

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/353324817>

Machine Learning Computational Fluid Dynamics

Conference Paper · June 2021

DOI: 10.1109/SAI553221.2021.9483997

CITATIONS

4

READS

2,608

6 authors, including:



Ali Usman

Luleå University of Technology

17 PUBLICATIONS 191 CITATIONS

[SEE PROFILE](#)



Muhammad Rafiq

Yeungnam University

16 PUBLICATIONS 67 CITATIONS

[SEE PROFILE](#)



Saeed Muhammad

Khalifa University

33 PUBLICATIONS 573 CITATIONS

[SEE PROFILE](#)



Ali Nauman P.hD.

Yeungnam University

21 PUBLICATIONS 679 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Turbocharger design and testing [View project](#)



Mini and micro-channel heatsinks [View project](#)

Machine Learning Computational Fluid Dynamics

Ali Usman
EISLAB Machine Learning
Luleå University of Technology
Luleå, Sweden
ali.usman@ltu.se

Muhammad Rafiq
Data Science Lab
Yeungnam University
Gyeongsan-si, South Korea
rafiq@ynu.ac.kr

Muhammad Saeed
Mechanical Engineering Department
Khalifa University of Science and Tech
Abu Dhabi, United Arab Emirates
muhammad.saeed1@ku.ac.ae

Ali Nauman
WINLab
Yeungnam University
Gyeongsan-si, South Korea
anauman@ynu.ac.kr

Andreas Almqvist
Division of Machine Element
Luleå Tekniska Universitet
Luleå, Sweden
andreas.almqvist@ltu.se

Marcus Liwicki
EISLAB Machine Learning
Luleå University of Technology
Luleå, Sweden
marcus.liwicki@ltu.se

Abstract— Numerical simulation of fluid flow is a significant research concern during the design process of a machine component that experiences fluid-structure interaction (FSI). State-of-the-art in traditional computational fluid dynamics (CFD) has made CFD reach a relative perfection level during the last couple of decades. However, the accuracy of CFD is highly dependent on mesh size; therefore, the computational cost depends on resolving the minor feature. The computational complexity grows even further when there are multiple physics and scales involved making the approach time-consuming. In contrast, machine learning (ML) has shown a highly encouraging capacity to forecast solutions for partial differential equations. A trained neural network has offered to make accurate approximations instantaneously compared with conventional simulation procedures. This study presents transient fluid flow prediction past a fully immersed body as an integral part of the ML-CFD project. MLCFD is a hybrid approach that involves initialising the CFD simulation domain with a solution forecasted by an ML model to achieve fast convergence in traditional CFD. Initial results are highly encouraging, and the entire time-based series of fluid patterns past the immersed structure is forecasted using a deep learning algorithm. Prepared results show a strong agreement compared with fluid flow simulation performed utilising CFD.

Keywords—Machine learning, fluid-structure interaction, computational fluid dynamics, numerical analyses, flow past a cylinder

I. INTRODUCTION

Multiphysics involved in fluid-structure analyses creates time and computational efficiency issues related to design approaches based on conventional numerical simulations. Such approaches typically discretise the governing equations in time and space, solve them using iterative numerical schemes, and computational time increases exponentially with an increase in mesh density. Similar to CFD and FSI, discretised numerical computations are involved in numerous engineering fields, e.g., electric- and electromagnetic- fields, acoustics, chemical species transport, fluid flow, heat transfer, optics, semiconductors, tribology and structural mechanics.

Recently, Berg and Nystrom [1] used deep feedforward artificial neural network (ANN) and unconstrained gradient-based optimisation to approximate partial differential equation (PDE). They showed an example where classical mesh-based methods could not be used, and neural network is an attractive alternative. Similarly, Long et al. used feedforward ANN to approximate a PDE solution and learned the PDE under consideration in an inverse approach [2]. They showed that the ability of ANN to learn the entire family of PDE based on available data is remarkable. Nabian and Meidani enforced initial and boundary conditions during the feedforward ANN training, and the loss function were optimised for randomly selected spatial points during each iteration [3]. This makes the outcome of ANN so-called mesh-less. Sun et al. proposed an approach for extracting governing PDEs for dynamic datasets fed to an ANN-based model [4].

Most recently, Li et al. [5] introduced a Fourier Neural Operator (FNO) to learn the entire family of parametric PDEs, including the Navier–Stokes equations. This work also presents ingeniously interpreting results between discrete mesh points. The method was shown to outperform traditional methods to solve PDEs in terms of error. Moreover, CPU-time required to get an output from a neural network for varying initial conditions is far lesser than that required by traditional numerical approaches, for instance, finite element method, to solve PDEs. Similarly, a study has reported successful ML utilisation to classify 2D surface pressure data [6]. The 2D surface data can be viewed as spatial and temporal variations in the output of a multivariable function similar to PDE output. This study shows the capabilities of ML to study a domain that is not directly considered eligible for transfer learning.

The Navier–Stokes equations are utilised extensively to estimate fluid-structure interaction in machine elements, for instance, air and blade interaction in a wind turbine. Therefore, the successful utilisation of ML to approximate and forecast fluid vortices and eddies, e.g., [7], is a proof of concept for the present study. Shortly, we foresee that computational mechanics involved in engineering design will rely heavily on ML because of the widespread benefit offered by ML, including time and efficiency issues related to the traditional design approach. Moreover, ML algorithms also suggest efficient strategies to reduce three-dimensional data to the planar transport network, such as turbulent CFD superstructures [8]. Knowledge gap: even though ANN-based physics-informed and data-driven models have been shown to accurately approximate and instantaneously forecast multivariable functions, the application of such models is

rarely explored toward developing an alternative framework to perform a computational hybrid ML-CFD approach in particular and computational engineering physics in general.

This study evaluates the utilisation of FNO [5] for FSI between incompressible fluid and a circular immersed body. A 2D FSI is computed utilising a commercially available CFD package. A comprehensive dataset is generated to train the FNO for varying boundary conditions at the inlet of the fluid domain and corresponding fluid dynamics developing in the computational domain the passage of time.

II. MODELLING

A. Computational fluid dynamic model

The physical description and the applied boundary conditions for the current work are shown in Fig. 1. The computational domain size is $30D \times 30D$, where D is the diameter of the cylinder. The Inlet boundary is placed $10D$ upstream of the cylinder while the outlet is positioned at $20D$ downstream. The distance of the side walls is maintained at $15D$ on each side of the cylinder. The structured mesh was generated using hexahedral elements, and an O-grid was introduced around the cylinder walls to ensure high-quality mesh in the vicinity of the boundary layer thickness providing 50 nodes within the boundary layer region. Mesh topology and node distribution near the cylinder wall are displayed in Fig. 1.

$$\frac{\partial}{\partial x}(\rho \mathbf{v}) = 0 \quad (1)$$

$$\sigma = -p\mathbf{I} + \mu(T)(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \quad (2)$$

$$\rho(D\mathbf{v}/Dt) = \text{div} \sigma \quad (3)$$

Transient and incompressible forms of the continuity and Navier-Stokes equations (Eq. 1 – 3) [9] are considered where σ is Cauchy stress, and are solved using a commercial software ANSYS-CFX. The normal velocity condition is imposed at the inlet boundary, and average pressure is specified at the outlet boundary. At the same time, the side walls are assigned with no-slip conditions. The time step and

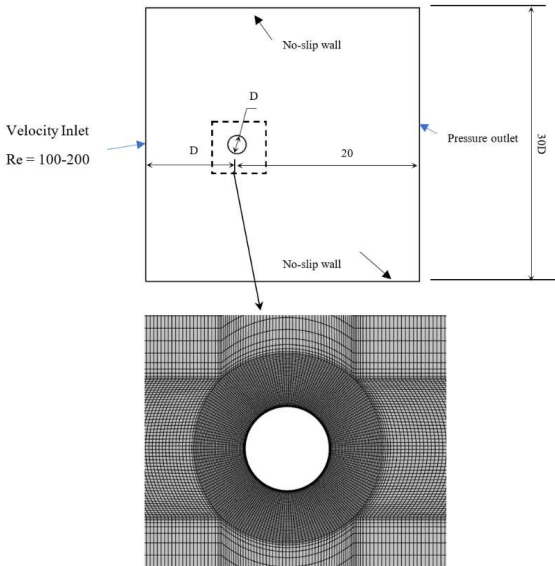


Fig. 1 Computational domain and boundary conditions (top), and mesh around the cylinder (bottom)

the total simulation time of 0.001s and 1s, respectively, were used for all calculations performed within the present study.

TABLE I. THE DATASET: FIRST FEW COEFFICIENT OF PRESSURES (CoP) FOR A TRANSIENT CFD AT VARYING REYNOLD NUMBERS. THE BLUE COLOUR REPRESENTS THE LOWEST CoP, AND RED REPRESENTS THE HIGHEST CoP FOR THE ENTIRE DATASET.

Re	t	t + 5T	t + 10T
100			
110			
120			
130			
140			
150			
160			
170			
180			
190			
200			

B. The dataset

A complete dataset is developed by numerical simulations of the aforementioned governing equations on the mesh shown in Fig.1b. The developing fluid flow pattern in the computational domain is divided into 1000 timesteps to capture detailed dynamics. The Reynolds number (Re) at the inlet of the boundary is varied from 100 to 200 with an increment of 10. Hence, a comprehensive dataset comprising 11000 distinct pressure field corresponding to each time step was created. The Coefficient of pressure at each timestep is non-dimensionalised by utilising all-time maximum CoP at $Re = 200$. The data for varying Reynolds number is shown in Table I.

C. The deep learning model

An FNO for the Navier–Stokes Equations [5] was adopted when designing the neural network architecture. This network consists of four spectral convolution blocks and two fully connected layers with 128 neurons each, and the architecture is shown in Fig. 2.

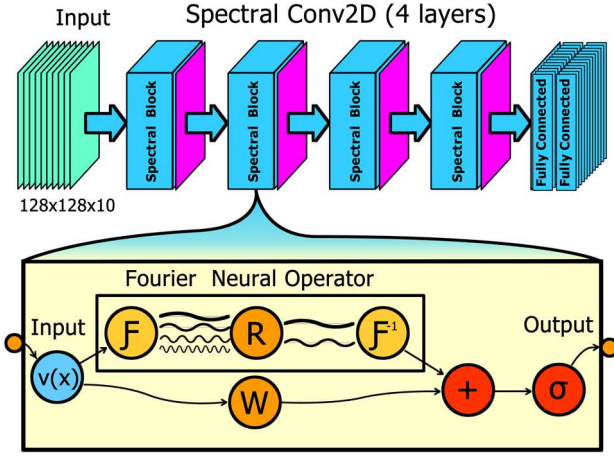


Fig. 2. Deep learning model based on Fourier Neural Operator (a part of this figure is adopted from [5] to illustrate the data flow).

a) *Input*: The model accepts a sequence of 10 images as a single input in the form of a 3D array; every single input is with a shape of 128x128x10. We employ an image sequence generated from the CDF system, as explained in the dataset section. The original sequence is processed to create input blocks with ten images in each sequence at a particular time t .

$$\text{Input_array}[t] = \text{ImageSequence}[t:t + S] \quad (3)$$

where t represent a particular input time, and S represents an input sequence size for one block, which is in our case, is set to 10.

b) *Spectral Convolutional Block*: A spectral convolution block consists of two streams. The FNO converts the input into the frequency domain and applies a low-pass filter to it, meaning that only the low-frequency content is considered in the multiplication between the Fourier transformed input and the systems Greens' function. Finally, the processed data is converted back to the existing domain. The weights used in the network to represent the spectral blocks are then combined with the weights that directly scale the input, as

illustrated in Fig. 2. An activation function is finally applied to obtain the block-wise output.

III. NUMERICAL EXPERIMENTATION

The flow of data and training method utilised for the present study is shown in Fig. 3. Data preprocessing is employed before further the data to FNO model. This includes the resizing, centering, clipping, and aligning of the input image sequence. The preprocessed images are further processed to generate input sequenced blocks to match the model input shape of 128x128x10. The prepared data is split into a training- and a test set sequences. The network is trained using a training loop, which employs loading the next batch, compute the loss function, updates optimiser and backpropagation derivatives for parameter updates. The model is saved at every best validation loss, and in case loss is not further improving, an early stop mechanism ends the training to avoid overfitting with the best weights in hand. Loss function for the training and test splits are shown in Fig. 4.

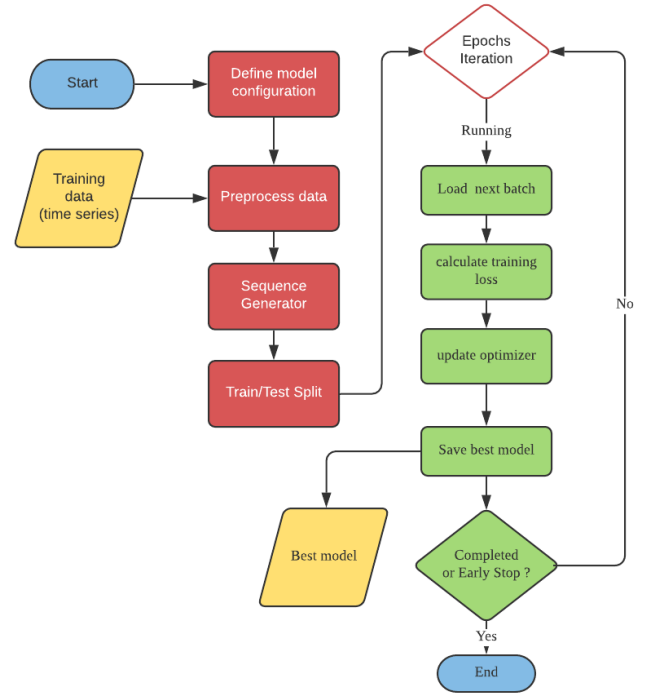


Figure 3. Illustration of model implementation and training procedure utilised in this study.

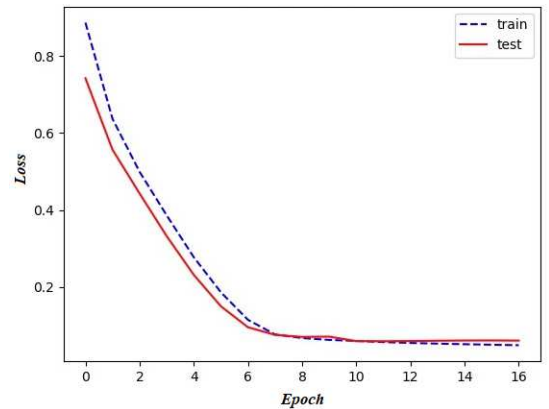
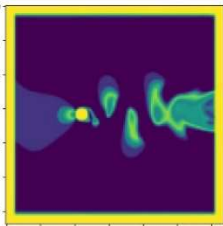
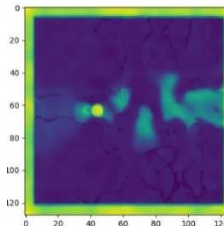
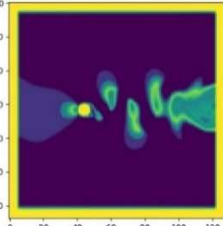
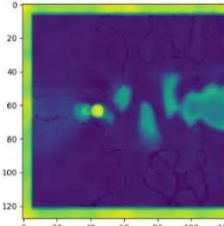
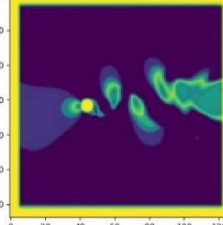
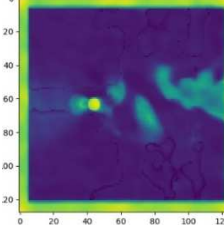
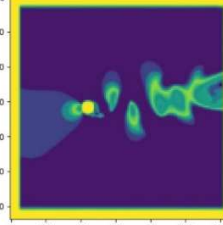
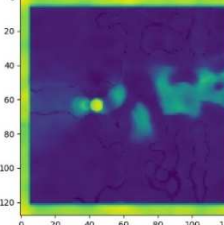
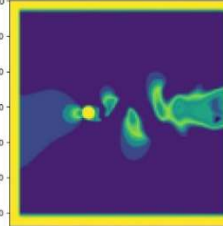
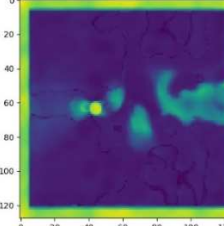


Fig. 4. Loss function value for training and test splits of the dataset.

TABLE II. PREDICTION OF DEEP LEARNING MODEL (DLP) COMPARED WITH THE GROUND TRUTH (GT) BASED ON THE TRADITIONAL NUMERICAL APPROACH OF COMPUTATIONAL FL.

Time	Ground Truth	Prepared
t (s)		
	Pixel to Pixel Comparison = 1.303711%	
t + 20 (ms)		
	Pixel to Pixel Comparison = 1.227539%	
t + 40 (ms)		
	Pixel to Pixel Comparison = 1.218750%	
t + 60 (ms)		
	Pixel to Pixel Comparison = 1.290039%	
t + 80 (ms)		
	Pixel to Pixel Comparison = 1.361328%	

IV. RESULTS AND DISCUSSION

The output sequence is generated given the input development of the solution. CoP solutions at selected instances are shown in Table II. Although there are apparent visual discrepancies between the DLM prediction and the GT, they agree qualitatively. The pixel-to-pixel binary image comparison with a threshold pixel value of 125, the solution prepared while utilising the aforementioned deep learning model, are in high agreement with the solution generated through traditional CFD computations. The model, remarkably, captured vortices generated in fluid past the

cylindrical structure. Here, it is noteworthy that the model forecasts the complete transient solution. Given the initial ten solutions at the beginning of the fluid-structure interaction, the following ten solutions are purely predicted. The solution at the earliest instant, i.e., 1st in the generated sequence, replaces the latest solution in the input sequence. This predict-and-replace process is continued, and after 11 iterations, the complete input sequence has been replaced with the predicted series of solutions. All the iterations from the 12th and until the 1000th are purely based on the predicted sequence of PDE solutions. Since it is an ongoing project and the results presented here are initial developments, the prepared results show diverged solution at some specific grid points. We believe this problem could be addressed by optimising the network architecture, which is our future goal in order to achieve state-of-the-art accuracy. Presently, the learned model is useful to the configuration of the flow field as presented in Table I with the circular body fully immerses in the fluid field; however, we are generating datasets with varying shapes of bodies with varying orientations to make predictions widely generalised.

ACKNOWLEDGMENT

The Swedish Kempe Foundation has funded parts of this work. The authors would also like to acknowledge the support from the Swedish Research Council: DNR 2019-04293.

REFERENCES

- [1] J. Berg and K. Nyström, "A unified deep artificial neural network approach to partial differential equations in complex geometries," *Neurocomputing*, vol. 317, pp. 28-41, 2018/11/23/ 2018, doi: <https://doi.org/10.1016/j.neucom.2018.06.056>.
- [2] Z. Long, Y. Lu, X. Ma, and B. Dong, "PDE-Net: Learning PDEs from Data," presented at the Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2018. [Online]. Available: <http://proceedings.mlr.press>.
- [3] M. A. Nabian and H. Meidani, "A deep learning solution approach for high-dimensional random differential equations," *Probabilistic Engineering Mechanics*, vol. 57, pp. 14-25, 2019/07/01/ 2019, doi: <https://doi.org/10.1016/j.probengmech.2019.05.001>.
- [4] Y. Sun, L. Zhang, and H. Schaeffer, "NeuPDE: Neural Network Based Ordinary and Partial Differential Equations for Modeling Time-Dependent Data," *arXiv pre-print server*, 2019-08-08 2019, arxiv:1908.03190.
- [5] Z. Li *et al.*, "Fourier Neural Operator for Parametric Partial Differential Equations," *arXiv pre-print server*, 2020-10-18 2020, arxiv:2010.08895.
- [6] Monit, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwicki, "Transforming Sensor Data to the Image Domain for Deep Learning - an Application to Footstep Detection," *arXiv pre-print server*, 2017-07-14 2017, arxiv:1701.01077.
- [7] D. Kochkov, Jamie, A. Alieva, Q. Wang, Michael, and S. Hoyer, "Machine learning accelerated computational fluid dynamics," *arXiv pre-print server*, 2021-01-28 2021, arxiv:2102.01010.
- [8] S. Pandey, J. Schumacher, and K. R. Sreenivasan, "A perspective on machine learning in turbulent flows," *Journal of Turbulence*, vol. 21, no. 9-10, pp. 567-584, 2020, doi: 10.1080/14685248.2020.1757685.
- [9] A. Almqvist, E. Burtseva, K. Rajagopal, and P. Wall, "On lower-dimensional models in lubrication, Part A: Common misinterpretations and incorrect usage of the Reynolds equation," *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, p. 135065012097379, 2020, doi: 10.1177/1350650120973792.