PINNs BASED ON THE BURGERS EQUATION

Vitória Biesek[1,*], Anderson B. Modena[2], Pedro H.A. Konzen[3,*]

PPGMAp*, IME-UFRGS, RS, Brazil

vitoriabiesek@gmail.com[1], andersonmodena@hotmail.com[2], pedro.konzen@ufrgs.br[3]

Abstract

A Physics-informed neural network (PINN) is a deep learning framework for solving partial differential equations (PDEs). In this work, we explore its application to the solution of the viscous Burgers equation. Two approaches are tested, one with finite differences and other with automatic differentiation. Numerical results are presented for a n-wave solution.

Keywords: physics-informed neural network; viscous Burgers equation; finite difference; automatic differentiation.

Introduction

A Physics-informed neural network (PINN, [3]) is a deep learning framework for solving partial differential equations (PDEs). Deep learning is a field of machine learning by multiple levels of composition [1]. In this work, we discuss on the application of PINNs to solve the viscous Burgers equation with Dirichlet boundary conditions

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2},\ x \in (-1,1), t > 0, \quad (1a)$$

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

$$u(t,-1) = u(t,1) = 0, t > 0. \quad (1c)$$

Burgers equation is a benchmark problem to test new numerical approaches for solving convective-diffusive PDEs. Since it has been introduced, it has been applied to the understanding of turbulent fluids, shock flows, wave propagation in combustion chambers, vehicular traffic movement, acoustic transmission and many other applications. For small values of ν, the convection term dominates and standard numerical discretization schemes are numerically unstable.

PINNs

We assume an Artificial Neural Network (ANN) of the type Multi-layer Perceptron (MLP, [2]). It has $(x,t) \in \overline{\mathcal{D}}$ as inputs and the estimate $\tilde{u} \approx u(x,t)$ as the output. It is denoted as

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

where $\left(W^{(l)},\boldsymbol{b}^{(l)},\boldsymbol{f}^{(l)}\right)$ is the triple of weights $W^{(l)}$, bias $\boldsymbol{b}^{(l)}$ and activation function $\boldsymbol{f}^{(l)}$ in the l-th layer of the network, $l = 1,2,\dots,n_l$.

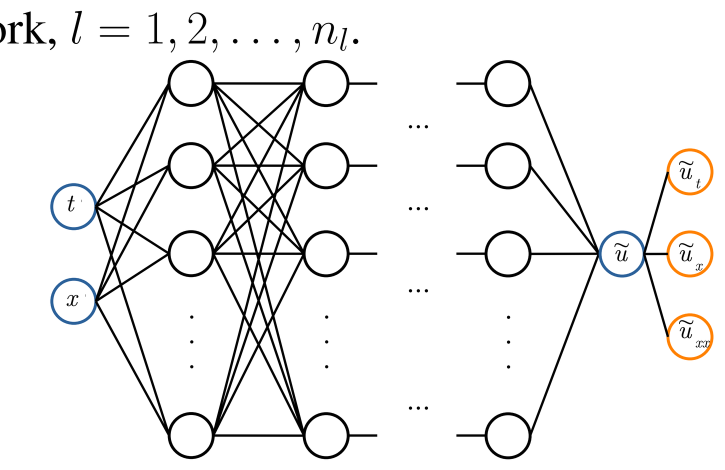

Figura 1: MLP scheme.

Following the PINNs approach, the training of the MLP is performed by solving the following minimization problem

$$\min_{\left\{\left(W^{(l)},\boldsymbol{b}^{(l)},\boldsymbol{f}^{(l)}\right)\right\}_{l=1}^{n_l}} \left\{\frac{1}{N_r}\sum_{i=1}^{N_r} r^2(x_{r,i},t_{r,i}) + \frac{p}{N_b}\sum_{i=1}^{N_b}\left[\tilde{u}(x_{b,i},t_{b,i}) - u(x_{b,i},t_{b,i})\right]^2\right\}, \quad (3)$$

where $\{(x_{r,i},t_{r,i})\}_{i=1}^{N_r}$ are N_r selected collocation points $x,t \in (-1,1)\times(0,T]$ and $\{(x_{b,i},t_{b,i})\}_{i=1}^{N_b}$ are N_b selected initial and boundary points. The PDE residual is given by

$$r(x,t) := \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in (-1,1), t > 0. \quad (4)$$

- FD Approach

$$\tilde{u}_t(t_s,x_s) = \frac{u(t_s,x_s) - u(t_s - h_t, x_s)}{h_t} \quad (5a)$$

$$\tilde{u}_x(t_s,x_s) = \frac{u(t_s,x_s+h_x) - u(t_s,x_s-h_x)}{2h_x} \quad (5b)$$

$$\tilde{u}_{xx}(t_s,x_s) = \frac{u(t_s,x_s-h_x) - 2u(t_s,x_s) + u(t_s,x_s+h_x)}{h_x^2} \quad (5c)$$

- AD Approach

$$\tilde{u}_t(t_s,x_s) = \mathcal{N}_t(t_s,x_s),\ \tilde{u}_x(t_s,x_s) = \mathcal{N}_x(t_s,x_s),\ \tilde{u}_{xx}(t_s,x_s) = \mathcal{N}_{xx}(t_s,x_s) \quad (6)$$

Results

- Test case $\nu = 0.01/\pi$
  - Initial condition

$$u_0(x) = -\sin(\pi x),\ -1 \le x \le 1 \quad (7)$$

  - Analytical solution

$$u(x) = \frac{-\int_{-\infty}^{\infty}\sin(\pi(x-\eta))f(x-\eta)e^{-\frac{\eta^2}{4\nu t}}\,d\eta}{\int_{-\infty}^{\infty} f(x-\eta)e^{-\frac{\eta^2}{4\nu t}}\,d\eta}, \quad (8)$$

  where $f(y) = e^{-\cos(\pi y)/(2\pi\nu)}$.
- MLP architecture, Adam optimizer

$$2 - 50\times 8 - 1,\ f^{(l)}(z) = \tanh(z),\ f^{(n_l)}(z) = z \quad (9)$$

- Uniform mesh

$$n_t = 50,\ n_x = 100 \quad (10)$$

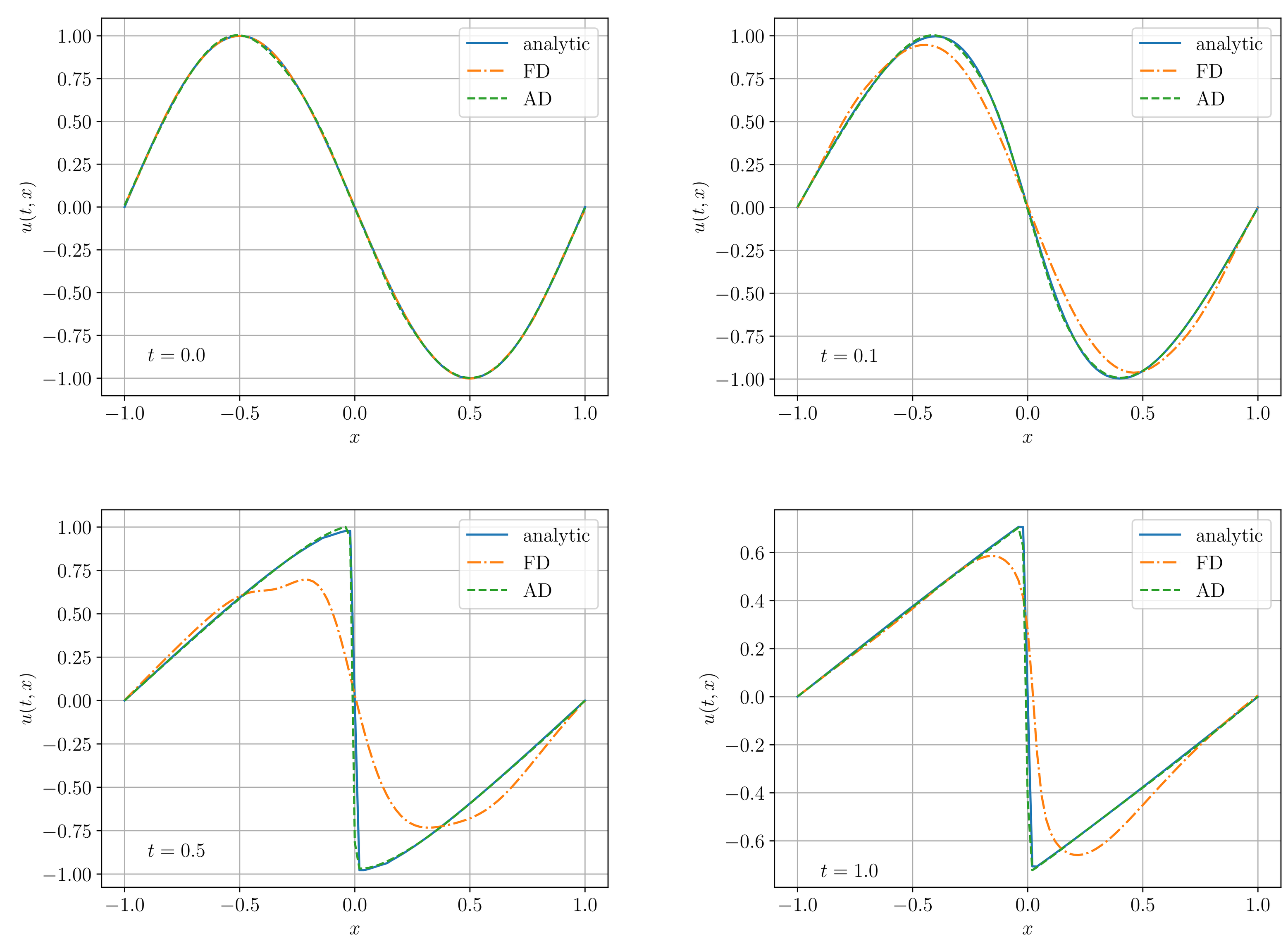

Figura 2: PINNs *versus* analytic solutions.

Final Considerations

We compared two approaches on the application of PINNs to solve the viscous Burgers equation. The FD approach consisted in estimating the residual of the Burgers equation with finite differences. The AD approach consisted in computing the residual by automatic differentiation on the ANN. Preliminary results indicate a much better performance of this second approach.

Referências

[1] Goodfellow, I.; Bengio, Y.; Courville, A. (2016), "*Deep Learning*", 1st ed., MIT Press, London.

[2] Haykin, S. (2008), "*Neural Networks and Learning Machines*", 3rd ed., Pearson, New Jersey.

[3] Raissi, M. et al. (2019), Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys., 378, 686-707.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.