



**HAL**  
open science

# DIFFLOW -A FRAMEWORK TO INCORPORATE THE PHYSICAL GRADIENT IN DEEP LEARNING MODELS FOR FLUID DYNAMICS

Eduardo Vital, Thibaut Munzer, Florent Bonnet, Morgane Bourgeois And  
Youssef Mesri, Youssef Mesri

► **To cite this version:**

Eduardo Vital, Thibaut Munzer, Florent Bonnet, Morgane Bourgeois And Youssef Mesri, Youssef Mesri. DIFFLOW -A FRAMEWORK TO INCORPORATE THE PHYSICAL GRADIENT IN DEEP LEARNING MODELS FOR FLUID DYNAMICS. 34th International Conference on Parallel Computational Fluid Dynamics, May 2023, Cuenca, Ecuador. hal-04510102

**HAL Id: hal-04510102**

**<https://hal.science/hal-04510102v1>**

Submitted on 18 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# DIFFLOW - A FRAMEWORK TO INCORPORATE THE PHYSICAL GRADIENT IN DEEP LEARNING MODELS FOR FLUID DYNAMICS

Eduardo VITAL\*, Thibaut MUNZER<sup>†</sup>, Florent BONNET<sup>†</sup>, Morgane BOURGEOIS<sup>†</sup> AND Youssef MESRI\*

\* Mines Paris - PSL University - CEMEF  
60 Bd Saint-Michel, 75272 Paris - France  
e-mail: eduardo.vitalbrasil@poli.ufrj.br, youssef.mesri@mines-paristech.fr  
Web page: <https://mines-paristech.eu/>

<sup>†</sup>Extrality  
19 Bd Poissonnière, 75002 Paris - France  
e-mail: thibaut,florent, morgane@extrality.ai - Web page: <https://www.extrality.ai/>

**Key words:** Deep Learning, CFD, Physically Informed Networks, Adjoint method

**Abstract.** Nowadays, Computational Fluid Dynamics (CFD) is a fundamental tool for industrial design. However, the computational cost of doing such simulations is expensive and can be detrimental for real-world use cases where many simulations are necessary, such as the task of shape optimization. Recently, Deep Learning (DL) has achieved a significant leap in a wide spectrum of applications and became a good candidate for physical systems, opening perspectives to CFD. To circumvent the computational bottleneck of CFD, DL models have been used to learn on Euclidean data, and more recently, on non-Euclidean data such as graphs and manifolds, allowing much faster and more efficient surrogate models. Nevertheless, DL presents the intrinsic limitation of extrapolating out of training data distribution. In this study, we present a pioneer work to increase the generalization capabilities of Deep Learning by incorporating the physical gradients (derivatives of the outputs w.r.t. the inputs) to the models. Our strategy has shown good results towards a better generalization of DL networks and our methodological/theoretical study is corroborated with empirical validation.

## 1 INTRODUCTION

In the context of computationally expensive CFD simulations, the progress of Deep Learning (DL) has made possible the analysis of data and proposed new approaches on how to handle fluid dynamics' problems [1, 2]. Therefore, using DL can significantly help to speed up the solution of numerical simulations with reasonable precision [3].

Moreover, since Deep Learning models are statistical, it is well known that the more data is available, the better models are. Nevertheless, the data is generated from costly processes, and new ways to improve generalization performance with the same amount of examples remain an open problem with several research directions. One possibility,

that is analyzed in this work, would be to supervise the network not only with the fluid dynamics but also its gradients. A similar procedure was applied in a financial context, called Differential Machine Learning (DML)[4].

The concept behind it is to use the gradients of the target labels w.r.t.the input data ( $\frac{\partial y^i}{\partial x^i}$ , where  $x^i$  is the input and  $y^i$  the output) during training. It was proven that, in regression problems, this can largely improve performance results and allow accurate functions to be learned from small datasets, decreasing also the need for regularization techniques. The model’s gradient used during supervision is the same computed by AD [5] by the Deep Learning framework for backpropagation. In this way, the gradient loss is also backpropagated, making the model twice differentiated.

To determine the derivatives of the fluid dynamics system the adjoint method can be used, allowing to compute the gradient of a specific function w.r.t. to its parameters when this function is constrained by a PDE. This method presents itself as a fast and efficient alternative, regardless of the quantity parameters. Hence, it is commonly used in the CFD industry for shape optimization.

Our work leverages the gradient data computed by the adjoint method in order to improve the generalization performance of currently used surrogate models in industry and academia. To summarize, our contributions are as follows: we propose improvements to Differential Machine Learning that can be applied in any complex or simpler cases; we also propose Diffflow: a framework that can drastically reduce errors on unseen data, mitigating the need to acquire/generate extra CFD simulations. To the best of our knowledge, this is the first application of Differential Machine Learning in fluid dynamics.

Although efficient surrogate models have been developed to tackle physics problems, a standard benchmarking dataset is not available. The data becomes even scarcer when also the adjoint gradient is required.

Therefore, we develop our own dataset but based on previous papers. A reliable, high-fidelity dataset constituting these conditions was developed by Bonnet *et al.* [6] and will serve as a reference for the primal equations (does not include the adjoint). The 2D airfoils presented are in the steady-state subsonic regime, while the incompressible flow presents a turbulent behavior and is modeled by the RANS equations with  $k - \omega$  SST turbulence.

## 2 Methodology

### 2.1 Learning scheme

In Differential Machine Learning the derivative ( $\frac{\partial \bar{y}_i}{\partial x_i}$ ) is not passed as target value, but computed from the predictions ( $\hat{y}_i$ ) with AD and used to add an extra penalization term to the loss function, the difference to the actual differential labels:  $\frac{\partial \bar{y}_i}{\partial x_i} - \frac{\partial \hat{y}_i}{\partial x_i}$ . This term has the form:

$$L_G = \frac{1}{N} \frac{\gamma}{\|\frac{\partial \hat{y}}{\partial \hat{x}}\|_b^2} \sum_{i=1}^N \left( \frac{\partial y_i}{\partial x_i} - \frac{\partial \hat{y}_i}{\partial \hat{x}_i} \right)^2 \quad (1)$$

The hyper-parameter  $\gamma$  is important to weight the loss of the gradient with the loss of the original labels. The final loss takes the following shape:

$$Loss = L_B + L_G \quad (2)$$

Where  $L_B$  is the baseline loss. When scaling the input and output, however, the ground truth gradient will no longer correspond to the one from the model. For the cases where the adjoint can achieve extreme values, the design space of the target and the differential labels is no longer the same. We propose then a new parameter  $\alpha$ :

$$L_G = f\left(\alpha \frac{\partial y}{\partial x}, \frac{\partial \hat{y}}{\partial x}\right) \quad (3)$$

The  $\alpha$  parameter can be optimized along with the network or beforehand. We encountered better results with the second option.

## 2.2 Dataset and baseline

The continuous adjoint method of the  $K - \omega SST$  model was chosen to generate the 2D NACA dataset. Most of the primal simulation parameters were inspired by [6]. The lift was selected as target, because its sensitivity presented a more informative distribution for the NACA wings than the drag.

In terms of design space, the main dataset was developed using 4-digits NACA wings. It included only shape variation, and thus, the freestream velocity  $U$  and  $AoA$  are maintained constant.

The target to predict, in its turn, is the lift force per point, as predicting surface variables instead of the total performance metric helps the network to better represent the function.

The entry data should describe the surface of the wing. Normally this is represented by a Point Cloud, where the coordinates  $x$  and  $y$  can be used to describe the geometry. In addition, as NACA wings present a sharp shape at the edges, its description can be improved by using also the normals  $(n_x, n_y)$  to the 2D surface. In this way, the features have 4 dimensions.

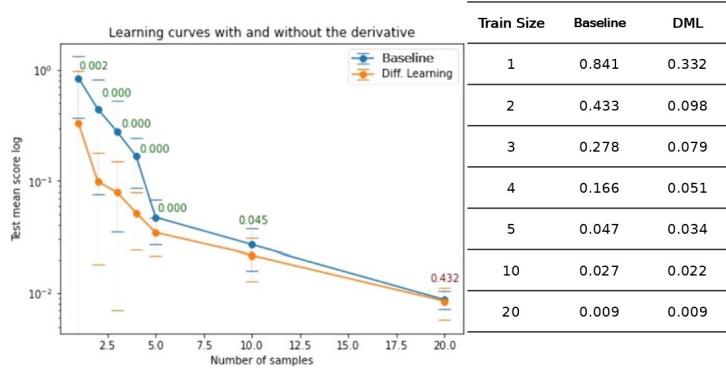
To represent this unstructured geometric data, PointNet [7] was chosen for being a simple and widely used architecture for benchmarking in computer vision.

## 2.3 Gradient processing framework

The continuous adjoint gradient is unstable and presents outliers on the trailing edge, where the shape is sharper. In addition, the difference in design space from the target and its derivatives makes the training/optimization process a challenge for the network. To allow the correct learning procedure, a processing framework was built for the adjoint.

Removing the adjoint by coordinate (trailing edge) during training produced convincing results, despite not being entirely reproducible to datasets containing other shapes. Adapted scaling and normalization techniques were also applied to the input and the output, during and before training.

### 3 Experiments



**Figure 1:** Comparison between DML and the baseline for the NACA dataset, with different shapes in test and train. Green and red annotations represent the p-value of the hypothesis that models are the same.

The performance of the developed framework is compared with the baselines, which do not consider the differential labels. An interesting piece of information to extract is how both methods compare when different amounts of data are available for train. As results presented an important variation depending on the seed, the test set remained constant while the train set was sampled and run multiple times with different initialization weights.

The regularization effects were stronger when certain shapes patterns were selected for training and the rest for evaluation (Fig. 1).

### 4 Conclusion

Using the gradient generated by the adjoint method proved to be an efficient method to improve the generalization performance of Deep Learning models for CFD.

In this work, we proposed improvements to Differential Machine Learning and leveraged it to use in fluid dynamics and graph models, achieving good results even when a high-quality gradient is not available. The developed framework, DiffFlow, allows, in some cases, the use of half the original quantity of data to achieve similar performances.

### REFERENCES

- [1] L. Wang, Y. Fournier, J.-F. Wald, and Y. Mesri, “Improving confidence on cfd by deep learning,” in *33rd Parallel CFD International Conference*, 2022.
- [2] L. Wang, Y. Fournier, J. Wald, and Y. Mesri, “A graph neural network-based framework to identify flow phenomena on unstructured meshes,” *Physics of Fluids*, vol. 35, no. 7, 2023.
- [3] Y. Yang and Y. Mesri, “Learning by neural networks under physical constraints for simulation in fluid mechanics,” *Computers & Fluids*, vol. 248, p. 105632, 2022.
- [4] B. Hugel and A. Savine, “Differential Machine Learning,” *arXiv:2005.02347 [cs, q-fin]*, Sept. 2020. arXiv: 2005.02347.

- [5] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. Society for Industrial and Applied Mathematics, second ed., Jan. 2008.
- [6] F. Bonnet, J. A. Mazari, P. Cinnella, and P. Gallinari, “AirfRANS: High Fidelity Computational Fluid Dynamics Dataset for Approximating Reynolds-Averaged Navier–Stokes Solutions,” in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, Sept. 2022.
- [7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” *arXiv:1612.00593 [cs]*, Apr. 2017. arXiv: 1612.00593.