ANN-MoC Approach for Solving First-Order Partial Differential Equations

Augusto Tchantchalam[1,*], Pedro C. dos Santos[2], Pedro H.A. Konzen[3,*]

PPGMAp*, IME-UFRGS, RS, Brazil

augustotchanchalam11@gmail.com[1], pedro.costa4137@gmail.com[2], pedro.konzen@ufrgs.br[3]

Abstract

In this work, we propose the ANN-MoC method for solving linear first-order partial differential equations. It consists of coupling an Artificial Neural Network (ANN) into the Method of Characteristics (MoC) for solving the equations. Alternatively to the usual interpolation approach, the ANN is applied to estimate the solution values on cell edges. Numerical experiments are shown to discuss the advantages and disadvantages of the proposed approach.

Keywords: artificial neural networks; linear first-order partial differential equations; method of characteristics.

Introduction

Linear first-order partial differential equations appears in the modeling of many important physical phenomena, such as heat radiative transfer [3] and neutron transport [4]. They have applications in high temperature manufacturing (e.g., class and ceramic manufactures), optical medicine, nuclear energy generation, and many others. In this work, we deal with equations of the following form

$$\Omega \cdot \nabla u + \sigma_t u = f \quad \text{in } \mathcal{D}, \tag{1}$$

where $u = u(\boldsymbol{x}) \in \mathbb{R}$, $\boldsymbol{x} = (x, y) \in \mathcal{D} = [a, b] \times [c, d]$, $\sigma_t > 0$ and a given direction $\Omega = (\mu, \eta)$ in the unitary disc centered at the origin. Incoming boundary condition is assumed as

$$u = u_{\text{in}} \quad \text{on } \Gamma^-, \tag{2}$$

where $u_{\text{in}} = u_{\text{in}}(\boldsymbol{x})$ is given on $\Gamma^- = \{\boldsymbol{x} \in \partial\mathcal{D} : \Omega \cdot \boldsymbol{n} < 0\}$, with $\boldsymbol{n}$ denoting the outward-pointing normal vector on the boundary.

The Method of Characteristics (MoC, [1]) is one of the approaches most used to solve (1)-(2). And, in this work, we propose the ANN-MoC, an application of Artificial Neural Networks (ANNs, [2]) to assist the MoC, by providing the needed estimates of the solution on the edge of mesh cells. The idea is to explore the ANNs as universal function approximators and gain advantage by a learning transfer strategy in the Ω direction.

ANN-MoC Approach

Lets assume a given uniform rectangular mesh $\mathcal{M}$ build from the Cartesian product of the partitions $\mathbb{P}_{h_x} = \{x_i = a + (i-1)h_x\}_{i=1}^{n_x}$, $h_x = (b-a)/(n_x - 1)$, and $\mathbb{P}_{h_y} = \{y_j = c + h_y(j-1)\}_{j=1}^{n_y}$, $h_y = (d-c)/(n_y-1)$, i.e. $\mathcal{M} = \mathbb{P}_{h_x} \times \mathbb{P}_{h_y}$. Without loss of generality, lets assume Ω is in the first quarter of the unitary disc, i.e. $\mu > 0$ and $\eta > 0$. The MoC solution of (1) has the form

$$u(\boldsymbol{x}_{ij}) = \tilde{u}(\overline{\boldsymbol{x}}_{ij})e^{-\int_0^s \sigma_t\, ds'} + \int_0^s f(s')e^{-\int_{s'}^s \sigma_t\, ds''}\, ds', \tag{3}$$

where $\boldsymbol{x}_{ij} = (x_i, y_j)$ and $\overline{\boldsymbol{x}}_{ij} = \boldsymbol{x}_{ij} - s\Omega$ is a point on one of the cell edges that has $\boldsymbol{x}_{ij}$ as a vertex.

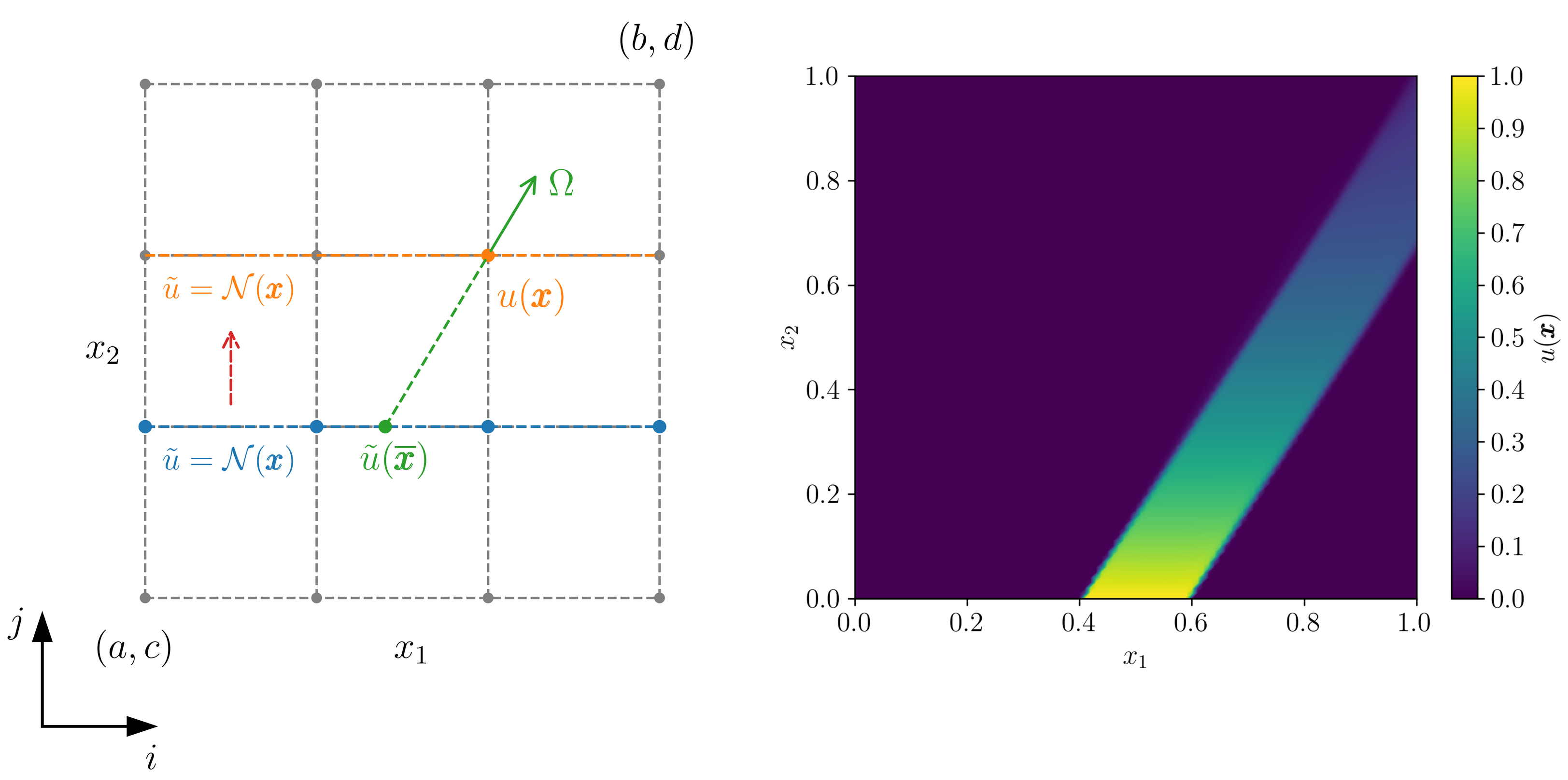

Figura 1: Left: ANN-MoC scheme. Right: Test case result.

In order to estimate the solution on cell edges, we apply an ANN of the type Multi-layer Perceptron (MLP)

$$\tilde{u}(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}; \left\{\left(W^{(l)}, \boldsymbol{b}^{(l)}, \boldsymbol{f}^{(l)}\right)\right\}_{l=1}^{n_l}\right), \tag{4}$$

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

By starting from the incoming boundary Γ^-, we train the neural network by solving the following minimization problem

$$\min_{\left\{\left(W^{(l)}, \boldsymbol{b}^{(l)}\right)\right\}_{l=1}^{n_l}} \frac{1}{n_s}\sum_{m=1}^{n_s} \left(\tilde{u}(\boldsymbol{x}_s) - u(\boldsymbol{x}_s)\right)^2, \tag{5}$$

where $\boldsymbol{x}_s$ are mesh nodes on Γ^-. The trained MLP is then applied to the computation of new mesh nodes with cell edges on Γ^-. Once the solution is known on all such new nodes, the MLP is retrained with warm initialization, i.e. we use transfer learning from the previous training to speed up computations.

Results

- Test case

$$\mathcal{D} = [0, 1]^2,\ \Omega = (0.3, 0.5),\ \sigma_t = 1.0,\ f \equiv 0$$
$$u(x_1, x_2) = u_{in}(x_1 - s\Omega_1, 0)e^{-\sigma_t s},\ s = \frac{x_2}{\Omega_2}$$

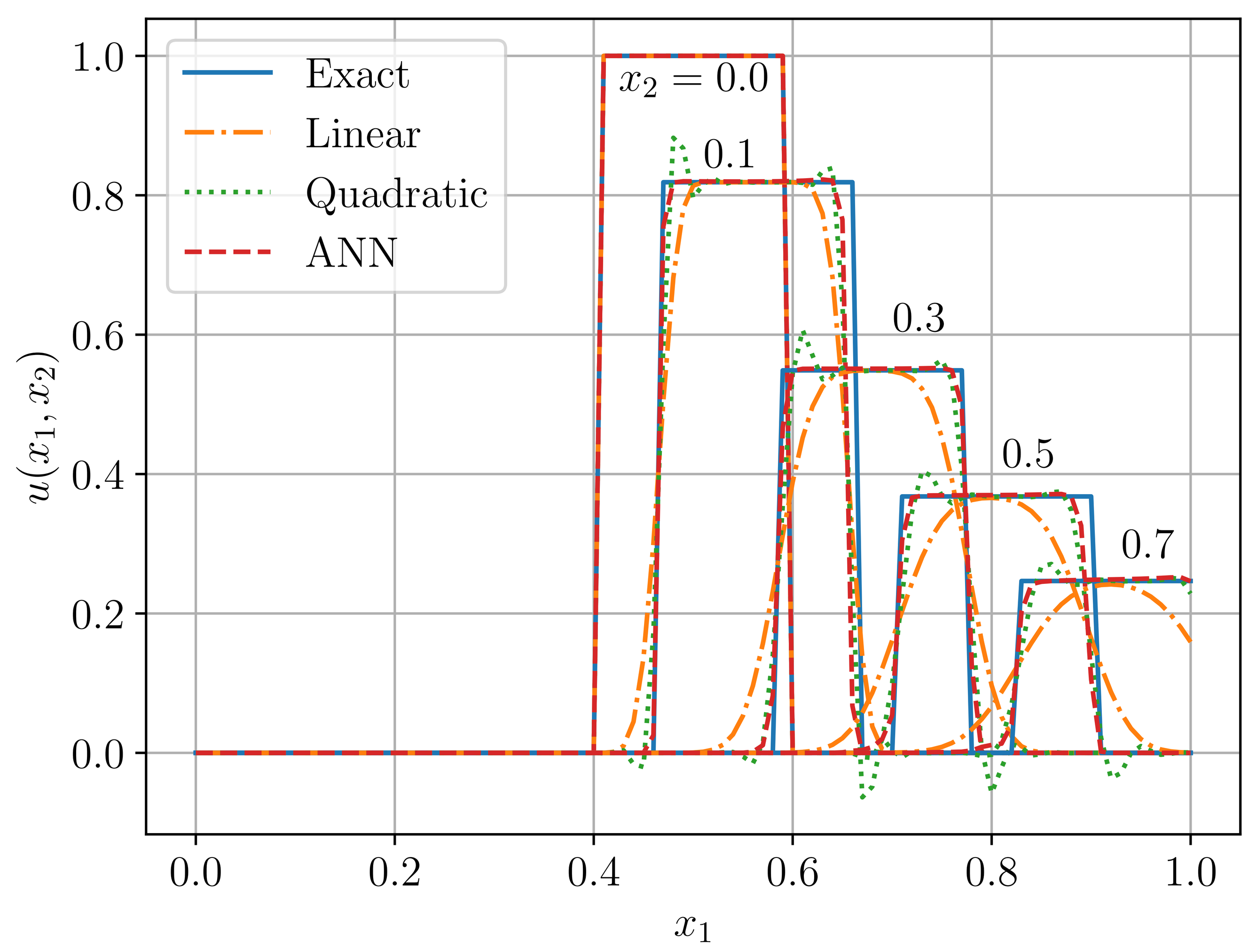

Figura 2: ANN-MoC solutions versus linear-MoC and quadratic-MoC solutions. MLP: $1 - 100 \times 2 - 1$ with sigmoid activation function.

Final Considerations

We presented the ANN-MoC method to the numerical solution of linear first-order partial differential equations. As an alternative to interpolation, it applies an ANN to estimates the solution on the edge of mesh cells. The numerical test case shows that the proposed method is less diffusive in comparison to the linear interpolation approach. In contrast to quadratic interpolation, it conserves the positiveness and has no observable spurious oscillations.

Referências

[1] Evans, L.C. (1998), “*Partial Differential Equations*”, AMS.

[2] Goodfellow, I.; Bengio, Y.; Courville, A. (2016), “*Deep Learning*”, MIT Press.

[3] Modest, M.F. (2013), “*Radiative Heat Transfer*”, Elsevier.

[4] Stacey, W.M. (2018), “*Nuclear Reactor Physics*”, Wiley.

Acknowledgements

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