Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 4, No. 2, Summer & Autumn 2024
Research paper

European option pricing underlying two assets using PINN

Kimiya Tavakoli[1], Abdolsadeh Neisy[2], Alireza Zamanpour[3]

[1] Department of Mathematics, Allameh Tabatabai University (ATU), Tehran, Iran.
kimtavakoli76@gmail.com

[2] Department of Mathematics, Allameh Tabatabai University (ATU), Tehran, Iran
a_neisy@atu.ac.ir

[3] Department of financial engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran

a.zamanpour@iau.ac.ir

Abstract:
Modeling and pricing European options are crucial tasks for financial companies seeking to determine the fair value of these instruments. Conventional methods, such as using Black-Scholes partial differential equations (PDEs), face challenges due to the high complexity involved and lack of data. To address these challenges, PINNs have recently emerged as a promising approach to solving the Black-Scholes PDEs for European options. In this paper, we tackle the two-dimensional Black-Scholes model to determine the price of a European exchange option. We employ a kind of ANNs (PINN) that is specifically designed to learn the option's value by minimizing an appropriately defined loss function. The data for our study were generated through simulations conducted in Python. Our results demonstrate the efficacy of the PINN approach by comparing the computed fair value of a European exchange option with the traditional solutions. The findings underscore the potential of PINNs in providing accurate and efficient pricing for complex financial derivatives.

Keywords: Physics-informed Neural Networks, Two-dimensional Black-Scholes Model, Loss Function, Activation Function, Artificial Neural Networks, European Option
Classification: 91Gxx, 91G99, 97Mxx

1 Introduction

Options are crucial financial instruments that have gained prominence over the years. Originating in the 19th century in the United States and Europe, their popularity led to the establishment of the Chicago Options Exchange by the Chicago

[2]Corresponding author
Received: 23/05/2024 Accepted: 29/09/2024
https://doi.org/10.22054/JMMF.2024.79962.1135

Board of Trade in April 1973, now a central global trading hub [16]. Develop a robust mathematical framework to model and price challenges, attracting diverse researchers in recent decades. Using the Black-Scholes partial differential equation (PDE) is a standard mathematical method for pricing and modeling financial instruments [5]. In this study, we expand the Black-Scholes framework to incorporate two assets, resulting in a two-dimensional parabolic partial differential equation (PDE) corresponding to the price dynamics of a European exchange option involving these two assets. However, applying it to real-world data can pose challenges due to uncertainties in data values and high complexity.

Various mathematical techniques, including numerical methods, have been developed to price the financial options. Numerical methods for option pricing, such as the Binomial Tree Model, Finite Difference Method, and Monte Carlo simulations, are essential tools in financial mathematics for valuing derivatives. These methods handle complex stochastic processes by simplifying them into manageable calculations, like discretizing stock price movements or solving the Black-Scholes equation numerically [11]. However, these methods face several drawbacks in real-world applications. Assumptions like constant volatility do not hold in volatile markets, leading to inaccuracies. High-dimensional problems, such as those involving basket or exotic options, introduce computational challenges and inefficiencies, often described as the "curse of dimensionality." Additionally, Monte Carlo simulations require extensive computational resources, complicating their use for real-time pricing [7]. More complex models have been developed to mitigate these issues, but at the cost of increased computational demands.

To address these challenges, This research specifically addresses the valuation of European exchange options. This derivative allows the holder to swap one asset for another on a specific future date, with the valuation dependent on two underlying assets, such as stocks [4]. Our novel contribution is the application of Physics-Informed Neural Networks (PINNs) to solve these complex two-dimensional Black-Scholes PDEs. PINNs employ a neural network architecture that learns to solve tasks through its structure rather than through rule-based programming. This network includes multiple layers from the input to the output, passing through several hidden layers that process the initial parameters and ultimately provide solutions [22]. This work builds on previous research that has successfully used artificial neural networks to tackle various financial derivative problems by integrating mathematical models with neural network technologies. For instance, neural network approaches have been preferred over traditional numerical methods for solving differential equations in financial applications, as outlined in "Neural Network Methods for Solving Differential Equations" [28]. Artificial neural networks are particularly adept at identifying patterns in economic data, including price forecasting for stocks and bond ratings [22].

Unlike traditional analytical-numerical methods, artificial neural networks offer a more practical alternative for approximating solutions to the Black-Scholes and

other differential equations prevalent in various scientific fields [2, 8, 10]. One of the key aspect of using neural networks in financial problem-solving is the formulation of an appropriate loss function [27]. Here we mentioned some advantages of Physics-Informed Neural Networks (PINNs) compared to traditional Artificial Neural Networks (ANNs). PINNs integrate physical laws and mathematical equations into the training process, allowing them to model complex physical phenomena. By leveraging the underlying physics, probably PINNs can achieve higher accuracy in solving differential equations and other mathematical problems compared to traditional ANNs. On the other hand PINNs can effectively utilize incomplete or limited datasets, as they rely on physical laws to guide the learning process, making them more robust in scenarios with sparse data. The use of PINNs often leads to more stable solutions, especially in dynamic systems. In contrast, traditional ANNs primarily rely on input-output mappings without considering the underlying physical principles, which can limit their effectiveness in solving complex problems.
In this paper, we structure our discussion: Section 2 outlines the formulation of financial PDE problems, including the necessary initial and boundary conditions. Section 3 discusses the structure of the loss function for training the PINN and the design of the network itself. Finally, Section 4 presents our results, showcasing the application of PINNs in financial mathematics and parameter estimation for options trading [14].

2 Methodology

2.1 Mathematical modeling

The reference option pricing PDE for the valuation of an European, put or call option is the BlackScholes equation. In this section, the pricing models for an European option with one underlying asset and European exchange option are presented. Then, all the initial and boundary conditions are considered.

European Options, One Underlying Asset

When it comes to options with one underlying asset, this means that the option is based on only one specific asset, such as a stock, commodity, or currency pair. This is in contrast to options with multiple underlying assets. Assume the underlying asset S follows the geometric Brownian motion:

$$dS = (\mu - \delta)Sdt + \sigma SdW_t, \tag{1}$$

W_t represents Weiner process, μ is the drift term, σ is the volatility, and the underlying asset S is assumed to pay a constant dividend yield δ, Assuming there are no arbitrage opportunities [3].

In the BlackScholes framework, the one-asset European option price, $v(S,t)$ satisfies the following PDE [22]:

$$\begin{cases} L(v) = \partial_t v + Av - rv = 0, & S \in \Omega^{\sim}, t \in [0,T), \\ v(S,T) = H(S), \end{cases} \tag{2}$$

where the operator A is defined as follows:

$$Av \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + (r-\delta) S \frac{\partial v}{\partial S}, \tag{3}$$

And the function H represents the payoff of the option at maturity. In other words:

$$\begin{cases} (K-S)^+ & for\, a\, put\, option, \\ (S-K)^+ & for\, a\, call\, option, \end{cases} \tag{4}$$

Here, K denotes the strike price agreed upon in the contract.

Boundary Conditions

European put option:

$$BC: \begin{cases} v(S_{1\infty}, t) = 0,\ 0 < t < T \\ v(0,t) = Ke^{-r(T-t)},\ 0 < t < T. \end{cases} \tag{5}$$

European call option:

$$BC: \begin{cases} v(0,t) = 0,\ 0 < t < T \\ v(S_{1\infty}, t) = S - Ke^{-r(T-t)},\ 0 < t < T. \end{cases} \tag{6}$$

The analytic solution for 2 is equal to [16]:

$$v_c(S,t) = Se^{(-\delta(T-t))} N_{0,1}(d_1) - Ke^{(-r(T-t))} N_{0,1}(d_2), \tag{7}$$

$$v_p(S,t) = Ke^{(-r(T-t))} N_{0,1}(-d_2) - Se^{(-\delta(T-t))} N_{0,1}(-d_1), \tag{8}$$

$N_{0,1}$ denotes cumulative distribution function for the standard normal. The first index 0 indicates the mean of the normal distribution, the second index 1 indicates the variance of the normal distribution. The parameters d_1 , d_2 are equal to:

$$d_1 = \frac{\log(\frac{S}{K}) + (r - \delta + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{(T-t)}}, d_2 = \frac{\log(\frac{S}{K}) + (r - \delta - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{(T-t)}}. \tag{9}$$

European exchange option, Two Underlying Assets

One of the simplest multi-asset options, is exchange option. Multi-asset options are a group of options whose pay-off depends on more than one underlying assets. In this sense, a two-asset option is a special case of multi-asset option, where the number of underlying assets is two. In this case, we say this option is two-dimensional and we use two-dimensional Black-Scholes framework. As it was mentioned, this type of option allows the holder to exchange one underlying asset to another. Let S_1 and S_2 be the asset price of the two underlying assets; the payoff the European exchange option on maturity date T is equal to:

$$v(S_1, S_2, T) = (S_1 - S_2)^+, \tag{10}$$

So by extending the dynamic for one underlying asset 1, suppose each asset price follows the following dynamics:

$$dS_1 = (\mu_1 - \delta_1)S_1 dt + \sigma_1 S_1 dW_1, \tag{11}$$

$$dS_2 = (\mu_2 - \delta_2)S_2 dt + \sigma_2 S_2 dW_2, \tag{12}$$

Let $v = v(S_1, S_2, t)$ be an infinitely differentiable function. By using Ito's lemma [18]:

$$dv = \frac{\partial v}{\partial t}dt + \frac{\partial v}{\partial S_1}dS_1 + \frac{\partial v}{\partial S_2}dS_2 + \frac{\partial^2 v}{\partial S_1 \partial S_2}dS_1 dS_2 + \frac{1}{2}\frac{\partial^2 v}{\partial S_1^2}(dS_1)^2 + \frac{1}{2}\frac{\partial^2 v}{\partial S_2^2}(dS_2)^2, \tag{13}$$

We know, $(dW_1)^2 = dt$, $(dW_2)^2 = dt$, $dtdW_1 = 0$, $dtdW_2 = 0$, $(dt)^2 = 0$ and $dW_1 dW_2 = \rho dt$ [18], then

$$\begin{cases} (dS_1)^2 = ((\mu_1 - \delta_1)S_1 dt + \sigma_1 S_1 dW_1)^2 = \sigma_1^2 S_1^2 dt, \\ (dS_2)^2 = ((\mu_2 - \delta_2)S_2 dt + \sigma_2 S_2 dW_2)^2 = \sigma_2^2 S_2^2 dt, \\ (dS_1 dS_2) = \sigma_1 \sigma_2 \rho S_1 S_2 dt. \end{cases} \tag{14}$$

Therefore, by substitution all these three equations in 13:

$$dv = (\frac{\partial v}{\partial t} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 v}{\partial S_1 \partial S_2} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 v}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 v}{\partial S_2^2})dt + \frac{\partial v}{\partial S_1}dS_1 + \frac{\partial v}{\partial S_2}dS_2, \tag{15}$$

Now, we will form a risk hedging portfolio like Π, including an option, Δ_1 share of S_1 and Δ_2 share of S_2. The value of this basket is equal to

$$\Pi = v(S_1, S_2, t) - \Delta_1 S_1 - \Delta_2 S_2, \tag{16}$$

Therefore, the change in the value of the basket will be in the following:

$$d\Pi = dv - \Delta_1(dS_1 + \delta_1 S_1 dt) - \Delta_2(dS_2 + \delta_2 S_2 dt), \tag{17}$$

Thus,

$$d\Pi = (\frac{\partial v}{\partial t} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 v}{\partial S_1 \partial S_2} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 v}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 v}{\partial S_2^2} - \Delta_1\delta_1 S_1 - \Delta_2\delta_2 S_2)dt$$

$$+ (\frac{\partial v}{\partial S_1} - \Delta_1)dS_1 + (\frac{\partial v}{\partial S_2} - \Delta_2)dS_2, \quad (18)$$

To have a risk-free portfolio, it should be:

$$\begin{cases} \Delta_1 = \frac{\partial v}{\partial S_1}, \\ \Delta_2 = \frac{\partial v}{\partial S_2}, \end{cases} \quad (19)$$

We assume the market is arbitrage-free. If arbitrage opportunities were present, market participants could take advantage of these price discrepancies to secure risk-free profits, resulting in pricing that is inconsistent and unsustainable. The absence of arbitrage guarantees that assets are priced equitably in relation to one another. The no-arbitrage condition plays a crucial role in the financial mathematical framework, as it ensures that discounted price processes behave as martingales, which is a fundamental concept in stochastic calculus and financial modeling [11]. In the absence of arbitrage, only one risk-free interest rate should be determined. Thus:

$$r\Pi dt = d\Pi, \quad (20)$$

Then

$$\frac{\partial v}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 v}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 v}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 v}{\partial S_1 \partial S_2} + (r-\delta_1)S_1 \frac{\partial v}{\partial S_1} + (r-\delta_2)S_2 \frac{\partial v}{\partial S_2} - rv = 0. \quad (21)$$

This equation represents the two-dimensional Black-Scholes equation for European exchange options with two underlying assets, which can be briefly written as follows [22]:

$$\begin{cases} L(v) = \partial_t v + \beta v - rv = 0 \qquad 0 < S_1, S_2 < \infty, 0 < t < T, \\ v(S_1, S_2, T) = H_1(S_1, S_2), \end{cases} \quad (22)$$

Where the operator β is equal to:

$$\beta v \equiv \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 v}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 v}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 v}{\partial S_1 \partial S_2} + (r-\delta_1)S_1 \frac{\partial v}{\partial S_1} + (r-\delta_2)S_2 \frac{\partial v}{\partial S_2}. \quad (23)$$

In the following, all the initial and boundary conditions are presented.

Initial and Boundary Conditions

Initial conditions for t=0:

$$IC : v(S_1, S_2, 0) = max(S_1 - S_2, 0) = (S_1 - S_2)^+, 0 < S_1, S_2 < \infty. \quad (24)$$

Boundary conditions [3]:

$$\begin{cases} BC : v(S_{1_{0_b}}, S_2, t) & = 0,\ 0 < t < T, \\ BC : v(S_1, S_{2_{0_b}}, t) & = S_1,\ 0 < t < T, \\ BC : v(S_{1_{m_b}}, S_2, t) & = max(S_{1_{m_b}} - S_2, 0),\ 0 < t < T, \\ BC : v(S_1, S_{2_{m_b}}, t) & = max(S_1 - S_{2_{m_b}}, 0),\ 0 < t < T. \end{cases} \quad (25)$$

Considering the above conditions from solving the two-dimensional Black-Scholes partial differential equation according to the Margrabes formula, the fair value of a European exchange option at time t is given by [15]:

$$v(S_1, S_2, t) = e^{-\delta_1(T-t)} S_1(t) N_{0,1}(d_1) - e^{-\delta_2(T-t)} S_2(t) N_{0,1}(d_2). \quad (26)$$

again $N_{0,1}$ denotes cumulative distribution function for the standard normal and parameters d_1 , d_2, and σ are equal to:

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}, \quad (27)$$

$$d_1 = \frac{\log(\frac{S_1}{S_2}) + (\delta_2 - \delta_1 + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{(T-t)}}, \quad (28)$$

$$d_2 = d_1 - \sigma\sqrt{(T-t)}. \quad (29)$$

3 physics-informed neural networks

Physics-Informed Neural Networks (PINNs) are a type of networks model that incorporates physical laws and constraints into the neural network architecture to solve forward and inverse problems involving partial differential equations (PDEs). The general idea of physics-informed neural networks (PINNs) is to solve problems where only limited data are available, e.g. noisy measurements from an experiment. The closed-form equation introduced in this study is the Margrabe formula [15]. Since the partial differential equation in this research is a second-order heat equation, its solution is faced with high complexity. To address this issue, we propose introducing Physics-Informed Neural Networks (PINNs). To compensate for the data scarcity, the algorithm is enriched with physical laws governing the problem at hand. Typically, those laws are described by parameterized non-linear partial differential equations of the form [12]:

$$\frac{\partial v}{\partial t} + N[v, \lambda] = 0,\ x \in \Omega,\ t \in T, \quad (30)$$

There are two different use-cases for physics-informed neural networks, namely data-driven inference and data-driven identification of partial differential equations. The first case addresses forward problems where the coefficients λ are known and the hidden solution $v(x,t)$ is computed based on initial and boundary data. The second one solves an inverse problem. The values of $v(x,t)$ are given, the goal is to identify the coefficients λ. In equation 30 $N[v,\lambda]$ represents a nonlinear differential operator with λ coefficients. This type of neural network is applicable to a wide range of problems, including reaction-diffusion equations of chemical systems to equations based on continuous mechanics.

In this section, we briefly introduce Physics-Informed Neural Networks for solving data-driven inference partial differential equations. A PINN consists of two main parts. The first part is a classical feedforward neural network approximating the value of $v(x,t)$. The second part, the physics-enriched part, presents the partial derivatives of the differential equation. In artificial neural networks, the goal is to find an appropriate answer that satisfies the corresponding partial differential equation. The output of the network depends on the set of weight and bias parameters. These weights and biases are learned by minimizing a loss function that includes several mean square error terms. The loss function can be written in general form as:

$$C = MSE_v + MSE_f. \tag{31}$$

The first term, denoted by MSE_v computes the error of the approximation $v(x,t)$ at known data points. This term entails data representing the boundary and initial conditions. The other term in the loss function, MSE_f, enforces the partial differential equation on a large set of randomly chosen collocation points inside the domain. Since the introduced mathematical model is the same as the two-dimensional heat equation, to better understand the problem, we first explain the PINN for the two-dimensional heat equation problem.

Suppose the rod is 1 meter long and has a thermal conductivity coefficient of $k = 1$. Initially, the temperature of the rod is 0řC throughout its length. Suddenly, the temperature at one end of the rod increases to 100řC. The goal is to calculate the temperature of the rod at any point and at any time. To solve this problem, a Physics-Informed Neural Network can be used. This network is trained using training data that provides the rod's temperature at specific points along the rod and at different times. After training, it can predict the rod's temperature at any point and at any time. A two-dimensional heat equation in general form is written as follows:

$$u_t = k \times (u_{xx} + u_{yy}), \tag{32}$$

where u is temperature at (x,y) at time t, u_{xx} and u_{yy} are second-order derivatives of temperature with respect to x and y, and k is the thermal conductivity coefficient. To solve this equation using a Physics-Informed Neural Network (PINN), we encode the equation into a neural network. In figure 1 a PINN for heat equation is represent: This artificial neural network consists of two main parts. The first part is a classical neural network that approximates the solution $u(x,y,t)$. The second part is the physics-informed component, where the differential operators appear. This artificial

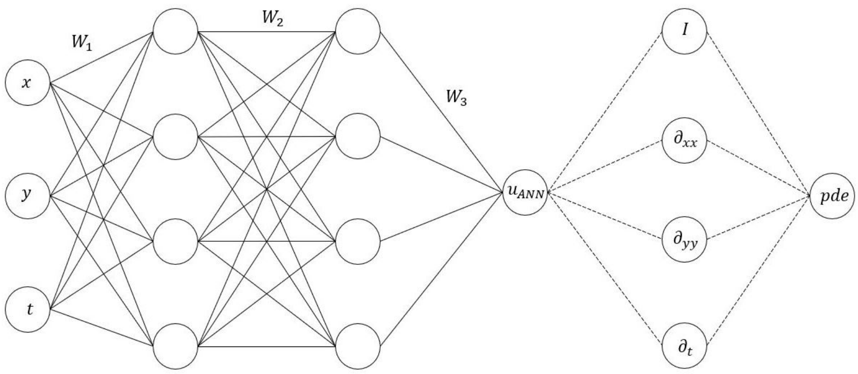

Figure 1: Diagram of a Physics-Informed Neural Network (PINN) for the 2D heat equation with 2 hidden layers

neural network has one input layer, two hidden layers, and one output layer. One of the goals of these artificial neural networks is to minimize the loss function, which is based on the boundary and initial conditions. Considering the 2D heat equation, we find out that it has 4 boundary conditions and one initial condition as follows:
Boundary conditions:

$$\begin{cases} u(x_{0_b}, y, t) & = 0,\ 0 \leqslant t \leqslant T, \\ u(x_{m_b}, y, t) & = 0,\ 0 \leqslant t \leqslant T, \\ u(x, y_{0_b}, t) & = 0,\ 0 \leqslant t \leqslant T, \\ u(x, y_{m_b}, t) & = 0,\ 0 \leqslant t \leqslant T, \end{cases} \tag{33}$$

Initial condition:

$$u(x, y, t) = f(x, y),\ 0 \leqslant x \leqslant x_m,\ 0 \leqslant y \leqslant y_m, \tag{34}$$

Considering the above conditions, the loss function can be written as:

$$Loss_{total} = \omega_I MSE_I + \omega_B MSE_B + \omega_{pde} MSE_{pde}, \tag{35}$$

where
$\mathbf{MSE_I}$ is mean squared Error of the initial condition, which is equal to:

$$MSE_I = \frac{1}{N_I} \sum_{i=1}^{N_I} |u_{ANN}(x_0^i, y_0^i, 0) - u(x_0^i, y_0^i, 0)|^2, \tag{36}$$

$\mathbf{MSE_B}$ is mean squared error of the boundary conditions, which is equal to:

$$MSE_B = \frac{1}{N_b} [(\sum_{i=1}^{N_b} |u_{ANN}(x_{0_b}, y_b^i, t_b^i)|^2 + \sum_{i=1}^{N_b} |u_{ANN}(x_{m_b}, y_b^i, t_b^i)|^2$$

$$+\sum_{i=1}^{N_b}|u_{ANN}(x_b^i, y_{0_b}, t_b^i)|^2 + \sum_{i=1}^{N_b}|u_{ANN}(x_b^i, y_{m_b}, t_b^i)|^2], \tag{37}$$

$\mathbf{MSE_{pde}}$ is is mean squared error of the collocation points of pde, which is equal to:

$$MSE_{pde} = \frac{1}{N}\sum_{i=1}^{N}|pde_{ANN}(x_{pde}^i, y_{pde}^i, t_{pde}^i)|^2, \tag{38}$$

By substituting MSE_I, MSE_B and MSE_{pde} in 35 the total loss function is equal to:

$$Loss_{total} = \frac{1}{N_I}\sum_{i=1}^{N_I}|u_{ANN}(x_0^i, y_0^i, 0) - u(x_0^i, y_0^i, 0)|^2$$
$$+\frac{1}{N_b}[(\sum_{i=1}^{N_b}|u_{ANN}(x_{0_b}, y_b^i, t_b^i)|^2 + \sum_{i=1}^{N_b}|u_{ANN}(x_{m_b}, y_b^i, t_b^i)|^2$$
$$+\sum_{i=1}^{N_b}|u_{ANN}(x_b^i, y_{0_b}, t_b^i)|^2 + \sum_{i=1}^{N_b}|u_{ANN}(x_b^i, y_{m_b}, t_b^i)|^2]$$
$$+\frac{1}{N}\sum_{i=1}^{N}|pde_{ANN}(x_{pde}^i, y_{pde}^i, t_{pde}^i)|^2, \tag{39}$$

To find the solution of the partial differential equation, we need to find the optimal u so:

$$arg\, min\, Loss_{total}(u) = \hat{u}, \tag{40}$$

This is the optimal solution that satisfies the 2D heat equation. So far, we have become familiar with the concepts of a physics-based neural network and how to construct the loss function based on the initial and boundary conditions. With attention to definitions related to the initial and boundary conditions, the total loss function for training this PINN in the 2D Black-Scholes framework is equal to:

$$Loss_{total} = \omega_I MSE_I + \omega_B MSE_B + \omega_{pde} MSE_{pde} = \frac{1}{N_I}\sum_{i=1}^{N_I}|v_{ANN}(S_{1,0}^i, S_{2,0}^i, 0) - v(S_{1,0}^i, S_{2,0}^i, 0)|^2$$
$$+\frac{1}{N_b}(\sum_{i=1}^{N_b}|v_{ANN}(S_{1_{0_b}}, S_{2b}^i, t_b^i) - v(S_{1_{0_b}}, S_2^i, t^i)|^2 + \sum_{i=1}^{N_b}|v_{ANN}(S_{1_{m_b}}, S_{2b}^i, t_b^i) - v(S_{1_{m_b}}, S_2^i, t^i)|^2$$
$$+\sum_{i=1}^{N_b}|v_{ANN}(S_{1b}^i, S_{2_{0_b}}, t_b^i) - v(S_1^i, S_{2_{0_b}}, t^i)|^2 + \sum_{i=1}^{N_b}|v_{ANN}(S_{1b}^i, S_{2_{m_b}}, t_b^i) - v(S_1^i, S_{2_{m_b}}, t^i)|^2)$$
$$+\frac{1}{N}\sum_{i=1}^{N}|pde_{ANN}(S_{1pde}^i, S_{2pde}^i, t_{pde}^i)|^2, \tag{41}$$

To reach a solution that satisfies the 2D Black-Scholes model, the total loss function must be optimized, so

$$arg\ min\ Loss_{total}(v_{ANN}) = \hat{v}, \tag{42}$$

4 Results

As mentioned earlier, based on minimizing the total loss function, the artificial neural network is capable of finding the solution to the partial differential equation. The implemented ANN has a 3-layer architecture. The first layer has 3 inputs, S_1, S_2 and t. The hidden layer has 105 neurons and the activation function used is $tanh$. It is obvious that $\frac{\partial v}{\partial t}$, $\frac{\partial^2 v}{\partial S_1^2}$, $\frac{\partial^2 v}{\partial S_2^2}$, $\frac{\partial^2 v}{\partial S_1 \partial S_2}$, $\frac{\partial v}{\partial S_1}$, $\frac{\partial v}{\partial S_2}$ are derivatives used to form the partial differential equation component. More specific details for setting up the design of this physics-informed neural network are available in table 1. Figure 2 presents a general overview of the design of this physics-informed neural network.

Table 1: Elements of PINN

Description and Configuration	PINNs Architecture
Inputs	3
Hidden Layers	1
Output Layers	1
Neurons in hidden layer	105
Activation Function	Hyperbolic Tangent (tanh)
Optimization Algorithm	Adam
Derivative Operators	$\frac{\partial v}{\partial t}$, $\frac{\partial^2 v}{\partial S_1^2}$, $\frac{\partial^2 v}{\partial S_2^2}$, $\frac{\partial^2 v}{\partial S_1 \partial S_2}$ $\frac{\partial v}{\partial S_1}$
Epochs	7000

Finally, after 7000 epochs, the final value of the loss function is equal to 5×10^{-3}

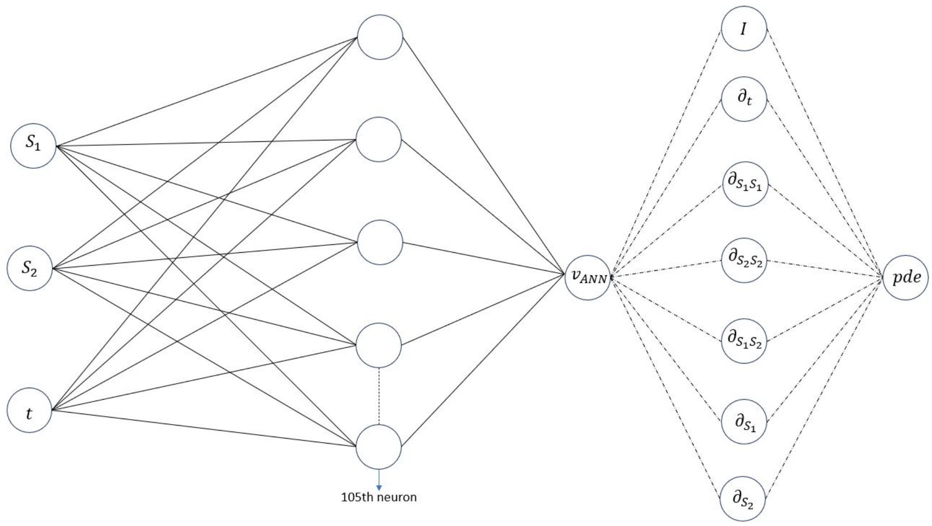

Figure 2: A physics-informed neural network for the two-dimensional Black-Scholes equation

Table 2: Comparison of the results between physics-informed neural network and Margarabe's formula related to the European exchange option

Input Values	Physics-informed Neural Network	Margarabe formula
$(36, 30, 3)$	5.4371	5.9095
$(42, 34, 3)$	7.4401	7.6511
$(29, 24, 3)$	4.2586	4.8840
$(53, 13, 3)$	39.0658	39.0091
$(49, 40, 3)$	8.4164	8.6693
$(53, 40, 3)$	12.3112	12.0392
$(20, 12, 3)$	7.1626	7.5068
$(50, 41, 3)$	8.4162	8.7063
$(27, 10, 3)$	16.6805	16.5387
$(31, 22, 3)$	8.2164	8.2922
$(53, 41, 3)$	11.3346	11.1901
$(43, 8, 3)$	34.4086	34.1357
$(24, 5, 3)$	18.0927	18.5306

and in figure 4, all the loss functions related to boundary and initial conditions, callocation points and the total loss function are displayed. In this section, a comparison is made between the solutions obtained from the physics-informed artificial neural network and the closed-form Margrabe formula. As shown in table 2, the network is capable of predicting the option price with high accuracy at given points. Here the evaluation metrics to calculate accuracy is R2 score. So

$$R^2 = 1 - \frac{RSS}{TSS} = 99.68\% \tag{43}$$

where

$$RSS = \sum (v - v^*)^2, \tag{44}$$

$$TSS = \sum (v - v^{\sim})^2, \tag{45}$$

In 44 v is the actual value, v^* is the predicted value and in 45 $v^{\sim}$ is the mean value of the variable /feature. In figure 3 , the equivalence between the price of European exchange option and the price obtained from PINN is shown.

5 Conclusions

This research explores the use of Physics-Informed Neural Networks (PINNs) to solve the complex two-dimensional Black-Scholes partial differential equation (PDE) for pricing European exchange options. Traditional methods for pricing options face challenges due to complex computations and uncertainties in real-world data. The

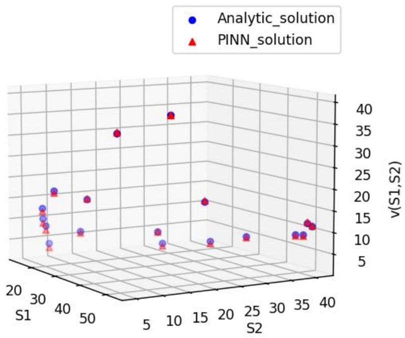

Figure 3: 3D scatter plotting for comparing the PINN solution and the analytic solution. Real data (blue points) means the values which obtained from Margrabe formula and the red points show the PINN solution. In both cases t is constant and its equal to 3.($t = 3$).

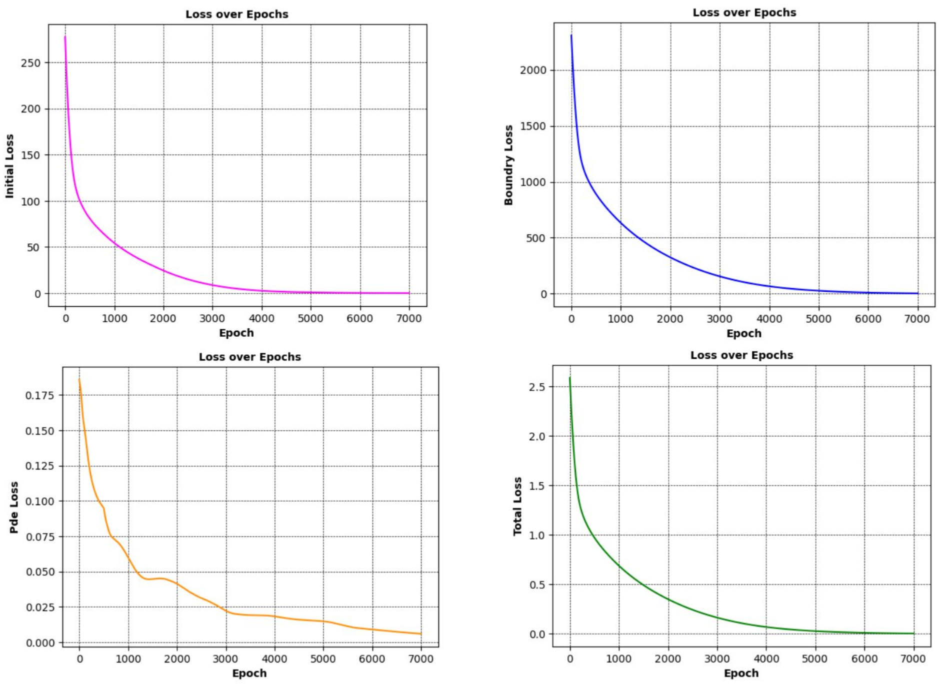

Figure 4: The process of minimizing the initial and boundary conditions, callocation points and the total loss functions for the 2D Black-Scholes model.

study's findings show the effectiveness of PINNs for addressing these challenges:

• PINNs offer an efficient and accurate approach for pricing complex financial derivatives like European exchange options.

• The PINN method, unlike conventional numerical methods, can handle complex models and high-dimensional problems without the ”curse of dimensionality”.
• PINNs demonstrate the potential to enhance the accuracy and efficiency of financial modeling and pricing, offering a practical alternative to traditional analytical-numerical methods.
Also the findings highlight that PINNs not only streamline the process of pricing complex financial derivatives (such as European exchange options) but also achieve high accuracy in their results, marking them as a robust alternative to traditional methodologies. As financial markets continue to evolve, the application of PINNs shows significant potential in enhancing the efficiency and reliability of financial modeling and option pricing, paving the way for more accurate evaluations in increasingly complex market environments.

For future work,
1) As said, the price of the European option with two underlying assets in this thesis has been obtained by Physics-informed Neural Network. You can use this method to find the price of American or Asian option.
2) Because artificial neural networks are used in a wide range of problems, the models of other artificial neural networks, such as RNN networks, can be used to solve different types of partial differential equations.
3) Considering the mathematical model used in this thesis, the model's parameters can be estimated instead of using artificial neural networks to find the option price.
4) The underlying asset dynamics were assumed to follow a geometric Brownian motion, which implies a continuous path without jumps. However, a potential direction for future research is to incorporate jumps into this model and explore the resulting implications.

Bibliography

[1] Azizi, E., Neisy, A., 2018, A New Approach in Geometric Brownian Motion Model, Fuzzy Information and Engineering and Decision, *Advances in Intelligent Systems and Computing Springer International Publishing AG*, **646**, 334-350.

[2] Amilon, H., 2003, A Neural Network Versus BlackScholes: A Comparison of Pricing and Hedging Performances, *J. Forecast*, **22**, 317335.

[3] Bjork, T, 2009, Arbitrage Theory in Continuous Time, *Oxford University Press* , 546 pp.

[4] Chen, Y., 2017, Numerical Methods for Pricing Multi-Asset Options, Department of Computer Science, University of Toronto, 85.

[5] Duffy, J., 2022, Numerical Methods in Computational Finance: A Partial Differential Equation (PDE/FDM) Approach, *Wiley*, 544 pp.

[6] Esfahani, M. K., Neisy, A., De Marchi, S., 2021, An RBF approach for oil futures pricing under the jump-diffusion model, *Journal of Mathematical Modeling*, **9**, 81-92.

[7] Glasserman, P., 2004, Monte Carlo Methods in Financial Engineering, *Springer Science and Business Media*, 596 pp.

[8] Grohs, P., Hornung, F., Jentzen, A., Wurstemberger, P.V., 2023, A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations, *Memoirs of the American Mathematical Society*, 47-109.

[9] Hagan, M., Demuth, H., Beale, M., De Jesús, O., 1996, Neural Network Design, *PWS Pub*, 744 pp.

[10] Han, J., Jentzenb, A., E, W., 2018, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. USA*, **34**, 85058510.

[11] Hull, J., 2000, Options, Futures Other Derivatives, Volume 1, *Prentice Hall*, 698 pp.

[12] Kohler, M., Krzyzak, A., Todorovic, N., 2010, Pricing of highdimensional American options by neural networks, *Math. Financ*, **20**, 383410.

[13] Kollmannsberger, S., Angella, D., Jokeit, M., Herrmann, L., 2021, Deep Learning in Computational Mechanics, *Springer*, 108 pp.

[14] Liu, S., Oosterlee, C.W., Bothe, S.M., 2019, Pricing options and computing implied volatilities using neural networks, *Risks*, **7**, 16.

[15] Margrabe, W, 1978, The value of an option to exchange one asset for another, *J. Financ*, **33**, 177186.

[16] Neisy, A., Salmani Gharaei, K., 2017, Financial Engineering and Markets Modeling: A MATLAB-based Approach, *Allameh Tabatabaei University* , 323 pp.

[17] Neisy, A., De Marchi, S., Jalili, R., 2018, A Radial Basis Function Method for Solving Bond Pricing Model, *Dolomites Research Week on Approximation (DRWA18)*, Alba di Canazei.

[18] Oksendal, B., 2003, Stochastic Differential Equations: An Introduction with Applications, *Springer*, 406 pp.

[19] Pasquale, A. Cuomo, S. Mariapia, R., 2024, A physicsinformed deep learning approach for solving strongly degenerate parabolic problems, *Engineering with Computers Springer*.

[20] Pettersson, U., Larsson, E., Marcusson, G., 2008, Improved radial basis function methods for multi-dimensional option pricing, *Computational and Applied Mathematics*, **222**, 8293.

[21] Raissi, M., Karniadakis, G.E., Perdikaris, P., 2017, Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations, *arXiv*, arXiv:1711.10561v1..

[22] Salvador, B., Oosterlee, C.W.,Van der Meer, R., 2021, Financial Option Valuation by Unsupervised Learning with Artificial Neural Networks, *Mathematics MDPI*, **9**,1-20.

[23] Sirignano, J., Spiliopoulos, K., 2018, A Deep Learning Algorithm for Solving Partial Differential Equations, *DGM: Journal of Computational Physics*, Amsterdam, The Netherlands.

[24] Shahrokhabadi, M.A., Neisy, A., Perracchione, E., Polato, M., 2019, Learning with reduced kernel-based methods: Environmental and financial applications, *Dolomites Research Notes on Approximation*, **12**, 12-27.

[25] Shidfar, A., Zakeri, A., Neisy, A., 2005, A two Dimension Inverse Heat Conduction Problem For Estimating Heat Source, *International Journal of Mathematics and Mathematical Science*, **10**, 1633-1641.

[26] Wilmott, P., Howison, S., Dewynne, J., 2012, The Mathematics of Financial Derivatives, *Cambridge University Press*, 312 pp.

[27] Van der Meer, R., Oosterlee, C., Borovykh, A., 2020, Optimally weighted loss functions for solving PDEs with Neural Netwoks, *arXiv*, arXiv:2002.06269.

[28] Yadav, N., Yadav, A., Kumar, M., 2015, Neural Network Methods for Solving Differential Equations, *Springer*, 114 pp.

[29] Zurada, J.M., 1992, Introduction to Artificial Neural Systems, *St. Paul : West*, 812 pp.

How to Cite: Kimiya Tavakoli[1], Abdolsadeh Neisy[2], Alireza Zamanpour[3], *European option pricing underlying two assets using PINN*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 4, No. 2, Pages:17–31, (2024).