

# PINNs para Resolução da Equação de Burgers com Transferência de Aprendizagem pelo Esquema Theta

Vitória Biesek<sup>1</sup> and Pedro Henrique de Almeida Konzen<sup>2</sup>

<sup>1,2</sup>PPGMAp, IME, UFRGS, Porto Alegre, RS, Brasil

## Resumo

A equação de Burgers é um caso de teste bem estabelecido na modelagem computacional de diversos fenômenos, como dinâmica de fluidos, dinâmica de gases, teoria do choque, cosmologia e outros. Neste trabalho, apresentamos a aplicação de rede neural fisicamente informada (PINNs, do inglês, *physics-informed neural networks*) com uma abordagem de transferência de aprendizagem pelo esquema  $\theta$  para resolver a equação de Burgers. A abordagem proposta consiste em buscar uma solução discreta no tempo por meio de uma sequência de redes neurais artificiais (ANNs, do inglês, *artificial neural networks*). A cada passo de tempo, a ANN anterior transfere seu aprendizado para o próximo modelo de rede, que aprende a solução no tempo corrente pela minimização de uma função de perda baseada na aproximação pelo esquema  $\theta$  da equação de Burgers. Para testar esta abordagem, é proposto um problema de referência com solução analítica. Em comparação com os modelos PINN usuais, a abordagem proposta tem a vantagem de exigir arquiteturas de redes neurais menores com resultados precisos semelhantes e potencialmente diminuir os custos computacionais.

## 1 Introdução

A equação de Burgers é encontrada na modelagem na dinâmica de fluidos, dinâmica de gases, teoria de choque, cosmologia, entre outras (Bonkile et al., 2018). É bastante utilizada para testes na análise matemática e numérica de equações diferenciais convectivas-difusivas (Konzen et al., 2017). Consideramos a equação de Burgers viscosa com condições inicial e de contorno do tipo Dirichlet, dadas por

$$u_t + uu_x = \nu u_{xx}, \quad (t, x) \in (0, t_f] \times (0, 1) \quad (1a)$$

$$u(0, x) = u_0(x), \quad x \in [0, 1], \quad (1b)$$

$$u(t, 0) = u(t, 1) = 0, \quad t \in [0, t_f], \quad (1c)$$

Por exemplo, em uma aplicação de dinâmica de fluidos, ela modela a velocidade  $u = u(t, x)$  (m/s) no tempo  $t$  (s) e ponto  $x$  (m), de um fluido com viscosidade cinemática  $\nu$  (m<sup>2</sup>/s).

Neste trabalho, investigamos a aplicação de redes neurais fisicamente informadas (PINNs, do inglês, *physics-informed neural networks*) com uma abordagem de transferência de aprendizagem pelo esquema  $\theta$  para resolver a equação de Burgers. PINNs (Raissi et al., 2019) são técnicas de aprendizagem profunda (em inglês, *deep learning*) (Goodfellow et al., 2016) para resolver equações diferenciais parciais (EDPs). Recentemente, elas foram aplicadas para resolver muitos problemas importantes, como equações incompressíveis de Navier-Stokes (Raissi et al., 2018; Jin et al., 2021), equações de Euler para fluxos aerodinâmicos de alta velocidade (Mao2020 et al., 2020), problemas de transferência de calor (Cai et al., 2021), e a equação de advecção (Vadyala et al., 2022).

Para uma EDP dependente do tempo, a PINN consiste em uma rede neural do tipo perceptron multicamadas (MLP, do inglês, *multilayer perceptron*) (Haykin, 2009) que aprende a solução  $u(t,x)$  das equações governantes. Usualmente, a MLP tem entradas  $(t,x)$  e saídas  $u(t,x)$ , ou entradas  $x$  e saídas  $\mathbf{u}(x) = \{u(t^{(k)},x)\}_{k=0}^{n_t}$  para dados  $n_t$  passos no tempo. Aqui, investigamos a aplicação de uma abordagem alternativa para a PINN, na qual a solução é dada por uma sequência de MLPs, uma a cada passo no tempo  $t^{(k)}$ . Dada a condição inicial, uma primeira MLP  $\mathcal{N}^{(0)}$  aprende a solução em  $t^{(0)} = 0$  treinando a aproximação da condição inicial  $u_0 = u_0(x)$ . Então, dado o tamanho do passo no tempo  $h_t > 0$ , o modelo  $\mathcal{N}^{(0)}$  transfere seu aprendizado para uma segunda MLP  $\mathcal{N}^{(1)}$ , que aprende a solução em  $t^{(1)} = h_t$  treinando com base na aproximação de Burgers com o esquema  $\theta$ . Este é um processo iterativo, onde o  $k$ -ésimo modelo  $\mathcal{N}^{(k)}$  transfere seu aprendizado para iniciar o modelo seguinte  $\mathcal{N}^{(k+1)}$ , que aprende a solução no tempo  $t^{(k+1)}$  através do esquema  $\theta$ .

Na sequência, apresentamos detalhes sobre a abordagem de PINNs com transferência de aprendizagem com o esquema  $\theta$ . Então, resultados para um estudo de caso são discutidos. Por fim, apresentamos algumas considerações finais.

## 2 PINNs

PINNs são técnicas de aprendizagem profunda para resolver EDPs. A solução é aprendida por uma ANN seguindo uma abordagem de aprendizagem supervisionada que incorpora a EDP na função de perda. Aqui, descrevemos uma abordagem PINN alternativa, onde uma sequência de MLPs, uma para cada passo de tempo discreto, é treinada com um esquema  $\theta$  de transferência de aprendizagem.

### 2.1 Perceptron Multicamadas

Neste trabalho, consideramos a rede da forma perceptron multicamadas (MLP)

$$\tilde{u} = \mathcal{N}\left(x; \left\{ \left( W^{(l)}, \mathbf{b}^{(l)}, \mathbf{f}^{(l)} \right) \right\}_{l=1}^{n_l} \right) \quad (2)$$

onde a tripla  $(W^{(l)}, \mathbf{b}^{(l)}, \mathbf{f}^{(l)})$  denota os pesos  $W^{(l)}$ , os *biases*  $\mathbf{b}^{(l)}$  e a função de ativação  $\mathbf{f}^{(l)}$  na  $l$ -ésima camada da rede,  $l = 1, 2, \dots, n_h + 1$ , com  $n_h$  um dado número de camadas escondidas. Como uma técnica de aprendizagem profunda, a saída da rede é computada através de sucessivas

composições

$$\mathbf{a}^{(l)} = \mathbf{f}^{(l)} \left( W^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \right), \quad (3)$$

com a entrada  $\mathbf{a}^{(0)} = x$ , a saída  $\mathbf{a}^{(n_h+1)} = \tilde{u}$  e  $l = 1, 2, \dots, n_h + 1$ . Assumindo a suavidade necessária de  $\mathbf{f}^{(l)}$ , as derivadas  $\tilde{u}_x$  e  $\tilde{u}_{xx}$  podem ser computadas por diferenciação automática como uma aplicação da regra da cadeia.

## 2.2 Esquema $\theta$

O esquema  $\theta$  para Eq. (1) consiste na iteração

$$\tilde{u}^{(0)}(x) = u_0(x), \quad (4a)$$

$$\tilde{u}^{(k)} = \tilde{u}^{(k-1)} + (1 - \theta)h_t \left( \nu \tilde{u}_{xx}^{(k-1)} - \tilde{u}^{(k-1)} \tilde{u}_x^{(k-1)} \right) + \theta h_t \left( \nu \tilde{u}_{xx}^{(k)} - \tilde{u}^{(k)} \tilde{u}_x^{(k)} \right), \quad (4b)$$

$$\tilde{u}^{(k)}(0) = \tilde{u}^{(k)}(1) = 0, \quad (4c)$$

onde  $\tilde{u}^{(k)}(x) \approx u(t^{(k)}, x)$  a cada tempo  $t^{(k)} = kh_t$ ,  $k = 0, 1, 2, \dots, n_t$ , com passo no tempo  $h_t = 1/n_t$ , para um dado número de passos no tempo  $n_t$ . Escolhendo  $\theta = 0$  temos o esquema de Euler Explícito, com  $\theta = 0.5$  temos o esquema de Crank-Nicolson e para  $\theta = 1$  temos o esquema de Euler Implícito.

A PINN proposta com esquema  $\theta$  de transferência de aprendizagem, consiste em treinar uma sequência de MLPs  $\tilde{u}^{(k)}(x) = \mathcal{N}^{(k)}(x)$ ,  $k = 0, 1, 2, \dots, n_t$ , com a entrada  $x$  e a saída sendo a estimativa de  $\tilde{u}^{(k)}$ .

Para a condição inicial Eq. (4a), a rede neural  $\mathcal{N}^{(0)}$  é treinada para minimizar a função perda

$$\mathcal{L}_0 := \frac{1}{n_s} \sum_{s=1}^{n_s} \left| \tilde{u}^{(0)}(x_s) - u_0(x_s) \right|^2, \quad (5)$$

com  $n_s$  amostras  $0 \leq x_s \leq 1$ . Sequencialmente,  $k = 1, 2, \dots, n_t$ , o aprendizado da rede neural  $\mathcal{N}^{(k-1)}$  é transferido para  $\mathcal{N}^{(k)}$ , a qual é treinada para minimizar a função perda

$$\mathcal{L} := \frac{1}{n_s - 2} \sum_{s=1}^{n_s-2} \left| \mathcal{R} \left( x; \tilde{u}^{(k)}, \tilde{u}^{(k-1)} \right) \right|^2 + \frac{1}{2} \left( \left| \tilde{u}^{(k)}(0) \right|^2 + \left| \tilde{u}^{(k)}(1) \right|^2 \right), \quad (6)$$

onde  $\mathcal{R}$  denota o resíduo

$$\mathcal{R} \left( x; \tilde{u}^{(k)}, \tilde{u}^{(k-1)} \right) := \tilde{u}^{(k)} - (1 - \theta)h_t \left( \nu \tilde{u}_{xx}^{(k-1)} - \tilde{u}^{(k-1)} \tilde{u}_x^{(k-1)} \right) - \theta h_t \left( \nu \tilde{u}_{xx}^{(k)} - \tilde{u}^{(k)} \tilde{u}_x^{(k)} \right). \quad (7)$$

As derivadas são computadas diretamente da rede neural por diferenciação automática.

O algoritmo básico da PINN com transferência de aprendizagem pelo esquema  $\theta$  é resumido como segue:

0. Determine o número  $n_t$  e o tamanho  $h_t$  do passo no tempo.
1. Determine a arquitetura  $\mathcal{N}^{(0)}$ .
2. Treine  $\mathcal{N}^{(0)}$  para minimizar a função perda inicial Eq. (5).
3. Para  $k = 1, 2, \dots, n_t$ :
  - a. (Transferência de aprendizagem.)  $\mathcal{N}^{(k)} \leftarrow \mathcal{N}^{(k-1)}$ .

b. Treine  $\mathcal{N}^{(k)}$  para minimizar a função perda Eq. (6).

Ao fim, a abordagem fornece a sequência de MLPs  $\{\mathcal{N}^{(k)}\}_{k=0}^{n_t}$ , cada uma fornecendo a solução estimada  $\tilde{u}^{(k)}(x) \approx u(t^{(k)}, x)$  da equação de Burgers (1). Normalmente, não há necessidade de armazenar toda a sequência, e o algoritmo precisa do armazenamento de apenas  $\mathcal{N}^{(k)}$  e  $\mathcal{N}^{(k-1)}$  em cada iteração.

### 2.3 Detalhes de implementação

Realizamos implementações em linguagem Python dos modelos de redes neurais com a ajuda da biblioteca PyTorch (Stevens et al., 2020). O modelo considerado é uma MLP com uma arquitetura  $1 - n_n \times n_h - 1$ , uma entrada,  $n_h$  camadas escondidas, cada uma com  $n_n$  unidades, e uma saída. Um estudo sobre as escolhas de  $n_h$  e  $n_n$  foi apresentado no trabalho de Biesek et al. (2023). Aqui, usamos  $n_h = 4$  e  $n_n = 50$  em todos os testes numéricos apresentados a seguir. Nas camadas ocultas, a tangente hiperbólica foi usada como função de ativação e na camada de saída a função identidade. Os modelos são treinados o método Adam (Kingma & Ba, 2014) como otimizador. Como critério de parada de treinamento, assumimos  $\mathcal{L}_0, \mathcal{L} < 10^{-6}$  (o cálculo foi realizado usando aritmética float 32-bits).

## 3 Resultados

Apresentamos experimentos numéricos para a equação de Burgers com condição inicial

$$u_0(x) = -\sin(\pi x). \quad (8)$$

No artigo de Basdevant et al. (1986), tem-se a seguinte solução analítica para este problema

$$u(x) = \frac{-\int_{-\infty}^{\infty} \sin(\pi(x-\eta))f(x-\eta)e^{-\frac{\eta^2}{4\nu t}} d\eta}{\int_{-\infty}^{\infty} f(x-\eta)e^{-\frac{\eta^2}{4\nu t}} d\eta}, \quad (9)$$

onde  $f(y) = e^{-\cos(\pi y)/(2\pi\nu)}$ . Assumimos  $\nu = 0.01/\pi$ .

Consideramos uma MLP com estrutura  $1 - 50 \times 4 - 1$ , uma entrada, 4 camadas escondidas com 50 unidades em cada camada, e uma saída. O modelo  $\mathcal{N}^{(k)}$  recebe os parâmetros iniciais de  $\mathcal{N}^{(k-1)}$  e seu treinamento depende do passo no tempo  $h_t$  e do número de amostras  $n_s$ . A fim de analisar a influência de tais parâmetros, realizamos testes numéricos variando os mesmos, para cada esquema  $\theta = 0$  (Euler explícito),  $\theta = 0.5$  (Crank-Nicolson) e  $\theta = 1$  (Euler implícito). Para  $\theta = 0$ , a Tabela 1 mostra os resultados  $n_e/\varepsilon_{\text{rel}}$ , sendo  $n_e$  o número total de épocas e o  $\varepsilon_{\text{rel}}$  valor final do erro relativo  $L^2$  dado por

$$\varepsilon_{\text{rel}}(t^{(k)}) := \frac{\|\tilde{u}^{(k)}(x) - u(t^{(k)}, x)\|_2}{\|u(t^{(k)}, x)\|_2}, \quad (10)$$

para  $k = n_t$ ,  $t = t_f = 1$ . Na Tabela 2 são apresentados resultados com o esquema  $\theta = 0.5$ . Resultados para o esquema  $\theta = 1$  são apresentados na Tabela 3.

Tabela 1: Euler explícito.

$h_t \backslash n_s$	20	200	2000
1E-1	-x-	-x-	-x-
1E-2	4E+4/7.5E-4	6E+4/2.1E-3	5E+4/3.0E-4
1E-3	16E+4/1.7E-2	14E+4/1.8E-2	1.3E+5/9.0E-3

Tabela 2: Crank-Nicolson.

$h_t \backslash n_s$	20	200	2000
1E-1	1E+4/2.8E-4	6E+4/3.1E-5	4E+4/4.2E-5
1E-2	4E+4/1.4E-2	4E+4/1.5E-3	3E+4/1.5E-3
1E-3	15E+4/4.0E-2	13E+4/6.9E-3	14E+4/4.2E-3

Tabela 3: Euler implícito.

$h_t \backslash n_s$	20	200	2000
1E-1	-x-	4E+4 / 4E-3	4E+4 / 4E-3
1E-2	3E+4 / 1E-3	3E+4 / 2E-4	5E+4 / 1E-3
1E-3	2E+5 / 1E-2	14E+4/3E-3	14E+4/5E-3

Os resultados apresentados nas Tabelas 1-3 indicam que a transferência de aprendizagem foi efetiva com todos os passos de tempo no esquema de Crank-Nicolson. Já, os esquemas de Euler demandaram um passo de tempo adequado, sendo que no caso de  $h_t = 10^{-1}$ , a aprendizagem das redes falhou em convergir. Da análise destes resultados, observamos que  $h_t = 10^{-2}$  e  $n_s = 200$  apresentam uma boa relação entre acurácia e custo de aprendizagem.

A Figura 1 mostra o gráfico das soluções PINNs (com transferência de aprendizagem com esquema  $\theta = 0, 0.5$  e  $1$ ), com  $n_s = 200$  e  $h_t = 10^{-2}$ , *versus* a solução analítica (linha contínua) do problema para os tempos  $t = 0.0, 0.1, 0.5, 0.9$  e  $1.0$ .

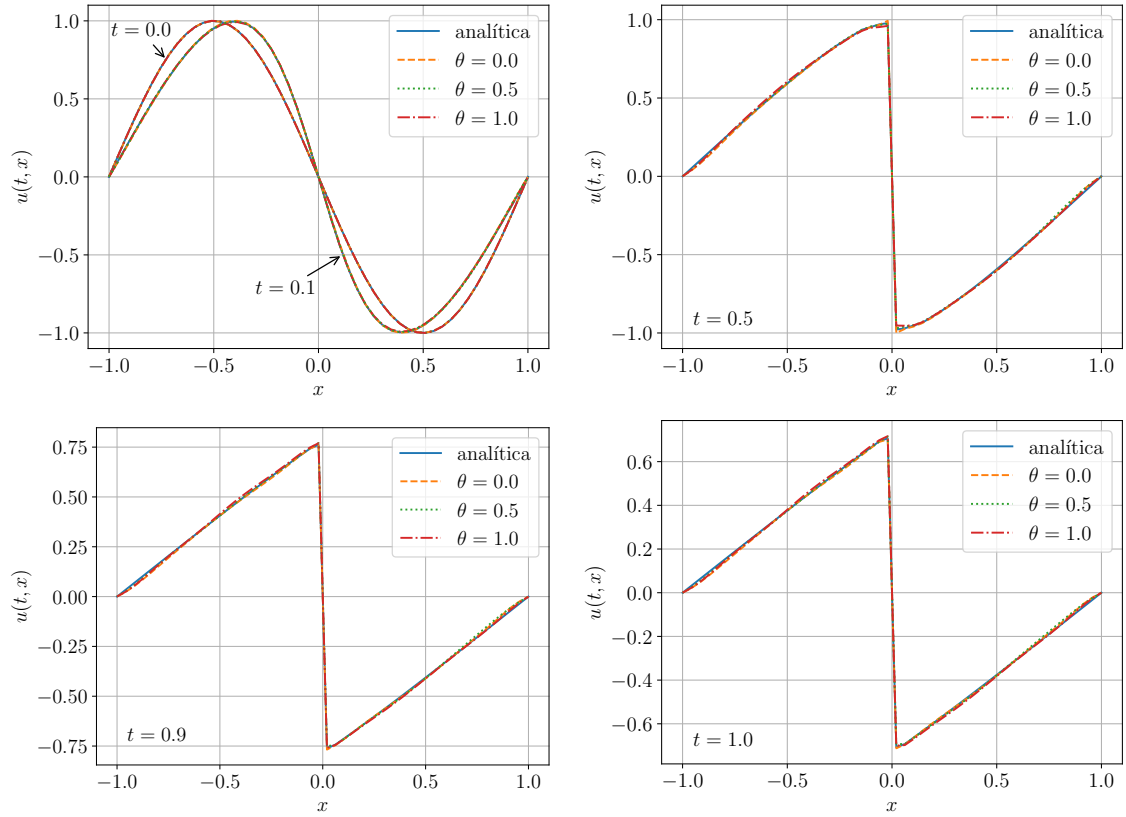


Figura 1: Comparações das soluções PINNs com transferência de aprendizagem *versus* soluções analíticas nos tempos  $t = 0.0, 0.1, 0.5, 0.9$  e  $1.$ . Linha contínua: solução analítica. Linha tracejada: PINN com esquema de Euler explícito. Linha pontilhada: PINN com esquema de Crank-Nicolson. Linha traço-pontilhada: PINN com esquema de Euler implícito.

## 4 Conclusões

Neste trabalho, propusemos a aplicação de PINNs com transferência de aprendizagem pelo esquema  $\theta$  para resolver a equação viscosa de Burgers com condições iniciais e de contorno de Dirichlet homogêneas. O método consiste na utilização de MLPs para estimar a solução em passos de tempo discretos. Como um processo iterativo, um primeiro modelo de rede neural é treinado para aprender a condição inicial. Então, o conhecimento é transferido para o próximo modelo, que aprende a solução no próximo passo de tempo, minimizando o resíduo do esquema  $\theta$  da equação de Burgers. O resultado é uma sequência de modelos de redes neurais que estimam as soluções do problema em passos de tempo discretos.

Em comparação com os modelos PINN usuais, a abordagem proposta tem a potencialidade de exigir arquiteturas de redes neurais menores com resultados precisos semelhantes e, por consequência, diminuir os custos computacionais. A comparação entre os valores  $\theta = 0.0, 0.5$  e  $1.$ , mostraram que todos os esquemas possuem uma acurácia similar, enquanto que o esquema de Euler explícito demanda um passo de tempo adequado para que a transferência de aprendizagem ocorra de forma efetiva.

## Referências Bibliográficas

- Basdevant, C., Deville, M., Haldenwang, P., Lacroix, J.M., Quazzani, J., Peyret, R., Orlandi, P., Patera, A.T., (1986). Spectral and finite difference solutions of the Burgers equation. *Computers & Fluids*, 14, 23-41.
- Biesek, V., Konzen, P.H.A., (2023) “Burgers’ PINNs with implicit Euler transfer learning”. Aceito em: *XXVI ENMC/XIV ECTM*, Nova Friburgo, Brasil.
- Bonkile, M.P., Aeasthi, A., Lakshmi, C., Mukundan, V., Aswin, V.S. (2018). A systematic literature review of Burgers’ equation with recent advances. *Pramana - Journal of Physics*, 90, 69.
- Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143, 06801.
- Goodfellow, I., Bangio, U., Courville, A. (2016), *Deep learning*, 1ed., MIT Press.
- Haykin, S. (2009), *Neural networks and learning machines*, 3ed., Pearson.
- Jin, X., Cai, S., Li, H., Karniadakis, G.E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426, 109951.
- Kingma, D.P., Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv*, arXiv:1412.6980.
- Konzen, P.H.A., Azevedo, F.S., Sauter, E., Zingano, P.R.A. (2017). Numerical simulations with the Galerkin least squares finite element method for the Burgers’ equation on the real line. *Trends in Computational and Applied Mathematics*, 18, 287-304.
- Mao, Z., Jagtap, D., Karniadakis, G.E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360, 112789.
- Raissi, M., Perdikaris, P., Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686-707.
- Raissi, M., Wang, Z., Triantafyllou, M., Karniadakis, G. (2019). Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 861, 119-137.
- Stevens, E., Antiga, L., Viehmann, T. (2020). *Deep learning with PyTorch*, 1ed., Manning Publications.
- Vadyala, S.R., Betgeri, S.N., Betgeri, N.P. (2022). Physics-informed neural network method for solving one-dimensional advection equation using PyTorch. *Array*, 13, 100110.
- Wood, W.L. (2026). An exact solution for Burger’s equation. *Communications in Numerical Methods in Engineering*, 22, 797-798.