Proceeding Series of the Brazilian Society of Computational and Applied Mathematics
Preprint

ANN-Flux Method for Solving Nonlinear Scalar Conservation Laws: Primal Aspects

Marcelo M. Mello[1], Arthur M. Espírito Santo[2], Pedro H. A. Konzen[3]
PPGMAp/IME/UFRGS, Porto Alegre, RS

Resumo. Artificial neural networks (ANNs) have been successfully applied to solve partial differential equations (PDEs). Conservation laws are PDEs that arise in a variety of fields such as fluid dynamics, traffic modeling, and fluid flow through porous media. We've recently proposed the original ANN-Flux method to solve the Riemann problem for nonlinear scalar conservation laws. The ANN-Flux method is designed for complex fluxes, which are costly to evaluate, differentiate, or invert. Based on the entropy solution form, the method applies two ANNs to approximate the flux F and its derivative inverse $G = [F']^{-1}$ for the shock and the rarefaction cases. The F is approximated by a multilayer perceptron (MLP) $\mathcal{N}_{\mathcal{F}}$. Automatic differentiation is then applied to approximate the derivative $F' \approx \mathcal{N}'_F$ and train a new MLP $\mathcal{N}_G \approx G$. Once the ANNs are trained, they can be applied to solve any Riemann problem in the flux convexity region. In this work we review the primal aspects of the generalization of the method to non-Riemann initial conditions by combining the ANN-Flux with the Godunov Method. The results show that the ANN-Flux produced precise results when compared with analytical (when available) or numerical alternatives.

Keywords: Deep Learning, Multilayer Perceptron, Conservation Law, Riemann Problem

1 Introduction

This work presents a summary of my master's thesis [7], bringing together the main aspects of the methodology and the highlighted results. Artificial neural networks (ANNs, [3]) have been successfully applied to solve partial differential equations, mainly after the emergence of physics-informed neural networks (PINNs, [9]). Applications to nonlinear conservation laws are also notable, including PINNs for high-speed flows ([6]), conservative PINNs (cPINNs, [4]), and weak PINNs (wPINNs, [10]). We address a hybrid deep learning approach for the solution of a nonlinear scalar conservation law

$$u_t + (F(u))_x = 0,\ x, t \in \mathbb{R} \times (0, t_f], \tag{1}$$

$$u(x, 0) = u_0, \tag{2}$$

with given flux function $F : \mathbb{R} \to \mathbb{R}$.

It is well known that the entropy solution of problem (1) with Riemann initial data can be a composition of shock and rarefaction waves, depending on the convexity of the flux function and on the left and right states. The shock wave (left) and rarefaction wave (right) can be written as

$$u(x,t) = \begin{cases} u_L, & x < t\sigma, \\ u_R, & x > t\sigma, \end{cases} \qquad u(x,t) = \begin{cases} u_L, & x < tF'(u_L), \\ G(x/t), & tF'(u_L) < x < tF'(u_R), \\ u_R, & x > tF'(u_R), \end{cases} \tag{3}$$

[1]marcelomokwademello@gmail.com
[2]arthurmes@ufrgs.br
[3]pedro.konzen@ufrgs.br

where $G(u) = [F']^{-1}(u)$, and σ is the shock speed satisfying the Rankine-Hugoniot condition $\sigma = \frac{F(u_L)-F(u_R)}{u_L-u_R}$.

The lack of convexity in the flux function implies that different regions of the flux may generate waves of distinct types, and the identification of admissible transitions between shocks and rarefactions is not straightforward. In such cases, the entropy condition is enforced through the construction of the convex (or concave) hull of the flux function, which determines the physically relevant wave structure.

2 ANN-Flux Method and the Godunov Scheme

The ANN-Flux method is designed for complicated fluxes, which are costly to evaluate, differentiate, or invert. Given that multilayer perceptrons (MLPs, [3]) neural networks are universal approximators, to build the Riemann problem solution the ANN-Flux estimates the flux F by a MLP $\tilde{y} = \mathcal{N}_F(u)$ classically trained to minimize the mean squared error (MSE) loss $\varepsilon = \varepsilon\left(\tilde{y}^{(s)}, y^{(s)}\right)$ with a generated data set $\left\{\left(u^{(s)}, y^{(s)} = F\left(u^{(s)}\right)\right)\right\}_{s=1}^{n_{s,F}}$, where $n_{s,F}$ is a given number of samples. For the simple shock wave case, the solution is then approximated by substituting F by $\mathcal{N}_F$ in (3). But, for a rarefaction case, a second MLP neural network $\mathcal{N}_G$ learns to approximate $[\mathcal{N}_F']^{-1} \approx [F']^{-1}$. The evaluation of $\mathcal{N}_F'$ can be efficiently computed by automatic differentiation (AD, [1]). The $\mathcal{N}_G$ is trained using randomly uniformly generated data set $\left\{\left(\delta\tilde{y}^{(s)}, u^{(s)}\right)\right\}_{s=1}^{n_{s,G}}$, where $n_{s,G}$ is a given number of samples. The data $\left\{u_L \leq u^{(s)} \leq u_m^{(1)}\right\}_{s=1}^{n_{s,G}}$, $\left\{u_m^{(1)} \leq u^{(s)} \leq u_m^{(2)}\right\}_{s=1}^{n_{s,G}}$, $\ldots$, $\left\{u_m^{(k)} \leq u^{(s)} \leq u_R\right\}_{s=1}^{n_{s,G}}$ is randomly generated, passed forward through $\mathcal{N}_F$ to give $\tilde{y}^{(s)} = \mathcal{N}_F\left(u^{(s)}\right)$, and then backpropagated to compute $\delta\tilde{y}^{(s)} = \mathcal{N}_F'\left(u^{(s)}\right)$. The set of inflection points $\left\{u_m^{(1)}, u_m^{(2)}, \ldots, u_m^{(k)}\right\}$ is computed using Newton's method. It is necessary to create $k+1$ $\mathcal{N}_G$-type networks so that all branches of the inverse of $\mathcal{N}_F'$ are represented. The training of the j-th neural network parameters $\tilde{u}^{(s)} = \mathcal{N}_G^{(j)}\left(\delta\tilde{y}^{(s)}\right)$ is obtained by minimizing the MSE loss $\varepsilon\left(\tilde{u}^{(s)}, u^{(s)}\right)$ on the neural network weights and biases. The solution of the rarefaction case where $u \in [u_m^{(j)}, u_m^{(j+1)}]$, is then given by substituting F' by $\mathcal{N}_F'$ and G by $\mathcal{N}_G^{(j)}$ in (3). In summary, the ANN-Flux solution for the Riemann problem of the conservation law is given by (left: shock; right: rarefaction)

$$u(x,t) = \begin{cases} u_L & ,x < t\sigma, \\ u_R & ,x > t\sigma, \end{cases} \qquad u(x,t) = \begin{cases} u_L & ,x < t\mathcal{N}_F'(u_L), \\ \mathcal{N}_G^{(j)}(x/t) & ,t\mathcal{N}_F'(u_L) < x < t\mathcal{N}_F'(u_R), \\ u_R & ,x > t\mathcal{N}_F'(u_R), \end{cases} \tag{4}$$

where $\sigma = \frac{\mathcal{N}_F(u_L)-\mathcal{N}_F(u_R)}{u_L-u_R}$ is Rankine-Hugoniot shock speed.

Unlike PINN-based approaches, where the neural network directly approximates the differential equation solution, the ANN-Flux methodology estimates only the flux function and the inverse of its derivative. This creates additional challenges for incorporating real data into training, since the flux represents only part of the governing equation. Now that we are able to solve the Riemann problem using the ANN-Flux method, we can employ the Godunov method to generalize the approach to arbitrary initial conditions, that is, not necessarily of the Riemann type.

To do this, we use the structure of the reconstruct-evolve-average (REA) scheme, in which any initial function is reconstructed as a piecewise constant function. In this process, each interface between two constant states defines a local Riemann problem. The solution to these Riemann problems is then obtained using the ANN-Flux method, allowing for local evolution of the solution. Finally, by applying the averaging step to each cell, we obtain an approximation of the solution for the general initial condition. For an extensive description of the methodology, see [7].

3 Results

For all the following results the neural networks have been trained with the Adam optimizer, the learning rate $l_r = 10^{-3}$ and the tolerance of $\tau = 5 \cdot 10^{-7}$ for the loss function, with hyperbolic tangent and the identity as activation functions at hidden and output layers, respectively.

Since the inverse of the flux derivative $G(u)$ is used to determine the solution in the rarefaction case, it is necessary that $F'(u)$ be monotonic to guarantee the existence of its inverse. Therefore, the neural network must be trained within the monotonic regions of $F'(u)$, so that each individual $\mathcal{N}_G$ behaves as a function.

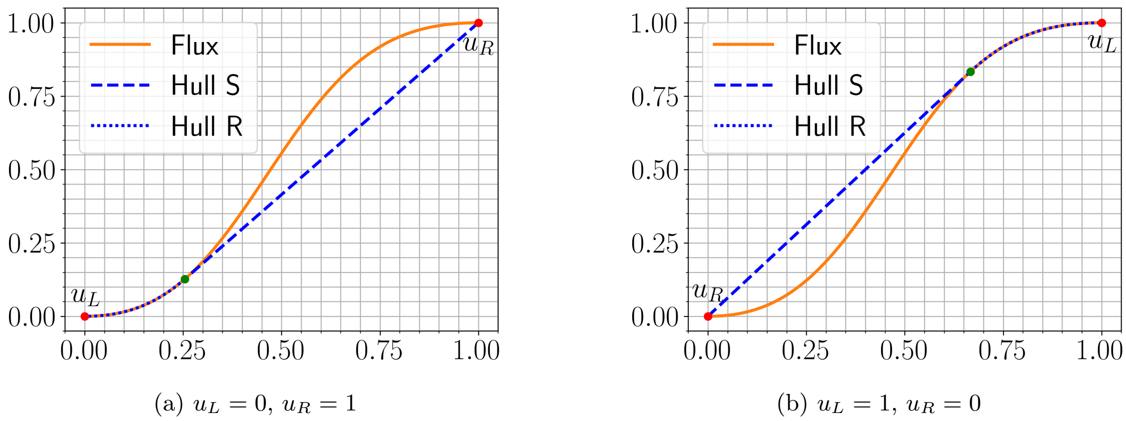


Figure 1: Construction of the upper (left) and lower (right) convex-hull for the Buckley-Leverett flux function where Hull S represents a shock wave, Hull R a rarefaction wave and the green dots are the points of interest. Source: [7].

Choosing the appropriate architecture and sample size for an MLP is done through an exploratory analysis. In this procedure, several architectures are tested, varying the number of hidden layers (n_h) and neurons (n_n), the network is trained and it is evaluated whether the stopping criterion, based on error or tolerance, is met, as well as the number of epochs required to reach this limit. At the end of this evaluation stage, the most efficient architecture is selected, namely the one that satisfies the stopping condition with the fewest epochs. Subsequently, different sample sizes are tested, and the fastest is chosen.

In the master's thesis [7], one of the benchmark tests chosen was the Inviscid Burgers Equation, whose flux is given by $f(u) = u^2/2$. However, since this flux function is convex, the solution to the Riemann problem becomes relatively simple, involving only pure shock or rarefaction waves. For this reason, the results presented in this work focus on equations whose flux function is non-convex, which makes determining the solution to the Riemann problem more complicated and, consequently, reinforces the relevance and flexibility of the proposed method. The results obtained with the Godunov method applied to Inviscid Burgers Equation can be found in the article [8].

3.1 Buckley-Leverett Equation

Now the results for the Buckley-Leverett equation, a model for two-phase immiscible and incompressible fluid flow through porous media [5]. It is given by the conservation law (1) with the flux

$$F(u) = \frac{u^2}{u^2 + \alpha(1-u)^2}, \tag{5}$$

where $u \in [0,1]$ is the saturation of one phase and x is the position along a one-dimensional porous medium. The constant α represents the ratio of the mobilities between the fluids, chosen here as 0.8. The flux function F models the fractional flow of one of the phases.

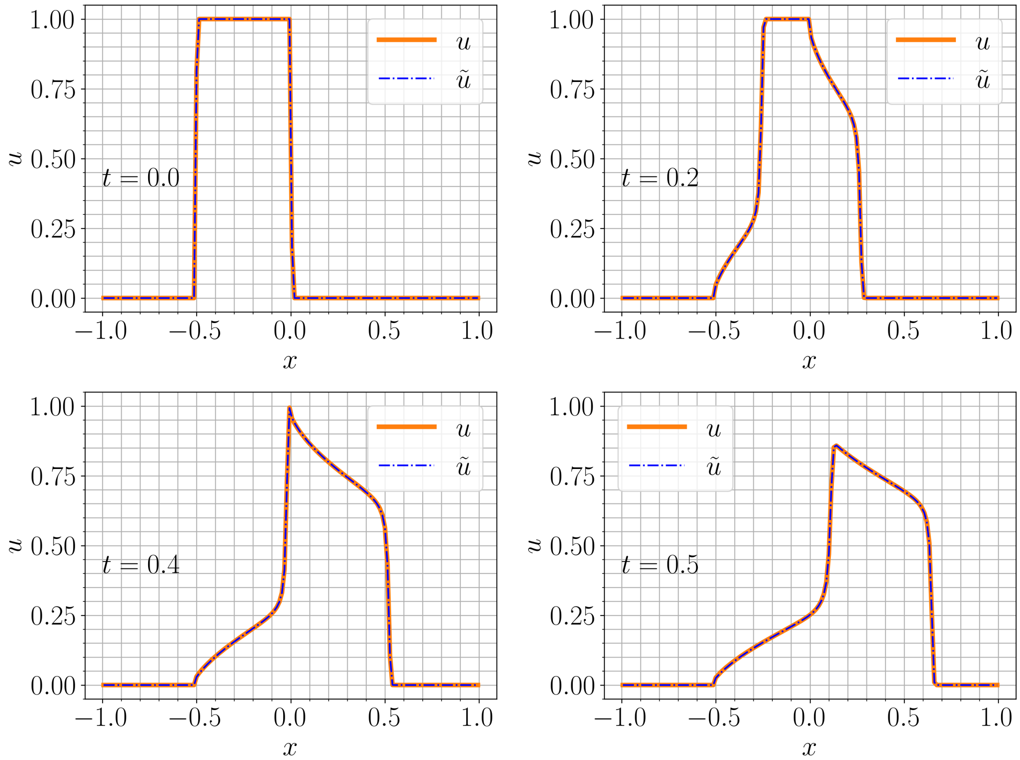


Figure 2: ANN-Flux ($\tilde{u}$) *versus* numerical solutions (u) of Buckley-Leverett equation, for different times. Source: [8].

The Buckley-Leverett flux exhibits a single change of concavity, which requires considering two distinct types of hulls, depending on the initial conditions. When $u_L < u_R$, an upper convex hull is constructed, as the solution involves a rarefaction followed by a shock, seen in Figure 1a. On the other hand, when $u_R < u_L$, the appropriate envelope is the lower convex hull, corresponding to the case where a shock followed by a rarefaction occurs, seen in Figure 1b.

Performing the same exploratory strategy applied in the Burgers case, we choose the architecture of $\mathcal{N}_F$ as $1 - 20 \times 3 - 1$ with $n_s = 200$, $\mathcal{N}_G^1$ as $1 - 40 \times 4 - 1$ with $n_s = 100$, and $\mathcal{N}_G^2$ as $1 - 50 \times 4 - 1$ with $n_s = 50$.

For the Godunov method, the three-state initial condition was used

$$u(x,0) = \begin{cases} 1, & -0.5 < x < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

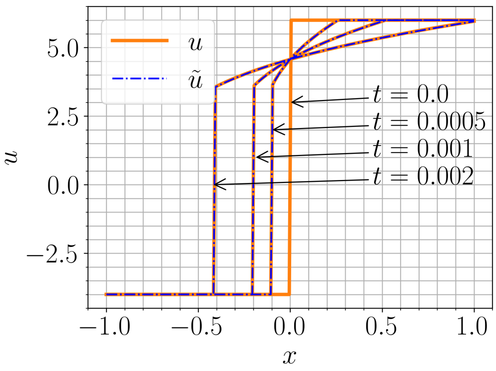
The reference numerical solution was computed using the flux function $F(u)$ and $F'(u)$ analytically, while the function G was obtained through Newton's method. The comparison between the final solution of the ANN-Flux method and the reference solution has a relative L_2-error of 10^{-5}. This setting is composed of two adjacent Riemann problems with wave structures that behave as the Riemann solution until interaction after the time $t = 0.4$. The solution in Figure 2 can be interpreted as a fluid propagating from the left to right, pushing another fluid of different density, such as water and oil.

3.2 Fourth-Order Polynomial Flux

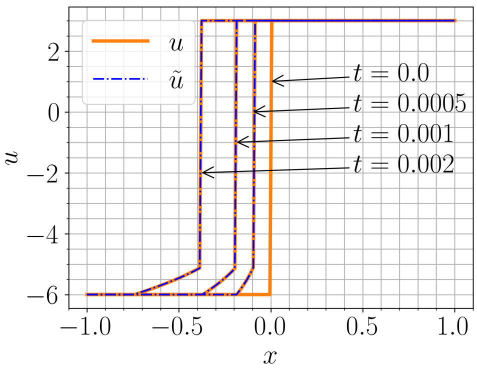
The next study case is inspired by the analysis presented in [2], which details the process of constructing the Riemann solution for a nonlinear conservation law with nonconvex flux, a fourth-order polynomial flux given by

$$F(u) = u^4 + au^3 + bu^2 + cu, \tag{7}$$

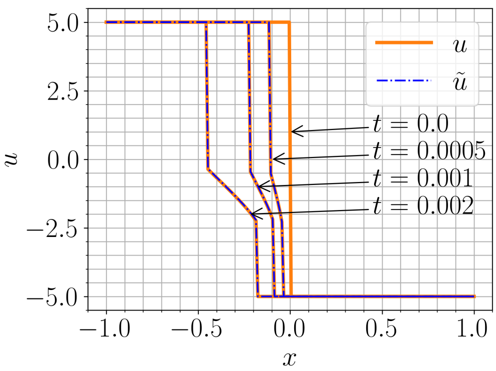
with coefficients $a = 3$, $b = -35$, and $c = -250$, defined in the interval $u \in [-7, 7]$.



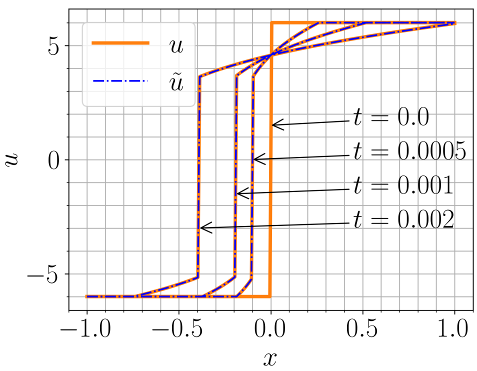
(a) Shock-Rarefaction: $u_L = -4$, $u_R = 6$



(b) Rarefaction-Shock: $u_L = -6$, $u_R = 3$



(c) Shock-Rarefaction-Shock: $u_L = 5$, $u_R = -5$



(d) Rarefaction-Shock-Rarefaction: $u_L = -6$, $u_R = 6$

Figure 3: ANN-Flux ($\tilde{u}$) *versus* numerical solutions (u) of the Polynomial Flux Equation, for different times. Source: [7].

The methodology developed in [2] was then expanded and applied to solve the isothermal Van der Waals gas equations, which are based on a modification of the ideal-gas model that takes into account particle volume and attractive forces between particles.

The flux $F(u)$ exhibits two changes of concavity, thus, three distinct neural networks $\mathcal{N}_G$ are needed to represent the inverse of the derivative of the flux. The chosen architecture for $\mathcal{N}_F$ was $1-80\times 5-1$ with $n_s=200$ and the architecture for $\mathcal{N}_G^1$, $\mathcal{N}_G^2$, and $\mathcal{N}_G^3$, are $1-30\times 5-1$ with $n_s=100$, $1-30\times 5-1$ with $n_s=50$ and $1-40\times 4-1$ with $n_s=50$ respectively.

Given the nonconvexity of the flux, six different types of solutions arise for the Riemann problem: four composite solutions and two additional solutions corresponding to pure shock and pure rarefaction waves. The comparison between the solutions obtained by the ANN-Flux method and the numerical reference solutions is presented in Figure 3. For all six cases considered, good approximations are observed between the solutions, with the relative error between the ANN-Flux method and the reference solution remaining consistently on the order of 10^{-4}.

For the Godunov method purposes, the initial condition is a smoothing of the discontinuity present in the Riemann problem, given by

$$\Phi(x) = -5\tanh\left(\frac{x}{\delta}\right), \tag{8}$$

with $\delta = 0.05$, resulting in a smoother and more gradual transition, seen in Figure 4. The relative error between the ANN-Flux solution and the reference numerical solution remained consistently of order 10^{-3}.

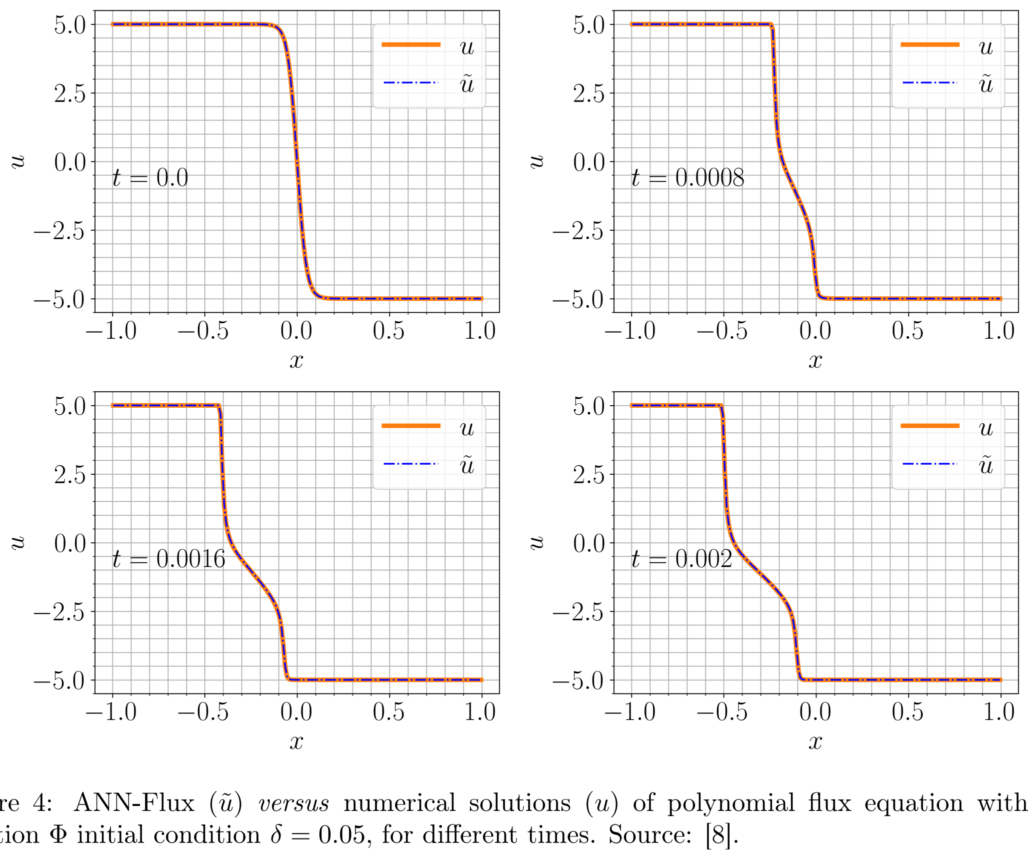


Figure 4: ANN-Flux ($\tilde{u}$) *versus* numerical solutions (u) of polynomial flux equation with the function Φ initial condition $\delta = 0.05$, for different times. Source: [8].

4 Concluding Remarks

The results demonstrate that the original ANN-Flux method is effective in approximating the flux function F and its derivative inverse $G = [F']^{-1}$ in the benchmark tests. The selected architectures were small, resulting in low inference cost. Training times remained short, with a maximum of 160 seconds for the polynomial flux equation.

This approach shows great promise for solving nonlinear conservation laws where approximating nonconvex fluxes may be computationally expensive or infeasible. Once the required ANNs are trained, the ANN-Flux method provides precise solution structures, including shocks, rarefactions, and combinations thereof.

Future directions include testing the ANN-Flux method on problems with more complicated fluxes, where direct evaluation is computationally expensive and operations like differentiation and inversion become unfeasible or extremely costly, extending the method to systems and higher dimensions of conservation laws, developing adaptive strategies for training the neural networks, and making a comparative study between the Riemann solvers used.

References

[1] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. "Automatic Differentiation in Machine Learning: a Survey". In: **Journal of Machine Learning Research** 18.153 (2018), pp. 1–43.

[2] M. Fossati and L. Quartapelle. **The Riemann Problem for Hyperbolic Equations under a Nonconvex Flux with Two Inflection Points**. 2014. arXiv: 1402.5906. URL: https://arxiv.org/abs/1402.5906.

[3] S. Haykin. **Neural Networks and Learning Machines**. 3a. ed. New York: Pearson, 2009. ISBN: 9780131471399.

[4] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis. "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems". In: **Computer Methods in Applied Mechanics and Engineering** 365 (2020), p. 113028. DOI: 10.1016/j.cma.2020.113028.

[5] R. J. LeVeque. **Numerical methods for conservation laws**. Vol. 214. Springer, 1992. DOI: 10.1007/978-3-0348-8629-1.

[6] Z. Mao, A. D. Jagtap, and E. G. Karniadakis. "Physics-informed neural networks for high-speed flows". In: **Computer Methods in Applied Mechanics and Engineering** 360 (2020), p. 112789. DOI: 10.1016/j.cma.2019.112789.

[7] M. M. Mello. **ANN-Flux Method: A Deep Learning Neural Network Approach for Solving Nonlinear Scalar Conservation Laws**. 2026.

[8] M. M. Mello, A. M. Espírito Santo, and P. H. A. Konzen. "A Finite Volume Application of ANN-Flux to Scalar Conservation Laws". In: **Ciência e Natura** (2026).

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: **Journal of Computational physics** 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[10] T. Ryck, S. Mishra, and R. Molinaro. **wPINNs: Weak Physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws**. 2022. DOI: 10.48550/arXiv.2207.08483. arXiv: 2207.08483 [math.NA].