# Usando Redes Neurais Guiadas pela Física (PINNs) para resolver EDPs


Laboratório de Dinâmica dos Fluidos Computacional

Prof. Roberto Ribeiro Santos Júnior
Prof. Thiago de Oliveira Quinelato

Guilherme Furquim
Guilherme Ozanski
Larry Steffen Bertoncello
Lucas Xavier
Samuel Kutz Paranhos

Universidade Federal do Paraná


8 de novembro de 2024

# LabFluid

# Apresentação

## Professores

- Professores:
  - Profa. Ailin Ruiz de Zarate Fabregas
  - Prof. Elias Alfredo Gudiño Rojas
  - Prof. Roberto Ribeiro Santos Jr.
  - Prof. Thiago de Oliveira Quinelato




# Sumário

# Equações Diferenciais Parciais (EDPs)

# Equações Diferenciais Parciais (EDPs)

## Introdução

No contexto de modelagem matemática, muitas vezes encontramos cenários onde modelamos um problema como a solução de equações.

Alguns tipos comuns de equações são:

- equações algébricas (soluções são números);
- relações de recorrência (soluções são sequências);
- **equações diferenciais ordinárias/parciais** (soluções são **funções**).

Em equações diferenciais, estamos à procura de **funções** que satisfazem as equações do nosso modelo.

# Equações Diferenciais Parciais (EDPs)

## Introdução

No contexto de modelagem matemática, muitas vezes encontramos cenários onde modelamos um problema como a solução de equações.

Alguns tipos comuns de equações são:

- equações algébricas (soluções são números);
- relações de recorrência (soluções são sequências);
- **equações diferenciais ordinárias/parciais** (soluções são **funções**).

Em equações diferenciais, estamos à procura de **funções** que satisfazem as equações do nosso modelo.

# Equações Diferenciais Parciais (EDPs)

## Introdução

No contexto de modelagem matemática, muitas vezes encontramos cenários onde modelamos um problema como a solução de equações.

Alguns tipos comuns de equações são:

- equações algébricas (soluções são números);
- relações de recorrência (soluções são sequências);
- **equações diferenciais ordinárias/parciais** (soluções são **funções**).

Em equações diferenciais, estamos à procura de **funções** que satisfazem as equações do nosso modelo.

# Equações Diferenciais Parciais (EDPs)

## Exemplo de EDO

**Equações diferenciais ordinárias** envolvem encontrar uma função de uma variável que satisfaz certas condições sobre suas derivadas, por exemplo:

- Problema de Valor Inicial (EDO):

$$\begin{cases} \boldsymbol{y}'(t) &= \lambda \boldsymbol{y}(t) \qquad \lambda \in \mathbb{R}, \\ \boldsymbol{y}(0) &= y_0, \end{cases}$$

com $y : \mathbb{R} \to \mathbb{R}$.

- Solução:

$$\boldsymbol{y}(t) = y_0 e^{\lambda t} \qquad \text{(uma função)}$$

# Equações Diferenciais Parciais (EDPs)

## Exemplo de EDO

**Equações diferenciais ordinárias** envolvem encontrar uma função de uma variável que satisfaz certas condições sobre suas derivadas, por exemplo:

- Problema de Valor Inicial (EDO):

$$\begin{cases} \boldsymbol{y}'(t) & = \lambda \boldsymbol{y}(t) \qquad \lambda \in \mathbb{R}, \\ \boldsymbol{y}(0) & = y_0, \end{cases}$$

  com $y : \mathbb{R} \to \mathbb{R}$.

- Solução:

$$\boldsymbol{y}(t) = y_0 e^{\lambda t} \qquad \text{(uma função)}$$

# Equações Diferenciais Parciais (EDPs)

## Exemplo de EDP

**Equações diferenciais parciais** envolvem encontrar uma função de várias variáveis que satisfaz condições sobre suas derivadas parciais. Mais especificadamente, tome o seguinte exemplo:

- Equação do Calor (EDP):

$$\begin{aligned} &u_t = \alpha u_{xx}, && (x,t) \in (-\pi,\pi)\times(0,6), \\ &u(x,0) = f(x), && x \in (-\pi,\pi), \qquad u(\pm\pi,t) = 0, \qquad t \in (0,6), \end{aligned}$$

  onde $u = u(x,t)$, $\alpha = 1$ e $f(x) = 3\,\mathrm{sen}(x)$.

- Solução:

$$u(x,t) = 3e^{-t}\,\mathrm{sen}(x)$$

  gif da equação

**Equações diferenciais parciais** envolvem encontrar uma função de várias variáveis que satisfaz condições sobre suas derivadas parciais. Mais especificadamente, tome o seguinte exemplo:

- Equação do Calor (EDP):

$$\begin{aligned} &u_t = \alpha u_{xx}, && (x,t) \in (-\pi,\pi) \times (0,6), \\ &u(x,0) = f(x), && x \in (-\pi,\pi), \qquad u(\pm\pi,t) = 0, \qquad t \in (0,6), \end{aligned}$$

onde $u = u(x,t)$, $\alpha = 1$ e $f(x) = 3\,\mathrm{sen}(x)$.

- Solução:

$$u(x,t) = 3e^{-t}\,\mathrm{sen}(x)$$

gif da equação

# Método de Diferenças Finitas

# Diferenças Finitas

## Construção do Método

O método de diferenças finitas, por exemplo, procura aproximar a solução de uma equação diferencial parcial substituindo o operador diferencial por um operador de diferenças.

Vamos aplicar o esquema de Diferenças Finitas na Equação do Transporte.

$$
\begin{aligned}
u_x + au_t &= 0 \qquad && (x,t) \in (-3,3) \times (0,1),\\
u(x,0) &= f(x) \qquad && x \in (-3,3),
\end{aligned}
$$

onde $u = u(x,t)$, $a = 2$ e $f(x) = e^{-10x^2}$.

# Diferenças Finitas

## Construção do Método

O método de diferenças finitas, por exemplo, procura aproximar a solução de uma equação diferencial parcial substituindo o operador diferencial por um operador de diferenças.

Vamos aplicar o esquema de Diferenças Finitas na Equação do Transporte.

$$
\begin{aligned}
u_x + a u_t &= 0 \qquad && (x,t) \in (-3,3) \times (0,1),\\
u(x,0) &= f(x) \qquad && x \in (-3,3),
\end{aligned}
$$

onde $u = u(x,t)$, $a = 2$ e $f(x) = e^{-10x^2}$.

# Diferenças Finitas

## Construção do Método

Primeiro, aproximamos as derivadas da equação:

$$u_t(x,t) = \lim_{\Delta t \to 0} \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}$$

$$u_x(x,t) = \lim_{\Delta x \to 0} \frac{u(x-\Delta x, t) - u(x,t)}{-\Delta x}$$

$$u_t(x,t) \approx \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}$$

$$u_x(x,t) \approx \frac{u(x,t) - u(x-\Delta x, t)}{\Delta x}$$

# Diferenças Finitas

## Construção do Método

Primeiro, aproximamos as derivadas da equação:

$$u_t(x,t) = \lim_{\Delta t \to 0} \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}$$

$$u_x(x,t) = \lim_{\Delta x \to 0} \frac{u(x-\Delta x, t) - u(x,t)}{-\Delta x}$$

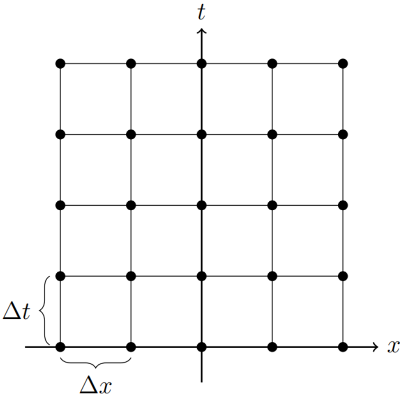
$$u_t(x,t) \approx \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t}$$

$$u_x(x,t) \approx \frac{u(x,t) - u(x-\Delta x, t)}{\Delta x}$$

# Diferenças Finitas

## Construção do Método

$$u(x,t) = u(j\Delta x, n\Delta t) = u_j^n$$



# Diferenças Finitas

## Construção do Método

$$[u_t]j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}; \; [u_x]_j^n \approx \frac{u_j^n - uj - 1^n}{\Delta x}$$

$$[u_t]_j^n + a[u_x]_j^n = 0$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0$$

Operador Diferencial:

$$\mathcal{D}u = u_t + au_x$$

Operador de Diferenças:

$$Du = \frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x}$$

# Diferenças Finitas

## Construção do Método

$$[u_t]j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}; \; [u_x]_j^n \approx \frac{u_j^n - uj - 1^n}{\Delta x}$$

$$[u_t]_j^n + a[u_x]_j^n = 0$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0$$

Operador Diferencial:

$$\mathcal{D}u = u_t + au_x$$

Operador de Diferenças:

$$Du = \frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x}$$

# Diferenças Finitas

## Construção do Método

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0$$

$$u_j^{n+1} = u_j^n - a\frac{\Delta t}{\Delta x}(u_j^n - u_{j-1}^n);\ \ \lambda = a\frac{\Delta t}{\Delta x}$$

$$u_j^{n+1} = u_j^n - \lambda(u_j^n - u_{j-1}^n)$$

Da condição inicial da EDP:

$$u(x,0) = f(x) = e^{-10x^2} \Leftrightarrow u_j^0 = e^{-10(j\Delta x)^2}$$

A partir dos valores $u_j^0$ conseguimos construir a solução em tempos maiores que zero.

# Diferenças Finitas

## Construção do Método

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0$$

$$u_j^{n+1} = u_j^n - a\frac{\Delta t}{\Delta x}(u_j^n - u_{j-1}^n);\ \ \lambda = a\frac{\Delta t}{\Delta x}$$

$$u_j^{n+1} = u_j^n - \lambda(u_j^n - u_{j-1}^n)$$

Da condição inicial da EDP:

$$u(x,0) = f(x) = e^{-10x^2} \Leftrightarrow u_j^0 = e^{-10(j\Delta x)^2}$$

A partir dos valores $u_j^0$ conseguimos construir a solução em tempos maiores que zero.

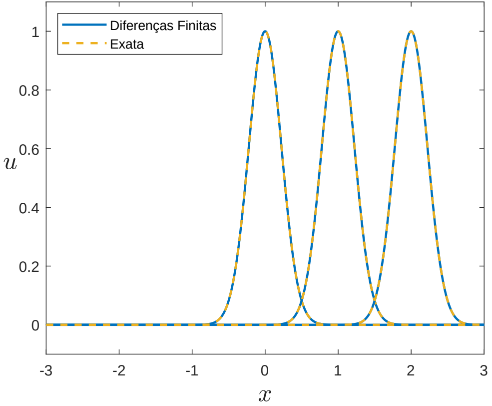



Figura: Comparação da Solução com $\lambda = 1$; $t = \{0, 0.5, 1\}$

# Diferenças Finitas

## Testes Numéricos

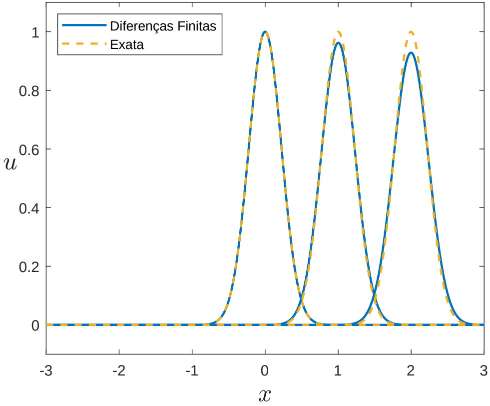



Figura: Comparação da Solução com $\lambda = 0.6$; $t = \{0, 0.501, 0.999\}$

# Diferenças Finitas

## Testes Numéricos

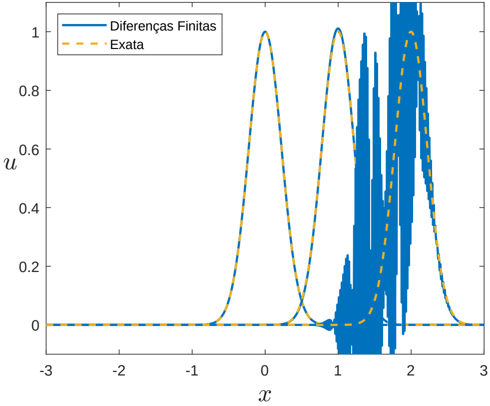



Figura: Comparação da Solução com $\lambda = 1.11$; $t = \{0, 0.4995, 0.999\}$

# Redes Neurais

# Redes Neurais

## Estrutura

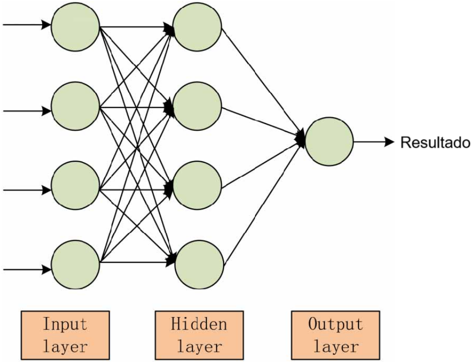



Figura: Rede Neural com 3 camadas

# Redes Neurais

## Estrutura

Essencialmente uma rede neural é uma **função parametrizada** $f(x;\theta)$:

- $x$ representa os dados de entrada
- $\theta$ representa os parâmetros do modelo (pesos e vieses).
- Através do ajuste de $\theta$, $f$ tentará **aproximar** uma saída desejada específica para cada $x$.

# Redes Neurais

## Estrutura

Essencialmente uma rede neural é uma **função parametrizada** $f(x;\theta)$:

- $x$ representa os dados de entrada
- $\theta$ representa os parâmetros do modelo (pesos e vieses).
- Através do ajuste de $\theta$, $f$ tentará **aproximar** uma saída desejada específica para cada $x$.

# Redes Neurais

## Estrutura

Essencialmente uma rede neural é uma **função parametrizada** $f(x;\theta)$:

- $x$ representa os dados de entrada
- $\theta$ representa os parâmetros do modelo (pesos e vieses).
- Através do ajuste de $\theta$, $f$ tentará **aproximar** uma saída desejada específica para cada $x$.

# Redes Neurais

## Estrutura

Uma rede neural simples pode ser representada como:

$$f(x) = \sigma\big(W_2\,\sigma(W_1 x + b_1) + b_2\big)$$

onde:

- $W_1$ e $W_2$ são as matrizes de pesos;
- $b_1$ e $b_2$ são os vetores de bias;
- $\sigma$ é uma função de ativação não linear.

# Redes Neurais

## Estrutura

Uma rede neural simples pode ser representada como:

$$f(x) = \sigma\big(W_2\,\sigma(W_1 x + b_1) + b_2\big)$$

onde:

- $W_1$ e $W_2$ são as matrizes de pesos;
- $b_1$ e $b_2$ são os vetores de bias;
- $\sigma$ é uma função de ativação não linear.

# Redes Neurais

## Estrutura

Uma rede neural simples pode ser representada como:

$$f(x) = \sigma\big(W_2\,\sigma(W_1 x + b_1) + b_2\big)$$

onde:

- $W_1$ e $W_2$ são as matrizes de pesos;
- $b_1$ e $b_2$ são os vetores de bias;
- $\sigma$ é uma função de ativação não linear.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada x**;
- A partir de uma configuração inicial de **pesos e viéses**($\theta$) a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada x**;
- A partir de uma configuração inicial de **pesos e viéses**($\theta$) a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada x**;
- A partir de uma configuração inicial de **pesos e viéses($\theta$)** a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada x**;
- A partir de uma configuração inicial de **pesos e viéses($\theta$)** a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada** $\mathbf{x}$;
- A partir de uma configuração inicial de **pesos e viéses**$(\theta)$ a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

**Modus Operanti:**

- Prover um banco de imagens como **entrada x**;
- A partir de uma configuração inicial de **pesos e viéses($\theta$)** a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo

Considere uma rede neural que tenta acertar o número escrito numa imagem.

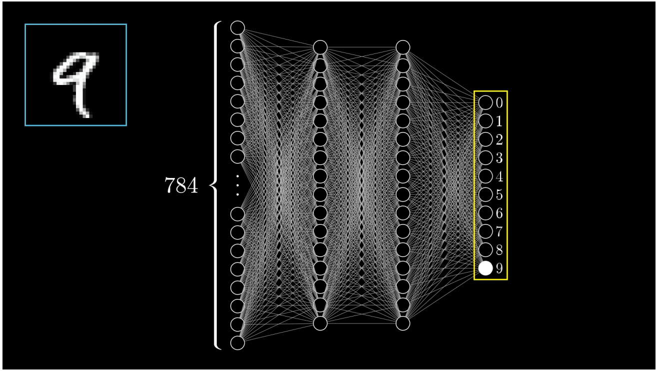
**Modus Operanti:**

- Prover um banco de imagens como **entrada** $\mathbf{x}$;
- A partir de uma configuração inicial de **pesos e viéses**$(\theta)$ a rede fará um chute;
- $\theta$ é alterado conforme a magnitude do erro.
- A rede chuta mais uma vez, e o processo segue.

Uma vez treinada, a rede está pronta para reconhecer imagens novas para ela.

# Redes Neurais

## Exemplo



# PINNs

# PINNs

## O que é?

PINN[1] é um acrônimo para Physics Informed Neural Network, ou Rede Neural Informada de Física, em português. Que basicamente significa uma rede neural capaz de encontrar uma solução numérica para uma EDP (Equação Diferencial Parcial) – daí o *informada de física*, pois muitos fenômenos físicos são descritos por EDPs.

[1] Maziar Raissi, Paris Perdikaris e George Em Karniadakis (2017). "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". Em: *arXiv preprint arXiv:1711.10561*.

# PINNs

## O que é?

- Considere uma equação diferencial parcial na forma geral:
- $u_t + \mathcal{N}[u; \lambda] = 0.$
- Onde $\mathcal{N}$ é um operador diferencial (possivelmente não-linear) parametrizado por $\lambda$
- Nessa formulação, a equação do transporte é descrita com $\mathcal{N}[u; a] = au_x$, assim $u_t + au_x = 0.$
- Sendo $u(x, t)$ a solução da EDP em questão, para $x \in \Omega \subset \mathbb{R}^D$ e $t \in [0, T]$.

# PINNs

## O que é?

- Considere uma equação diferencial parcial na forma geral:
- $u_t + \mathcal{N}[u; \lambda] = 0.$
- Onde $\mathcal{N}$ é um operador diferencial (possivelmente não-linear) parametrizado por $\lambda$
- Nessa formulação, a equação do transporte é descrita com $\mathcal{N}[u; a] = au_x$, assim $u_t + au_x = 0.$
- Sendo $u(x,t)$ a solução da EDP em questão, para $x \in \Omega \subset \mathbb{R}^D$ e $t \in [0, T]$.

# PINNs

## O que é?

- Considere uma equação diferencial parcial na forma geral:
- $u_t + \mathcal{N}[u; \lambda] = 0.$
- Onde $\mathcal{N}$ é um operador diferencial (possivelmente não-linear) parametrizado por $\lambda$
- Nessa formulação, a equação do transporte é descrita com $\mathcal{N}[u; a] = au_x$, assim $u_t + au_x = 0.$
- Sendo $u(x, t)$ a solução da EDP em questão, para $x \in \Omega \subset \mathbb{R}^D$ e $t \in [0, T]$.

# PINNs

## O que é?

- Considere uma equação diferencial parcial na forma geral:
- $u_t + \mathcal{N}[u; \lambda] = 0.$
- Onde $\mathcal{N}$ é um operador diferencial (possivelmente não-linear) parametrizado por $\lambda$
- Nessa formulação, a equação do transporte é descrita com $\mathcal{N}[u; a] = au_x$, assim $u_t + au_x = 0.$
- Sendo $u(x, t)$ a solução da EDP em questão, para $x \in \Omega \subset \mathbb{R}^D$ e $t \in [0, T]$.

# PINNs

## O que é?

- Considere uma equação diferencial parcial na forma geral:
- $u_t + \mathcal{N}[u; \lambda] = 0.$
- Onde $\mathcal{N}$ é um operador diferencial (possivelmente não-linear) parametrizado por $\lambda$
- Nessa formulação, a equação do transporte é descrita com $\mathcal{N}[u; a] = au_x$, assim $u_t + au_x = 0.$
- Sendo $u(x, t)$ a solução da EDP em questão, para $x \in \Omega \subset \mathbb{R}^D$ e $t \in [0, T]$.

O que difere uma PINN de outras redes neurais é a forma como a função do erro é definida. Defina $D := u_t + \mathcal{N}[u; \lambda]$ como o lado esquerdo da equação. Vamos resolver um problema generico da forma:

$$
\begin{aligned}
&\min |D|, && (x,t) \in \Omega \times [0,T],\\
&u(x,0) = u_0(x), && x \in \Omega,\\
&u(x,t) = u_b(x,t), && (x,t) \in \partial\Omega \times [0,T].
\end{aligned}
$$

Uma PINN é capaz de aproximar $u(x,t)$ por minimizar a seguinte função de erro.

# PINNs

## Formulação

- $\text{EMQ} = \text{EMQ}_0 + \text{EMQ}_b + \text{EMQ}_{\mathcal{D}}$
- $\text{EQM}_{\mathcal{D}} = \dfrac{1}{N_{\mathcal{D}}} \displaystyle\sum_{i=1}^{N_{\mathcal{D}}} \big|\mathcal{D}(x_i, t_i)\big|^2$, onde $(x_i, t_i)$, $i = 1, \cdots, N_{\mathcal{D}}$, representa $N_{\mathcal{D}}$ pontos de colocação;
- $\text{EQM}_0 = \dfrac{1}{N_0} \displaystyle\sum_{i=1}^{N_0} \big|(u(x_i, 0) - u_0(x_i)\big|^2$, onde $x_i$, $i = 1, \cdots, N_0$ representa $N_0$ pontos iniciais;
- $\text{EQM}_b = \dfrac{1}{N_b} \displaystyle\sum_{i=1}^{N_b} \big|u(x_i, t_i) - u_b(x_i, t_i)\big|^2$, onde $(x_i, t_i)$, $i = 1, \cdots, N_b$ corresponde a $N_b$ pontos na borda;

# PINNs

## Formulação

- $\text{EMQ} = \text{EMQ}_0 + \text{EMQ}_b + \text{EMQ}_{\mathcal{D}}$
- $\text{EQM}_{\mathcal{D}} = \dfrac{1}{N_{\mathcal{D}}} \displaystyle\sum_{i=1}^{N_{\mathcal{D}}} \left| \mathcal{D}(x_i, t_i) \right|^2$, onde $(x_i, t_i)$, $i = 1, \cdots, N_{\mathcal{D}}$, representa $N_{\mathcal{D}}$ pontos de colocação;
- $\text{EQM}_0 = \dfrac{1}{N_0} \displaystyle\sum_{i=1}^{N_0} \left| (u(x_i, 0) - u_0(x_i) \right|^2$, onde $x_i$, $i = 1, \cdots, N_0$ representa $N_0$ pontos iniciais;
- $\text{EQM}_b = \dfrac{1}{N_b} \displaystyle\sum_{i=1}^{N_b} \left| u(x_i, t_i) - u_b(x_i, t_i) \right|^2$, onde $(x_i, t_i)$, $i = 1, \cdots, N_b$ corresponde a $N_b$ pontos na borda;

DeepXDE é uma biblioteca científica para aprendizado de máquina para redes neurais informadas pela física.

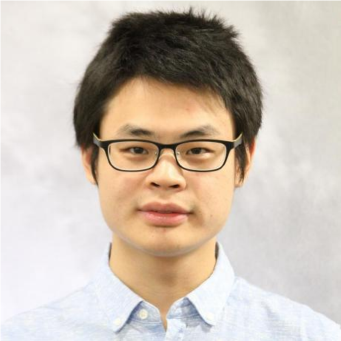


Figura: Lu Lu, Yale University.

Resolvemos via PINNs a equação do transporte para $a = 2$: $u_t + 2u_x = 0$.

| **Parâmetro** | **Valor** |
|---|---|
| Domínio espacial ($x$) | $[-5, 15]$ |
| Domínio temporal ($t$) | $[0, 7.5]$ |
| Pontos de colocação no domínio ($N_{\mathcal{D}}$) | 10 000 |
| Pontos de colocação na fronteira ($N_b$) | 4 000 |
| Pontos de colocação no valor inicial ($N_0$) | 4 000 |
| Função de Ativação | tanh |
| Neurônios por camada | 60 |
| Camadas intermediárias | 5 |
| Iterações do otimizador | 10 000 |
| Taxa de Aprendizado | 0.001 |

# PINNs - Exemplo

## Exemplo: Equação do Transporte

Obtemos um erro L2 relativo de no máximo $5 \times 10^{-3}$, com a seguinte evolução ao longo do tempo.

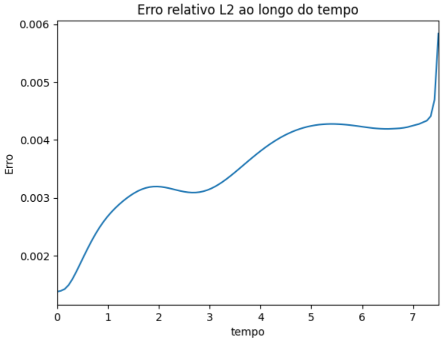



Figura: Erro Relativo L2 ao longo do tempo

# PINNs

## Equação do Transporte

gifs da solução