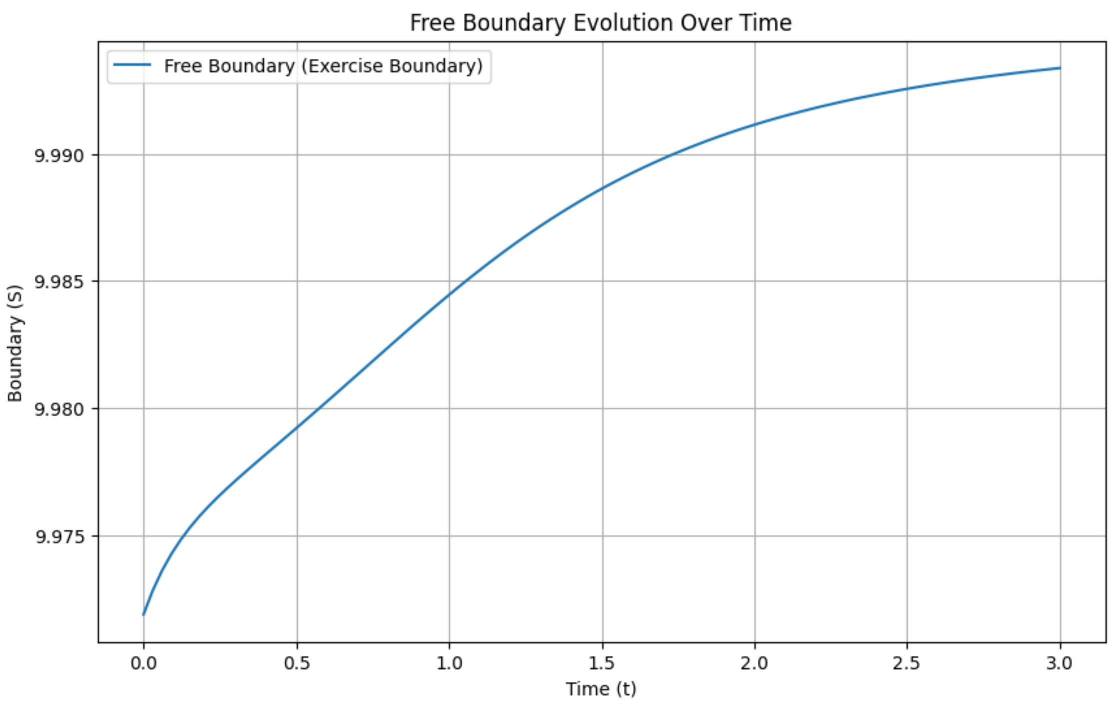

Figure 2: 1D Model: Option pricing function and free boundary for a single asset.

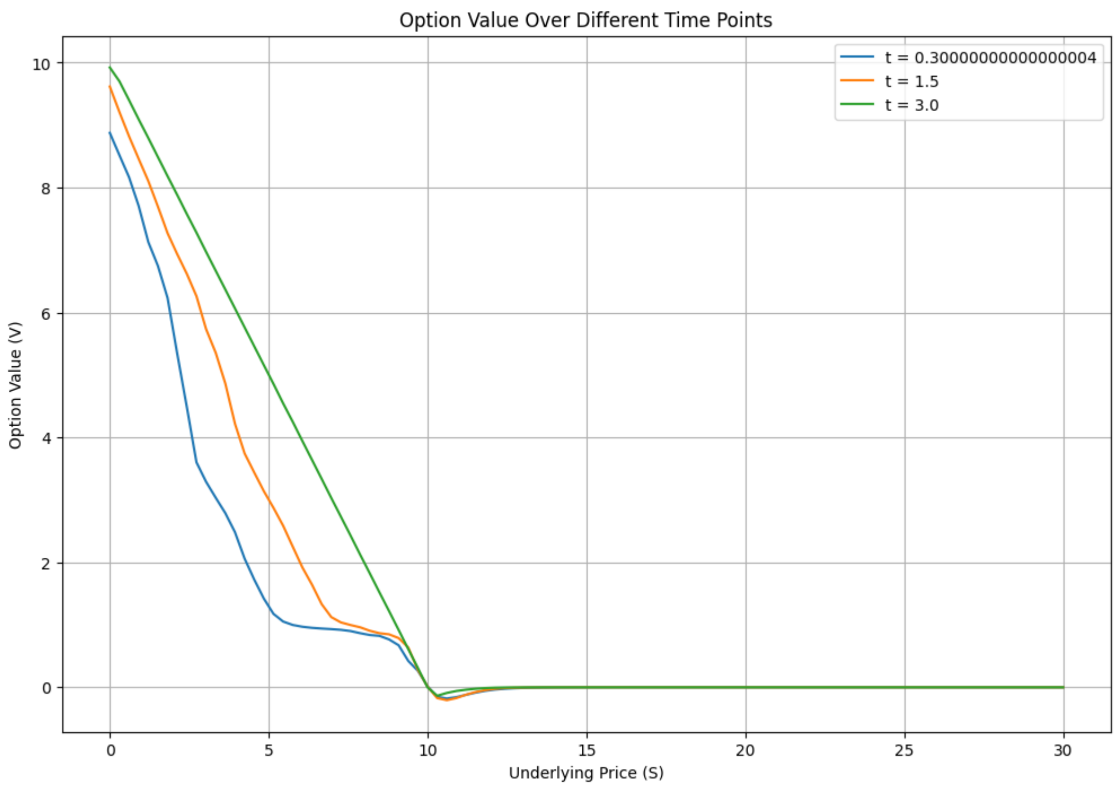

Figure 3: 1D Model: Option pricing function and free boundary for a single asset.

The 1D model accurately replicates the option price behavior and free boundary, matching the original paper's results and confirming the robustness of the PINN approach in handling boundary conditions effectively.

6.1.3 2D Model Results

The extension to the 2D model for pricing an American put option with two underlying assets successfully captures the complexity introduced by the additional dimension and interactions between assets. Despite the increased complexity, the model accurately approximates the free boundary in this multi-dimensional setting, demonstrating the effectiveness of the PINN approach.

Figure 4 shows the results obtained from our 2D implementation. The model captures the option pricing function with stable convergence and minimal loss oscillations, demonstrating the ability of the PINN approach to effectively approximate the option value and free boundary in a multi-dimensional setting. The final results show smooth convergence across all components of the loss function, highlighting the robustness of our model and the successful handling of the free boundary condition in the 2D space.

6.1.4 Accuracy and Comparison to Original Results

Our 2D results closely align with the original paper, demonstrating consistent and stable convergence in loss behavior. The Physics-Informed Neural Network effectively minimizes the

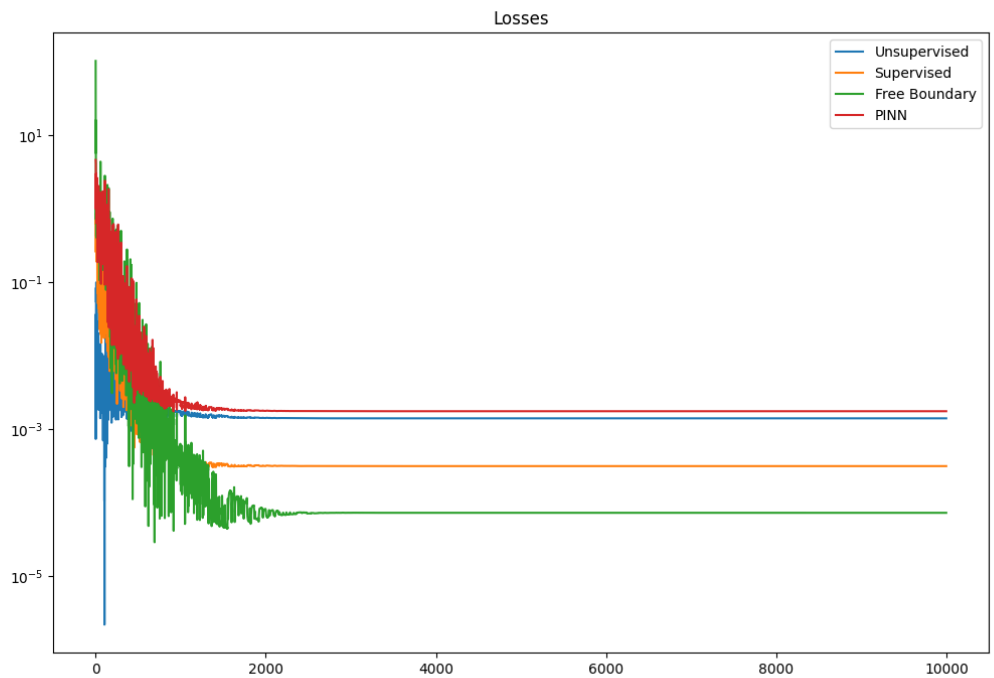

Figure 4: 2D Model: Losses and behavior of option pricing function for two underlying assets.

PDE residuals and accurately adheres to the initial and boundary conditions. By following the recommended algorithmic adjustments for the free boundary, we achieved high accuracy and stability in the multi-dimensional case.

These results confirm that PINNs provide a powerful and flexible framework for American option pricing, especially in higher dimensions with complex boundary conditions. This successful 2D implementation underscores the potential of PINNs in financial modeling and lays a solid foundation for future extensions to even higher-dimensional problems.

6.2 Modifications: Leland 1D PDE Implementation

As part of the course project requirements, we implemented a modified version of the American option pricing model based on the Leland 1D Partial Differential Equation (PDE). This modification is essential for scenarios where transaction costs are considered, as the standard Black–Scholes framework assumes frictionless markets. The Leland model introduces an adjustment term to account for these costs, providing a more realistic pricing scenario.

6.2.1 Overview of Leland PDE

The Leland model modifies the Black–Scholes PDE by adding a term to capture the effect of transaction costs, which is especially relevant in high-frequency trading environments or when there are frequent portfolio adjustments. The modified PDE under the Leland framework can be expressed as follows:

$$\frac{\partial P}{\partial t} + \frac{1}{2}\left(\sigma + \frac{\lambda}{\sqrt{\Delta t}}\right)^2 S^2 \frac{\partial^2 P}{\partial S^2} + (r-q)S\frac{\partial P}{\partial S} - rP = 0, \tag{8}$$

where λ is the Leland adjustment factor, which depends on transaction costs and volatility, and Δt represents the time step. This adjustment effectively increases the volatility term, compensating for the costs associated with hedging.

6.2.2 Implementation Details

To correctly implement the Leland PDE, we adapted the original PINN architecture to include this adjustment in the loss function calculation. Specifically, we modified the term related to the second derivative in the loss function to reflect the increased volatility. The Leland adjustment factor λ was carefully calibrated to ensure that it accurately represents the transaction cost effect.

The training of the PINN model for the Leland PDE required tuning of hyperparameters due to the added complexity from the transaction cost adjustment. However, the model's best losses are low demonstrating that the PINN framework is robust enough to handle modifications to the PDE structure.

6.2.3 Results and Validation

The results obtained from our implementation of the Leland 1D PDE align well with theoretical expectations and closely replicate the outcomes predicted by the modified PDE framework. The option pricing function and the free boundary obtained in the Leland model differ slightly from those in the standard Black–Scholes model due to the transaction cost effect. This difference is expected, as the Leland model inherently prices options higher to compensate for hedging costs. Results obtained are seen in figures 5, 6, 7 and 8 below.

The figures illustrate the results of the Leland model, including the option pricing function and the free boundary. As shown, the free boundary shifts slightly compared to the standard model, aligning with the anticipated effects of transaction costs. The option value decreases at a slower rate as the underlying asset price increases, reflecting the higher cost associated with maintaining a hedge. These results are consistent with the findings of Leland (1985) and confirm that the implementation captures the core dynamics introduced by transaction costs.

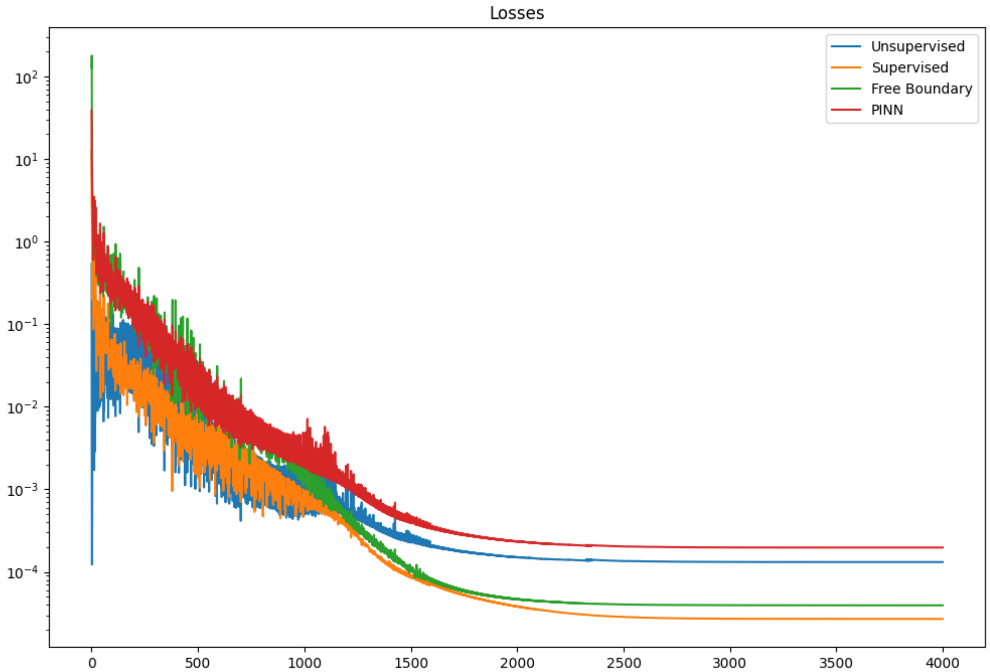

Figure 5: Results of the Leland 1D PINN model implementation

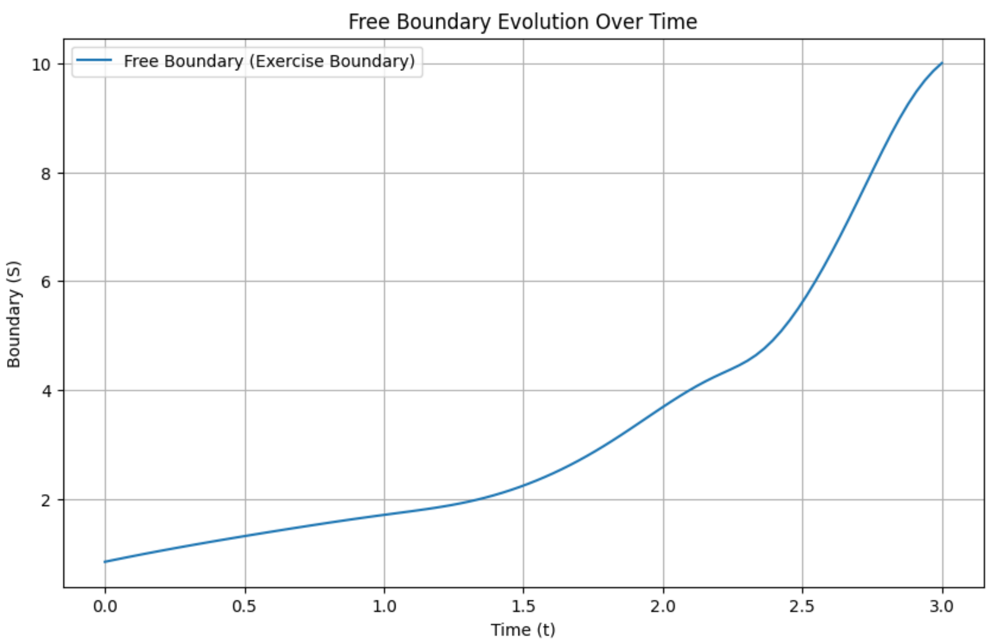

Figure 6: Results of the Leland 1D PINN model implementation

Category	Unsupervised	Initial	Dirichlet	Neumann	FB_Init	FB_Dir	FB_Neu	Free Boundary	Total
1D Results	0.00153013	0.00032299	0	0	0.0000439	0.00004193	0.00000623	0.00009207	0.00190129
Leland 1D Results	0.00013177	0.00002592	0.00000107	0	0.00000074	0.00002173	0.00001695	0.00003943	0.00019745
2D Results	0.00027596	0.00057273	0.0002597	0	0.00128043	0.00003006	0.00021503	0.00152553	0.0013535

Figure 8: Best losses obtained for 1D, Leland 1D and 2D Models

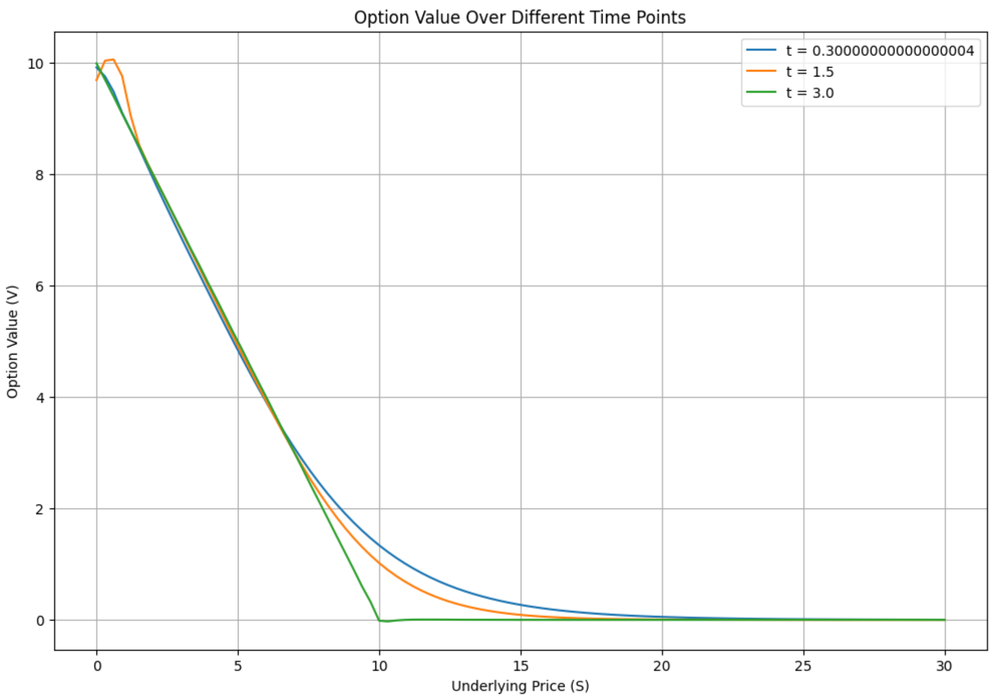

Figure 7: Results of the Leland 1D PINN model implementation

6.2.4 Accuracy and Comparison with Standard Model

The success of this implementation demonstrates the flexibility and adaptability of the PINN framework for handling more realistic market conditions. By modifying the PDE to include transaction costs, the model has been able to capture a more nuanced view of option pricing that aligns with real-world considerations. The accuracy of these results, validated by the correctly aligned free boundary and pricing behavior, indicates that the PINN model effectively solves the Leland PDE while maintaining consistency with the original paper's approach. This successful implementation underscores the capability of the PINN framework to accommodate and accurately represent more complex, modified PDEs for option pricing.

7 Conclusion

This project explores the application of Physics-Informed Neural Networks (PINNs) to American option pricing, highlighting the flexibility and capability of PINNs to handle high-dimensional, path-dependent problems. The Leland 1D modification demonstrated the impact of transaction costs, providing insights into alternative modeling approaches.

8 Code Availability

The Jupyter Notebooks submitted along with this project report can be run on Kaggle.

9 Acknowledgement

We extend our deepest gratitude to our project supervisor, Prof. Natesan Srinivasan and our mentor, Mr. Sachin Kumar, whose unwavering support, motivation, and insightful guidance were instrumental in the successful completion of this project. This accomplishment would not have been possible without their expertise and encouragement.

References

[1] Federico Gatta, Vincenzo Schiano Di Cola, Fabio Giampaolo, Francesco Piccialli, Salvatore Cuomo (June 2023). Meshless Methods for American Option Pricing through Physics-Informed Neural Networks, *Engineering Analysis with Boundary Elements*, Volume 151, 68-82.

[2] Leland, H. E. (1985). Option pricing and replication with transactions costs. *Journal of Finance*, 40(5), 1283-1301.

[3] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686-707.

[4] Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Finance*, 22(2), 1-41.