COMSOL
CONFERENCE
2024 FLORENCE

Neural Network Surrogates For Backward-Facing Step Flow Up To Re=900 In COMSOL Multiphysics®

M. Rabenanahary[1]
1. R&D and Simulation, PROTOSTEP, Cergy-Pontoise, France.

Abstract

Over the past 10 years, scientific, technical, and technological advancements have significantly propelled the integration of artificial intelligence and machine learning technologies within the field of physics, particularly in numerical simulations and the design of autonomous and efficient Digital Twins. In 2023, COMSOL Multiphysics® introduced, for the first time, functionalities of Surrogate Models, including the ability to train non-informed neural networks using experimental data or models within the software. By training on data from velocity and pressure fields, we evaluated the accuracy and performance of these networks in predicting the backward-facing step steady-state flow for Reynolds numbers (Re) up to ~900, comparing the results to experimental data. We tested various configurations of these networks and the Laminar Flow CFD model samples generated to train them. We measured both the validation accuracy (e.g., for Re<580 and Re<400) and the generalization accuracy of the trained networks in Reynolds number and spatial domains. Our preliminary work indicates that the most optimized neural networks are able to predict the extension of the primary recirculation zone with a minimum validation accuracy of 0.5% and a generalization accuracy ranging between 5.8% and 14.4%. We demonstrated that a neural network trained on 2D numerical simulation data for Re < 400 - where results align within 5% of experimental data - is able to produce generalization predictions consistent with experimental results in the range of 400<Re<920, where the flow becomes three-dimensional. With a computing time of ~0.6 seconds, the neural networks are 12.5 to 14 times faster than a non-linear stationary PARDISO solver operating on the same mesh within COMSOL Multiphysics®.

Keywords: Numerical simulation, computational fluid dynamics, artificial intelligence, Deep Neural Networks for Physics, Digital Twin

Introduction

From elementary particles identification [1, 2] to chemical processes simulations [3], progresses in machine and deep learning over the past decades have increasingly entered fields of physics, therefore benefiting to both industrial sectors and academic research.

Those advances have improved the fidelity and efficiency of alternative neural network (NN) models in Computational Fluid Dynamics (CFD), when confronted to well-grounded predictions of non-deep learning models [4, 5, 6, 7]. In 2023, COMSOL® released surrogate model features, including the ability to train deep neural networks (DNNs) on COMSOL®-generated models or experimental data.

The present work aims to study the predictions of those non-informed networks for the simulation of a steady-state laminar flow over a Backward-Facing Step (BFS) and a Reynolds Number Re ranging from 70 to 900. Such phenomenon widely occurs around buildings, in aerodynamics, heat transfer, engines, condensers, or vehicles (e.g. affecting air-filter performances [8]). Thus, over the last 40 years, this reference CFD problem has been thoroughly studied in 2D/3D experiments and numerical simulations [9, 10, 11, 12, 13, 14, 15, 16], prompting well-established comparison data to the present work. This paper first describes the experimental setup, and assumptions made for the simulations. Then, the general governing model and methods set throughout the simulations are posed. The predictions accuracy of the resulting DNNs are then measured against reference numerical/experimental values, as well as the computing time of the DNNs in contrast to a non-deep learning PARDISO solver in COMSOL®. Finally, ideas of improvements and the envisaged following steps to the present paper are discussed.

General theory, setup, and assumptions

Inspired by the experiment of [9], the 2D stationary simulations here are based on [17]. A channel, whose inlet is injected with a fully developed Poiseuille flow of viscosity μ[Pa · s] and maximum velocity $U_{max} = Re \cdot \frac{\mu}{\rho h_S}$, abruptly expands by a $E = 2.0$ ratio, from width h_1 to $h_2 = E \cdot h_1$ at a step $x = S$ of height h_S (Figure 1). Henceforth, combined effects of free shear layer separation, walls frictions, and adverse pressure gradient (due to an expanding flow) form reattachment zones at the walls [9, 10]. Those delimit flow recirculation regions. There is a primary one sticking to the bottom step. Then, there is an upper secondary one which forms from $Re > 300$.

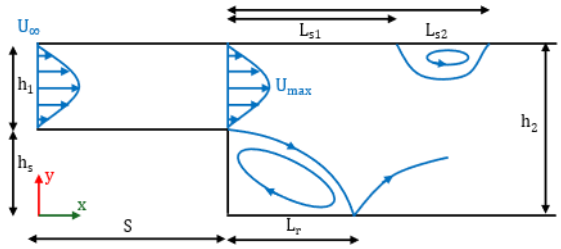

Figure 1. Sketch of the backward-facing step (BFS) flow field.

Governing equations and numerical setup

CFD and simulation model

Under COMSOL®, the 2D stationary, gravity-free, isothermal, incompressible Navier-Stokes equations are solved through the 'Laminar Flow' physics of the 'CFD Module' on the same geometry as in the COMSOL® application [17]:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0\,, \qquad (1)$$

$$\rho(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\boldsymbol{\nabla} p + \mu \boldsymbol{\nabla}^2 \mathbf{u}\,, \qquad (2)$$

where $\mathbf{u} = (u_x, u_y), \rho, p$ are the flow velocity, density, and pressure fields. As the laws of similitude place significant emphasis on dimensionless parameters such as Re_h, U/U_{max}, or y/h, water with $\rho = 997\ kg \cdot m^{-3}$ and $\mu = 8.905 \cdot 10^{-4}\ Pa \cdot s$ is arbitrarily simulated at ambient room temperature [9]. Using second order Lagrange elements to model velocity and linear elements to model pressure helps stabilize the latter through the P1+P2 discretization scheme. Computing speed is improved by disabling the solution-stabilizing streamline diffusion, as the involved Re_h remain low in this case.

Inlet boundary conditions are set to 'Normal flow velocity' $U_\infty = 4\frac{y}{h_1}\left(1 - \frac{y}{h_1}\right) \cdot \frac{\mu}{\rho h_S} Re$ [11], whereas null static pressure is imposed along the Outlet. No-slip conditions are applied to each wall.

For numerical cost-accuracy efficiency, quadrant mesh elements are prioritized far from the step's edge, whose vicinity is meshed with more refined triangular elements (Figure 2). In any case, 5-layer thick boundary layers are resolved near every wall.

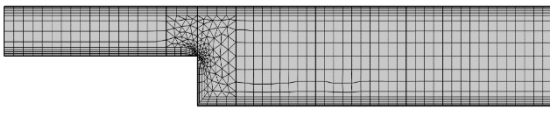

Figure 2. Geometry mesh of the solved CFD model.

Velocity and pressure are solved using a more robust nonlinear Fully Coupled, Stationary, Direct PARDISO solver set with an 'Automatic highly nonlinear (Newton)' damping method.

Surrogate Model Training and DNNs

The newly released 'Surrogate Model Training' node is added to the study. Subsequent sets of models (hereafter called training model datasets) are therefore generated for the DNN trainings via a method of Latin Hypercube Sampling (LHS) of the Quantity of Interest (QoI) Re between $Re_{min} = 70$ and Re_{max}. Following the same LHS method, and for each sampled Re, the values $(\mathbf{u}, p)$ over the whole geometry are spatially sampled to build the full training model dataset of a first base DNN named AI#0. Subsequently, four other DNNs (AI#1, AI#2, AI#3, AI#4) are derived from AI#0, except that this time the $(\mathbf{u}, p)$ values are more finely sampled at the one-point order spatial nodes of the Lagrange elements that form the mesh. Differences between the settings of those 5 successively improved DNNs and their datasets are detailed in the following section.

Based on COMSOL®'s 'Tubular Reactor Surrogate Model Application' [18], all DNNs are built with 3-components inputs (x, y, Re), 3-components outputs (u_x, u_y, p), and 4 hidden layers accounting for 50, 40, 30, and 20 neurons, respectively. Throughout the DNN trainings, an Adaptative Moment Estimation (Adam) optimization algorithm is used to update the learnable parameters of each network, with a propagated batch size of 512, a learning rate of 10^{-3}, and a maximum of 2000 epochs. For each DNN, a randomly picked 10% fraction of its training model dataset is extracted for the validation loss, while the remaining 90% is used for training loss and updates of the learnable parameters. Both losses are expressed as 'Root mean-square errors'. For consistent comparison between the DNNs, their 'Random seeds' used for operations with Random Number Generators (RNGs) are fixed by default to 0. The same way, the 'Initial Random Seeds' of the sets generated by the 'Surrogate Model Training' node are fixed by default to 1014.

Results and discussion

AI#0: base DNN and accuracy metrics

The training model dataset for the DNN AI#0 contains 4000 unique data values (x, y, Re, u_x, u_y, p), each associated to a unique Re value within the range 70<Re<580. Figure 3 and Figure 4 confront the predicted velocity and pressure maps between a Finite Element Method (FEM) COMSOL® simulation, and AI#0/AI#1/AI#2. They qualitatively show that all three trained DNNs can reproduce the global expected flow behaviours: deflection of the main inflowing streamlines, formation of 2 recirculation regions, velocity/pressure distribution, and the structure of the adverse pressure gradient.

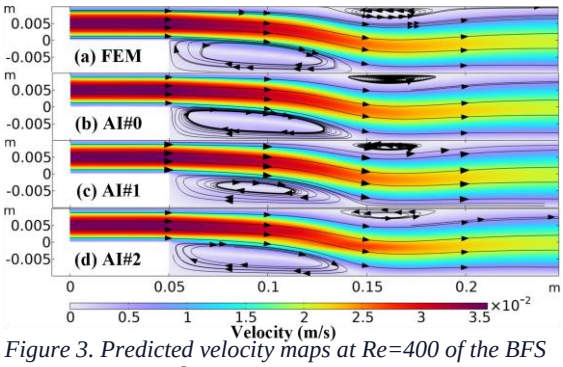

Figure 3. Predicted velocity maps at Re=400 of the BFS flow by COMSOL® simulations and DNNs AI#0, AI#1, AI#2.

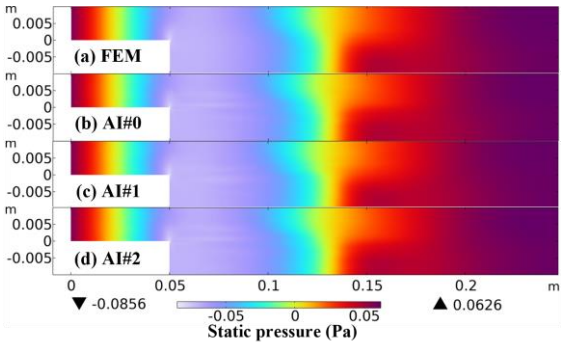

Figure 4. Same as Figure 3, but with static pressure.

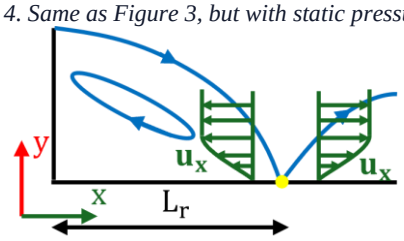

Figure 5. Sketch of velocity profiles around reattachment zone.

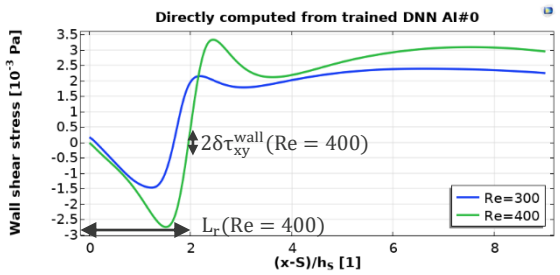

Figure 6. Wall shear stress profiles inferred by AI#0 along the downstream bottom wall.

As pictured in Figure 5 and illustrated by Figure 6 for AI#0, the reattachment positions that delimit the primary and secondary recirculation regions along the walls are characterized by a null wall shear stress (switching between negative and positive τ_{xy}^{wall}):

$$\tau_{xy}^{wall}(x) = \mu \cdot \partial_y u_x\big|_{(x,y)\text{ on wall}} = 0 \qquad (3)$$

This way, as in [9, 10, 12], the accuracy of the DNNs' predictions against the numerical and experimental reference values can be gauged in

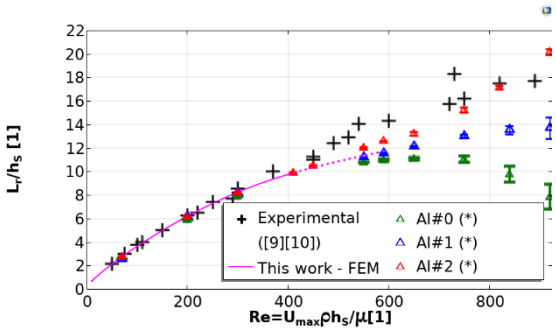

Figure 7. Width of the primary recirculation region against Re, from experimental references [9, 10] *(black), a 2D finite-element COMSOL® simulation (magenta), AI#0 (green), AI#1 (blue), and AI#2 (red). Dashed line shows data inside the additional training range 400<Re<580 for AI#0 and AI#1, as AI#2 is only trained on Re<400 (full line). *: $L_r = x - S$ such that $\left|\tau_{xy}^{wall}(x)\right| = \left|\partial_y u_x\right|_{(x,y)\text{ on wall}} \leq 0.05 \cdot \delta\tau_{xy}^{wall}$, where: $\delta\tau_{xy}^{wall} = \max\left(\left|\min_{x\text{ on wall}}(\partial_y u_x)\right|, \left|\max_{x\text{ on wall}}(\partial_y u_x)\right|\right)$*

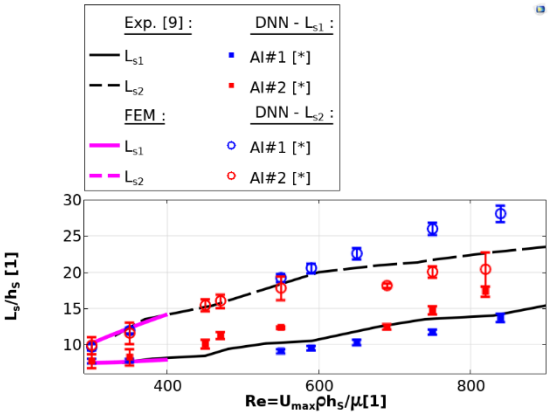

Figure 8. Same as Figure 7, but with the extents of the secondary recirculation region.

reattachment positions L_r, L_{s1}, and L_{s2}. On one hand, AI#0's validation accuracy on L_r performs well on unseen Re<580 data (corresponding to the range within which the DNN is trained), with relative differences from the references ranging between 5.6% and 7.5% (averaging to 6.8%), as presented in . On the other hand, AI#0 fails to generalize accurate predictions on unseen data above Re=580, with relative differences between 23.2% and 44.1% (averaging to 30.6%). One explanation could be that the DNN is underfitting due to the insufficient size and model complexity of its training model dataset, making it unable to capture enough patterns that correlate the DNN's input and output variables.

AI#1: increasing the size of the training data

The sampling of the $(\mathbf{u}, p)$ values on the spatial nodes of the mesh's elements increases the size of AI#1's training model dataset to $\sim 2.3 \times 10^6$ unique (x, y, Re, u_x, u_y, p) values, which accounts to $\sim$600 times the size for AI#0. In contrary to the latter, for each sampled Re in [70 ; 580], thousands

more points over the whole geometry are thus sampled.

As shown on Figure 7, this significantly improves both the validation ($\mathrm{Re} \leq 580$) and the generalization ($\mathrm{Re} > 580$) accuracies of the DNN AI#1. In fact, relative differences to references on L_r range from 0.3% to 4.4% (averaging to 2.7%) on validation, and 15.2% to 22.6% (averaging to 18.8%) on generalization.

In contrast to AI#0, AI#1 now preserves the monotonously increasing length of the primary recirculation region, as expected from experimental data below $\mathrm{Re} < 900$ (a faster flow prompting stronger inertial effects which move the reattachment zones further from the step [10]).

Besides, as expected, Figure 8 confirms that the trained DNNs can reproduce the apparition of the secondary recirculation region above $\mathrm{Re} > 300$.

AI#1 reproduces the increase of the reattachment positions L_{s1} and L_{s2} with increasing Re, even though less accurately than with the primary recirculation region. In fact, AI#1's relative differences to references on $L_{s2} - L_{s1}$ range from 17.3% to 29.2% on validation (averaging to 23.8%), and 39.6% to 75% on generalization (averaging to 59.1%).

On L_r and $L_{s2} - L_{s1}$, differences between the predictions of AI#1 and the expected results may partly be due to the training dataset part within $400 < \mathrm{Re} < 580$. In fact, as experimentally seen by [9, 12], the beforehand bi-dimensional flow becomes strongly three-dimensional above $\mathrm{Re} > 400$, therefore producing disagreement between physical and 2D computational experiments [9], hence the latter's divergence with the predictions of AI#1 on Figure 7.

AI#2: a more pertinent preselection on training data

In order to prove the last assumption, the training dataset of AI#2 is subsequently restricted to $70 < \mathrm{Re} < 400$, while maintaining the same size of $\sim 2.3 \times 10^6$ unique values than AI#1. Therefore, AI#2 is only trained on data that fit with both physical and 2D computational experiments. Figure 7 and Figure 8 show that this preselection prompts clear improvement of generalization accuracy both on the primary and secondary recirculation regions ($\mathrm{Re} > 580$). In fact, relative differences to references on L_r drop between 5.8% and 14.4% (averaging to 10.6%), whereas they drop between 32.2% and 38.3% on $L_{s2} - L_{s1}$ (averaging to 35.1%).

Besides, for validation accuracy ($\mathrm{Re} < 580$), the relative differences to references on L_r range between 0.5% and 4.1% (averaging to 2.0%).

AI#3, AI#4: accuracy on spatial generalization

This section gauges how accurately the DNNs can generalize to regions of space which are unseen to them during the training process.

AI#3 and AI#4 are based on AI#1 and AI#2, respectively. However, the sampled training points (x, y) a re here sampled on less broad regions of the geometry, namely restricted to beyond the step and to the vicinity of the two recirculation regions (hereafter named the generalization region). This way, AI#3 and AI#4 are tested in Figure 9 over the unseen upstream inlet of the channel, for $\mathrm{Re} = 650$.

Qualitatively, the predicted flows over the generalization region successfully remain horizontal and parallel.

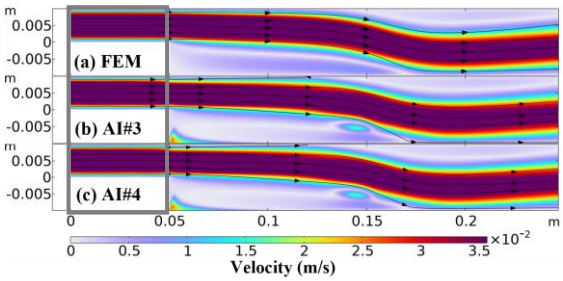

Figure 9. Generalization over the inlet channel of AI#3 and AI#4 (grey frame), for Re=650.

However, more precise measures show wide localized discrepancy between the predicted velocity magnitudes of the 2 DNNs and the FEM solver over the generalization region. For AI#3, the absolute gaps in velocity range from 10^{-3} % to 91.5% of the maximum velocity U_{max} that is measured from the FEM solver. Nevertheless, all the gaps above 30% are concentrated inside a horizontal 1mm-thick layer which is stuck all the way along the bottom wall of the generalization region. For AI#4 (whose training model dataset is restricted to $\mathrm{Re} < 400$), the gaps in velocity drop between 0.01% and 59.8% of U_{max} from the FEM solver, hence again the importance of the preselection on the training data.

Besides, for not yet understood reasons, all the gaps beyond 45% moved this time in a 0.5mm-thick layer all the way along the upper wall of the generalization region.

In any case, the DNNs constructed under COMSOL® in the present work have more difficulties in generalizing over (unseen) space than over (unseen) kinematics Re, especially near the walls where the boundary layers are less accurately reproduced. Forthcoming works may explore improvements to deal with these issues, perhaps through more robust and complex methods of deep learning or through fine-tuning on the training model dataset and hyperparameters of the DNNs (e.g. number of hidden layers or neurons per layer). Those investigations are not held in the present study.

Computing time against a non-deep learning solver

To finish, the execution times of the DNNs are measured for different values of Re<1000, and compared to computing times of a nonlinear Fully Coupled, Stationary, Direct PARDISO solver that is set with an 'Automatic highly nonlinear (Newton)' damping method, all executed on the same mesh within COMSOL®. To take account of statistical fluctuations (e.g. from hardware responses), each measure is repeated 10 times in a row.

First, no variation beyond 1% is measured between the execution times of the 5 tested DNNs, for the same any value of Re.

Second, Table 1 and Figure 10 show that the DNNs execute between 12.5 and 14 times faster than the PARDISO solver, even dropping under 1 s on an Intel® Core™ i7-1185G7 @ 3.00GHz processor. Furthermore, the DNNs keep a more constant computing time than the PARDISO solver over the explored range $Re < 1000$. All these results are consistent with the fact that, aside from the hardware, the execution time of each DNN used here rather depends on the architecture and topology of the neural network, through the number of neurons and hidden layers. In other words, the same number of operations are repeated by the computer with any input value of Re.

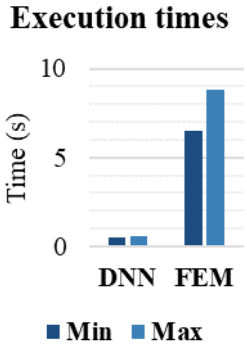

Figure 10. Full range of execution and computing times.

Re (1)	t^{exec}_{DNN} (s)	$t^{exec}_{PARDISO}$ (s)
650	0.56 ± 0.04	7.50 ± 1.00
750	0.57 ± 0.08	7.60 ± 1.10
1000	0.60 ± 0.08	8.10 ± 0.70

Table 1: *Computing and executing times*

Conclusions

Under COMSOL®, we were able to set efficient, fast, and increasingly accurate deep neural networks which can reproduce physical and computational experiments of a steady-state BFS laminar flow.

We showed that throughout the training process of the DNNs, improved accuracy compared to validated references needs a careful attention to: the size and model complexity of the training dataset; the pertinence of the sampled data through preselection.

We illustrated with our test case that a neural network trained on 2D numerical simulation data - within the range of Reynolds number where results align with experimental data - can produce generalization predictions consistent with experimental results in the range of 400<Re<920 - where the flow becomes three-dimensional.

The built DNNs have low execution time compared to non-deep learning methods. This propriety lays an interesting base for possible applications and integrations of such DNNs into reactive devices or codes that probe BFS flows (and more generally, various problems of physics).

Even though we've seen that the DNNs introduced in 2023 within COMSOL® are already powerful, we may note that they remain non-informed. Therefore, more advanced studies may explore the use of more complex methods of deep learning, such as recurrent neural networks [3] or physics-informed neural networks [7, 19, 20]. For forthcoming works, such methods may also help in accurately solving unsteady flows [20]. Finally, as the present work didn't explore the variation of the networks' architecture and algorithms, a thorough fine-tuning of the hyperparameters for the present DNNs might also ensure better optimization of the accuracy of their predictions [5].

References

[1] N. Tamir, I. Bessudo, B. Chen, H. Raiko and L. Barak, "Neural networks for boosted di-τ identification," *Journal of Instrumentation*, vol. 19, no. 07, p. P07004, 2024.

[2] The ATLAS Collaboration, "Transformer Neural Networks for Identifying Boosted Higgs Bosons decaying into $b\bar{b}$ and $c\bar{c}$ in ATLAS," *CERN Notes*, 2023.

[3] Y. Zheng, C. Hu, X. Wang and Z. Wu, "Physics-informed recurrent neural network modeling for predictive control of nonlinear processes," *Journal of Process Control*, vol. 128, p. 103005, 2023.

[4] Y. Zhang, W.-J. Sung and D. Mavris, "Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient," *arXiv (Cornell University)*, 2018.

[5] E. Ajuria-Illaramendi, A. Alguacil, M. Bauerheim, A. Misdariis and B. Cuenot, "Towards a hybrid computational strategy based on Deep Learning for incompressible flows," *AIAA AVIATION FORUM*, pp. 1-17, 2020.

[6] O. Hjortstam, Á. Konrádsson, Y. Serdyuk and C. Häger, "Physics-Informed Neural Networks for Studying Charge Dynamics in Air," 2023.

[7] E. H. W. Ang, G. Wang and B. F. Ng, "Physics-Informed Neural Networks for Low Reynolds Number Flows over Cylinder," *Energies*, vol. 16, no. 12, pp. 45-58, 2023.

[8] Y. Shenghong, "Two dimensional backward facing single step flow preceding an automotive air-filter," PhD Thesis, Oklahoma State University, Stillwater, OK, USA, 2000.

[9] B. F. Armaly, F. Durst, J. C. F. Pereira and B. Schönung, "Experimental and theoretical investigation of backward-facing step flow," *Journal of Fluid Mechanics*, vol. 127, pp. 473-496, 1983.

[10] S. Thangam and D. Knight, "Effect of step height on separated flow past a backward-facing step," *Physics of Fluids*, vol. 1, no. 3, pp. 604-606, 1989.

[11] D. K. Gartling, "A test problem for outflow boundary conditions-flow over a backward-facing step," *International Journal for Numerical Methods in Fluids*, vol. 11, no. 7, pp. 953-967, 1990.

[12] T. Lee and D. Mateescu, "EXPERIMENTAL AND NUMERICAL INVESTIGATION OF 2-D BACKWARD-FACING STEP FLOW," *Journal of Fluids and Structures*, vol. 12, no. 6, pp. 703-716, 1998.

[13] B. Armaly, A. Li and J. Nie, "Measurements in three-dimensional laminar separated flow," *International Journal of Heat and Mass Transfer*, vol. 46, no. 19, pp. 3573-3582, 2003.

[14] X.-D. Yang, H.-Y. Ma and Y.-N. Huang, "Prediction of homogeneous shear flow and a backward-facing step flow with some linear and non-linear K–ε turbulence models," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, no. 3, pp. 315-328, 2005.

[15] H. H. Choi, V. T. Nguyen and J. Nguyen, "Numerical Investigation of Backward Facing Step Flow over Various Step Angles," *Procedia Engineering*, vol. 154, pp. 420-425, 2016.

[16] B. Zajec, M. Matkovič, N. Kosanič, J. Oder, B. Mikuž, J. Kren and I. Tiselj, "Turbulent Flow over Confined Backward-Facing Step: PIV vs. DNS," *Applied Sciences*, vol. 11, no. 22, p. 10582, 2021.

[17] COMSOL's Application Gallery, "Turbulent Flow over a Backward-Facing Step," [Online]. Available: https://www.comsol.com/model/turbulent-flow-over-a-backward-facing-step-228. [Accessed 2024].

[18] COMSOL, "Tubular Reactor Surrogate Model Application," [Online]. Available: https://www.comsol.com/model/tubular-reactor-surrogate-model-application-120381. [Accessed 2024].

[19] M. Raissi, P. Perdikaris and G. 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.

[20] R. Quita, Y.-S. Chen, H.-Y. Lee, A. C. Hu and J. M. Hong, "Conservative Physics-Informed Neural Networks for Non-Conservative Hyperbolic Conservation Laws Near Critical States," *arXiv (Cornell University)*, 2023.