



**UNIVERSIDADE FEDERAL DO MARANHÃO**

Fundação Instituída nos termos da Lei 5.152 de 21/10/1966 - São Luís - MA

**Centro de Ciências Exatas e Tecnologia**  
**Curso de Matemática – Bacharelado**

Davi Komura de Castro Lobato

# **Solução da Equação de Poisson usando Redes Neurais**

São Luís - MA  
2023

Davi Komura de Castro Lobato 

## **Solução da Equação de Poisson usando Redes Neurais**

Monografia (Trabalho de Conclusão de Curso) apresentada à Coordenadoria dos cursos de Matemática, da Universidade Federal do Maranhão, como requisito parcial para obtenção do grau de Bacharel em Matemática.

Curso de Matemática – Bacharelado

Universidade Federal do Maranhão

Orientador: Prof. Me. Cleber Araújo Cavalcanti

São Luís - MA

2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Diretoria Integrada de Bibliotecas/UFMA

Komura de Castro Lobato, Davi.  
Solução da Equação de Poisson usando Redes Neurais /  
Davi Komura de Castro Lobato. - 2023.  
41 p.

Orientador(a): Prof. Me. Cleber Araújo Cavalcanti.  
Monografia (Graduação) - Curso de Matemática,  
Universidade Federal do Maranhão, São Luís, 2023.

1. Diferenças Finitas. 2. Equação de Poisson. 3.  
Equações Diferenciais Parciais. 4. Métodos Numéricos. 5.  
Redes Neurais Informadas pela Física. I. Araújo  
Cavalcanti, Prof. Me. Cleber. II. Título.

Davi Komura de Castro Lobato 

## Solução da Equação de Poisson usando Redes Neurais

Monografia (Trabalho de Conclusão de Curso) apresentada à Coordenadoria dos cursos de Matemática, da Universidade Federal do Maranhão, como requisito parcial para obtenção do grau de Bacharel em Matemática.

Trabalho **APROVADO**. São Luís - MA, 23/02/2024

---

**Prof. Me. Cleber Araújo Cavalcanti**  
DEMAT/UFMA  
Orientador

---

**Prof. Dr. João de Deus Mendes da Silva**  
DEMAT/UFMA  
Primeiro Examinador

---

**Prof. Me. Gabriel Braga Marques Teobaldo**  
FEP/IF/USP  
Segundo Examinador

*Aos apaixonados por Matemática e redes neurais, desvendando os segredos do universo.*

# Agradecimentos

Gostaria de iniciar meus agradecimentos expressando minha gratidão a Deus, cuja presença e bênçãos estiveram presentes em todos os momentos deste trabalho.

Aos meus pais, Luana Komura e Robson Ferreira, sou imensamente grato pelo amor, apoio incondicional e pela constante inspiração que me proporcionaram. Agradeço também à minha namorada, Isis Oliveira, pela compreensão, paciência e incentivo que foram fundamentais para minha jornada acadêmica.

Aos meus colegas do Grupo Sarrus, Ygor Carvalho, Renata França, Carla Beatriz, Rafael Vieira, Claudio Rickielly, Yangerfeson Silva e Ronaldo Pinheiro, agradeço pela parceria, troca de experiências e apoio mútuo ao longo desses anos.

Ao meu orientador, Cleber Cavalcanti, agradeço pela orientação dedicada, pelos ensinamentos valiosos e pela confiança em meu trabalho. Sua orientação foi fundamental para o desenvolvimento desta pesquisa.

A todos os professores do curso de Matemática da UFMA, agradeço pela qualidade do ensino, pelas oportunidades de aprendizado e pela inspiração que proporcionaram ao longo da minha formação acadêmica.

Por fim, agradeço a todos que, de alguma forma, contribuíram para a realização desta monografia. Seu apoio e colaboração foram essenciais para este trabalho. Muito obrigado.

*“A matemática é a linguagem com a qual Deus escreveu o universo.”*

Galileu Galilei (1564 – 1642)

# Resumo

Este trabalho aborda a resolução da Equação de Poisson, uma Equação Diferencial Parcial (EDP), utilizando métodos numéricos e Redes Neurais Informadas pela Física (PINNs). Inicia-se com uma introdução às EDPs e à Equação de Poisson em uma e duas dimensões, incluindo as condições de contorno. Em seguida, explora-se as Redes Neurais, sua arquitetura e função de ativação, destacando a abordagem PINN.

A monografia compara o método de Diferenças Finitas com as PINNs, abordando precisão, eficiência computacional e facilidade de implementação. O método de Diferenças Finitas é detalhado, incluindo sua aplicação na Equação de Poisson. A comparação quantitativa dos métodos é discutida em termos de robustez numérica.

Por fim, são apresentados experimentos e resultados da solução da Equação de Poisson com ambos os métodos em uma e duas dimensões. Os resultados destacam a eficácia e eficiência dos métodos, oferecendo insights sobre sua aplicabilidade. O trabalho foi realizado utilizando Python.

**Palavras-chave:** Equações Diferenciais Parciais, Métodos Numéricos, Diferenças Finitas, Redes Neurais Informadas pela Física, Equação de Poisson.

# Zusammenfassung

Diese Arbeit behandelt die Lösung der Poisson-Gleichung, einer partiellen Differentialgleichung (PDE), unter Verwendung numerischer Methoden und Physik-informierten neuronalen Netzen (PINNs). Sie beginnt mit einer Einführung in PDEs und die Poisson-Gleichung in einer und zwei Dimensionen, einschließlich der Randbedingungen. Anschließend wird auf neuronale Netze, ihre Architektur und Aktivierungsfunktion eingegangen, wobei der Fokus auf dem PINN-Ansatz liegt.

Die Monographie vergleicht die Methode der Finite Differenzen mit PINNs hinsichtlich Genauigkeit, Rechenleistung und Implementierungsaufwand. Die Methode der Finite Differenzen wird im Detail erläutert, einschließlich ihrer Anwendung auf die Poisson-Gleichung. Der quantitative Vergleich der Methoden wird im Hinblick auf numerische Robustheit diskutiert.

Abschließend werden Experimente und Ergebnisse zur Lösung der Poisson-Gleichung mit beiden Methoden in einer und zwei Dimensionen präsentiert. Die Ergebnisse heben die Wirksamkeit und Effizienz der Methoden hervor und bieten Einblicke in ihre Anwendbarkeit. Die Arbeit wurde unter Verwendung von Python durchgeführt.

**Schlüsselwörter:** Partielle Differentialgleichungen, Numerische Methoden, Finite Differenzen, Physik-Informierte Neuronale Netze, Poisson-Gleichung.

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>2</b>	<b>EQUAÇÕES DIFERENCIAIS PARCIAIS (EDPS)</b>	<b>12</b>
<b>2.1</b>	<b>Introdução às EDPs</b>	<b>12</b>
2.1.1	Principais Tipos de EDPs	12
2.1.2	Métodos de Resolução de EDPs	13
<b>2.2</b>	<b>Equação de Poisson 1D</b>	<b>14</b>
2.2.1	Condições de Contorno para a Equação de Poisson 1D	15
<b>2.3</b>	<b>Equação de Poisson 2D</b>	<b>15</b>
2.3.1	Condições de Contorno para a equação de Poisson 2D	16
<b>3</b>	<b>REDES NEURAIS</b>	<b>17</b>
<b>3.1</b>	<b>Introdução às Redes Neurais</b>	<b>17</b>
<b>3.2</b>	<b>Tipos de Rede neurais</b>	<b>18</b>
<b>3.3</b>	<b>Arquitetura de Redes Neurais</b>	<b>19</b>
3.3.1	Camadas de uma Rede Neural	19
3.3.2	Neurônios	20
3.3.2.1	Estrutura de um Neurônio	20
3.3.3	Funções de Ativação	21
<b>4</b>	<b>MATERIAIS E MÉTODOS</b>	<b>23</b>
<b>4.1</b>	<b>Rede neurais Informadas Pela Física</b>	<b>23</b>
4.1.1	A arquitetura da rede neural informada pela física	23
<b>4.2</b>	<b>Função de Perda</b>	<b>25</b>
4.2.1	Função de Ativação	26
4.2.2	A dinâmica dos PINNs e o processo de otimização	27
<b>5</b>	<b>COMPARAÇÃO ENTRE O MÉTODO DE DIFERENÇAS FINITAS E AS REDES NEURAIS INFORMADAS PELA FÍSICA</b>	<b>28</b>
<b>5.1</b>	<b>O Método de Diferenças Finitas</b>	<b>28</b>
5.1.1	Diferenças finitas na equação de Poisson 1D	28
5.1.2	Aplicação das Diferenças Finitas na Equação de Poisson 2D	30
<b>5.2</b>	<b>Comparação Quantitativa</b>	<b>30</b>
5.2.1	Precisão	30
5.2.2	Eficiência Computacional	31
5.2.3	Robustez Numérica	31

5.2.4	Facilidade de Implementação . . . . .	31
<b>6</b>	<b>EXPERIMENTOS E RESULTADOS . . . . .</b>	<b>32</b>
<b>6.1</b>	<b>Solução da Equação de Poisson 1D . . . . .</b>	<b>32</b>
<b>6.2</b>	<b>Solução da Equação de Poisson 2D . . . . .</b>	<b>36</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>41</b>

# 1 Introdução

As Equações Diferenciais Parciais (EDPs) desempenham um papel fundamental na modelagem matemática de diversos fenômenos físicos, biológicos e engenharias. A resolução dessas equações é frequentemente desafiadora e requer o emprego de métodos numéricos eficazes. Neste contexto, as Redes Neurais (RNs) têm se destacado como uma abordagem poderosa e inovadora para resolver EDPs de forma eficiente e precisa.

No âmbito deste estudo, exploraremos a aplicação das RNs na resolução da Equação de Poisson, tanto em uma dimensão (1D) quanto em duas dimensões (2D). A Equação de Poisson é uma EDP importante que descreve fenômenos de difusão eletrostática, temperatura e potencial gravitacional, entre outros. O uso de RNs para resolver a Equação de Poisson tem sido objeto de intensa pesquisa devido à sua capacidade de lidar com problemas complexos de forma eficaz.

Para tanto, utilizaremos Python como principal ferramenta de implementação, aproveitando as bibliotecas Numpy, Pandas e Matplotlib para manipulação de dados, cálculos numéricos e visualização de resultados. Além disso, exploraremos a arquitetura das RNs informadas pela física, que incorporam conhecimento físico ao processo de aprendizagem da rede, tornando-as mais eficientes e precisas na resolução de EDPs.

Por fim, este estudo também comparará quantitativamente o método de Diferenças Finitas, amplamente utilizado na resolução de EDPs, com as RNs informadas pela física. Serão analisados aspectos como precisão, eficiência computacional, robustez numérica e facilidade de implementação, a fim de avaliar o desempenho relativo desses métodos na resolução da Equação de Poisson em diferentes dimensões.

Dessa forma, esta monografia pretende contribuir para o avanço do conhecimento na área de resolução numérica de EDPs, apresentando uma abordagem inovadora e comparativa entre métodos tradicionais e novas técnicas baseadas em RNs informadas pela física.

## 2 Equações Diferenciais Parciais (EDPs)

Nesta seção, as equações diferenciais parciais serão brevemente apresentadas, juntamente com alguns conceitos matemáticos que são úteis ao estudá-las. As Equações Diferenciais Parciais (EDPs) representam um dos pilares fundamentais da matemática aplicada e desempenham um papel crucial na modelagem de fenômenos complexos que evoluem no espaço e no tempo. Neste capítulo, exploraremos os conceitos básicos das EDPs, classificaremos os tipos de EDPs e concluiremos, mostrando alguns métodos para a resolução de EDPs.

### 2.1 Introdução às EDPs

Uma equação diferencial parcial, conforme (EVANS, 2022), “é uma equação que envolve uma função desconhecida de duas ou mais variáveis e algumas de suas derivadas parciais”.

**Definição 2.1.** *Uma equação diferencial parcial é apresentada na seguinte forma*

$$F(D^k u(x), D^{k-1} u(x), \dots, Du(x), u(x), x) = 0 \quad (2.1)$$

na qual  $k$  representa a ordem da equação diferencial parcial,

$$F : \mathbb{R}^{N^k} \times \mathbb{R}^{N^{k-1}} \times \dots \times \mathbb{R}^N \times \mathbb{R} \times U \rightarrow \mathbb{R} \quad (2.2)$$

com  $U \subseteq \mathbb{R}^N$  aberto conexo (domínio).

Na solução de Equações Diferenciais Parciais (EDPs), o objetivo é encontrar todas as funções  $u$  que satisfaçam a equação (2.1), considerando as possíveis restrições impostas pelas condições de contorno em uma região específica  $\Gamma$  da fronteira do domínio  $U$ . A expressão “resolver” implica, de maneira ideal, na obtenção de soluções que sejam simples e explícitas ou, quando essa abordagem não for factível, na dedução da existência e de outras propriedades das soluções.

#### 2.1.1 Principais Tipos de EDPs

Nesta subseção, iremos abordar os quatro principais tipos de equações diferenciais parciais: a equação de Laplace, a equação de transporte, a equação do calor e a equação da onda. Estes modelos matemáticos desempenham um papel crucial na descrição de fenômenos físicos em diversas áreas da ciência e engenharia.

## a) Equação de Laplace

A equação de Laplace é uma das mais importantes equações diferenciais parciais para representar fenômenos estacionários. Na qual é representada da seguinte forma

$$\Delta u = 0 \quad (2.3)$$

## b) Equação do Transporte

Representada por (EVANS, 2022) como

$$u_t + b \cdot Du = 0 \quad (2.4)$$

## c) Equação (da Difusão) do Calor

Como mostrado em (EVANS, 2022) a equação do calor “também conhecida como equação de difusão, descreve em aplicações típicas a evolução no tempo da densidade  $U$  de alguma quantidade, como calor, concentração química *etc.*” que é definida como

$$u_t - c^2 \Delta u = 0 \quad (2.5)$$

e a equação de calor não homogênea

$$u_t - c^2 \Delta u = f \quad (2.6)$$

ambas sujeitas a condições iniciais e a condições de contorno.

## d) Equação das Ondas

Representada em (EVANS, 2022) como

$$u_{tt} - c^2 \Delta u = 0 \quad (2.7)$$

e a equação da onda não homogênea

$$u_{tt} - c^2 \Delta u = f \quad (2.8)$$

ambas sujeitas a condições iniciais e a condições de contorno.

## 2.1.2 Métodos de Resolução de EDPs

Resolver equações diferenciais parciais (EDPs) é um desafio que requer uma compreensão profunda de várias técnicas matemáticas. Existem vários métodos para resolver EDPs, cada um com suas próprias vantagens e desvantagens. Vamos explorar alguns desses métodos em detalhes.

- Métodos Analíticos

### 1. Método das Características

O método das características é uma técnica poderosa para resolver EDPs de primeira ordem. Ele transforma a EDP em uma série de equações diferenciais ordinárias (EDOs) ao longo de curvas chamadas características. Ao resolver essas EDOs, podemos obter a solução da EDP original.

### 2. Método de Separação de Variáveis

O método de separação de variáveis é uma técnica comum para resolver EDPs lineares homogêneas. Este envolve assumir que a solução pode ser escrita como o produto de funções, cada uma das quais depende apenas de uma das variáveis independentes. Isso transforma a EDP em um conjunto de EDOs que podem ser resolvidas separadamente.

### 3. Método de Transformada de Fourier

A transformada de Fourier é uma ferramenta poderosa para resolver EDPs. Ela transforma a EDP de um domínio espacial ou temporal para um domínio de frequência, onde a equação pode ser mais fácil de resolver. A solução é então transformada de volta para o domínio original usando a transformada inversa de Fourier.

- Métodos Numéricos

#### 1. Método das Diferenças Finitas

O método das diferenças finitas é uma técnica numérica para resolver EDPs. Ele aproxima a EDP por uma equação de diferenças finitas, que pode ser resolvida usando técnicas de Álgebra Linear. Este método é particularmente útil quando a solução exata é difícil de obter.

#### 2. Método dos Elementos Finitos

O método dos elementos finitos é outra técnica numérica para resolver EDPs. Ele divide o domínio da solução em uma malha de elementos finitos e aproxima a solução dentro de cada elemento. Este método é amplamente utilizado em simulações de engenharia e física.

## 2.2 Equação de Poisson 1D

A equação de Poisson unidimensional é uma equação diferencial parcial (EDP) elíptica não homogênea que desempenha um papel fundamental em diversas áreas da física, como eletrostática, eletrodinâmica, mecânica dos fluidos e transferência de calor. Esta é uma extensão direta da equação de Laplace unidimensional e pode ser escrita como

$$\frac{d^2u}{dx^2} = -f(x), \quad x \in [a, b] \subseteq \mathbb{R} \quad (2.9)$$

Nesta equação,  $u$  é a função desconhecida que estamos tentando determinar, que pode representar quantidades físicas como o potencial elétrico em um fio condutor. A função  $f$  é a função de fonte conhecida, que representa a distribuição de carga ou matéria ao longo da linha unidimensional.

A solução da equação de Poisson unidimensional nos dá a distribuição do potencial ao longo da linha em resposta à distribuição da fonte  $f$ .

### 2.2.1 Condições de Contorno para a Equação de Poisson 1D

As condições de contorno são essenciais para a solução da equação de Poisson unidimensional, pois elas especificam o comportamento da solução nas extremidades do intervalo  $x \in [a, b]$ , que define o domínio do problema.

As condições de contorno mais comuns para a equação de Poisson 1D são as condições de Dirichlet, onde o valor da função é especificado nas extremidades do intervalo, e as condições de Neumann, onde a derivada da função é especificada nas extremidades. Por exemplo, em um problema eletrostático, as condições de Dirichlet podem ser usadas para especificar o potencial elétrico em cada extremidade do fio condutor, enquanto as condições de Neumann podem ser usadas para especificar a densidade de corrente elétrica nas extremidades.

A escolha adequada das condições de contorno é crucial para garantir que a solução da equação de Poisson unidimensional seja física e útil para o problema em questão.

Em resumo, a equação de Poisson unidimensional é uma ferramenta poderosa para descrever o comportamento de sistemas físicos em uma dimensão, e as condições de contorno desempenham um papel crucial na determinação da solução desta equação.

## 2.3 Equação de Poisson 2D

A equação de Poisson é uma equação diferencial parcial (EDP) elíptica não homogênea que desempenha um papel fundamental em várias áreas da física. Essas áreas incluem, mas não se limitam à teoria do potencial, a teoria da elasticidade e a teoria do eletromagnetismo. A equação de Poisson é uma extensão direta da equação de Laplace, que é uma EDP homogênea.

Na forma bidimensional, a equação de Poisson pode ser escrita como

$$\nabla^2 u = f(x, y), \quad (x, y) \in U \subseteq \mathbb{R}^2 \quad (2.10)$$

na qual  $U$  é um aberto conexo. Nesta equação,  $\nabla^2$  é o operador Laplaciano, o qual em coordenadas cartesianas bidimensionais é dado por  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ . Este operador é uma medida da curvatura de uma função em um espaço multidimensional.

A função  $u$  é a função desconhecida que estamos tentando resolver. Esta função pode representar uma variedade de quantidades físicas, dependendo do contexto. Por exemplo, na teoria do potencial,  $u$  pode representar o potencial gravitacional ou elétrico.

Finalmente,  $f$  é a função de fonte conhecida. Esta função representa a distribuição de matéria (na teoria do potencial gravitacional) ou carga (na teoria do potencial elétrico) no espaço. A solução da equação de Poisson nos dá a distribuição do potencial em resposta a esta distribuição de fonte.

Resolver a equação de Poisson é um problema comum em física e engenharia. Existem várias técnicas para resolver esta equação, incluindo métodos analíticos e numéricos. A escolha do método depende das condições de contorno do problema e da natureza da função de fonte  $f$ .

### 2.3.1 Condições de Contorno para a equação de Poisson 2D

A equação de Poisson é uma equação diferencial parcial elíptica que desempenha um papel fundamental em várias áreas da física. Em particular, a versão bidimensional dessa equação é frequentemente encontrada ao resolver problemas de potencial eletrostático, fluxo de calor e outros fenômenos físicos que podem ser descritos por um campo escalar.

As condições de contorno são essenciais para a solução da equação de Poisson. Elas especificam o comportamento da solução na fronteira do domínio do problema. Existem vários tipos de condições de contorno que podem ser aplicadas, incluindo condições de Dirichlet, onde o valor da função é especificado na fronteira, e condições de Neumann, onde a derivada normal da função é especificada na fronteira.

Ao resolver a equação de Poisson 2D, é importante escolher as condições de contorno apropriadas para o problema físico em questão. Por exemplo, em um problema de potencial eletrostático, as condições de contorno de Dirichlet podem ser usadas para especificar o potencial na fronteira, enquanto as condições de Neumann podem ser usadas para especificar o campo elétrico na fronteira.

Além disso, em alguns casos, pode ser necessário usar uma combinação de condições de contorno de Dirichlet e Neumann, conhecida como condição de contorno mista. A escolha das condições de contorno adequadas é crucial para garantir que a solução da equação de Poisson seja fisicamente significativa e útil para o problema em questão.

Em resumo, as condições de contorno desempenham um papel crucial na solução da equação de Poisson 2D. Elas determinam o comportamento da solução na fronteira do domínio e devem ser escolhidas com cuidado para refletir o problema físico que está sendo resolvido. A escolha correta das condições de contorno garante que a solução da equação de Poisson seja fisicamente relevante e forneça *insights* úteis sobre o fenômeno físico em estudo.

## 3 Redes Neurais

Neste capítulo, faremos uma introdução às redes neurais, um componente essencial da inteligência artificial. As redes neurais são inspiradas no funcionamento do cérebro humano e são usadas para resolver uma ampla gama de tarefas, desde reconhecimento de padrões até processamento de linguagem natural. Exploraremos seus elementos fundamentais, como camadas, neurônios e conexões. Ao longo deste capítulo, explica-se como as redes neurais são treinadas, otimizadas e aplicadas em problemas do mundo real, preparando o terreno para uma compreensão mais profunda deste campo empolgante.

### 3.1 Introdução às Redes Neurais

Redes neurais artificiais, às vezes abreviadas como RNAs ou simplesmente redes neurais, são um elemento essencial da aprendizagem de máquina e inteligência artificial. Elas são modeladas a partir do funcionamento do cérebro humano e são utilizadas para resolver uma variedade ampla de problemas complexos. Em sua forma mais básica, uma rede neural consiste em unidades de processamento interconectadas, conhecidas como neurônios artificiais ou unidades, que trabalham juntas para realizar tarefas específicas.

A ideia das redes neurais foi inicialmente inspirada pelo funcionamento do cérebro biológico, composto por bilhões de neurônios interconectados. Da mesma maneira, nas redes neurais artificiais, os neurônios artificiais são organizados em camadas, incluindo a camada de entrada, uma ou mais camadas ocultas e a camada de saída. Cada neurônio está conectado a outros nas camadas adjacentes através de conexões ponderadas.

As redes neurais são especialmente adequadas para tarefas de aprendizado supervisionado, onde algoritmos são treinados com dados conhecidos tanto na entrada quanto na saída para aprender a mapear os dados inseridos nos resultados desejados. Durante o treinamento, a rede ajusta automaticamente os pesos das conexões entre os neurônios para otimizar o desempenho na tarefa.

Em tarefas de classificação, por exemplo, uma rede neural pode aprender a distinguir diferentes categorias com base em características extraídas dos dados inseridos. No contexto de regressão, a rede tem a capacidade de fazer previsões de valores numéricos com base nos padrões identificados nos dados utilizados para treinamento.

## 3.2 Tipos de Rede neurais

Nesta seção, vamos explorar diferentes tipos de redes neurais, que desempenham papéis cruciais em diversas aplicações de aprendizado de máquina. As redes neurais são modelagens matemáticas inspiradas na estrutura e função do cérebro humano, capazes de aprender padrões complexos em dados. Vamos focar em três arquiteturas distintas que desempenham um papel fundamental em várias áreas do aprendizado de máquina e processamento de dados. São elas

### I. Redes Neurais Feedforward (FNN):

#### 1. Redes Feedforward de Camada Única:

Em (HAYKIN, 2009) define-se uma rede neural de feedforward de camada única como uma rede neural em que as neurônios são organizados em duas camadas: uma camada de entrada e uma camada de saída. Na camada de entrada, os neurônios não executam nenhuma computação, apenas passam os dados de entrada para a camada de saída. Na camada de saída, os neurônios executam uma operação de soma ponderada dos dados de entrada e aplicam uma função de ativação para produzir a saída da rede.

#### 2. Redes Feedforward Multicamadas:

Conforme (HAYKIN, 2009), as Redes Feedforward Multicamadas distinguem-se pela presença de uma ou mais camadas ocultas, cujos nós de computação são denominados neurônios ocultos. Esses neurônios ocultos têm a função de intervir entre a entrada externa e a saída da rede de alguma forma útil. Ao adicionar uma ou mais camadas ocultas, a rede é capacitada a extrair estatísticas de ordem superior de sua entrada. Em um sentido bastante amplo, a rede adquire uma perspectiva global apesar de sua conectividade local, devido ao conjunto extra de conexões sinápticas e à dimensão extra de interações neurais.

### II. Redes Neurais Recorrentes (RNN):

(HAYKIN, 2009) afirma que uma rede neural recorrente distingue-se de uma rede neural feedforward por ter pelo menos um laço de retroalimentação. Isso significa que a saída de um neurônio é alimentada de volta para suas próprias entradas, ou para as entradas de outros neurônios na rede. Essa estrutura permite que as redes recorrentes aprendam padrões sequenciais de dados, o que as torna particularmente úteis para tarefas como reconhecimento de fala e tradução de idiomas.

### III. Redes Neurais Convolucionais (CNN):

(GOODFELLOW; BENGIO; COURVILLE, 2016) afirmam que as redes convolucionais (apud LeCun, 1989) são um tipo especializado de rede neural para processar dados que possuem uma topologia de grade conhecida. Exemplos incluem dados de séries temporais, que podem ser pensados como uma grade 1-D com amostras tomadas em intervalos de tempo regulares, e dados de imagens, que podem ser pensados como uma grade 2-D de pixels. As redes convolucionais têm sido tremendamente bem-sucedidas em aplicações práticas. O nome “rede neural convolucional” indica que a rede emprega uma operação matemática chamada convolução. A convolução é um tipo especializado de operação linear. As redes convolucionais são simplesmente redes neurais que usam a convolução no lugar da multiplicação de matriz geral em pelo menos uma de suas camadas.

## 3.3 Arquitetura de Redes Neurais

Nesta seção, explora-se em detalhes a arquitetura de redes neurais utilizada para abordar a resolução da equação de Poisson. Como mencionado anteriormente, as redes neurais têm se mostrado uma ferramenta poderosa para resolver uma ampla variedade de problemas, incluindo equações diferenciais parciais. A escolha adequada da arquitetura de rede desempenha um papel fundamental no desempenho e na eficácia do modelo. Portanto, é crucial entender como a estrutura da rede neural é projetada para atender às necessidades específicas do problema em questão.

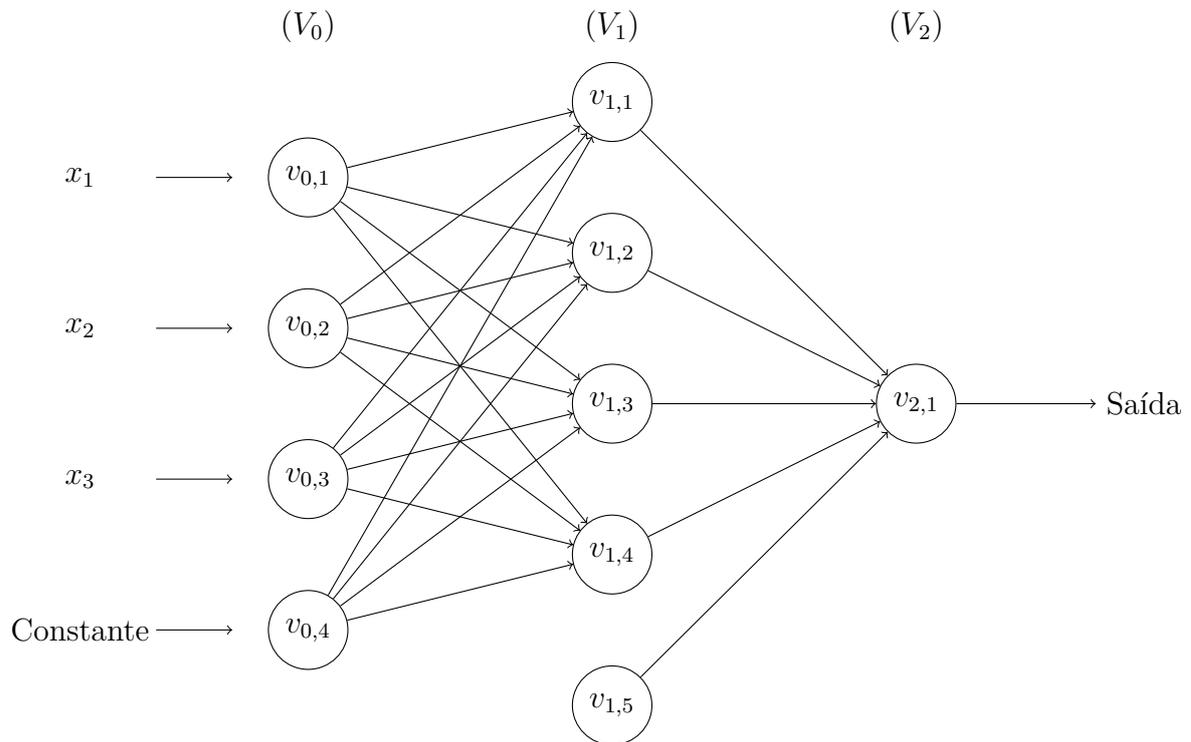
Começaremos discutindo os componentes essenciais de uma rede neural, como camadas, neurônios e funções de ativação. Em seguida, descreveremos a arquitetura geral da rede neural, incluindo o número de camadas, o dimensionamento e a organização dessas camadas. Além disso, abordaremos as escolhas de hiperparâmetros, como a taxa de aprendizado, a função de custo e o otimizador, que desempenham um papel crítico no treinamento da rede neural.

### 3.3.1 Camadas de uma Rede Neural

Uma rede neural é composta por várias camadas que desempenham funções específicas no processamento de informações. Compreender a organização e a função de cada camada é fundamental para projetar uma arquitetura eficaz na resolução da equação de Poisson. Nesta subseção, exploraremos as principais camadas de uma rede neural e seu papel no contexto do nosso problema. A ilustração visual das camadas de uma rede neural é apresentada abaixo.

A primeira camada, denominada de camada de entrada, é representada por  $V_0$  e é

Figura 3.1 – Arquitetura da Rede Neural



Fonte: Baseado em (SHALEV-SHWARTZ; BEN-DAVID, 2014)

composta por um total de  $n + 1$  neurônios, onde  $n$  corresponde à dimensionalidade do espaço de entrada. As camadas denotadas como  $V_1$  até  $V_{T-1}$  são comumente referidas como camadas ocultas em redes neurais. A última camada, designada como  $V_T$ , é especificamente denominada camada de saída. Em problemas de previsão de complexidade reduzida, a camada de saída consiste em um único neurônio, cuja saída representa o resultado da rede neural. Denominamos por  $T$  o número de camadas na rede, excluindo-se  $V_0$ , ou seja, referimo-nos à “profundidade” da rede.

### 3.3.2 Neurônios

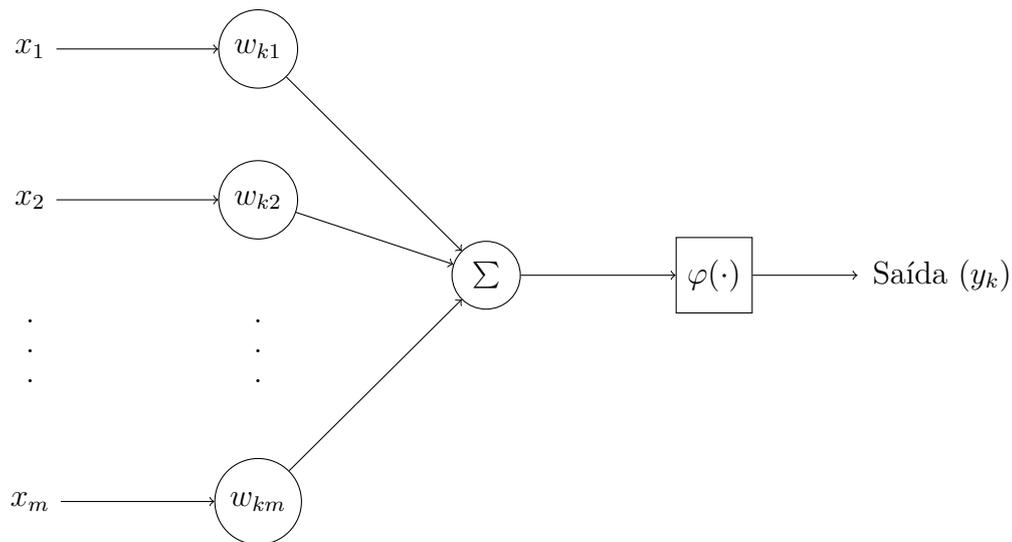
De acordo com (HAYKIN, 2009), um neurônio é uma unidade de processamento de informações que é fundamental para a operação de uma rede neural. Compreender o funcionamento dos neurônios é essencial para entender como a rede neural aprende e como isso se aplica à resolução da equação de Poisson.

#### 3.3.2.1 Estrutura de um Neurônio

Cada neurônio em uma rede neural é composto por esses componentes

- a) **Sinais de entrada:** São os valores  $x_1, x_2, \dots, x_n$  que constituem os dados que nutrem o modelo preditivo em questão.

Figura 3.2 – Estrutura de um neurônio



Fonte: (HAYKIN, 2009)

- b) **Pesos sinápticos:**  $(w_1, w_2, \dots, w_n)$  São os pesos atribuídos às entradas da rede neural, e esses valores são adquiridos por meio do processo de treinamento.
- c) **Combinador linear:**  $(\Sigma)$  Soma-se os sinais de entrada, ponderados pelos pesos sinápticos correspondentes, a fim de gerar um potencial de ativação.
- d) **Função de ativação:**  $(\varphi)$  O propósito consiste em restringir a amplitude de saída de um neurônio dentro de um intervalo predefinido de valores.
- e) **Saída:**  $(y_k)$  O valor final de saída, em um contexto neural, constitui-se como um ponto crucial, passível de utilização como entrada em neurônios subsequentes que se encontram interconectados de maneira sequencial.

### 3.3.3 Funções de Ativação

Conforme (HAYKIN, 2009),

uma função de ativação é uma função que limita a amplitude da saída de um neurônio. A função de ativação também é conhecida como função de compressão, pois comprime (limita) a faixa de amplitude permitida do sinal de saída para algum valor finito. (HAYKIN, 2009)

#### a) Função Sigmoide:

Em (HAYKIN, 2009) encontra-se que a função sigmoide é amplamente utilizada em redes neurais devido às suas propriedades de suavização e não-linearidade. Esta é definida como uma função estritamente crescente que equilibra de maneira elegante o comportamento linear e não linear. Sua fórmula é dada por

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3.1)$$

ou

$$\sigma'(z) = f(z) (1 - f(z)) \quad (3.2)$$

A função sigmoide, definida na equação (3.1), abrange uma faixa contínua de valores no intervalo de 0 a 1. Além disso, é importante notar que a função sigmoide é diferenciável, o que a torna uma escolha relevante em diversas aplicações em que a suavidade e a diferenciação são características desejáveis.

**b) Função ReLU (Rectified Linear Unit):**

A função ReLU é uma função de ativação linear para entradas positivas e zero para entradas negativas. Sua fórmula é dada por

$$f(z) = \max(0, z) \quad (3.3)$$

ou

$$f'(z) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases} \quad (3.4)$$

De acordo com (GÉRON, 2019), a função de ativação ReLU é contínua, mas não é diferenciável em  $z = 0$ . No entanto, na prática ela funciona muito bem e tem a vantagem de o cálculo ser rápido. Mais importante ainda, o fato dela não possuir um valor máximo de saída também ajuda na redução de alguns problemas durante o Gradiente Descendente.

**c) Função Tangente Hiperbólica (tgh):**

A função tangente hiperbólica mapeia a entrada para o intervalo  $(-1, 1)$ . Sua fórmula é dada por

$$\text{tgh}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (3.5)$$

ou

$$(\text{tgh})'(z) = 2\sigma(2z) - 1 \quad (3.6)$$

De acordo com (GÉRON, 2019), a função de ativação tangente hiperbólica

tem o formato de  $S$ , é contínua e diferenciável, mas seu valor de saída varia de  $-1$  a  $1$  no início do treinamento, o que tende a tornar cada saída da camada mais ou menos normalizada, ajudando frequentemente a acelerar a convergência.

## 4 Materiais e métodos

Este capítulo destina-se a apresentar e discutir as principais ferramentas e métodos empregados na construção de redes neurais para a resolução da Equação de Poisson em duas dimensões e da Equação de Poisson em uma dimensão. O objetivo é fornecer uma compreensão detalhada das técnicas utilizadas nesse contexto.

### 4.1 Rede neurais Informadas Pela Física

Como Segundo (ALKHADHR; ALMEKKAWY, 2023), as Redes Neurais de Aproximação de Equações Diferenciais Parciais (PINNs, do inglês “*Physics-Informed Neural Networks*”) são uma classe de redes neurais profundas que funcionam como ferramentas de aproximação de equações diferenciais parciais (EDP).

As redes neurais artificiais (RNA) representam modelos matemáticos capazes de serem treinados para executar tarefas complexas, como a classificação de imagens e a previsão de séries temporais. Essas redes têm demonstrado sucesso em diversas aplicações em engenharia e ciência, inclusive na resolução de problemas envolvendo Equações Diferenciais Parciais (EDPs).

As redes neurais informadas pela física (PINNs) são uma técnica que integra de forma contínua as informações tanto das medições quanto das equações diferenciais parciais (EDPs). As EDPs podem ser de ordem inteira, equações integro-diferenciais, EDPs fracionárias ou EDPs estocásticas (KARNIADAKIS *et al.*, 2021).

#### 4.1.1 A arquitetura da rede neural informada pela física

A metodologia empregada nessa monografia se baseia nas contribuições de ALKHADHR; ALMEKKAWY (2023) e RAISSI; PERDIKARIS; KARNIADAKIS (2019), cujos trabalhos serviram como fonte de inspiração para a estrutura arquitetônica adotada. Essa estrutura é delineada essencialmente por dois elementos-chave: a rede aproximadora e a rede residual.

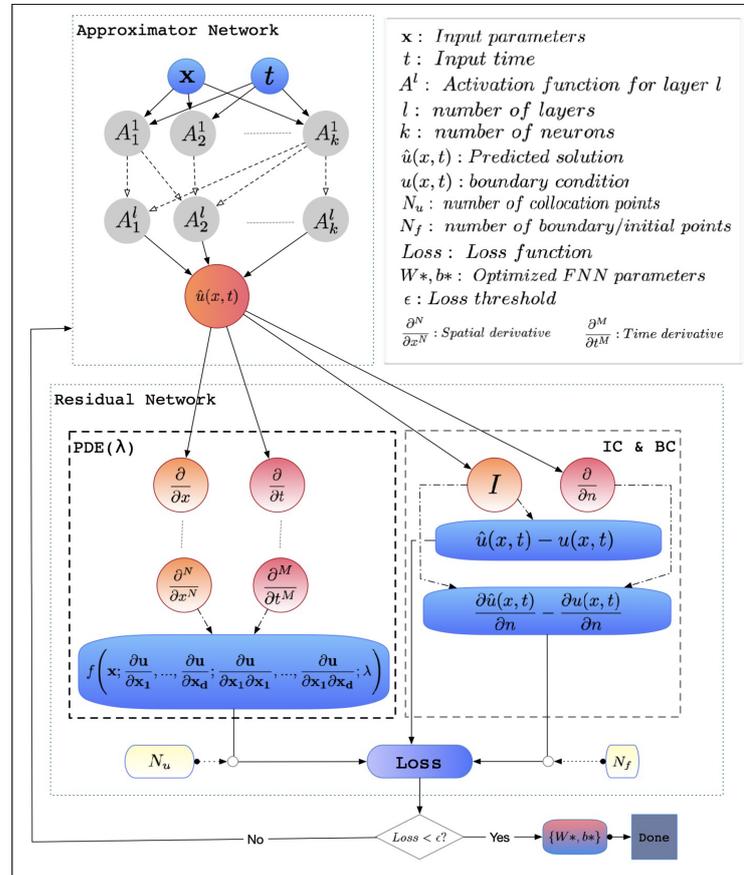
A rede aproximadora visa a previsão do resultado do modelo  $u$  no ponto de colisão  $x$  após ser treinada. Já a rede residual aplica a fórmula da equação física governante ao resultado da rede neural aproximadora como parte da função de perda. Esse processo incorpora as principais características das *Physical-Informed Neural Networks* (PINNs).

As PINNs combinam a capacidade de aproximação de uma rede neural com a incorporação direta das leis físicas subjacentes, representadas pela equação governante do problema. Isso possibilita não apenas a predição dos resultados, mas também a garantia

de que as soluções obtidas respeitem os princípios físicos subjacentes.

Na figura abaixo tem-se a ilustração para a arquitetura de uma PINN

Figura 4.1 – Uma arquitetura PINN



Fonte: (ALKHADHR; ALMEKKAWY, 2023)

A integração das redes aproximadoras com as redes residuais dentro das PINNs oferece uma abordagem poderosa para resolver problemas de física, possibilitando a modelagem e a predição de fenômenos complexos. Essa técnica torna-se essencial em diversas áreas, incluindo ciências naturais, engenharias e até mesmo em ciência de dados, onde a combinação de modelos baseados em física com métodos de aprendizado de máquina é de grande relevância para a obtenção de resultados precisos e confiáveis.

A topologia da rede é tal que a entrada é composta por duas variáveis, correspondentes às coordenadas espaciais necessárias para a solução da equação diferencial parcial. Uma característica importante é a normalização das entradas para o intervalo  $[-1, 1]$ , realizada através da aplicação de uma escala adequada baseada nos limites espaciais da região de interesse, que são derivados dos dados físicos do problema.

Ademais, a arquitetura da rede é complementada pelo uso de otimizadores adequados, neste caso, o algoritmo Adam com uma taxa de aprendizado de  $1 \cdot 10^{-3}$ , visando

minimizar a função de perda definida para capturar tanto a consistência com as condições de contorno quanto a solução precisa da equação diferencial parcial.

## 4.2 Função de Perda

Na prática do treinamento de redes neurais, a função de perda desempenha um papel crucial. Essa função é definida com base em dois componentes primários: a perda associada à Equação Diferencial Parcial (PDE do inglês *Partial Differential Equations*) e a perda nas Condições de Contorno (BC, do inglês *Boundary Conditions*).

A expressão que define a função de perda, comumente conhecida como Erro Quadrático Médio (MSE, do inglês Mean Squared Error), pode ser representada pela seguinte fórmula como mostrado em (ALKHADHR; ALMEKKAWY, 2023), (KARNIADAKIS *et al.*, 2021) e (RAISSI; PERDIKARIS; KARNIADAKIS, 2019):

$$\text{MSE} = w_f \cdot \text{MSE}_f + w_b \cdot \text{MSE}_b \quad (4.1)$$

na qual

- MSE é o Erro Quadrático Médio total;
- $w_f$  e  $w_b$  são pesos atribuídos às perdas das equações diferenciais parciais e das condições de contorno, respectivamente;
- $\text{MSE}_f$  é o Erro Quadrático Médio associado à equação diferencial parcial;
- $\text{MSE}_b$  é o Erro Quadrático Médio referente às condições de contorno.

Para calcular  $\text{MSE}_f$ , utiliza-se a seguinte fórmula

$$\text{MSE}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \left\| f(\hat{u}(x_u^i, t_u^i)) - f(x_u^i, t_u^i) \right\|^2 \quad (4.2)$$

na qual

- $N_f$  é o número de pontos relacionados à equação diferencial parcial;
- $\hat{u}(x_u^i, t_u^i)$  representa a solução aproximada; diferenciais parciais e das condições de contorno, respectivamente;
- $f(\hat{u}(x_u^i, t_u^i))$  e  $f(x_u^i, t_u^i)$  são respectivamente as funções aproximadas e reais associadas à equação diferencial parcial.

Já para calcular  $\text{MSE}_b$ , utiliza-se a seguinte expressão

$$\text{MSE}_b = \frac{1}{N_b} \sum_{j=1}^{N_b} \left\| f_b(\hat{u}(x_u^j, t_u^j)) - f_b(x_u^j, t_u^j) \right\|^2 \quad (4.3)$$

na qual

- $N_b$  é o número de pontos associados às condições de contorno;
- $\hat{u}(x_u^j, t_u^j)$  é a solução aproximada para os pontos das condições de contorno;
- $f_b(\hat{u}(x_u^j, t_u^j))$  e  $f_b(x_u^j, t_u^j)$  representam as funções aproximadas e reais para as condições de contorno.

Essas formulações do Erro Quadrático Médio ponderam as discrepâncias entre as soluções aproximadas e reais tanto das equações diferenciais parciais quanto das condições de contorno. O uso de pesos  $w_f$  e  $w_b$  permite atribuir importâncias relativas a essas duas parcelas da função de perda, de acordo com a relevância de cada aspecto no treinamento da rede neural.

### 4.2.1 Função de Ativação

Nesta seção, descrevemos a implementação das funções de ativação utilizadas na arquitetura da Rede Neural Artificial (RNA) empregada para a solução de problemas de física via Redes Neurais de Diferenciação (PINNs).

A RNA foi projetada com duas camadas ocultas, cada uma contendo 1500 neurônios. Esses neurônios são ativados pela função tangente hiperbólica (tgh), conforme mostrado em (3.5).

A escolha da função de ativação tgh foi motivada pela sua capacidade de lidar com a não-linearidade dos problemas físicos e sua habilidade de mapear os valores de entrada para um intervalo entre  $-1$  e  $1$ , promovendo estabilidade numérica durante o treinamento. (FIORIN *et al.*, 2011)

A função de ativação tgh é aplicada sequencialmente às camadas da rede, permitindo a aprendizagem de representações complexas dos dados de entrada. Essa função apresenta um comportamento suave e contínuo, o que facilita a propagação do gradiente durante o processo de otimização.

A utilização da função de ativação tgh na arquitetura da RNA demonstrou ser adequada para a resolução eficiente de equações diferenciais parciais (EDPs) por meio dos PINNs, fornecendo resultados satisfatórios e convergência durante o treinamento. (FIORIN *et al.*, 2011)

Esta escolha da função de ativação foi baseada em considerações teóricas e experimentais, visando a obtenção de resultados precisos e a eficiência na resolução de problemas físicos modelados por EDPs.

### 4.2.2 A dinâmica dos PINNs e o processo de otimização

A dinâmica dos PINNs é fundamentada na minimização de uma função de perda composta por duas componentes principais: a perda associada à satisfação da equação diferencial parcial em pontos internos do domínio e a perda relacionada ao atendimento das condições de contorno impostas.

Segundo (RAISSI; PERDIKARIS; KARNIADAKIS, 2019), o processo de otimização busca ajustar os parâmetros da rede neural de forma a minimizar a discrepância entre a solução aproximada pela rede e as condições de contorno impostas, ao mesmo tempo em que satisfaz a equação diferencial nos pontos internos do domínio.

Para tanto, a rede neural é treinada iterativamente utilizando métodos de otimização, neste caso, empregando o algoritmo do Método do Gradiente (Método do Máximo Declive). Segundo (BOTTOU, 2010), durante o treinamento, os pesos da rede são atualizados em direção ao gradiente da função de perda em relação aos parâmetros treináveis da rede. Essa otimização é realizada com o intuito de encontrar valores para os parâmetros da rede neural que minimizem simultaneamente a discrepância entre a solução aproximada e as condições impostas, garantindo a precisão da solução dentro do domínio.

A classe PINN encapsula essa dinâmica, definindo a arquitetura da rede neural, as funções de perda associadas à equação diferencial e às condições de contorno, bem como os métodos para cálculo dos gradientes e atualização dos pesos da rede. Assim, essa implementação ilustra a interação entre a arquitetura da rede neural, a formulação matemática das equações diferenciais e o processo iterativo de otimização para resolver tais problemas físicos.

(RAISSI; PERDIKARIS; KARNIADAKIS, 2019) afirmam que essa abordagem oferece uma forma flexível e eficaz de resolver EDPs, representando um avanço significativo no campo da modelagem matemática e computacional de problemas físicos complexos.

# 5 Comparação entre o Método de Diferenças Finitas e as Redes Neurais Informadas pela Física

Este capítulo é dedicado à exploração e análise comparativa entre duas abordagens distintas para a resolução de equações diferenciais parciais (EDPs): o método de diferenças finitas e as redes neurais informadas pela física (PINNs). Em particular, focaremos na aplicação dessas técnicas às equações de Poisson 1D e Poisson 2D. Ambas as abordagens têm suas vantagens e desvantagens, e a escolha entre elas depende em grande parte do problema específico que estamos tentando resolver.

## 5.1 O Método de Diferenças Finitas

O método de diferenças finitas é uma técnica numérica amplamente utilizada para a resolução de equações diferenciais. Este método baseia-se na ideia de substituir as derivadas que aparecem em uma equação diferencial por diferenças finitas, que são formas discretas das derivadas. A principal vantagem deste método é sua simplicidade e flexibilidade. Pode ser aplicado a uma ampla gama de problemas, e é particularmente útil quando a solução analítica para a equação diferencial é desconhecida ou difícil de obter.

### 5.1.1 Diferenças finitas na equação de Poisson 1D

Existem vários esquemas de diferenças finitas que podem ser usados para resolver a equação de Poisson em uma dimensão. Vamos explorar esses esquemas de diferenças finitas em detalhes.

#### 1. Diferenças Centrais de Segunda Ordem

Esta abordagem usa uma fórmula de diferença central de quarta ordem para a segunda derivada. A equação resultante para um ponto  $x_i$  é mais complexa, mas oferece maior precisão.

$$u''_{x_i} = \frac{1}{h^2} (u_{x_{i-1}} - 2u_{x_i} + u_{x_{i+1}}) - \frac{1}{12} h^2 u^{(iv)} - \frac{1}{360} h^4 u^{(vi)} + O(h^6) \quad (5.1)$$

#### 2. Diferenças Centrais de Quarta Ordem

Esta abordagem usa uma fórmula de diferença central de quarta ordem para a segunda derivada. A equação resultante para um ponto  $x_i$  é mais complexa, mas oferece maior precisão.

$$u''_{x_i} = \frac{1}{h^2} (u_{x_{i-1}} - 2u_{x_i} + u_{x_{i+1}}) + \frac{1}{12} h^2 f''_{x_i} - \frac{1}{360} h^4 u^{(vi)}_{x_i} + O(h^6) \quad (5.2)$$

### 3. Método de Relaxação

O método de relaxação é um método iterativo usado para resolver equações diferenciais, como a equação de Poisson, discretizadas por diferenças finitas. A ideia básica é começar com uma estimativa inicial para a solução e, em seguida, atualizar iterativamente essa estimativa até que ela convirja para a solução precisa.

$$u^{(k+1)}(x_i) = \frac{1}{2} \left( u^{(k)}(x_{i-1}) + u^{(k)}(x_{i+1}) - h^2 f(x_i) \right) \quad (5.3)$$

na qual  $f$  é a função conhecida na equação de Poisson.

Este é um método simples e amplamente utilizado para resolver equações diferenciais parciais, mas a convergência pode ser lenta em alguns casos. Existem variantes mais avançadas do método de relaxação, como o método de sobre-relaxação sucessiva (SOR), que pode acelerar a convergência.

### 4. Método de Diferenças Finitas Não-Estruturadas

O método de diferenças finitas não estruturadas é uma técnica para resolver equações diferenciais em geometrias complexas, onde a grade não segue uma estrutura regular, como uma grade retangular ou triangular típica. Em vez disso, ele usa uma malha arbitrária que se ajusta à geometria do problema. A formulação geral do método de diferenças finitas não estruturadas é semelhante à dos métodos estruturados, mas a discretização da equação diferencial é realizada em uma malha irregular.

Uma formulação geral para um ponto  $P$  em uma malha não estruturada poderia ser

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{A_P} \sum_{N_P} \frac{(u_P - u_N)}{(x_P - x_N)} \cdot \frac{(x_P - x_N)}{(x_P - x_N)^2} \quad (5.4)$$

Essa expressão é uma forma de discretizar a segunda derivada parcial em relação a  $x$  para torná-la solucionável numericamente em métodos de elementos finitos e outros métodos numéricos. onde  $A_P$  é a área do polígono em torno do ponto  $P$ , a soma é feita sobre todos os pontos vizinhos  $N_P$  de  $P$ , e  $u_P$  e  $u_N$  são os valores da função  $u$  nos pontos  $P$  e  $N_P$ , respectivamente.

Essa é apenas uma aproximação geral e a formulação exata dependerá da configuração específica da malha não estruturada e da equação diferencial sendo resolvida. O método de diferenças finitas não estruturadas é mais complexo de implementar do que os métodos estruturados, mas pode ser necessário para resolver problemas em geometrias complexas.

Estes são apenas alguns dos esquemas comuns utilizados para resolver a equação de Poisson com diferenças finitas. A escolha do esquema depende da precisão desejada, da complexidade da geometria e de outras considerações específicas do problema.

### 5.1.2 Aplicação das Diferenças Finitas na Equação de Poisson 2D

Nesta seção, vamos considerar a aplicação do método de diferenças finitas à equação de Poisson 2D. Esta equação é dada por  $u_{xx} + u_{yy} = f(x, y)$ , onde  $u_{xx}$  e  $u_{yy}$  representam as segundas derivadas parciais de  $u$  em relação a  $x$  e  $y$ , respectivamente, e  $f$  é uma função dada.

Seguindo a abordagem em (STRIKWERDA, 2004), considera-se a equação de Poisson no quadrado unitário. Em (STRIKWERDA, 2004) apresenta-se uma aproximação de diferenças centrais para as segundas derivadas em duas direções. Esta aproximação leva à seguinte fórmula de diferença

$$\delta_x^2 v_{\ell,m} + \delta_y^2 v_{\ell,m} = f \quad (5.5)$$

ou, de forma equivalente,

$$\frac{1}{h^2} (v_{\ell+1,m} + v_{\ell-1,m} + v_{\ell,m+1} + v_{\ell,m-1} - 4v_{\ell,m}) = f_{\ell,m} \quad (5.6)$$

O operador de diferença no membro esquerdo de (5.6) é chamado de laplaciano discreto de cinco pontos. Este operador desempenha um papel fundamental na formulação e resolução de problemas de diferenças finitas.

## 5.2 Comparação Quantitativa

Para realizar uma comparação quantitativa entre o método de diferenças finitas (MDF) e as Redes Neurais Informadas pela Física (PINNs), vamos considerar diversos critérios relevantes para a resolução de equações diferenciais parciais (EDPs). Estes critérios incluem precisão, eficiência computacional, robustez numérica e facilidade de implementação.

### 5.2.1 Precisão

A precisão é um aspecto crucial na resolução de EDPs, pois determina quão próximas as soluções numéricas estão da solução analítica (quando disponível) ou de soluções de referência. O método de diferenças finitas oferece precisão de ordem superior, especialmente quando esquemas de alta ordem são utilizados. No entanto, a precisão das PINNs também pode ser alta, especialmente quando treinadas com dados suficientes e arquiteturas adequadas.

### 5.2.2 Eficiência Computacional

Em termos de eficiência computacional, as PINNs podem oferecer vantagens significativas. Uma vez treinadas, as PINNs podem fornecer soluções rápidas para uma ampla gama de problemas, sem a necessidade de discretização do domínio espacial ou temporal. Por outro lado, o método de diferenças finitas requer a discretização do domínio, o que pode resultar em malhas computacionais volumosas e tempos de computação mais longos, especialmente em problemas tridimensionais ou com geometrias complexas.

### 5.2.3 Robustez Numérica

A robustez numérica refere-se à capacidade de um método numérico lidar com uma variedade de condições e características do problema, como singularidades, discontinuidades ou não linearidades. Embora o método de diferenças finitas seja conhecido por sua robustez em muitos casos, ele pode enfrentar dificuldades em problemas com geometrias irregulares ou singularidades fortes. As PINNs, por outro lado, têm a capacidade de aprender automaticamente características complexas do problema durante o treinamento, o que as torna mais robustas em certos cenários.

### 5.2.4 Facilidade de Implementação

Em relação à facilidade de implementação, as diferenças finitas tendem a ser mais diretas e simples de implementar, especialmente para aqueles familiarizados com métodos numéricos clássicos. Por outro lado, a implementação de PINNs pode exigir um conhecimento mais profundo de aprendizado de máquina e frameworks específicos de *deep learning*. No entanto, com o crescente desenvolvimento de bibliotecas de código aberto para *deep learning*, como TensorFlow e PyTorch, a implementação de PINNs está se tornando mais acessível.

Em resumo, a escolha entre o método de diferenças finitas e as PINNs depende das características específicas do problema em questão, bem como das considerações relacionadas à precisão, eficiência computacional, robustez numérica e facilidade de implementação. Em muitos casos, uma abordagem híbrida que combina o melhor dos dois métodos pode ser a mais adequada.

## 6 Experimentos e Resultados

Este capítulo descreve os experimentos e resultados alcançados ao resolver a equação de Poisson 1D e a equação de Poisson 2D usando redes neurais do tipo Physics-Informed Neural Networks (PINNs), utilizando Python e suas bibliotecas TensorFlow, NumPy e Matplotlib. Os resultados são comparados com os obtidos pelo método das diferenças finitas, com o objetivo de analisar a eficácia e precisão das redes neurais na resolução de problemas de física computacional em comparação com métodos tradicionais. Os experimentos são realizados em diversas configurações para avaliar o desempenho e a generalização das redes neurais em diferentes cenários de problemas. As seções seguintes detalham os experimentos realizados e os resultados obtidos, seguidos de uma discussão sobre as conclusões derivadas desses resultados.

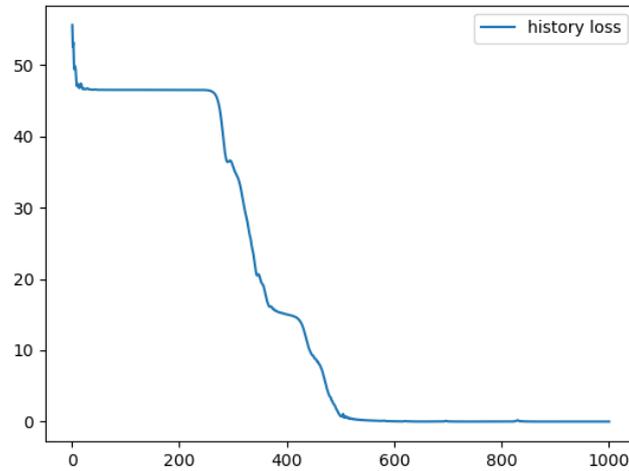
### 6.1 Solução da Equação de Poisson 1D

Nesta seção, abordamos a resolução da equação de Poisson unidimensional, um problema fundamental na física que descreve a distribuição de um campo escalar em um meio unidimensional. Utilizamos exemplos relevantes como referência para ilustrar a aplicação de técnicas de solução. Os resultados dos experimentos realizados e uma discussão sobre as conclusões alcançadas são apresentados detalhadamente.

- Exemplo 1

$$\left\{ \begin{array}{ll} \nabla^2 u(x) = \sum_{k=1}^4 k \operatorname{sen}(kx) + 8 \operatorname{sen}(8x), & x \in (0, \pi) \\ u(x, 0) = 0, & x \in (0, \pi) \\ u(x, \pi) = \pi, & x \in (0, \pi) \end{array} \right. \quad (6.1)$$

Figura 6.1 – Treinamento do exemplo 1 com 1000 Épocas



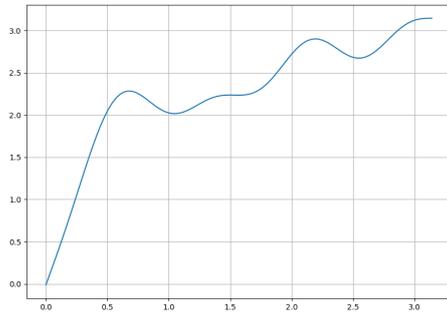
Fonte: Próprio Autor

Para o exemplo 1, o treinamento da Physics-Informed Neural Network (PINN) foi realizado ao longo de 1000 épocas, conforme representado no gráfico anteriormente apresentado. A seguir, são apresentados os resultados obtidos com o treinamento da PINN para o exemplo 1, juntamente com uma comparação com a solução obtida utilizando o método das diferenças finitas.

O treinamento da PINN resultou em um conjunto de parâmetros otimizados, os quais foram empregados para calcular a solução aproximada da equação em questão. A precisão e a eficácia da solução obtida foram avaliadas mediante a diferença entre a solução utilizando diferenças finitas e a solução da rede neural.

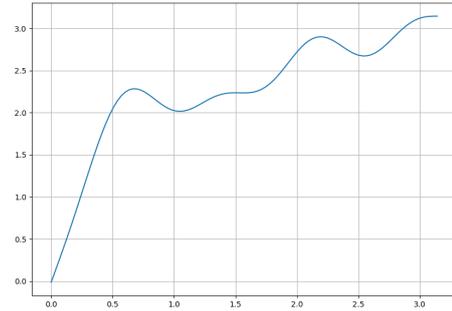
Para o treinamento, foi utilizado uma taxa de aprendizado de 0.001. A rede neural implementada foi uma rede densa com três camadas: duas camadas ocultas, cada uma com 1500 neurônios e ativação tangente hiperbólica (tanh), e uma camada de saída com um neurônio, que não possui função de ativação especificada, o que implica em uma saída linear.

Figura 6.2 – Solução do exemplo 1 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.3 – Solução do exemplo 1 com a PINN

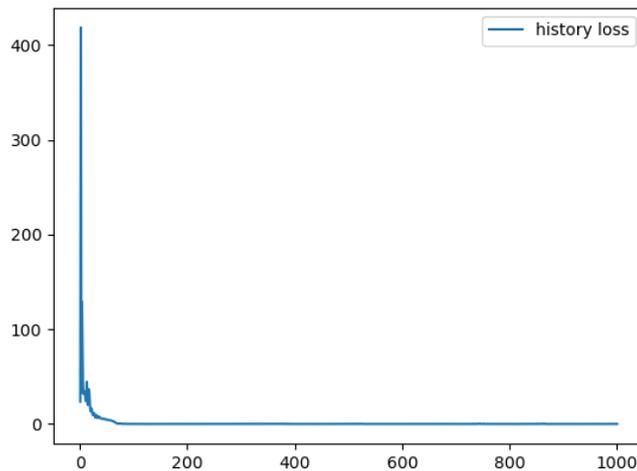


Fonte: Próprio Autor

- Exemplo 2:

$$\begin{cases} \nabla^2 u(x) = (3x + x^2)e^x, & x \in (0, 1) \\ u(x, 0) = 0, & x \in (0, 1) \\ u(x, 1) = 0, & x \in (0, 1) \end{cases} \quad (6.2)$$

Figura 6.4 – Treinamento do exemplo 2 com 1000 Épocas

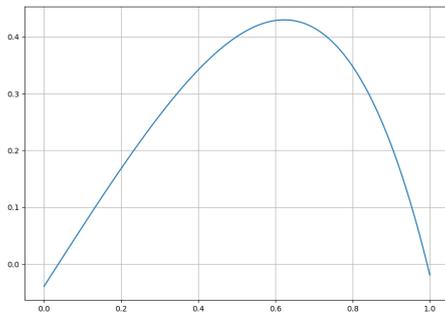


Fonte: Próprio Autor

No exemplo 2, foi empregado o mesmo método utilizado no exemplo 1, que consiste na aplicação da Physics-Informed Neural Network (PINN). Assim como no exemplo anterior, o treinamento da PINN foi conduzido ao longo de 1000

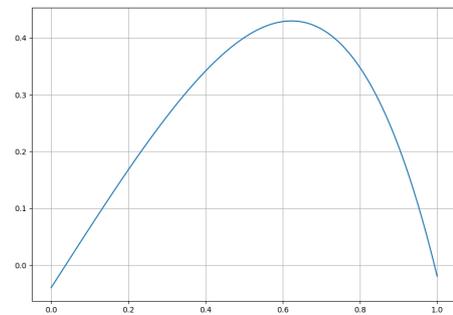
épocas, e os resultados foram comparados com a solução obtida pelo método das diferenças finitas. O objetivo foi avaliar a precisão e a eficácia da solução obtida pela rede neural em relação à solução numérica tradicional. Além disso, no exemplo 2 também foi utilizada uma taxa de aprendizado de 0.001 e a mesma arquitetura de rede neural densa, com três camadas: duas camadas ocultas, cada uma com 1500 neurônios e ativação tangente hiperbólica (tanh), e uma camada de saída com um neurônio, sem função de ativação especificada, resultando em uma saída linear. Essa abordagem foi adotada com o intuito de comparar o desempenho da PINN em diferentes exemplos e validar sua aplicabilidade em problemas diversos.

Figura 6.5 – Solução do exemplo 2 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.6 – Solução do exemplo 2 com a PINN

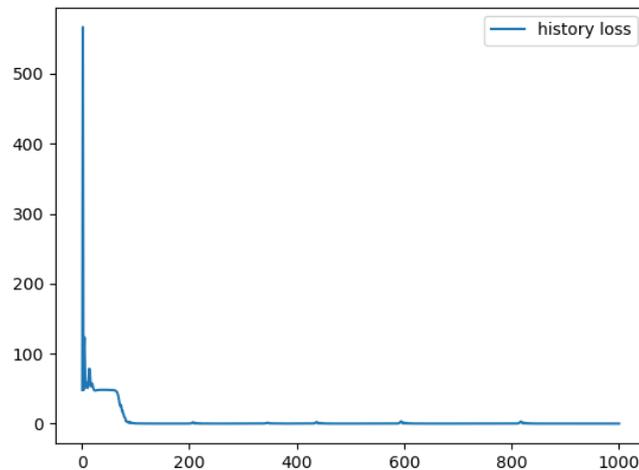


Fonte: Próprio Autor

- Exemplo 3:

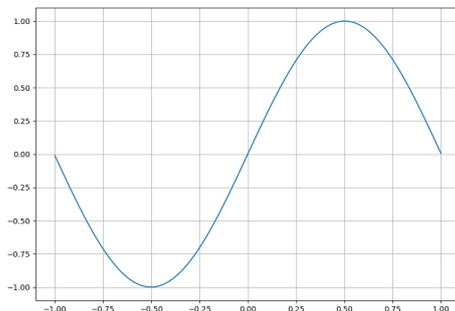
$$\left\{ \begin{array}{ll} \nabla^2 u(x) = \pi^2 \text{sen}(\pi x), & x \in (-1, 1) \\ u(x, -1) = 0, & x \in (-1, 1) \\ u(x, 1) = 0, & x \in (-1, 1) \end{array} \right. \quad (6.3)$$

Figura 6.7 – Treinamento do exemplo 3 com 1000 Épocas



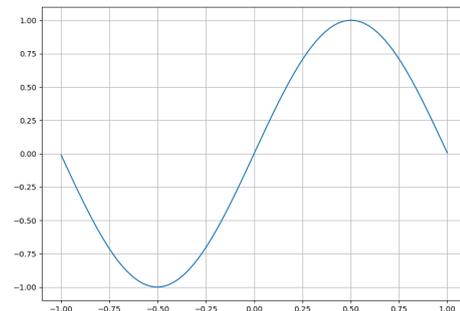
Fonte: Próprio Autor

Figura 6.8 – Solução do exemplo 3 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.9 – Solução do exemplo 3 com a PINN



Fonte: Próprio Autor

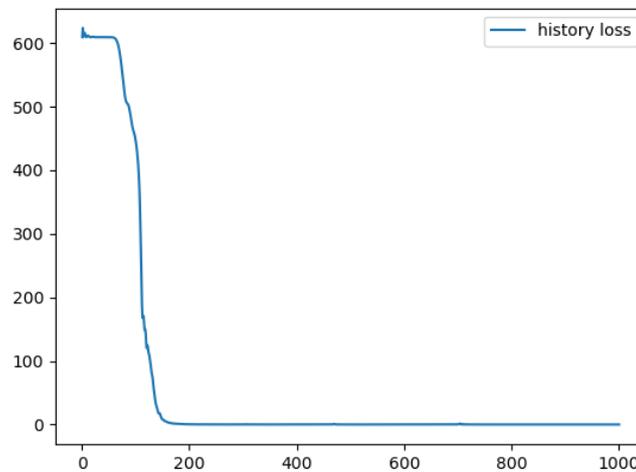
## 6.2 Solução da Equação de Poisson 2D

Nesta seção, exploramos a aplicação da rede neural do tipo Physics-Informed Neural Networks (PINNs) para resolver a equação de Poisson em duas dimensões. Utilizamos exemplos relevantes apresentados em (AGGARWAL; UGAIL, 2019) como base para demonstrar a eficácia e a precisão do método proposto. Por meio da abordagem de PINNs, buscamos não apenas obter soluções precisas para a equação de Poisson, mas também explorar a capacidade das redes neurais em lidar com problemas de física computacional de forma eficiente e versátil. Nossos experimentos e resultados são apresentados detalhadamente nesta seção, seguidos de uma análise crítica das descobertas.

- Exemplo 1:

$$\begin{cases}
 \nabla^2 u(x, y) = -5\pi^2 \text{sen}(\pi x) \cos(2\pi y), & (x, y) \in (0, 1) \times (0, 1) \\
 u(x, 0) = \text{sen}(\pi x), & x \in (0, 1) \\
 u(x, 1) = \text{sen}(\pi x), & x \in (0, 1) \\
 u(0, y) = 0, & y \in (0, 1) \\
 u(1, y) = 0, & y \in (0, 1)
 \end{cases} \quad (6.4)$$

Figura 6.10 – Treinamento do Exemplo 1 com 1000 Épocas



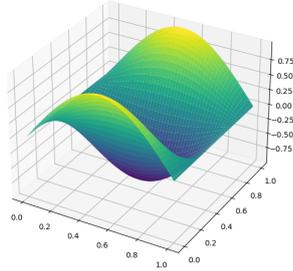
Fonte: Próprio Autor

Para o exemplo 1, o treinamento da Physics-Informed Neural Network (PINN) foi realizado ao longo de 1000 épocas, conforme representado no gráfico anteriormente apresentado. A seguir, são apresentados os resultados obtidos com o treinamento da PINN para o exemplo 1, juntamente com uma comparação com a solução obtida utilizando o método das diferenças finitas.

O treinamento da PINN resultou em um conjunto de parâmetros otimizados, os quais foram empregados para calcular a solução aproximada da equação em questão. A precisão e a eficácia da solução obtida foram avaliadas mediante a diferença entre a solução utilizando diferenças finitas e a solução da rede neural.

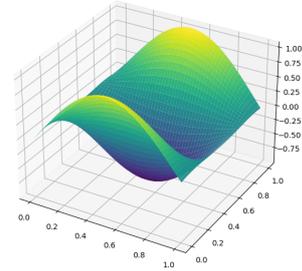
Para o treinamento, foi utilizado uma taxa de aprendizado de 0,001. A rede neural implementada foi uma rede densa com três camadas: duas camadas ocultas, cada uma com 1500 neurônios e ativação tangente hiperbólica (tanh), e uma camada de saída com um neurônio, que não possui função de ativação especificada, o que implica em uma saída linear. Na camada de entrada, a rede espera dados de entrada bidimensionais.

Figura 6.11 – Solução do Exemplo 1 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.12 – Solução do Exemplo 1 com a PINN

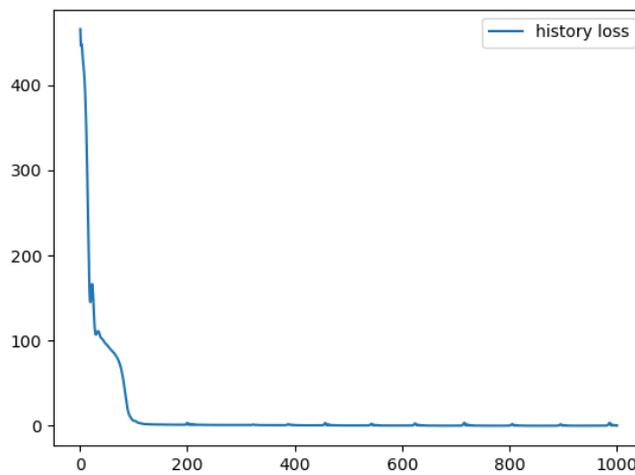


Fonte: Próprio Autor

- Exemplo 2:

$$\left\{ \begin{array}{ll} \nabla^2 u(x, y) = -4 e^{x^2+y^2} (x^2 + y^2 + 1), & (x, y) \in (0, 1) \times (0, 1) \\ u(x, 0) = e^{x^2}, & x \in (0, 1) \\ u(x, 1) = e^{x^2+1}, & x \in (0, 1) \\ u(0, y) = e^{y^2}, & y \in (0, 1) \\ u(1, y) = e^{y^2+1}, & y \in (0, 1) \end{array} \right. \quad (6.5)$$

Figura 6.13 – Treinamento do Exemplo 2 com 1000 Épocas

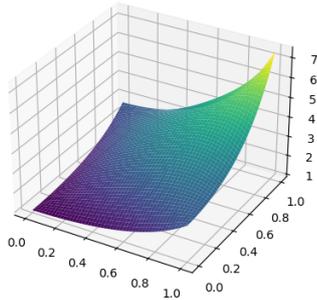


Fonte: Próprio Autor

Assim como foi feito no Exemplo 1 repetiu-se o processo no Exemplo 2, o treinamento da Physics-Informed Neural Network (PINN) foi realizado ao longo

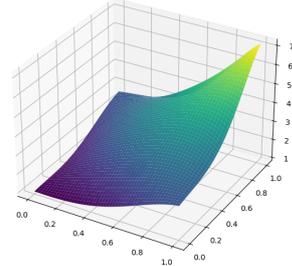
de 1000 épocas, conforme representado no gráfico anteriormente apresentado. A seguir, são apresentados os resultados obtidos com o treinamento da PINN para o Exemplo 2, juntamente com uma comparação com a solução obtida utilizando o método das diferenças finitas.

Figura 6.14 – Solução do Exemplo 2 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.15 – Solução do Exemplo 2 com a PINN

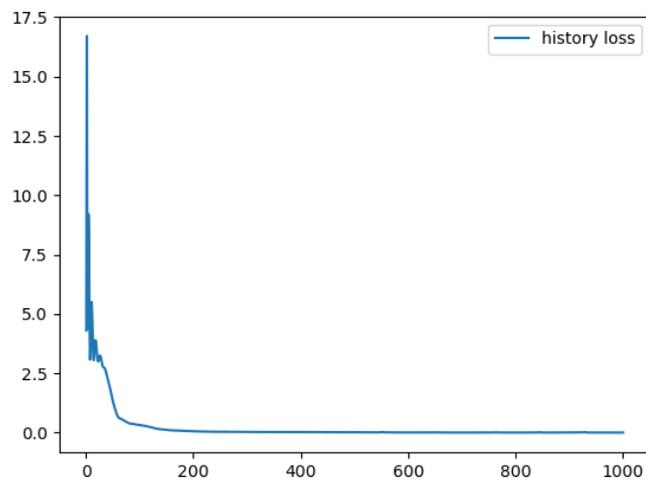


Fonte: Próprio Autor

- Exemplo 3:

$$\left\{ \begin{array}{ll} \nabla^2 u(x, y) = e^{-x} (x - 2 + y^3 + 6y), & (x, y) \in (0, 1) \times (0, 1) \\ u(x, 0) = xe^{-1}, & x \in (0, 1) \\ u(x, 1) = e^{-x} (x + 1), & x \in (0, 1) \\ u(0, y) = y^3, & y \in (0, 1) \\ u(1, y) = (1 + y^3) e^{-1}, & y \in (0, 1) \end{array} \right. \quad (6.6)$$

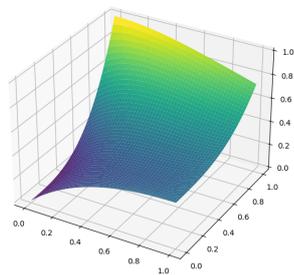
Figura 6.16 – Treinamento do Exemplo 3 com 1000 Épocas



Fonte: Próprio Autor

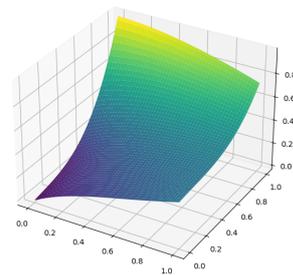
Assim como foi feito no Exemplo 1 e no exemplo 2 repetiu-se o processo no Exemplo 3, o treinamento da Physics-Informed Neural Network (PINN) foi realizado ao longo de 1000 épocas, conforme representado no gráfico anteriormente apresentado. A seguir, são apresentados os resultados obtidos com o treinamento da PINN para o Exemplo 3, juntamente com uma comparação com a solução obtida utilizando o método das diferenças finitas.

Figura 6.17 – Solução do Exemplo 3 com Diferenças Finitas



Fonte: Próprio Autor

Figura 6.18 – Solução do exemplo 3 com a PINN



Fonte: Próprio Autor

# Referências

AGGARWAL, R.; UGAIL, H. On the solution of poisson's equation using deep learning. In: **2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)**. [S.l.: s.n.], 2019. p. 1–8. Citado na página 36.

ALKHADHR, S.; ALMEKKAWY, M. Wave equation modeling via physics-informed neural networks: Models of soft and hard constraints for initial and boundary conditions. **Sensors**, v. 23, n. 5, 2023. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/23/5/2792>>. Citado 3 vezes nas páginas 23, 24 e 25.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: LECHEVALLIER, Y.; SAPORTA, G. (Ed.). **Proceedings of COMPSTAT'2010**. Heidelberg: Physica-Verlag HD, 2010. p. 177–186. ISBN 978-3-7908-2604-3. Citado na página 27.

EVANS, L. C. **Partial differential equations**. Second edition. Providence, Rhode Island: American Mathematical Society, 2022. (Graduate studies in mathematics, 19). ISBN 9781470469429 9780821849743. Citado 2 vezes nas páginas 12 e 13.

FIORIN, D. V.; MARTINS, F. R.; SCHUCH, N. J.; PEREIRA, E. B. Aplicações de redes neurais e previsões de disponibilidade de recursos energéticos solares. **Revista Brasileira de Ensino de Física**, Sociedade Brasileira de Física, v. 33, n. 1, p. 01–20, Jan 2011. ISSN 1806-1117. Disponível em: <<https://doi.org/10.1590/S1806-11172011000100009>>. Citado na página 26.

GÉRON, A. **Mãos à obra: aprendizado de máquina com scikit-learn & tensorflow**. Alta Books, 2019. ISBN 9788550803814. Disponível em: <<https://books.google.com.br/books?id=7Y7izQEACAAJ>>. Citado na página 22.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge, Massachusetts: The MIT Press, 2016. (Adaptive computation and machine learning). ISBN 9780262035613. Citado na página 19.

HAYKIN, S. **Neural networks and learning machines**. 3rd ed. ed. New York: Prentice Hall, 2009. OCLC: ocn237325326. ISBN 9780131471399. Citado 3 vezes nas páginas 18, 20 e 21.

KARNIADAKIS, G. E.; KEVREKIDIS, I. G.; LU, L.; PERDIKARIS, P.; WANG, S.; YANG, L. Physics-informed machine learning. **Nature Reviews Physics**, v. 3, n. 6, p. 422–440, maio 2021. ISSN 2522-5820. Disponível em: <<https://www.nature.com/articles/s42254-021-00314-5>>. Citado 2 vezes nas páginas 23 e 25.

RAISSI, M.; PERDIKARIS, P.; KARNIADAKIS, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. **Journal of Computational Physics**, v. 378, p. 686–707, 2019. ISSN 0021-9991. Disponível em: <<https://www.sciencedirect.com/journal/journal-of-computational-physics>>.

---

[//www.sciencedirect.com/science/article/pii/S0021999118307125](http://www.sciencedirect.com/science/article/pii/S0021999118307125)>. Citado 3 vezes nas páginas 23, 25 e 27.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: from theory to algorithms**. New York, NY, USA: Cambridge University Press, 2014. ISBN 9781107057135. Citado na página 20.

STRIKWERDA, J. C. **Finite difference schemes and partial differential equations**. 2nd ed.. ed. Philadelphia, Pa.: Society for Industrial and Applied Mathematics (SIAM), 2004. OCLC: 693942995. ISBN 9780898717938. Citado na página 30.