A Finite Volume Application of ANN-Flux to Scalar Conservation Laws

Marcelo M. Mello[1], Arthur M. Espírito Santo[2], Pedro H. A. Konzen[3]

[1]IME PPGMAp, Universidade Federal do Rio Grande do Sul, marcelomokwademello@gmail.com
[2]IME PPGMAp, Universidade Federal do Rio Grande do Sul, arthurmes@ufrgs.br
[3]IME PPGMAp, Universidade Federal do Rio Grande do Sul, pedro.konzen@ufrgs.br

Abstract. Artificial neural networks (ANNs) have recently gained attention for solving nonlinear partial differential equations, particularly in data-driven approaches to conservation laws. In this work, we present a combined method that incorporates the ANN-Flux model into a Godunov-type finite volume scheme to solve nonlinear scalar conservation laws. The method replaces the classical Riemann solver with a neural surrogate composed of two multilayer perceptrons: one approximates the flux function, and the other learns the inverse of its derivative, using automatic differentiation. At each cell interface, the approximate solution of the Riemann problem is used to compute the numerical flux, preserving the structure of Godunov methods. We validate the approach on benchmark problems, including the inviscid Burgers and Buckley–Leverett equations. The ANN-Flux method is particularly advantageous in problems with intricate nonconvex fluxes, where traditional methods may become costly or unstable.

Keywords. Deep Learning; Multilayer Perceptron; Conservation Laws; Finite Volume Methods.

1. INTRODUCTION

Artificial neural networks [2] have been successfully applied to solve partial differential equations, mainly after the emergence of the physics-informed neural networks [7]. Applications to nonlinear conservation laws are also notable, including PINNs for high-speed flows [5], conservative PINNs [3], and weak PINNs [8]. In this work, we propose the new ANN-Flux method to solve the Riemann problem for the nonlinear scalar conservation law

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

$$u(x,0) = u_L,\ x \in \mathbb{R}_-,\ u(x,0) = u_R,\ x \in \mathbb{R}_+, \tag{2}$$

with given left and right states $u_L, u_R \in \mathbb{R}$, and flux function $F : \mathbb{R} \to \mathbb{R}$.

It is a well-known fact that the problem (1) with a nonconvex flux function has an entropic solution that is can be a composition of shock and rarefaction waves, depending on the left and right states. The shock wave solution can be written as

$$u(x,t) = \begin{cases} u_L & ,x < t\sigma, \\ u_R & ,x > t\sigma, \end{cases} \tag{3}$$

and the rarefaction wave solution can be written as:

$$u(x,t) = \begin{cases} u_L & ,x < tF'(u_L), \\ G(x/t) & ,tF'(u_L) < x < tF'(u_R), \\ u_R & ,x > tF'(u_R), \end{cases} \tag{4}$$

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

We present the most recent version of the ANN-Flux method capable of handling nonlinear conservation laws with nonconvex flux functions. The key idea is to represent the solution of the Riemann problem as a composition of elementary shock and rarefaction waves, determined by the convex (or concave) envelope that characterizes the corresponding entropy solution. This neural network-based formulation allows for accurate modeling of complicated wave structures without resorting to traditional analytic solvers. We conclude this section by describing a Godunov-type finite volume framework in which the ANN-Flux method provides an approximate Riemann solution used to compute the intercell numerical flux, preserving the Godunov framework and thus forming the proposed combined method.

2. METHODOLOGY

2.1. ANN-Flux Method

The ANN-Flux method is designed for intricate fluxes, which are costly to evaluate, differentiate, or invert. Given that multilayer perceptrons [2] neural networks are universal approximators, the ANN-F consists of approximating 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 group of MLP $\mathcal{N}_G^{(k)}$, where k is the number of concavity changes of the flux, learns to approximate $[\mathcal{N}_F']^{-1} \approx [F']^{-1}$. The evaluation of $\mathcal{N}_F'$ can be efficiently computed by automatic differentiation [1]. The $\mathcal{N}_G$ is trained using a directly 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 $\left\{u_L \leq u^{(s)} \leq u_m^1\right\}_{s=1}^{n_{s,G}}$,$\left\{u_m^1 \leq u^{(s)} \leq u_m^2\right\}_{s=1}^{n_{s,G}}$, ..., $\left\{u_m^k \leq u^{(s)} \leq u_R\right\}_{s=1}^{n_{s,G}}$ is randomly generated, forward through N_F to give $\tilde{y}^{(s)} = N_F\left(u^{(s)}\right)$, and then backward propagated to compute $\delta\tilde{y}^{(s)} = \mathcal{N}_F'\left(u^{(s)}\right)$. The change in concavity $\left\{u_m^1, u_m^2, \ldots, u_m^k\right\}$ ensure that each $\mathcal{N}_G^{(k)}$ between them behaves as a function, allowing the inversion of the flux, or which would be impossible without this separation and are computed using Newton's method (finding the roots of the second derivative of the flux). The training of $k+1$ different neural networks $\tilde{u}^{(s)} = \mathcal{N}_G^{(k)}\left(\delta\tilde{y}^{(s)}\right)$ are obtained by minimizing the MSE loss $\varepsilon = \varepsilon\left(\tilde{u}^{(s)}, u^{(s)}\right)$ on the neural network weights and biases. The solution of the rarefaction case is then given by substituting F' by $\mathcal{N}_F'$ and G by $\mathcal{N}_G$ in (4). In summary, the

ANN-Flux solution of the conservation law is given by (left: shock; right:rarefaction)

$$u(x,t) = \begin{cases} u_L & , x < t\sigma, \\ u_R & , x > t\sigma, \end{cases} \qquad u(x,t) = \begin{cases} u_L & , x < t\mathcal{N}_F'(u_L), \\ \mathcal{N}_G(x/t) & , t\mathcal{N}_F'(u_L) < x < t\mathcal{N}_F'(u_R), \\ u_R & , x > t\mathcal{N}_F'(u_R), \end{cases} \tag{5}$$

where $\sigma = \frac{\mathcal{N}_F(u_L)-\mathcal{N}_F(u_R)}{u_L-u_R}$ is Rankine-Hugoniot shock speed. By combining the new shock and rarefaction waves (5) we can solve Riemann problems with nonconvex fluxes using the ANN-Flux method.

2.2. Numerical Framework

In order to numerically solve the conservation law (1) with potentially nonconvex fluxes, we employ a finite volume formulation based on a modified Godunov scheme, in which the exact Riemann problem solution – approximated via a neural network – is used at each cell interface. Let us consider the one-dimensional grid composed of cells $C_i = [x_{i-1/2}, x_{i+1/2}]$, with spatial step Δx. The discrete solution approximation of $u(x,t)$ at time t^n is denoted by $U_i^n \approx u(x_i, t^n)$. Time evolution is obtained by the standard update equation:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(\hat{F}_{i+1/2}^n - \hat{F}_{i-1/2}^n\right),$$

where $\hat{F}_{i+1/2}^n$ is the numerical flux at the interface $x_{i+1/2}$, obtained by solving a local Riemann problem between values U_i^n (to the left) and U_{i+1}^n (to the right). In this work, this resolution is performed using the ANN-Flux solution described in subsection 2.1.

Specifically, for each pair (U_i^n, U_{i+1}^n), we evaluate the solution of equation (5), selecting the appropriate profile (shock, rarefaction, or a combination of those in the case of nonconvex fluxes) based on the initial conditions. The flux is then estimated as $\hat{F}_{i+1/2}^n = F(u(0^+, x_{i+1/2}))$, where u is given by the ANN-Flux method.

To guarantee stability, we impose a CFL condition of the form:

$$\Delta t \leq \frac{\Delta x}{\max_i |\lambda_i|},$$

in which $\lambda_i \approx \max\left(|N_F'(U_i^n)|, |N_F'(U_{i+1}^n)|\right)$ represents the largest local characteristic speed, estimated from the derivative of the neural network that approximates the flux function. After this restricted time step, we project the solution back onto the original mesh, ensuring that it remains within the discrete finite volume space. This projection step is necessary to preserve consistency with the numerical mesh structure [4].

Finally, we note that the proposed numerical framework fits within the class of Godunov-type methods [9], as it retains the essential feature of solving Riemann problems between adjacent constant states to compute interface fluxes. The originality of this approach lies in replacing traditional analytical or numerical solvers with pre-trained neural networks. This allows greater flexibility and generalization potential, while maintaining the conceptual structure of the classical method.

3. RESULTS

Here numerical tests with the ANN-F method are applied to selected inicial condition functions. 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.

3.1. Inviscid Burgers Equation

We first present two sets of initial data for the benchmark problem of the inviscid Burgers equation, where $F(u) = u^2/2$. It is a well-established test case in the computational modeling of several physical phenomena such as fluid dynamics, gas dynamics, and others. The Burgers case is just a proof of concept for validating the method in question. To choose the ANN architecture, we performed numerical tests, in [6], varying the number of units (n_n) and layers (n_h), with $n_s = 200$ samples per epoch. From these, we concluded that an MLP of architecture $1 - 30 \times 2 - 1$ (one input, 30 units per hidden layer, and one output) is adequate to approximate the flux with a L_2-error of 2×10^{-4}.

Figure 1 shows the comparison between the construction of the solution using the Godunov's scheme with ANN-F versus the same scheme but with the analytical solutions of the inviscid Burgers equation. The initial condition is $u(x{,}0) = sin(\pi x)$ with domain $[-1{,}1]$. The $\mathcal{N}_G$ has been set as a simple perceptron with the identity as the activation function. We note that the ANN-F has obtained very good results, the comparison between the two solutions has a relative L_2-error of 10^{-6}

3.2. Buckley-Leverett Equation

Now the results for the Buckley-Leverett equation, a model for two-phase immiscible and incompressible fluid flow through porous media [4]. It is given by the conservation law (1) with the flux

$$F(u) = \frac{u^2}{u^2 + \alpha(1-u)^2}, \tag{6}$$

where $u \in [0, 1]$ is the saturation of one phase and x is the position along a one-dimensional porous medium. The flux function F models the fractional flow of one of the phases. The Buckley-Leverett flux has one change of concavity, so the training of two different $\mathcal{N}_G$

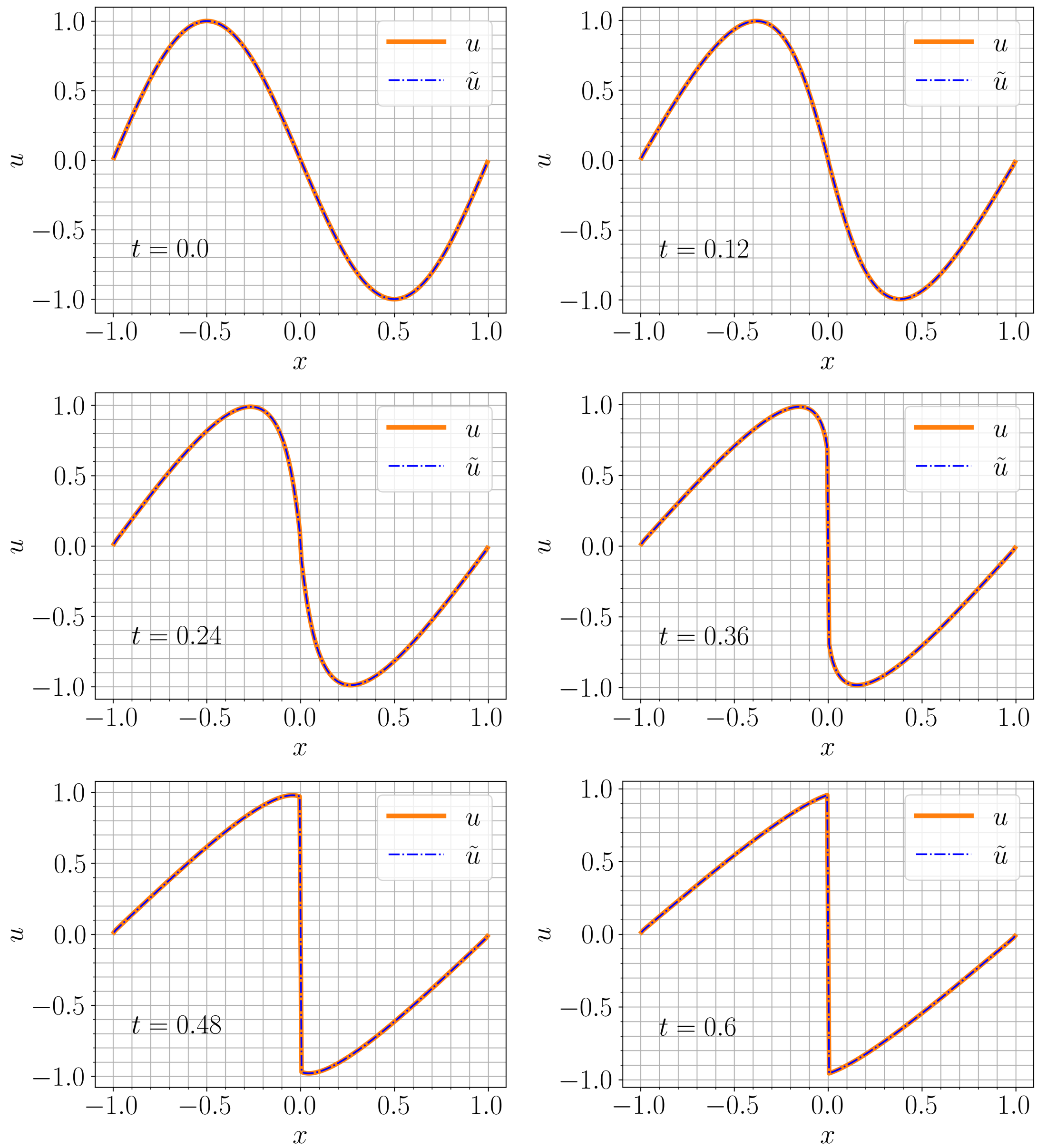

Figure 1. ANN-F ($\tilde{u}$) *versus* analytical solutions (u) of the inviscid Burgers equation. Source: From the authors.

is necessary. The numerical solution was calculated using the flux function $F(u)$ and $F'(u)$ analytically, while the function G was obtained through Newton's method. The three-state initial condition used in such problem was

$$u(x,0) = \begin{cases} 1, & -0.5 < x < 0, \\ 0, & otherwise. \end{cases} \tag{7}$$

composed of two adjacent Riemann problems whit wave structures that will interact after time $t = 0.375$. Performing the same numerical test as in the Burgers case we choose the architecture of $\mathcal{N}_G^1$ in Table 1 and $\mathcal{N}_G^2$ in Table 2, as $1 - 40 \times 4 - 1$ and $1 - 50 \times 4 - 1$ respectively .The comparison of the final solutions of the ANN-Flux method and the numerical one has a relative L_2-error of 10^{-5}.

Table 1. Choosing the architecture for the neural network $\mathcal{N}_G^1$ for the Buckley-Leverett. The results are n_e/ϵ, where n_e is the number of epochs and ϵ is the error using L_2 norm. Source: From the authors.

$n_n \backslash n_h$	1	2	3	4
20	50000/1.5E − 2	50000/4.9E − 3	50000/6.4E − 3	48424/3.8E − 3
30	50000/1.4E − 2	50000/4.5E − 3	45247/5.3E − 3	50000/4.7E − 3
40	50000/1.1E − 2	50000/4.3E − 3	50000/3.8E − 3	40640/3.5E − 3
50	50000/1.3E − 2	50000/4.4E − 3	44650/3.8E − 3	50000/4.7E − 3

Table 2. Choosing the architecture for the neural network $\mathcal{N}_G^2$ for the Buckley-Leverett. The results are n_e/ϵ, where n_e is the number of epochs and ϵ is the error using L_2 norm. Source: From the authors.

$n_n \backslash n_h$	1	2	3	4
20	50000/4.6E − 3	48872/1.0E − 3	48846/1.2E − 3	50000/1.2E − 3
30	47506/3.7E − 3	49388/1.3E − 3	39573/9.1E − 4	47506/7.8E − 4
40	50000/4.2E − 3	50000/1,8E − 3	48392/1.8E − 3	42929/1.9E − 3
50	50000/4.4E − 3	50000/1.7E − 3	46584/8.6E − 4	32039/9.9E − 4

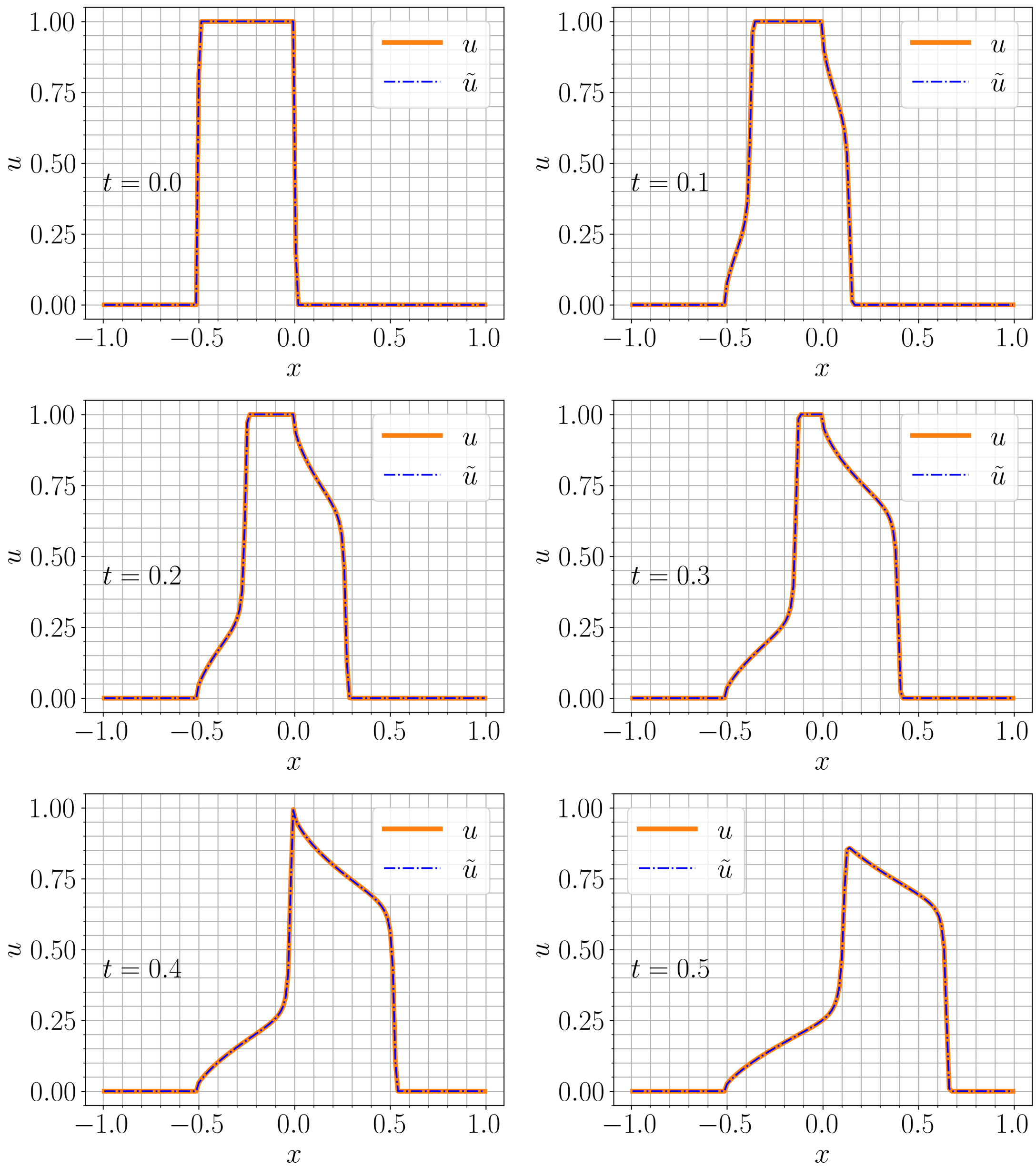

Figure 2. ANN-F ($\tilde{u}$) *versus* numerical solutions (u) of Buckley-Leverett equation. Source: From the authors.

4. CONCLUDING REMARKS

The results demonstrate that the ANN-Flux method is effective in approximating the flux function F and its derivative inverse $G = [F']^{-1}$ in benchmark cases such as the inviscid Burgers and Buckley–Leverett equations. This approach shows great promise for solving nonlinear conservation laws where approximating convex and nonconvex flux functions 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 extending the method to systems of conservation laws and developing adaptive strategies for training the neural networks on-the-fly during simulation.

ACKNOWLEDGMENT

M. Mello thanks financial support by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). A. Espírito Santo was supported by Rio Grande do Sul Research Support Foundation- FAPERGS, Brazil, grant number 23/2551-0000772-0.

REFERENCES

[1] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic Differentiation in Machine Learning: a Survey", *Journal of Machine Learning Research*, vol. 18, no. 153, pp. 1–43, 2018.

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

[3] 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", *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 113 028, 2020. DOI: 10.1016/j.cma.2020.113028.

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

[5] Z. Mao, A. D. Jagtap, and E. G. Karniadakis, "Physics-informed neural networks for high-speed flows", *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112 789, 2020. DOI: 10.1016/j.cma.2019.112789.

[6] M. M. Mello, A. M. Espirito Santo, and P. H. A. Konzen, "ANN-Flux Methods to Nonlinear Conservations Laws: Selected Shock and Rarefaction Cases", in *Anais da 10ª Conferência Sul em Modelagem Computacional (10º MCSul) e X Seminário e Workshop em Engenharia Oceânica (X SEMENGO)*, vol. 10, Rio Grande(RS) Universidade Federal do Rio Grande (FURG), 2024, ISBN: 978-65-272-1184-6.

[7] 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", *Journal of Computational physics*, vol. 378, pp. 686–707, 2019. DOI: 10.1016/j.jcp.2018.10.045.

[8] 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].

[9] E. F. Toro, *Godunov methods: Theory and applications*. Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4615-0663-8.