



PRECIFICAÇÃO DE OPÇÕES E CALIBRAÇÃO DE VOLATILIDADE IMPLÍCITA NO MERCADO BRASILEIRO USANDO REDES NEURAIIS INFORMADAS PELA FÍSICA (PINNS)

Rubens Molina Loures do Nascimento

Universidade do Estado do Rio de Janeiro – FAT /

rubens.nascimento@discentes.fat.uerj.br

Vahid Nikoofard

Universidade do Estado do Rio de Janeiro – FAT / vahid@fat.uerj.br

Resumo

Desafios em finanças quantitativas, como precificação de opções e calibração de volatilidade implícita, são recorrentes. O modelo de Black-Scholes, ao assumir volatilidade constante, não captura fenômenos como o volatility smile. Este trabalho propõe uma metodologia baseada em Redes Neurais Informadas pela Física (PINNs) para precificar opções e inferir a superfície de volatilidade implícita. A equação diferencial parcial (EDP) de Black-Scholes é incorporada diretamente na função de perda da rede, permitindo que o modelo aprenda com dados de mercado observados (prêmio da opção) e com a lei física que rege sua dinâmica. Foram utilizados dados de opções de compra europeias de ativos negociados no mercado brasileiro, adquiridos via API, entre 2021 e 2023. Para validação, empregaram-se dados de opções negociadas em 2024, envolvendo ativos de sete grandes empresas nacionais. Os resultados analisados demonstraram que o PINN obteve alta acurácia na precificação de preços de mercado e, aprendeu a dinâmica da EDP com resíduos com valores próximos a zero. Além disso, o modelo é capaz de inferir uma superfície de volatilidade implícita suave e financeiramente coerente com a realidade do mercado, capturando com sucesso o "volatility smile", como também derivar as "Greeks" (como o Delta) através de diferenciação automática.

Palavras-chave: Volatilidade implícita, Precificação de opções, Redes Neurais PINNs, Equação de Black-Scholes, Volatility smile.

Abstract

Challenges in quantitative finance, such as option pricing and the calibration of implied volatility, are recurrent. The Black-Scholes model, by assuming constant volatility, fails to capture phenomena like the volatility smile. This work proposes a methodology based on Physics-Informed Neural Networks (PINNs) to price options and infer the implied volatility surface. The Black-Scholes partial differential equation (PDE) is directly embedded into the network's loss function, allowing the model to learn from observed market data (option premiums) and the physical law governing its dynamics. Data from European call options on assets traded in the Brazilian market, acquired via API between 2021 and 2023, were used. For validation, option data from 2024 were employed, involving assets from seven major national companies. The analyzed results showed that the PINN achieved high accuracy in pricing market options and learned the PDE dynamics with residuals close to zero. Furthermore, the model can infer a smooth and financially consistent implied volatility surface that successfully captures the volatility smile, and also derive the Greeks (such as Delta) through automatic differentiation.

Keywords: Implied volatility, Option pricing, Physics-Informed Neural Networks, Black-Scholes equation, Volatility smile.

Resumen

Los desafíos en las finanzas cuantitativas, como la valoración de opciones y la calibración de la volatilidad implícita, son recurrentes. El modelo de Black-Scholes, al asumir una volatilidad constante, no logra capturar fenómenos como el "volatility smile". Este trabajo propone una metodología basada en Redes Neuronales Informadas por la Física (PINNs) para valorar opciones e inferir la superficie de volatilidad implícita. La ecuación diferencial parcial (EDP) de Black-Scholes se incorpora directamente en la función de pérdida de la red, permitiendo que el modelo aprenda tanto de los datos observados del mercado como de la ley física que rige su dinámica. Se utilizaron datos de opciones de compra europeas de activos del mercado brasileño, obtenidos vía API entre 2021 y 2023. Para la validación, datos negociadas en 2024, involucrando activos de siete grandes empresas nacionales. Los resultados analizados demostraron que la PINN logró alta



Anais do Simpósio Acadêmico de Engenharia de Produção (SAEPRO) da EEL-USP

IX SAEPRO – 25 e 26 de novembro de 2025

precisión en la valoración de precios de mercado y aprendió la dinámica de la EDP con residuos cercanos a cero. Además, el modelo puede inferir una superficie de volatilidad implícita suave y coherente con la realidad financiera del mercado, capturando exitosamente el "volatility smile", y también derivar las "Greeks" (como el Delta) mediante diferenciación automática.

Palabras clave: Volatilidad implícita, Valoración de opciones, Redes Neuronales PINNs, Ecuación de Black-Scholes, Volatility smile.

1 INTRODUÇÃO

Modelos matemáticos, como o modelo de Black-Scholes, que são capazes de descrever como os preços dos contratos evoluem são fundamentais para quem atua no mercado financeiro (HULL, 2016). O modelo introduzido por Black, Scholes e Merton foi capaz de fornecer a primeira solução analítica em tempo contínuo para a precificação de opções europeias, revolucionando o campo na época.

No entanto, o modelo foi construído sobre premissas que o simplificam, sendo a mais importante, assumir que a volatilidade (σ) do ativo subjacente permanecerá constante ao longo de todo tempo, independente do preço de exercício (strike) (HULL, 2016). Após anos de negociações no mercado, essa premissa se demonstrou que esta suposição não é válida, pois o mercado demonstra um fenômeno conhecido como “volatility smile”, ou “skew”, onde ocorre com opções que contenham o mesmo vencimento, mas diferentes strikes, implicando assim em diferentes valores de volatilidade.

Essa premissa é capaz de gerar um problema ao tentar precificar uma opção, uma vez que há a necessidade de uma calibração de uma superfície de volatilidade implícita que seja consistente com os preços de mercado observados, porém, métodos tradicionais para isso são muitas vezes computacionalmente intensivos ou carecem de robustez.

As Redes Neurais Informadas pela Física (PINNs) surgiram como uma classe de modelos de *deep learning* que são capazes de resolver equações diferenciais parciais (EDPs) (SANTOS, 2024), onde a principal ideia é utilizar a própria EDP, junto com suas condições de contorno e iniciais, como um termo que é responsável por regularizar a função de perda da rede, junto com as observações realizadas através dos dados, penalizando toda vez que a rede se desvie do esperado ao violar as leis físicas, ou financeiras, impostas e dos dados reais.

Neste presente trabalho, é proposto e demonstrado a eficácia do modelo PINN com o objetivo de não só ser capaz de precificar opções, como também capaz de gerar o

peso de calibração da volatilidade observado no mercado brasileiro, tendo aprendido a respeitar a EDP de Black-Scholes e dados do mercado financeiro brasileiro.

2 FUNDAMENTAÇÃO TEÓRICA

Neste tópico, vamos tratar a respeito de como se fundamenta o modelo matemático utilizado. Antes de tudo, precisamos entender sobre a equação diferencial parcial (EDP) de Black-Scholes, que é responsável por ser a parte da física do PINN e que governa a precificação no mercado. Em seguida, a formulação da Rede Neural é detalhada para a fim de explicar como ela é capaz de integrar a física ao seu processo de treinamento e aprendizagem.

2.1 A Equação Diferencial Parcial de Black-Scholes

A Equação de Black-Scholes (BS), uma EDP (Equação Diferencial Parcial) parabólica de segunda ordem e que é responsável por precificar uma opção europeia V , e que depende do preço do ativo subjacente, S , e do tempo, t , pode ser descrita pela equação diferencial parcial de Black-Scholes a seguir:

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

Onde:

- $V(S, t)$ é o preço da opção.
- S é o preço do ativo subjacente.
- t é o tempo, onde é utilizado o tempo até o vencimento, $\tau = T - t$, o que inverte o sinal do termo $\frac{\partial V}{\partial \tau}$.
- σ é a volatilidade do ativo subjacente.
- r é a taxa de juros livre de risco.

No modelo clássico de Black-Scholes, é levado em questão a premissa de que tanto a taxa de juros r e quanto a volatilidade σ são constantes, no entanto, a suposição de uma volatilidade constante é uma suposição falha quando o objetivo é capturar a realidade presente no mercado ao não ser capaz de capturar o fenômeno do "sorriso da

volatilidade" (ou volatility smile), onde opções Out The Money (OUT) e In The Money (ITM) não podem ser precificados de forma correta (HULL, 2016).

Com isso, para resolver esta limitação, podemos formular a Equação de Black-Scholes incluindo a volatilidade local $\sigma(S, t)$. Com esta abordagem utilizando PINN permite com que este termo não seja tratado como uma constante, mas sim como uma função alvo dependente do estado (S, t) que pode ser aprendida pela rede, ao fazer a resolução do problema de forma inversa (LIU; MA, 2025), sendo capaz de aprender uma estrutura que é complexa e não-constante a partir dos dados do mercado utilizados durante o treino, fazendo assim com que seja gerado um modelo capaz de realizar uma precificação mais realista do que a formulação analítica padrão.

2.2 Redes Neurais como Aproximadores de Funções

Uma Rede Neural (Neural Networks) tem como seu princípio a ideia de um Perceptron de Múltiplas Camadas (MLP), que tem como papel ser um aproximador de funções. Uma MLP que contenha uma quantidade suficiente de neurônios e ao menos uma camada oculta com uma função de ativação não linear que seja capaz de se aproximar de qualquer função contínua em um domínio em um determinado domínio compacto com um certo grau de liberdade de precisão, isso de acordo com o Teorema de Aproximação Universal (HORNIK et al. apud SANTOS, 2024).

O modelo trabalho no presente trabalho teve como base esta arquitetura estrutural, onde a rede recebe um vetor de entrada $x = V_{\theta}(S, K, \tau, r)$ e que o processa através uma sequência de camadas ocultas, e por sua vez, a cada camada, estes dados de entrada passam por uma transformação linear e sendo multiplicado pelos pesos e adição de viés, e em seguida, passando por uma função de ativação não linear. A camada final produz uma saída escalar V_{θ} que é a aproximação esperada da rede com os preços reais de opções, que são o nosso alvo. A figura abaixo demonstra um exemplo de estrutura utilizada no projeto.

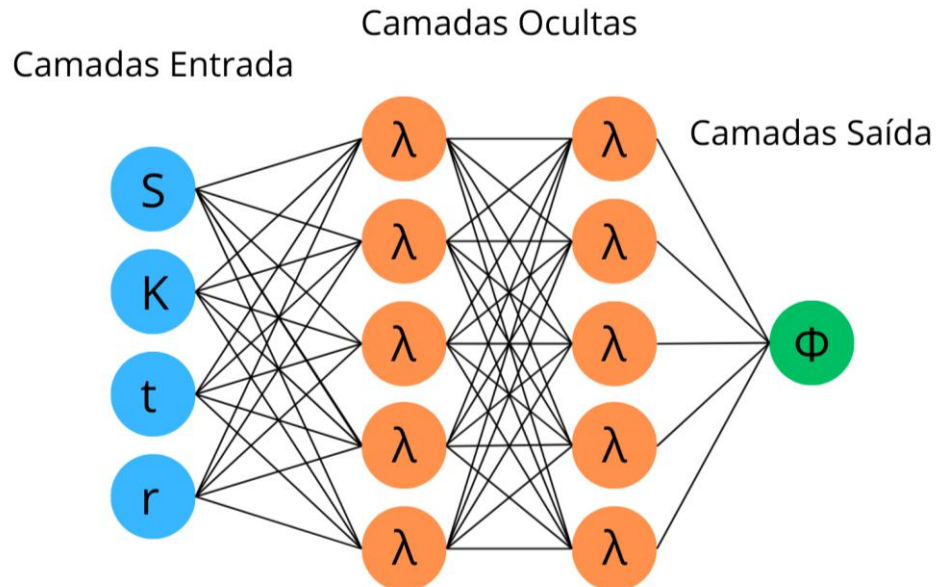


Figura 1 Esquema do funcionamento de uma Rede Neural (NN).

A capacidade de uma NN em aprender através de mapeamentos complexos de um $\mathbb{R}^n \rightarrow \mathbb{R}^m$ a torna uma ferramenta qualificada capaz de aprender a função objetivo do trabalho $V(S, K, \tau, r)$, que é uma função não linear. Porém, uma Rede Neural utilizando apenas os dados do mercado, exige uma quantidade demasiada de dados suficiente para poder cobrir todo o espaço de precificação e ainda poderia não ser capaz de ter uma consistência física, pois ela poderia por exemplo violar as condições de não-arbitragem ou falhar em reproduzir corretamente a curva de volatilidade de uma forma suave. Para poder solucionar isso, se faz necessário a utilização de estrutura PINN.

2.3 Redes Neurais Informadas pela Física (PINNs)

Uma rede neural PINN, que são um tipo de deep learning, e que é capaz de incorporar o conhecimento físico através de soluções de Equações Diferenciais Parciais no processo de aprendizado, como ocorre no presente trabalho, onde a rede do tipo $V_{\theta}ge$, parametrizada pelos pesos θ , onde a mesma é treinada e capaz de prever valores de preço de opções $V(S, t)$.

O uso do PINN se destaca na sua função de perda composta, onde ao invés de apenas ser treinada através da observação dos dados reais, como ocorre em aprendizado supervisionados tradicionais, a rede se utiliza também do aprendizado através da física

imposta sobre si, no caso do presente trabalho, EDP de Black-Scholes. A sua função de perda logo total (L_{total}), é uma soma de duas componentes, a perda gerada pelos dados (L_{data}), e a perda guiada pela física (L_{PDE}) e que é dado por:

$$L_{total} = \lambda_{data} \cdot L_{data} + \lambda_{PDE} \cdot L_{PDE}$$

- **Perda dos dados (L_{data}):** Este fator segue a mesma lógica utilizada em treino de modelos de aprendizagem supervisionado, onde ela é capaz de medir a diferença entre os preços previstos pela rede V_θ e os valores dos preços observado no mercado V_{Market} , e através do Erro Quadrático Médio (MSE) sobre um conjunto com N_{data} pontos de dados presente:

$$L_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} |V_\theta(S_i, t_i) - V_\theta|^2$$

Este termo tem como objetivo garantir com que o modelo seja ajustado aos preços reais do mercado, o que foi validado na avaliação do treinamento como visto na imagem abaixo:

- **Perda da Física (L_{PDE}):** Este termo de perda é regido por um modelo fisicamente consistente, onde ele é capaz de fazer a medição do quanto a saída que o modelo teve V_θ e quanto esta viola a EDP de Black-Scholes. Sendo assim, é definido o resíduo da EDP como:

$$f(S, t) = \frac{\partial V_\theta}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V_\theta}{\partial S^2} + rS \frac{\partial V_\theta}{\partial S} - rV_\theta$$

onde as derivadas parciais não são calculadas por diferenças finitas, mas sim de forma analítica utilizando a diferenciação automática, que se torna uma vantagem pois nos é permitido calcular as gregas, como o Delta ($\Delta = \frac{\partial V_\theta}{\partial S}$) como subproduto da rede.

A perda L_{PDE} é então calculada como o MSE deste resíduo, sendo calculado sobre um grande número de N_{PDE} pontos de colocação, que foram amostrados de maneira aleatória dentro do domínio da EDP (S, t):

$$L_{PDE} = \frac{1}{N_{PDE}} \sum_{j=1}^{N_{PDE}} |f(S_j, t_j)|^2$$

Sendo assim, o objetivo do termo $f(S, t) \rightarrow 0$ é forçado sobre todo o domínio, garantindo assim que o modelo possa aprender a EDP de Black-Scholes. Nos resultados gerados a partir do treino da rede utilizada neste presente trabalho, são válidos conforme visto na imagem abaixo.

2.4 Dados Experimentais

Para poder validar corretamente um modelo treinado de maneira supervisionada, o uso de dados reais, caso se aplique, é um dos pontos fundamentais do projeto. Para o caso de precificação de opções, a performance do modelo depende diretamente de dados de mercado reais, pois a componente L_{data} do PINN requer um conjunto de informações para que o modelo possa encontrar a melhor solução.

2.4.1 Fonte e Justificativa dos Dados

Os dados foram adquiridos via API fornecido pela empresa OPLab, uma provedora de dados do mercado financeiro, e foram consolidados em um conjunto de dados que abrange o período de 2021 a 2023 de uma cesta de ativos, sendo eles das empresa petrolífera Petrobrás (PETR4), da bolsa de valores (B3SA3), dos bancos Banco do Brasil (BBAS3), banco Bradesco (BBDC4) e o banco Itaú (ITUB4), a empresa do setor de energia (ELET3), a empresa varejista Magazine Luiza (MGLU3) e da mineradora Vale (VALE3). A escolha destes ativos foi definida devido à alta liquidez dos mesmos no período citado, o que garante que os preços observados (premium) pelo modelo sejam confiáveis diante da realidade do mercado.

2.4.2 Estrutura e Pré-processamento dos Dados

O conjunto de dados fornecido via API contém informação de uma série de contratos negociados diariamente para cada opção. Os dados presentes neste conjunto são:

- *symbol*: O ticker do contrato daquela opção.
- *time*: O registro de data/hora da observação.
- *spot_price*: O preço do ativo subjacente negociado naquele momento, S .
- *strike*: O preço de exercício do contrato, K .
- *premium*: O último preço negociado (prêmio) da opção, V .

- *days_to_maturity*: O número de dias até o vencimento e que foi convertido para unidade em ano utilizando $\tau = \text{número de dias até o vencimento} / 252$, sendo 252 o número de dias comerciais padrão de acordo com o mercado de ações brasileiro.
- *moneyness*: Uma classificação pré-calculada de um valor intrínseco e que serve para descrever o estado que aquela opção se encontra, e que diz quanto aquela opção teria de valor se ela pudesse ser exercida naquele momento, sendo elas:
 - Out The Money (OUT)
 - At the money (ATM)
 - In The Money (ITM).
- *volatility*: A volatilidade implícita (IV) de mercado naquele instante observado.
- delta, gamma, vega, theta, rho: São as "Greeks", ou gregas, que são calculadas pelo mercado de acordo com os valores apresentados de *days_to_maturity*, *strike* e *premium*.
- *option_type*: O tipo de contrato de opção que está sendo negociado, sendo elas:
 - Call (Opção de compra) - ela concede ao comprador o direito de compra de um ativo sobre o negociador. Ela ocorre quando o investidor tem a análise de que aquele ativo em questão irá subir de valor no momento do exercício.
 - Put (Opção de venda) - concede ao comprador o direito de venda sobre um ativo analisado. Ocorre quando o investidor tem a análise de que o preço daquele ativo observado irá cair no momento do exercício daquela opção.

Na arquitetura do projeto, o script responsável por preparar e disponibilizar para o modelo é o `data_loader.py`, onde ele executa:

1. **Filtragem de Tipo:** Seleccionamos exclusivamente opções do tipo CALL para manter a consistência do modelo aplicado como física no modelo.
2. **Filtragem de Liquidez:** Contratos com volume de negociação zero ou que não tiveram alteração de preço (indicando falta de liquidez) são descartados. Isso se dá devido ao executar as operações de treinamento, mostrou-se que valores

próximos a zero faziam que os resultados dentensem ao infinito, tornando inconsistente as operações por violar as condições de contorno

3. **Filtragem de Vencimento:** Opções com vencimento muito curto como $\tau < 5 \text{ dias}$, são removidas, pois, nesses regimes, o "time decay" (Theta) são extremamente altos e os preços podem se tornar errados, não seguindo a dinâmica suave esperada da EDP.
4. **Filtragem de "Moneyness":** Opções ITM e OTM, por exemplo $|\Delta| > 0,99$ ou $|\Delta| < 0,01$, são descartadas devido ao fato de seus prêmios consistem quase inteiramente de valor intrínseco ou são muito próximos de zero, oferecendo assim pouca informação para o ajuste esperado da volatilidade.

2.4.3 Variáveis de Treinamento e Validação

Após ocorrer o pré-processamento dos dados, eles são divididos em duas categorias essenciais para ocorrer o treinamento, sendo elas:

- **Variáveis de Treinamento:** O vetor de entrada x que alimenta a rede neural é composta pelas variáveis:
 - *spot_price* (S): O preço do ativo subjacente negociado naquele momento.
 - *strike* (K): O preço de exercício do contrato.
 - *taxa de juros* (r): Taxa de juros livre de risco, sendo padronizado para o cenário brasileiro a taxa Selic, com os dados no formato diário e obtidos via Banco Central.
 - *time_to_maturity* (τ): Tempo de maturidade daquele ativo calculado a partir de $\tau = \text{número de dias até o vencimento} / 252$ como dito anteriormente.

Portanto, a rede é capaz de calcular o preço a partir do vetor $V_\theta =$

$V_\theta(S, K, \tau, r)$.

- **Variável Target:** O preço do mercado *premium*, ou V_{market} , é o dado supervisionado, tendo como função única ser comparada com a saída da rede V_θ e para calcular a perda dos dados em L_{data} . É esta perda que é responsável por ajustar a volatilidade implícita que a rede tem como objetivo aprender.

3 MÉTODO

Esta seção é responsável por detalhar a arquitetura computacional e o pipeline de treinamento que foi implementado.

3.1 Pipeline de Dados e Pré-processamento

O primeiro passo no que se refere à arquitetura do modelo é o script responsável pelo Pipeline que é responsável por preparar os dados transformando em tensores e disponibilizá-los para a rede. Os dados seguem um fluxo bem estabelecido da seguinte forma:

1. **Geração e Coleta de Dados:** O estudo utilizando PINNs pode ser aplicado tanto utilizando dados sintéticos gerados com a finalidade de simular a dinâmica de preço sobre um framework, como o de Black-Scholes, quanto em dados reais, como o do que foi aplicado no projeto, que foram utilizados dados de uma cesta de ativos. A partir disto, o script responsável pelo gerenciamento dos dados aplica filtros como discutido anteriormente, como a Filtragem por Moneyness e a inclusão dos dados via API do Banco Central para obter as informações da taxa de risco (r).
2. **Extração de Variáveis:** Após o gerenciamento dos dados, ocorre uma separação dos dados que serão as nossas colunas de entrada (*features*), que é o próprio vetor $x = [S, K, \tau, r]$ onde S é o preço do exercício daquele ativo, K o preço de exercício, τ é o tempo até o vencimento e r a taxa de risco livre, e a nossa coluna do alvo supervisionado V_{market} (*premium*).
3. **Normalização de Entradas:** Para garantir a estabilidade adequada do treinamento e pelo fato de redes neurais terem sensibilidade dos hiperparâmetros do framework PINN e também na escala dos dados de entrada, é realizado uma normalização destes, utilizando o MinMaxScaler para tornar os dados em uma escala de $[0,1]$. As estatísticas de normalização, como por exemplo S_{min} e S_{max} , são armazenadas para a desnormalização das variáveis de entrada no momento que os dados serão utilizados para a física e do forward pass.
4. **Divisão e Estratificação de Dados:** O dataset é então dividido em subconjuntos de Treinamento, Teste e Validação. Para poder ter a garantia de que o modelo

possa ter uma distribuição balanceada de regimes de preço, é aplicado uma amostragem estratificada nestes dados, onde o conjunto dos dados é categorizado com base no parâmetro moneyness, e assim, é garantido essa proporção das categorias e diminui o viés sistemáticos.

5. **Preparação de Tensores:** Por fim, estes dados que são separados em lotes (batches) de tamanho de 1024, para garantir que não ocorra um problema de alocação demasiada de espaço na memória do GPU devido a grande quantidade de operações realizadas durante o treinamento. Após isso, estes lotes são convertidos em tensores e assim alimentando a entrada do modelo.

3.2 Arquitetura da Rede Neural

A arquitetura da Rede Neural PINN é uma Multilayer Perceptron (MLP) como já discutido, que incorpora alguns mecanismos com o objetivo de obter uma melhora no desempenho e estabilidade durante o treinamento:

1. **Mapeamento de Features de Fourier (FFM):** O vetor de entrada $x = [S, K, \tau, r]$ já normalizado é transformado para um espaço de alta dimensão utilizando FFM. Essa técnica é implementada com o objetivo de fazer com que a rede aprenda funções de alta frequência como a superfície da volatilidade. Isso se dá pois esta técnica visa diminuir o viés espectral (spectral bias) que é algo inerente às MLPs. Esse mapeamento ocorre através de uma camada linear simples e com pesos fixos, e que é inicializada aleatoriamente com o desvio padrão da $\sigma_{fourier}$ e faz com que ocorra uma transformação seno e cosseno dos resultados.
2. **Estrutura Shallow-Deep Híbrida:** Após ocorrer o FFM, a rede parte a ser dividida em duas redes paralelas, sendo elas uma rasa (Shallow), que é composta por duas camadas ocultas com 256 e 64 neurônios cada, e uma profunda (Deep) que é composta por quatro camadas ocultas, compostas por 521 neurônios cada, cuja as suas saídas são combinadas em uma camada linear intermediária.
3. **Funções de Ativação Localmente Adaptativas (LAAF):** Para cada camada presente na rede neural, foi aplicado uma função de ativação *SiLU* (*Sigmoid Weight Linear Unit*). O seu uso tem como objetivo fazer com que o gradiente

tenha um fluxo mais estável em redes mais profundas, em comparação a ativações tradicionais como o ReLU ou Tanh que tem valores de ativações fixos.

4. **Arquitetura de Saída Dupla:** O vetor de features da última camada de saída é direcionado para duas camadas de saída independentes que formam o principal ponto do projeto que é a implementação de uma solução direta e outra como uma solução do problema inverso:
 - a. Na solução direta, a volatilidade é tratada como um valor constante na entrada, como é solucionado a equação Black-Scholes na realidade.
 - b. Na segunda camada linear de saída, é a implementação proposta para tratar a volatilidade não como constante, mas sim como uma variável que será ajustada e capaz de obter um valor intrínseco e mais assertivo. Isso faz com que o modelo obtenha um valor do premium mais ajustado à realidade do mercado.

3.3 Implementação da Física

Em frameworks PINN, a principal característica é a implementação de uma física que seja capaz de transformar Equações Diferenciais Parciais (EDP) em um termo que é responsável por regularizar a função de perda.

1. **Definição da EDP:** No presente trabalho, a EDP em questão é a equação de Black-Scholes, onde o modelo se baseia para precificação de opções. A função $V(S, \tau)$ deve satisfazer o resíduo da EDP F , sendo dado por:

$$F = \frac{\partial V_{\theta}}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V_{\theta}}{\partial S^2} + rS \frac{\partial V_{\theta}}{\partial S} - rV_{\theta}$$

Onde temos que $V_t = \frac{\partial V_{\theta}}{\partial t}$ (Theta), $V_S = \frac{\partial V_{\theta}}{\partial S}$ (Delta), e $V_{\tau} = \frac{\partial^2 V_{\theta}}{\partial S^2}$ (Gamma).

2. **Diferenciação Automática (AD):** Os cálculos das derivadas parciais apresentadas, tanto de primeira quanto de segunda ordem, são realizados a partir da Diferenciação Automática, que é uma estrutura presente no módulo da biblioteca Pytorch, onde é utilizado o módulo apropriado para poder realizar o cálculo do gradiente do preço alvo V em relação às features do treinamento. A AD é recomendada pois é um método preciso, que não ocorre erros relacionados

a truncamentos ou arredondamento, que é comum em métodos aplicando diferenças finitas.

3. **Aplicação da Regra da Cadeia:** Um passo fundamental durante a realização dos cálculos envolve empregar diretamente a Regra da Cadeia. Visto que o aprendizado e o cálculo das derivadas ocorrem nas features de entrada já normalizadas, torna-se fundamental ajustar essas derivadas pela mesma escala da normalização realizada durante o gerenciamento dos dados. Desta forma, podemos garantir que as derivadas físicas corretas sejam obtidas em relação a S e t .
4. **Cálculo da Perda da Física:** Para cada ponto dentro do conjunto de alocação, é determinado um Resíduo da EDP F . A partir disto, a Perda da Física (L_{PDE}) é dada pelo Erro Quadrático Médio (MSE) de F em relação a zero, de modo que este cálculo é responsável por quantificar o quão bem o modelo obedece às leis da física que foram impostas para o modelo.

3.4 Orquestração do Treinamento

O pipeline completo do aprendizado da PINN reúne métodos de otimização e seleção de dados para contornar algumas dificuldades de convergência comuns presentes neste framework, como a disparidade nos gradientes.

1. **Otimização e Aprendizado Progressivo:** Para garantir um aprendizado progressivo do modelo, um método de treinamento com taxas de aprendizado (Learning Rate) decrescente foi utilizado. O treino inicia com uma taxa alta de $1e^{-4}$, o que permite que o modelo seja capaz de uma convergência rápida inicial, aprendendo o comportamento dos dados e da física presente, e a cada etapa pré-definida, essa taxa de aprendizado decai até a ordem de $1e^{-6}$, onde o é o momento que o modelo é capaz de realizar ajustes a cada ciclo de treinamento. Essas etapas mudam a cada vez que não ocorre uma melhoria significativa na taxa de Perda total L_{Total} na ordem de $1e^{-7}$ depois de uma quantidade de ciclos definido de ciclos definidos por um fator dentro de um Early Stopping (patience), logo, durante o loop de treinamento para cada fator de aprendizado é atualizado a cada vez que não ocorre uma melhoria significativa, diminuindo esta taxa de

aprendizado até que ao atingir o último estágio, o treinamento é encerrado, mesmo sem atingir o número máximo de ciclos pré-definido. Esta técnica empregada evita que o modelo execute uma quantidade de ciclos sem uma melhoria significativa na aprendizagem.

2. **Ponderação Adaptativa de Perdas (Adaptive Loss Weighting):** Uma forma de mitigar uma patologia do gradiente desequilibrado, que ocorre como consequência de uma soma estática das perdas e que é comum nestes modelos de redes neurais, é utilizar um autobalanceamento das perdas. Essa técnica visa implementar os pesos λ_{data} e λ_{PDE} como parâmetros treináveis $\log_lambda_data$ e $\log_lambda_PDE$, de forma que possa garantir que a patologia é mitigada e garanta que os termos das perdas sejam balanceados de uma forma dinâmica.
3. **Balanceamento dos Gradientes:** Um problema comum que ocorre nos treinamentos de redes PINNs é a “patologia dos gradientes”, onde as magnitudes dos termos L_{data} e L_{PDE} acabam disputando entre si, fazendo com que ocorra um desequilíbrio, pois esta diferença faz com que a rede otimize um termo de perda em detrimento ao outro. Se L_{data} dominar, a rede sofrerá *overfitting*, pois haverá uma memorização dos dados e ela não irá aprender a física. Já se ocorrer o contrário, a rede pode aprender a como obter uma solução fisicamente plausível, porém, esta solução não consegue se ajustar aos preços reais do mercado. Como solução para este problema, foi implementado uma técnica de balanceamento adaptativo de perdas, onde os pesos λ_{data} e λ_{PDE} são ajustados de forma dinâmica a cada época. O objetivo é fazer com que o gradiente de cada termo de perda tenha magnitudes comparáveis, garantido assim que a rede leve em consideração tanto as informações observadas através dos dados, como também na solução a física imposta sobre o modelo. O gráfico a seguir demonstra como os pesos λ_{data} e λ_{PDE} foram ajustados ao longo do treinamento.
4. **Fluxo de Gradiente e Ativações de Rede:** É comum que PINNs exijam redes profundas para poderem serem capazes de se aproximarem de soluções complexas de EDP. As funções de ativações presentes nas redes como a “ReLU” e “tanh”,

podem sofrer com o desaparecimento de gradientes, problema conhecido como “*vanishing gradients*”, ou até mesmo com o problema de algum neurônio saturar e prejudicar o fluxo de informação. Para isso, ao invés das funções de ativações citadas anteriormente, foi utilizada a função “SiLU” (Sigmoid-weighted Linear Unit), que também é conhecida como Swish. O uso desta função permite que, localmente, ocorra um fluxo de gradiente mais estável, evitando assim uma saturação de forma prematura e uma melhora na capacidade da rede em aprender mapeamentos complexos.

5. **"Spectral Bias" e Funções de Alta Frequência:** Em redes neurais padrões (Multilayer Perceptrons - MLPs), podem sofrer de uma tendência a aprender primeiro componentes que tenham baixa frequência da função de destino, tendo dificuldade em capturar os detalhes presentes nas funções que tenham alta frequência. Este problema é chamado como um problema de "spectral bias". No mundo das finanças, isso se torna um problema muito crítico, uma vez que quando isso ocorre, não seja possível captar detalhes de fenômenos como o "volatility smile" ou o comportamento do preço próximo ao vencimento ($t \rightarrow T$), pois estes são por natureza caracterizados por serem de alta frequência. Para solucionar isto, foi aplicado um Mapeamento com "Fourier Features" (Fourier Feature Mapping). Esta técnica faz com que antes de passar as variáveis de entrada (S, t, K) para alimentar a primeira camada da rede, estas variáveis são mapeadas para um espaço vetorial utilizando uma expansão baseada em senos e cossenos, por exemplo, uma entrada $x \rightarrow [\sin(f_1 x), \cos(f_1 x), \dots, \sin(f_n x), \cos(f_n x)]$. Desta forma, esta transformação durante o pré processamento da rede permite com que ela aprenda de uma forma mais fácil as funções com alta frequência, e é capaz de capturar as curvaturas acentuadas vistas nas superfícies de volatilidade e preço.
6. **Regularização da Superfície de Volatilidade para Problemas Inversos:** A regularização do parâmetro da volatilidade (σ) a partir dos preços observados do mercado (V) é por definição um problema inverso mal posto (ill-posed inverse problem). Esta condição surge pelo fato de perda de dados (L_{data}) focar unicamente no erro de precificação, não tendo assim um gradiente que é capaz de

representar o parâmetro σ . Como consequência, a rede pode convergir para uma solução que tem alta precisão em prever os preços (V), ela falha em inferir uma superfície de volatilidade que seja suave e economicamente plausível, pois o otimizador encontra um atalho onde a saída σ acaba se tornando ruidosa ou pode colapsar em valores que não representam a realidade, visto que o impacto na perda total é ignorado pelos termos de preço, gerando assim uma patologia similar à dos gradientes desequilibrados citados anteriormente (WANG; TENG; PERDIKARIS, 2020). Para solucionar este problema, houvesse a necessidade de implementar um terceiro termo de perda, focado na Regularização de Suavidade da volatilidade (L_{reg}), onde ele tem como função penalizar soluções onde a superfície da volatilidade é complexa ou ruidosa. Esta implementação consiste em um cálculo das “gregas da volatilidade”, calculando as suas derivadas parciais a partir da própria saída σ da rede em relação às variáveis de entrada $x = [S, K, \tau]$, com o auxílio da diferenciação automática, o termo é definido como a média dos quadrados dessas derivadas (MSE) de acordo com a equação abaixo, visando minimizar este termo em conjunto com os demais termos, fazendo com que o otimizador encontre uma superfície com menores gradientes, o tornando mais suave e que satisfaça a física da EDP e se ajuste ao mercado:

$$L_{reg} = E \left[\left(\frac{\partial \sigma}{\partial S} \right)^2 + \left(\frac{\partial \sigma}{\partial K} \right)^2 + \left(\frac{\partial \sigma}{\partial \tau} \right)^2 \right]$$

4 RESULTADOS

4.1 Panorama das Opções no Brasil:

Antes de mergulhar nas estratégias e dinâmicas do mercado de opções, é essencial compreender algumas características únicas do ambiente brasileiro, especialmente na B3 — a bolsa de valores oficial do Brasil.

Uma das primeiras peculiaridades que chama atenção é a predominância das opções de compra, conhecidas como calls. Elas representam a maior parte das negociações, o que influencia diretamente o comportamento dos investidores e a formação de preços no mercado.

Outro ponto relevante é a concentração de liquidez em poucos ativos. Papéis como PETR4 (Petrobras) e VALE3 (Vale) dominam o volume de negociações, tornando-se os principais alvos de estratégias com opções. Essa concentração pode limitar a diversidade de operações, mas também oferece maior previsibilidade e profundidade para quem opera nesses ativos.

Por fim, o calendário de vencimento das opções na B3 possui uma regra específica: elas expiram sempre na terceira segunda-feira de cada mês. Esse detalhe é fundamental para o cálculo do tempo até a expiração representado pela variável T e impacta diretamente o valor das opções e a escolha das estratégias mais adequadas.

4.2 Avaliação da Convergência:

Durante o processo de treinamento, a convergência do modelo foi monitorada cuidadosamente por meio de dois indicadores principais, representados em gráficos históricos que ajudam a visualizar o desempenho e a evolução da otimização.

- O gráfico de perdas oferece uma visão clara da eficácia da otimização das diferentes componentes da função de perda. Observa-se uma queda rápida e consistente tanto na perda associada aos dados de mercado quanto na perda relacionada à equação diferencial parcial. Essa trajetória indica que o otimizador conseguiu equilibrar com sucesso o ajuste aos dados reais com o respeito às leis físicas que regem o modelo. A estabilização das curvas em níveis baixos, sem oscilações abruptas ao final do treinamento, sugere que o modelo convergiu de forma robusta, sem cair na armadilha da memorização excessiva dos dados.

Histórico de Perdas e Taxa de Aprendizado

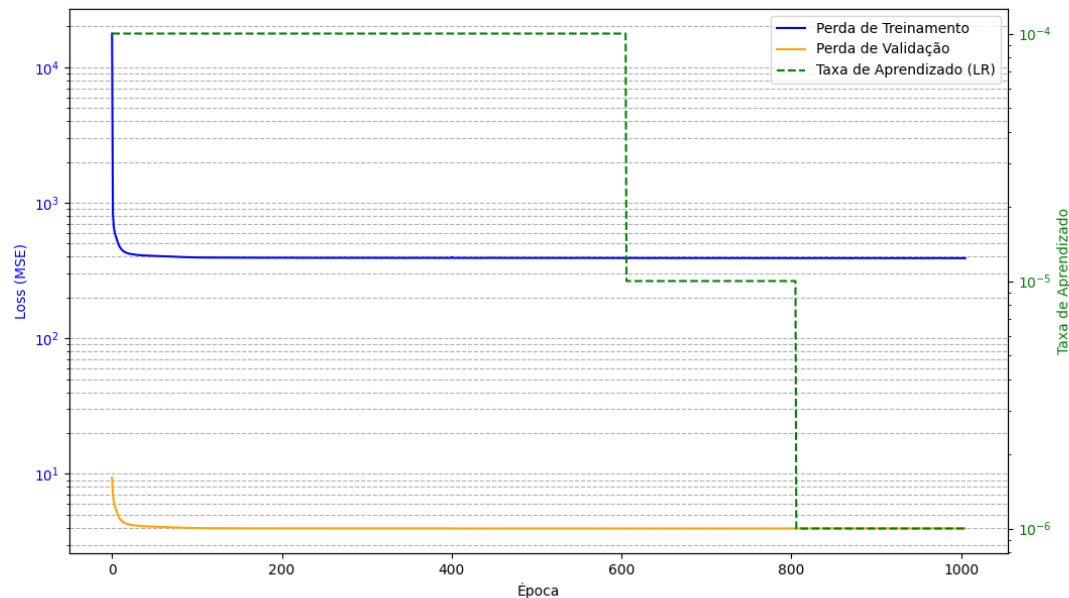


Figura 2 Histórico de Perdas durante o treinamento.

- Já o gráfico de evolução dos pesos ilustra como o mecanismo de balanceamento adaptativo atua ao longo do treinamento. Inicialmente, o peso atribuído aos dados de mercado (*weight_data*) é elevado, o que permite ao modelo aprender rapidamente a precificar os ativos com base nas observações reais. Com o avanço do treinamento, os pesos associados à equação física (*weight_pde*) e à regularização (*weight_reg*) ganham relevância, redirecionando o foco da otimização para a consistência matemática e a suavidade da superfície de volatilidade. Essa transição é essencial para garantir que o modelo não apenas reproduza os preços observados, mas também incorpore os princípios fundamentais da teoria de Black-Scholes.

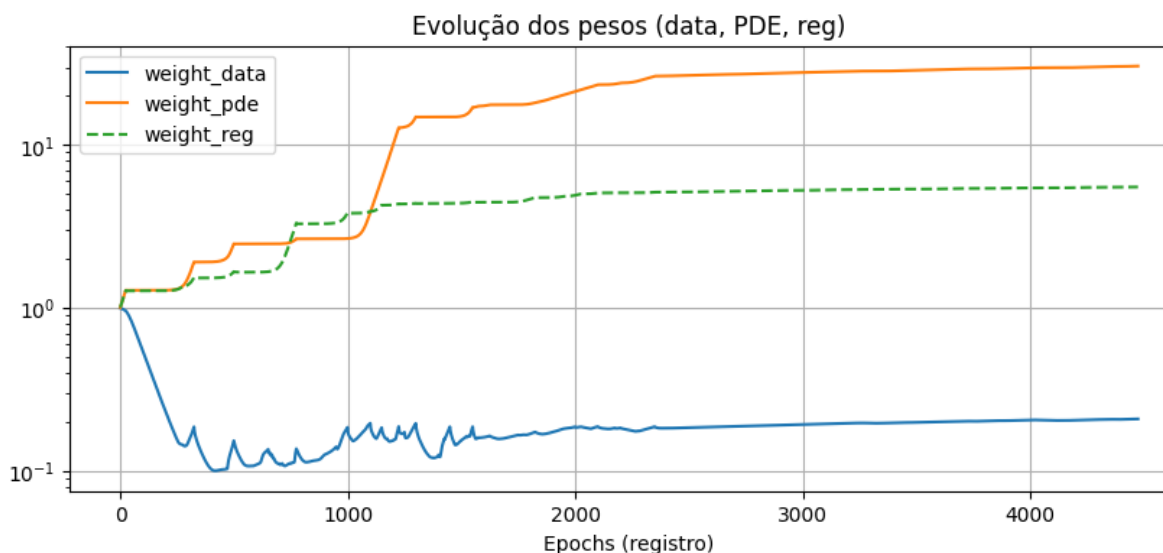


Figura 3 Evolução dos Pesos Dinâmicos.

4.3 Validação da Precificação

Para avaliar a capacidade do modelo em precificar opções com precisão, foram realizadas comparações diretas entre os preços previstos e os preços reais observados no mercado. Três visualizações principais foram utilizadas para essa análise:

- O gráfico de dispersão *Previsto vs. Real* oferece uma visão clara da performance do modelo. Nele, os preços previstos (eixo Y) são comparados diretamente com os preços de mercado (eixo X). A proximidade dos pontos à linha diagonal $y=x$ revela uma correlação quase perfeita, indicando que o modelo consegue replicar com alta fidelidade os preços reais em uma ampla gama de valores. Essa aderência sugere que o modelo não apenas aprendeu os padrões do mercado, mas também

os generalizou de forma eficaz.

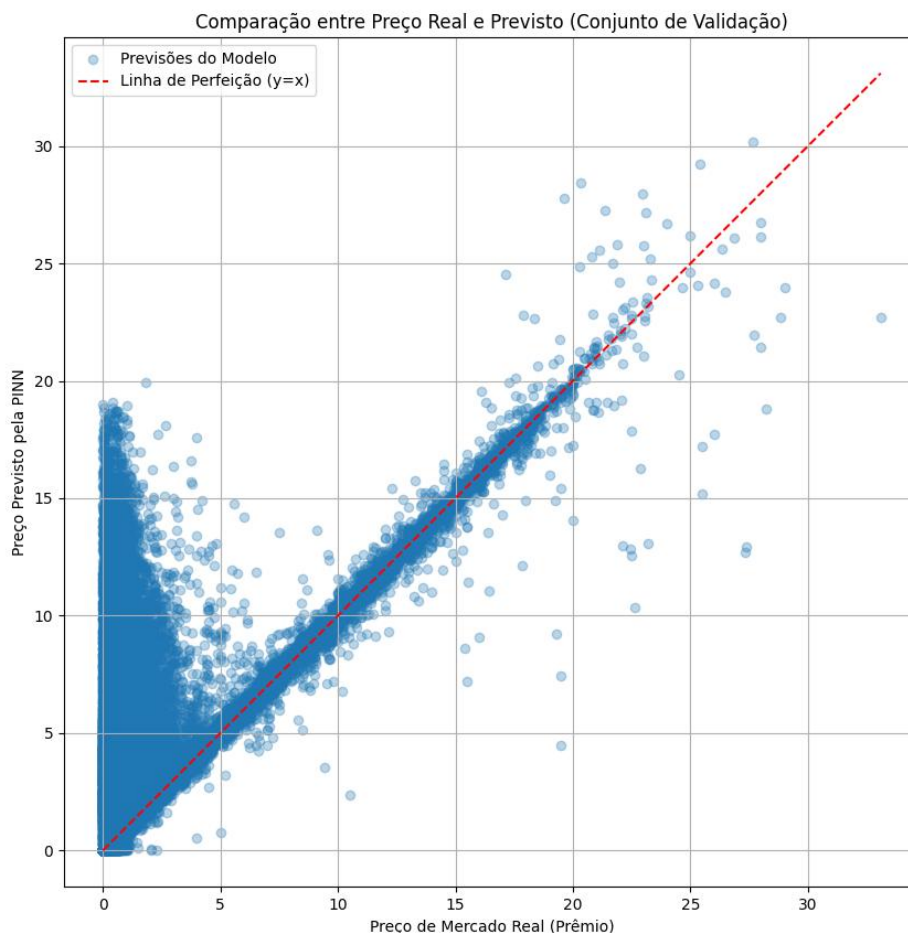


Figura 4 Comparação dos Preços previstos.

- A comparação entre a superfície de preços gerada pelo modelo PINN e a superfície real do mercado mostra uma impressionante semelhança. O modelo foi capaz de capturar nuances importantes da estrutura de preços, como a curvatura e a forma da superfície em função do *moneyness* e do tempo até o vencimento. Essa aderência visual reforça a robustez do modelo na representação da dinâmica de precificação observada no mercado brasileiro.

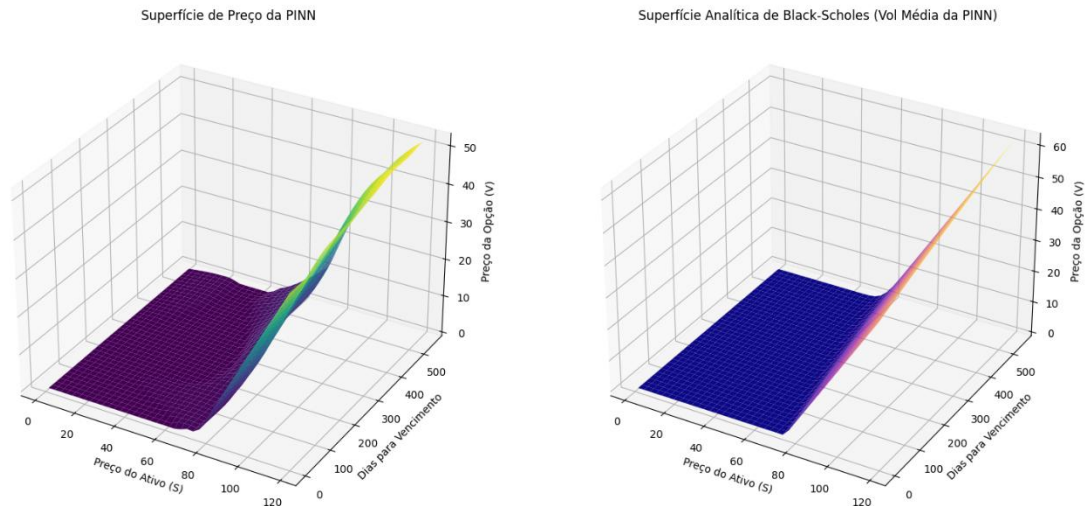


Figura 5 Superfície de Black-Scholes Gerada pelo PINN e Resolução Analítica.

4.4 Validação da Física

Um dos aspectos mais importantes na validação de modelos baseados em equações diferenciais é verificar se eles respeitam a estrutura matemática que os fundamenta. No caso do modelo PINN aplicado à precificação de opções, essa verificação foi feita por meio da análise da superfície de resíduo da equação de Black-Scholes.

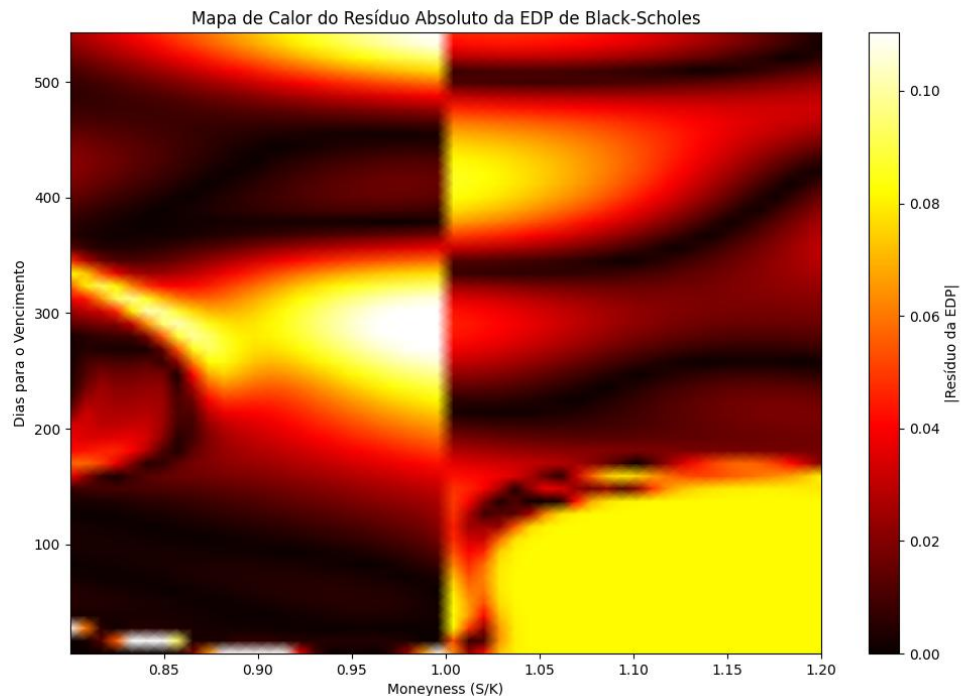


Figura 6 Heatmap do Resíduo da EDP.

O gráfico acima apresenta o valor do resíduo da equação diferencial parcial ao longo de todo o domínio de entrada, ou seja, em diferentes combinações de preço do ativo subjacente (S) e tempo até o vencimento (t). A superfície resultante se mantém uniformemente próxima de zero, o que indica que o modelo não apenas aprendeu a precificar corretamente, mas também incorporou com sucesso a dinâmica da equação de Black-Scholes.

Essa aderência à estrutura matemática é um sinal claro de que o modelo não está apenas ajustando os dados de forma superficial, mas sim internalizando os princípios físicos que regem o comportamento dos preços das opções. Em outras palavras, o modelo não apenas "imita" o mercado ele entende sua lógica.

4.5 Descoberta de Estruturas Financeiras

Ao tratar a volatilidade σ como uma saída da rede, o modelo resolveu um problema inverso: inferiu diretamente dos preços estruturas complexas do mercado. Isso permitiu capturar a dinâmica da volatilidade sem suposições prévias, respeitando a lógica da equação de Black-Scholes.

- Os gráficos abaixo mostram que o modelo PINN conseguiu capturar o fenômeno conhecido como "sorriso da volatilidade" uma variação típica ignorada pelo modelo clássico de Black-Scholes, que assume volatilidade constante. O PINN inferiu uma superfície onde opções *out-of-the-Money* (OTM) e *In-the-Money* (ITM) apresentam volatilidade mais alta que opções *At-the-Money* (ATM), refletindo com precisão uma estrutura de mercado essencial para a precificação de derivativos exóticos e estratégias de gestão de risco.

Superfície de Volatilidade Implícita Aprendida (PINN)

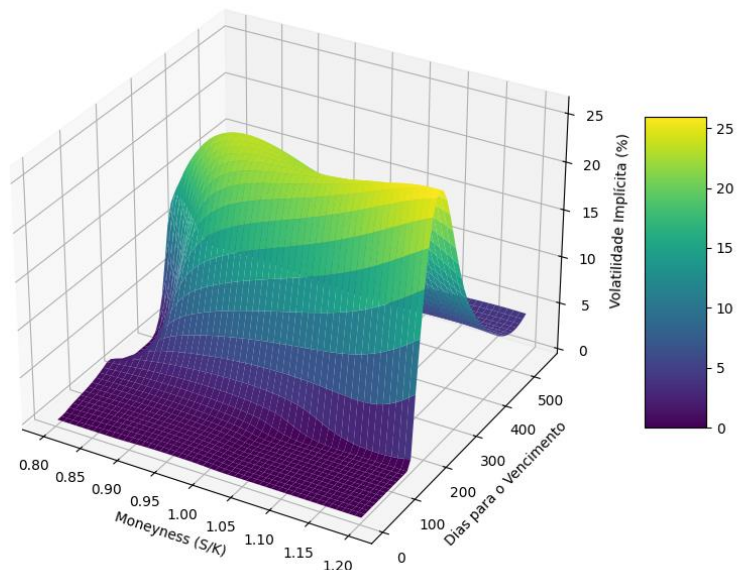


Figura 7 Superfície de Volatilidade.

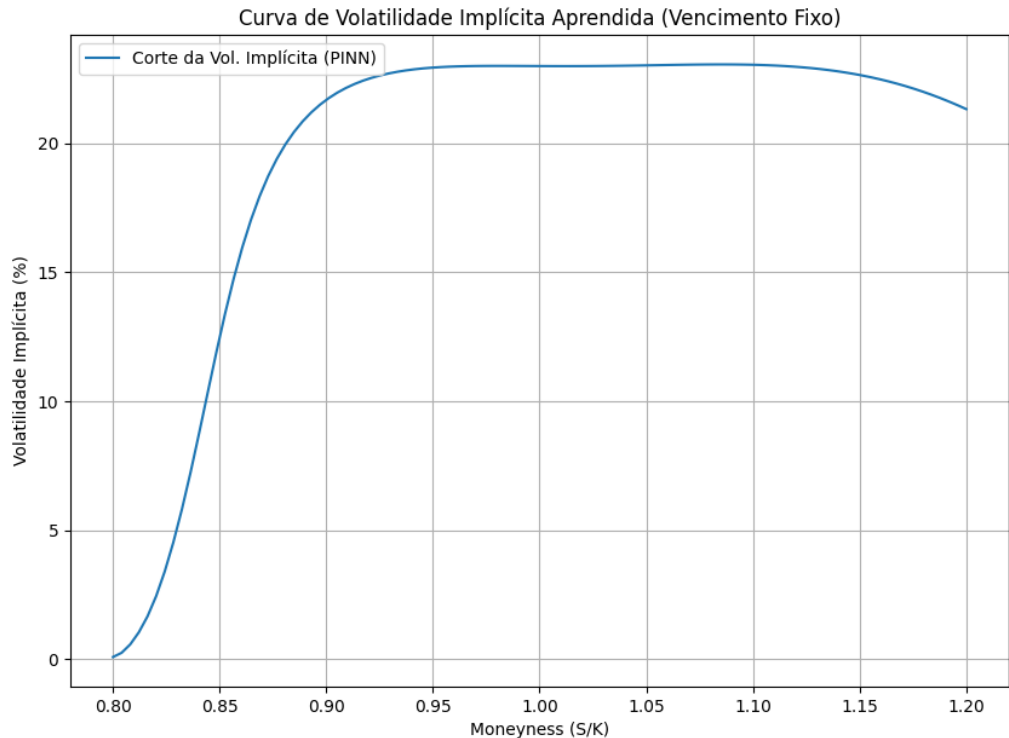


Figura 8 Curva "Volatility Smile".

5 DISCUSSÕES

Os resultados demonstram que o modelo PINN desenvolvido é acurado, ao replicar os preços de mercado com baixo erro (L_{data}); fisicamente consistente, ao obedecer à equação de Black-Scholes (L_{PDE}); e capaz de inferência, ao resolver o problema inverso de descobrir a superfície de volatilidade implícita uma estrutura complexa e não trivial do mercado (L_{reg}). Esses três aspectos comprovam a robustez da abordagem, que combina precisão, aderência à teoria financeira e capacidade de extrair padrões diretamente dos dados.

A abordagem PINN se destaca em relação aos métodos tradicionais por dois motivos principais. Primeiro, ela não parte da suposição de volatilidade constante, como ocorre na calibração clássica, permitindo aprender o *volatility smile* diretamente dos dados. Segundo, diferentemente de modelos puramente de machine learning, como MLPs padrão, o PINN incorpora restrições físicas neste caso, a equação de Black-Scholes o que assegura previsões financeiramente coerentes, evita resultados implausíveis e melhora a capacidade de generalização do modelo.

O treinamento do modelo PINN demanda alto custo computacional, exigindo GPUs e longos períodos de otimização. Além disso, sua implementação atual está restrita à precificação de opções europeias de compra, conforme a formulação da EDP



em *physics.py*. Essas limitações, no entanto, abrem oportunidades claras para avanços futuros, como:

1. adaptação do modelo para opções americanas, exigindo a reformulação da EDP como uma inequação variacional;
2. expansão para outros ativos e tipos de derivativos;
3. incorporação de modelos mais sofisticados, como o de Heston, que trata a volatilidade como um processo estocástico e substituiria a EDP de Black-Scholes na função de perda L_{PDE} .

6 CONSIDERAÇÕES FINAIS

Em resumo, este projeto validou com êxito a aplicação de um framework PINN para a precificação de opções. O modelo alcançou alta acurácia em dados fora da amostra, demonstrou consistência com a equação de Black-Scholes e foi capaz de inferir parâmetros ocultos, como a superfície de volatilidade. Esses resultados reforçam o potencial da abordagem híbrida que combina dados e conhecimento físico como uma solução promissora para desafios complexos em finanças quantitativas.

7 REFERÊNCIAS BIBLIOGRÁFICAS

DHIMAN, Ashish; HU, Yibei. Physics Informed Neural Network for Option Pricing. *arXiv preprint arXiv:2312.06711*, 2023.

HULL, John C. *Opções, futuros e outros derivativos*. 9. ed. Porto Alegre: Bookman, 2016.

JAGTAP, Ameya D.; KAWAGUCHI, Kenji; KARNIADAKIS, George Em. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. *arXiv preprint arXiv:1909.12228*, 2020.

LIU, Jixiao; MA, Yecheng. Predicting Implied Volatility Via Improved Pinns-Informer to the BS Equation. *Mathematical Modeling and Algorithm Application*, v. 4, n. 1, 2025.

NOGUER I ALONSO, Miquel; ANTOLÍN CAMARENA, Julián. Physics-Informed Neural Networks (PINNs) in Finance. *SSRN*, 2023. Disponível em: <https://ssrn.com/abstract=4598180>.

SANTOS, Daniel de Souza; FERREIRA, Tiago Alessandro Espínola. Neural Network Learning of Black-Scholes Equation for Option Pricing. *arXiv preprint arXiv:2405.05780*, 2024.

SANTOS, Henrique Guthierry Ataíde. *SIMULAÇÕES COMPUTACIONAIS DE ENGENHARIA ATRAVÉS DE REDES NEURAIS INFORMADAS POR FÍSICA*. 2024. Monografia (Trabalho de Conclusão de Curso) - Universidade Federal de Pernambuco, Recife, 2024.

TANCIK, Matthew et al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *arXiv preprint arXiv:2006.10739*, 2020.



Anais do Simpósio Acadêmico de Engenharia de Produção (SAEPRO) da EEL-USP

IX SAEPRO – 25 e 26 de novembro de 2025

WANG, Sifan; TENG, Yujun; PERDIKARIS, Paris. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv preprint arXiv:2001.04536*, 2020.

[AN INTUITIVE EXPLANATION OF BLACK-SCHOLES]. An Intuitive Explanation of Black–Scholes. Publicado em: 28 set. 2024. Disponível em: <https://www.gauthmath.com/solution/1826601225493602/01-E090032-7-Imagine-que-voc-navegando-por-um-site-de-not-cias-tenha-encontrado->.