The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

HYBRID FINITE VOLUME - GRAPH NEURAL NETWORK METHOD FOR COMPRESSIBLE INVISCID FLUID FLOW SIMULATION

Ondřej Bublík[1], **Václav Heidler**[1] **and Jan Vimmr**[1,2]

[1] NTIS - New Technology for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 30100 Pilsen, Czech Republic
[2] Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 30100 Pilsen, Czech Republic

Key words: graph neural network, CFD, machine learning, compressible fluid flow

Summary. Graph neural networks (GNNs) are gaining traction in computational fluid dynamics (CFD) due to their compatibility with unstructured meshes, unlike traditional convolutional neural networks (CNNs). However, GNNs and other data-driven approaches often violate conservation laws. To address this, we propose a hybrid finite volume-graph neural network (FV-GNN) method, which combines the traditional finite volume method (FVM) with machine learning. This method is applied to the problem of compressible fluid flow in a cascade of blades.

1 INTRODUCTION

Recent advancements in deep learning have revolutionized computational fluid dynamics (CFD). For example through the use of graph neural networks (GNNs). These networks enable modeling flow characteristics and also can handle complex geometries. They excel in scenarios involving unstructured meshes by representing data as graphs. GNNs demonstrate significant potential in CFD simulations by effectively learning from irregular connectivity patterns inherent in unstructured meshes [1]. However, challenges such as scalability and conservation law errors still need to be addressed [2, 3].

To mitigate these issues, hybrid models that integrate traditional numerical methods with neural networks have emerged. Notable examples include the work of Finn [4], which combine the finite volume method (FVM) with neural networks. This hybrid approach enhances the handling of boundary conditions and ensures the conservation of physical quantities.

This paper introduces a hybrid finite volume - graph neural network (FV-GNN) method. In this approach, the neural network is used to predict fluxes at volume interfaces, traditionally determined by methods such as Rusanov or Van-Leer approximations. This innovation reduces computational demands and leverages GPU compatibility, significantly enhancing performance. The FV-GNN method not only simplifies implementation but also ensures conservativity, presenting a promising solution for CFD applications on unstructured meshes.

2 HYBRID FINITE VOLUME - GRAPH NEURAL NETWORK METHOD

As the mathematical model we consider the system of Euler equations, which describe compressible flow of an inviscid fluid. The fundamental scheme of the FVM is outlined as follows

[6]

$$\boldsymbol{w}_i^{n+1} = \boldsymbol{w}_i^n - \frac{\Delta t}{A_i} \sum_{j=1}^{n_e} \boldsymbol{H}(\boldsymbol{w}_i^n, \boldsymbol{w}_j^n, \boldsymbol{n}_j) \Delta l_j, \tag{1}$$

where Δt is the time step, A_i is the area of the finite volume, n_e denotes the number of faces and $\boldsymbol{H}$ is the numerical flux. Symbols $\boldsymbol{n}_j$ and Δl_j denote the normal vector and the length corresponding to the j-th face, respectively.

GNNs solve differential equations on unstructured meshes by updating node values based on neighboring nodes variables. This process typically involves two steps: predicting values at the graph edges connecting nodes and using these values to update the node variables. Both steps are modeled via a multilayer perceptron (MLP) [1].

In this study, we integrate GNNs with the cell-centered FVM approach. By considering the dual graph of the FVM mesh (Fig. 1), the FVM scheme mimics GNN behavior. Our hybrid FV-GNN method replaces the second GNN step with the FVM scheme

$$\boldsymbol{w}_i^{n+1} = \boldsymbol{w}_i^n - \frac{\Delta t}{A_i} \sum_{j=1}^{n_e} \mathcal{H}_j \Delta l_j, \tag{2}$$

retaining the first GNN step as an neural network approximation $\mathcal{H}$ of the FVM numerical flux $\boldsymbol{H}$. This hybrid method ensures conservativity, unlike pure GNNs, and simplifies implementation by substituting the complex numerical flux calculation in FVM with MLP predictions.

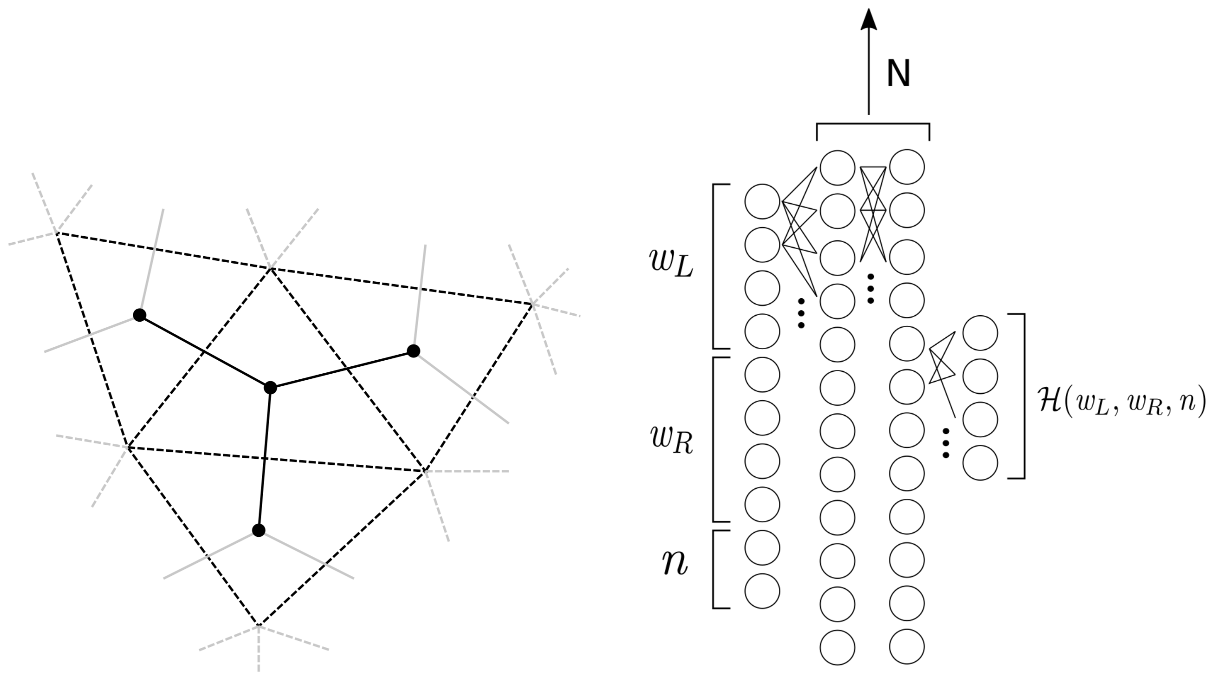

Figure 1: FVM mesh (dashed lines) and the corresponding dual graph (solid lines) (left). Architecture of the employed Multilayer Perceptron (right).

3 CASE STUDY

To validate the proposed computational model, we consider the problem of inviscid compressible fluid flow through a blade cascade. The blade profiles are defined using two Bezier curves, illustrated in Fig. 2. A spacing of 1 unit between the blades is maintained, and a structured quadrilateral computational mesh is used to discretize the area between the blades, as shown in Fig. 2.

To handle boundary conditions, we introduce a ghost cell, serving as an extra layer of cells. At the inlet, we specify the vector of conservative variables within this ghost layer. And for the outlet, density and velocity components are extrapolated from the flow field here. On the solid impermeable wall, we determine the normal momentum and subsequently construct the vector of conservative variables. For the periodic boundary, values are simply copied.

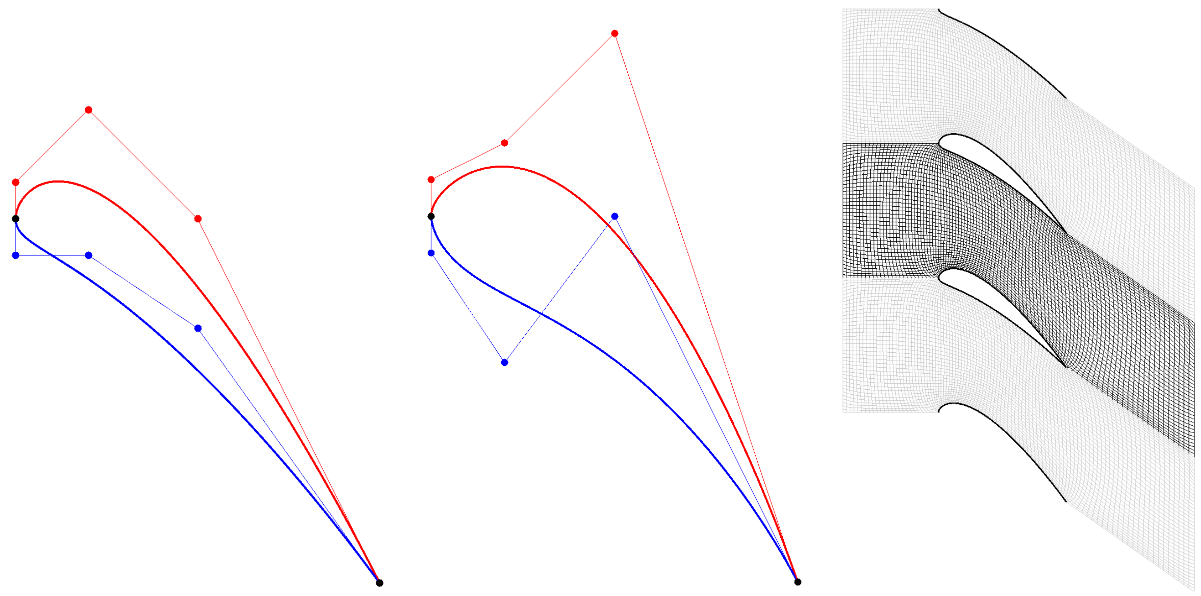

Figure 2: Considered test profiles. Profile 1 (left) and Profile 2 (middle). Computational mesh (right).

4 TRAINING DATASET

Creating an effective training dataset requires diversity and randomness in generating the vectors $\boldsymbol{w}_R$ and $\boldsymbol{w}_L$. This is achieved by defining specific ranges for the variables: $\varrho = [\varrho_{min}, \varrho_{max}]$, u, $v = [-u_{max}, u_{max}]$ and $p = [p_{min}, p_{max}]$. Within these intervals, random values ϱ_r, u_r, v_r, and p_r are generated. These random values are then used to construct a vector of conservative variables:

$$\boldsymbol{w} = \left[\varrho_r,\ \varrho_r u_r,\ \varrho_r v_r,\ \frac{p_r}{\kappa - 1} + \frac{1}{2}\varrho_r(u_r^2 + v_r^2)\right].$$

Additionally, the unit normal vector $\boldsymbol{n} = [\sin(\alpha_r), \cos(\alpha_r)]$ is generated by selecting a random angle $\alpha_r \in [0, 2\pi]$.

With these randomly generated vectors $\boldsymbol{w}_R$, $\boldsymbol{w}_L$, and $\boldsymbol{n}$, the training dataset is constructed. To ensure consistency, all values within the input and output vectors are normalized to the interval $[0, 1]$.

5 NUMERICAL RESULTS

For the purpose of validating the proposed approach, we trained three neural network models designed for the approximation of numerical fluxes. The hyperparameters of the used models are illustrated in Table 1. These models differs in hidden layer sizes and are designed to encompass Rusanov flux type [5].

Table 1: Hyper-parameters of used neural network

input neurons	10
output neurons	4
hidden layers	2
neurons in hidden layer	N [64, 128, 256]
activation function	ReLU
activation function at output layer	sigmoid

The created models were tested on the problem of flow over a cascade of blades, as described in previous section. The simulations were conducted for two different airfoil profiles and two distinct pressure ratios, $\Pi = 0.5$ and $\Pi = 0.7$.

Fig 3 display the pressure field distributions for both profiles and pressure ratio $\Pi = 0.7$, alongside their comparison with CFD calculations. Further, the table 2 presents the error in drag and lift forces computed based on the pressure distribution along the airfoil.

Fig 4 compare the pressure field for airfoil profile 2 and a pressure ratio of $\Pi = 0.7$ with the CFD results.

The consistent trend observed in all obtained results indicates that with an increasing number number of neurons increase the model's capability to capture the physical phenomena under consideration. The results demonstrate that model with 256 neurons in each hidden layer is sufficiently accurate with the error in lift and drag remaining below 7%.

6 CONCLUSIONS

In this paper, we introduce a hybrid FV-GNN method that merges the traditional finite volume approach for discretizing partial differential equations governing compressible inviscid fluid flow with a neural network to approximate the numerical flux across boundaries. This study aims to examine how varying the number of neurons in the neural network affects the resulting pressure field in specific scenarios. The findings reveal that a neural network with 256 neurons in the hidden layers closely approximates the numerical flux, producing results comparable to the exact numerical solution. With 128 neurons, the solution maintained relatively low error, although some deviations in the pressure field were observed in certain regions. A configuration with 64 neurons proved insufficient. Given the complexity of the problem, these results suggest the broad applicability of the proposed method. A significant advantage of this approach over

Table 2: Rusanov flux

	profile 1		profile 2	
pressure ratio	$\Pi = 0.7$	$\Pi = 0.5$	$\Pi = 0.7$	$\Pi = 0.7$
size 64				
Lift error [%]	24.7	32.9	14.2	11.9
Drag error[%]	15.4	23.2	9.7	10.6
size 128				
Lift error [%]	1.1	3.9	1.4	0.7
Drag error[%]	1.0	8.2	1.1	2.0
size 256				
Lift error [%]	1.4	6.7	2.7	0.6
Drag error[%]	0.6	3.4	1.3	2.2

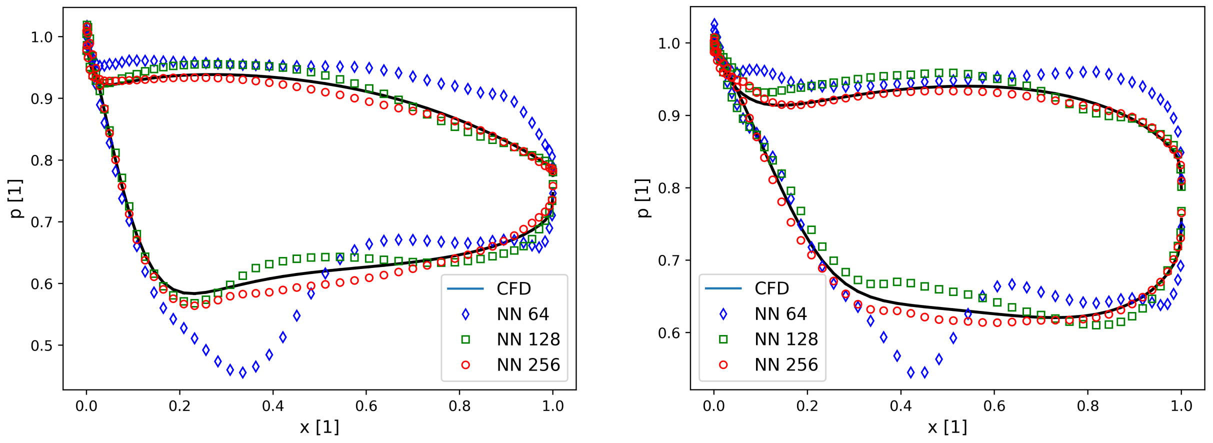

Figure 3: Pressure distribution along Profile 1 (left) and 2 (right) with the pressure ratio of $\Pi = 0.7$

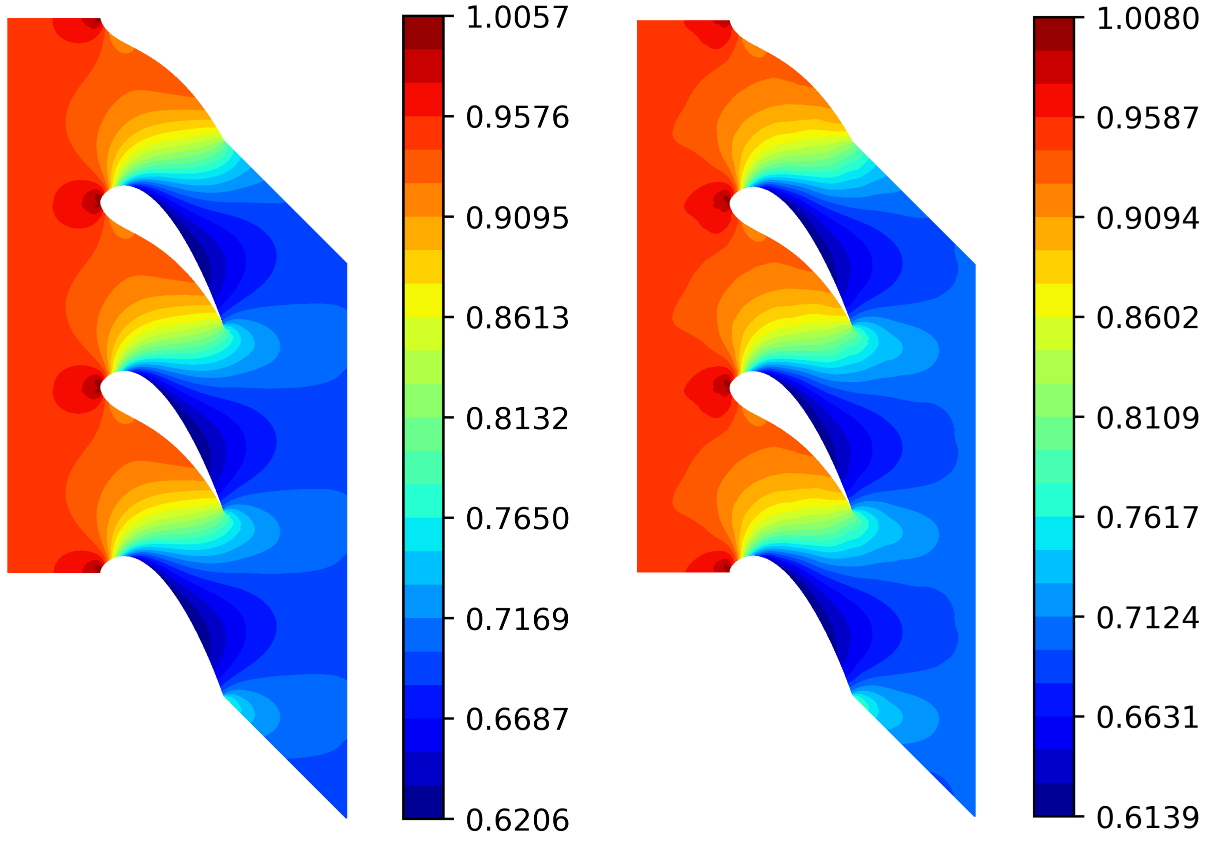

Figure 4: Comparison of the pressure distribution for the CFD calculation (left) and the model with 256 neurons (right) for the Rusanov flux, Profile 2, and the pressure ratio of $\Pi = 0.7$

classical FVM is its replacement of complex flux functions with a neural network, simplifying the implementation of the computational scheme.

REFERENCES

[1] Battaglia Peter W., Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. "Relational Inductive Biases, Deep Learning, and Graph Networks." arXiv preprint arXiv:1806.01261. https://doi.org/10.48550/arXiv.1806.01261

[2] Pfaff Tobias, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. 2020. "Learning Mesh-Based Simulation with Graph Networks." arXiv preprint arXiv:2010.03409. https://doi.org/10.48550/arXiv.2010.03409

[3] Shao Xuqiang, Zhijian Liu, Siqi Zhang, Zijia Zhao, and Chenxing Hu. 2023. "PIGNN-CFD: A Physics-Informed Graph Neural Network for Rapid Predicting Urban Wind Field Defined on Unstructured Mesh." Building and Environment 232: 110056. https://doi.org/10.1016/j.buildenv.2023.110056.

[4] Praditia Timothy, Matthias Karlbauer, Sebastian Otte, Sergey Oladyshkin, Martin V. Butz, and Wolfgang Nowak. 2021. "Finite Volume Neural Network: Modeling Subsurface Contaminant Transport." arXiv preprint arXiv:2104.06010. https://doi.org/10.48550/arXiv.2104.06010

[5] Toro E.F. 2009. Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Berlin, Heidelberg: Springer. DOI:10.1007/b79761.

[6] Hirsch Charles. 2007. "Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics." Journal of Transportation Technologies, Vol.5 No.4. https://doi.org/10.1016/B978-0-7506-6594-0.X5037-1.