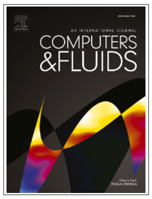
Computers and Fluids 306 (2026) 106946

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid



RANS-CNN: A physics-informed convolutional neural network for solving reynolds-averaged Navier-Stokes equations in duct flows ★

Gaurav Bokil *, Sebastian Merbold, Stefanie De Graaf

German Aerospace Center (DLR), Institute of Electrified Aero Engines, Cottbus, 03046, Germany

ARTICLE INFO

Keywords:
Physics-informed machine learning
Convolutional neural network
Turbulent flow
Reynolds averaged navier stokes equations

ABSTRACT

Classical Computational Fluid Dynamics (CFD) simulations of turbulent flows in aerospace applications are computationally demanding and limit rapid design exploration. Convolutional Neural Networks (CNN) are being employed as surrogate models to overcome this challenge. Physics-informed approaches have also been applied to CNNs albeit only for simple flow fields such as laminar flow and heat conduction. This study advances Physics-Informed Convolutional Neural Networks (PICNN) to solve the steady incompressible Reynolds-Averaged Navier-Stokes (RANS) equations in wall-bounded geometries. The proposed method employs a higher-order finite difference scheme for computing spatial gradients, thus enhancing numerical accuracy. Additionally, the Dirichlet boundary conditions are strongly enforced in the network architecture using custom output layers and boundary masks. Numerical stabilisation is incorporated to enable the CNN to simulate high Reynolds number flows without losing stability. To assess the capabilities of this approach on aerospace use cases, it is tested on three data-free cases: S-shaped duct, a ducted body force heat exchanger, and flow over a forward facing step along with a backward facing step geometry with sparse labelled data. Moreover, a comparison between zero-equation and one-equation turbulence models is presented when employed in this framework. The RANS-CNN models performed with over 95 % accuracy on geometries with attached flow and 80 % on separated flow cases. The results obtained from the case studies confirm the capability of the RANS-CNN method in developing a robust and computationally efficient surrogate model with sparse data for smooth ducts.

1. Introduction

Presently, electrification of aircraft propulsion systems is being examined as a sustainable alternative to combustion engines. Propulsion topologies primarily powered by hydrogen fuel-cells and batteries are sensitive with regards to operational temperatures and temperature gradients. With fuel cells operating at approximately 50 % efficiency, managing heat rejection poses a significant challenge for the Thermal Management System (TMS). Thus, the design of the cooling ducts and heat exchangers to efficiently dissipate the heat is of critical importance. To design ducts for specific applications, sensitivity analyses and geometry optimisation needs to be carried out across a large design space. However, the computational cost of conducting CFD simulations acts as a bottleneck. To overcome this, surrogate models to predict fluid flow from geometry parameters are developed. By training a surrogate model on simulations and experimental data, interpolating the design space and evaluating more configurations is possible. In recent trends, various Machine Learning (ML) methods to construct surrogate models based on Artificial Neural Networks (ANN), Random Forest regression (RF), and Gaussian Processes (GP) are being put forth. Such techniques require tremendous datasets to accurately capture the design space, thus adding the cost of simulations to the cost of training. In most cases, where only sparse data or no data is available, these approaches cannot be applied. To address this challenge, physics-based deep learning methods are explored as an alternative. For a physics-informed surrogate model to be effective, it is necessary for the model to be capable of predicting fluid flow for a single geometry accurately.

In this paper, Section 2 outlines the literature regarding physics-informed deep learning methods and its limitations. Section 3 describes the methodology which extends the current state-of-the-art enabling them to simulate problems with high Reynolds numbers. It is followed by Section 4 in which the method is tested on four different case studies and is evaluated against reference data. In Section 5, the observations from the case studies are interpreted and the performance of this method is evaluated. Finally, future prospects in expanding this research, and capability of this approach in surrogate modelling, are discussed.

★ This work is carried out in the context of Trustworthy physics Informed Ai foR Aerospace and transportation (TIARA) internal DLR project.

* Corresponding author.

E-mail addresses: gaurav.bokil@dlr.de (G. Bokil), sebastian.merbold@dlr.de (S. Merbold), stefanie.degraaf@dlr.de (S. De Graaf).

https://doi.org/10.1016/j.compfluid.2025.106946
Received 31 March 2025; Received in revised form 8 December 2025; Accepted 11 December 2025

2. State-of-the-art

When it comes to physics-informed or physics-guided approaches, Physics-Informed Neural Networks (PINN) [1] are observed to be the popular choice in solving PDEs trained on sparse experimental data. Heger et al. [2] trained a PINN to predict the flow through parameterised 3D geometries in aerothermal vehicle simulations with the help of boundary data. Wassing et al. [3] also showed the capabilities of PINN in addressing the sub- and supersonic problems by solving the compressible Euler equations. Xiao et al. [4] developed a PINN framework for solving the Reynolds-Averaged Navier-Stokes (RANS) equations for Rayleigh-Taylor turbulent mixing as an inverse problem. Pioch et al. [5] compared PINNs utilising different RANS turbulence models in simulating the backward facing step supported by sparse data. Wandel et al. [6] took the help of Hermite-spline interpolation in an attempt to further improve the PINN technique. Eivazi et al. [7] demonstrated the use of PINNs for solving RANS equations, while Ghosh et al. [8] expanded on this approach by proposing a data-driven surrogate model that integrates PINNs to handle flows at different Reynolds numbers. Despite the rapid development, PINNs are not easily scalable for surrogate modelling as they use fully connected networks and automatic differentiation. On the other hand, Convolutional Neural Network (CNN) are known to require less parameters to learn representations.

Recently, CNNs have shown high potential in predicting fluid flows as data-driven surrogates. They have been applied to various applications such as airfoils, flow through blood vessels, seismic modelling, heat exchangers and vehicles [9–12]. A comparison of CNN with ANN, GP, and Radial Basis Function (RBF) methods for revealed that the CNN achieved greater generalisability than other methods [13]. Armas et al. [14] trained a CNN to predict the Unsteady-RANS results at each time-step given the flow at the previous time-step. Holistic reviews on various physics-informed machine learning techniques collectively estimate their advantages, limitations and trends [15–17]. Efforts are being made in incorporating physics-based constraints on CNNs. Zhang et al. [18] developed a physics-informed CNN (PICNN) for simulating Darcy flow without labelled data using finite volume discretisation scheme. Yuan et al. [19] proposed a PICNN for spatiotemporal PDEs with boundary encoding and showed its capabilities on cases such as Burgers equation and advection-diffusion equation. The gradients were computed using second order finite difference scheme spatially and up to fourth order discretisation schemes like the Runge-Kutta temporally. Zhao et al. [20] trained a U-net to predict the temperature on a heat source layout using the heat conduction equation in the loss function and hard constrained boundary conditions. It was shown by Sun et al. [21] and Berrone et al. [22] that hard enforcement of boundary conditions improve convergence and accuracy of CNNs trained without labelled data. Zhang et al. and Bokil et al. observed that supporting a data-driven model with physics-guided constraints results in a more accurate surrogate model [23,24]. Grimm et al. [25] extensively studied the facets of PICNN and demonstrated the effects of higher-order spatial gradients, treatment of boundary conditions, and pressure stabilisation on the predictions.

Despite the research on PICNNs, it is still limited to simple partial differential equations such as laminar flow, porous media, heat conduction, and advection-diffusion due to numerical challenges. This work aims to overcome these limitations and develop a physics-informed CNN model to solve the steady state incompressible Reynolds-Averaged Navier-Stokes equations (RANS-CNN) with and without sparse labelled data, specifically in the context of confined duct flows.

3. Methodology

The process of constructing the RANS-CNN consists of six parts namely geometry representation as inputs, the CNN architecture, enforcement of boundary conditions, computation of spatial gradients, the governing partial differential equations, and lastly the physics-informed loss function for training. To simplify the PDEs, the non-dimensional version of the RANS equations is solved and is detailed in Section 3.5.

3.1. Input geometry representation

To train a CNN model, appropriate geometry representation is required as an input. Similar to the state-of-the-art, a binary mask [9–12] is used for this purpose. The coordinates of the duct walls are sampled in the range [0, 1] in Y direction and [0, 2] in X-direction. These can be either the exact coordinates or coordinates parameterised using spline control points. Considering the duct wall coordinates as a closed boundary, a binary mask depicting the value 0 outside the duct walls and 1 inside is constructed as shown in Fig. 1.

Further, the distance function (DF) representation is computed from the binary mask as a closest distance to the level set duct wall boundaries via,

$$f_{\text{distance}} = \min(d(x, \Gamma_{\text{walls}})). \tag{1}$$

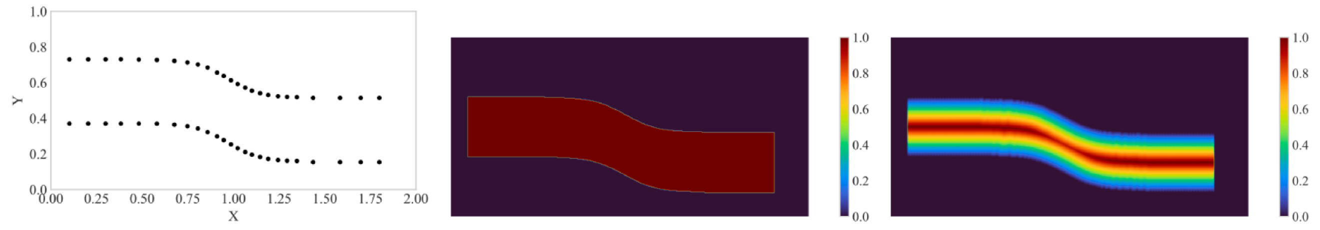
Typically, CNNs are seen to be used for external aerodynamic problems [9,10,12]. In such cases, grid points inside the obstacle (e.g., airfoil) are represented by a Signed Distance Function (SDF) with negative values inside and positive values outside the obstacle boundary. Here, since internal flow inside the duct is of interest, the external distance field is zeroed out, thus obtaining a distance function field which is sufficient to represent the geometric features of a duct.

3.2. Network architecture

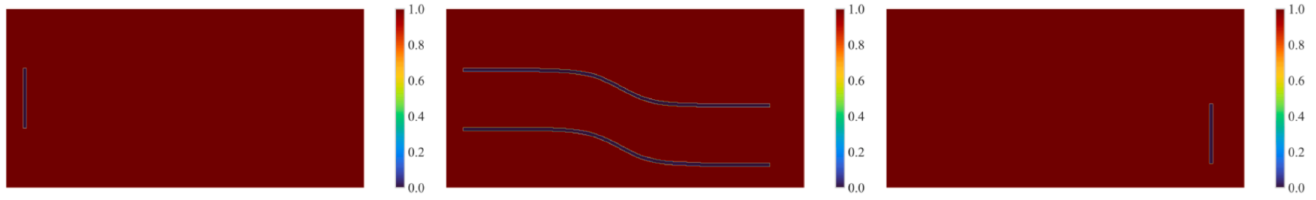
The binary mask and DF field of a chosen resolution [NY, NX] act as primary inputs to the CNN model. The U-net [26] is chosen to construct the CNN architecture. It consists of a downsampling and a symmetric upsampling part with skip connections. Skip-connections enable the information from the downsampling part of the network to be carried forward. In the state-of-the-art, the U-net has shown substantial potential in mapping geometric inputs to flow fields [12,14,24,27–29]. Here, two separate U-nets are constructed. The velocity-net predicts two fields namely the horizontal and vertical velocity components and the pressure-net predicts only the pressure field. Both the U-nets are constructed similarly. Fig. 2 outlines the architecture with filter counts and layers. Each U-net consists of four layers with filters 16-32-64-256 and stride 2 compressing the inputs. It is followed by four symmetric upsampling and convolution layers with filters 256-64-32-16 concatenated with the skip-connections from the downsampling part. A filter size of $[3, 3]$ and Rectified Linear Unit (ReLU) activation function is used, although it depends on the problem at hand. Inputs to the U-net are the binary mask and the DF field with a tensor shape of [NY, NX, 2] for a grid resolution of [NY, NX]. The two outputs of velocity-net with shape [NY, NX, 2] correspond to the X- and Y-component of velocity. The output of the pressure-net is the pressure field with shape [NY, NX, 1]. The reason for using separate U-nets lies in the fact that the distribution of velocities and pressure are different. Using separate encoders and decoders for them allows better capturing of the flow variables [11]. The complete CNN with two such U-nets consists of 1,891,971 trainable parameters.

3.3. Dirichlet boundary conditions

Contrary to the common procedure in the state-of-the-art, where the CNN predictions are directly fed to the physics-informed loss function, a more robust approach is sought. Here, the Dirichlet boundary conditions of the problem are strongly enforced onto the U-net outputs via a customised output layer. This is achieved through supplementary inputs to the CNN. These inputs include a wall mask, an inlet mask and an outlet mask. Each of these masks are essentially binary fields with value zero at the wall boundaries, inlet boundary, and outlet boundary respectively, and one everywhere else. First, the coordinates of the respective



(a) Duct coordinates (b) Binary mask (c) DF field



(d) Inlet mask (e) Wall mask (f) Outlet mask

Fig. 1. Geometry representation from duct coordinates and supplementary boundary masks for enforcing boundary conditions.

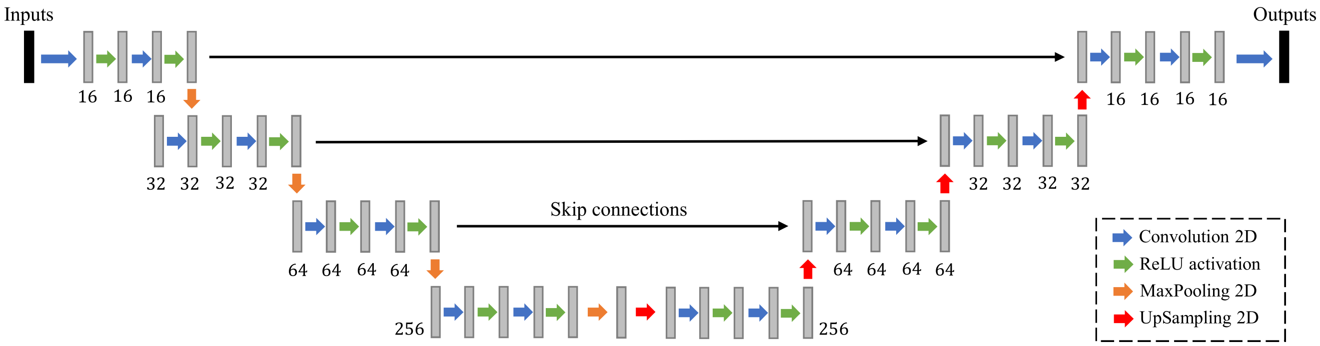


Fig. 2. The U-net architecture for velocity and pressure prediction with filters of size [3, 3] and depicted layer operations.

boundaries are sampled on the structured grid. Using the nearest neighbours functionality from SciPy-KDTree, the grid points corresponding to that boundary are isolated and a mask is constructed. The supplementary mask inputs are shown in Fig. 1(d–f). The time to construct the CNN inputs and supplementary masks is observed as 48.6 ms ± 440 µs and is thus negligible. To apply the no-slip boundary condition on duct walls, the wall boundary mask (M_{W}) is applied to the velocity prediction from the velocity U-net as,

$$u_1 = u_{\mathrm{CNN}} \cdot M_{\mathrm{W}}, \tag{2}$$

$$v_1 = v_{\mathrm{CNN}} \cdot M_{\mathrm{W}}. \tag{3}$$

As a result, the velocities at the walls become zero. Similarly, the outlet boundary mask (M_{O}) is used to set the zero pressure boundary condition via,

$$p = p_{\mathrm{CNN}} \cdot M_{\mathrm{O}}. \tag{4}$$

For the additional inlet velocity BC, since the inlet velocity is the reference velocity, a value of 1 is set using the inlet mask (M_{I}),

$$u = u_1 \cdot M_{\mathrm{I}} + 1 \cdot (1 - M_{\mathrm{I}}), \tag{5}$$

$$v = v_1 \cdot M_{\mathrm{I}}. \tag{6}$$

Since all the Dirichlet BCs are “exactly” imposed on the network, they are inherently satisfied and need not be included in the physics-informed loss function.

This approach not only reduces the computational effort in evaluating the loss function, but also inherently improves the convergence to the solution. Moreover, any discrepancies emerging from errors in weakly imposed boundary conditions can be avoided [21,22].

3.4. *Spatial gradients*

To calculate the physics-informed losses, the spatial gradients of the boundary encoded predictions (u, v, p from Eqs. 2–6) need to be computed. The finite difference scheme is employed for this purpose. Various options such as forward, backward or central difference schemes with different orders of accuracy can be exploited. Majority of the literature on PICNNs use either first or second order schemes. Instead, a higher-order central difference scheme is employed to achieve sufficient accuracy in solving the RANS equations. The effect of using such a scheme is detailed in Section 4.1.

For example, the sixth-order gradient in the X-direction of a field ϕ at every point in the uniform grid is computed as,

$$\frac{\partial \phi}{\partial x_i} = \frac{1}{h} \cdot \left(\frac{-1}{60}\phi_{i-3} + \frac{3}{20}\phi_{i-2} + \frac{-3}{4}\phi_{i-1} + \frac{3}{4}\phi_{i+1} + \frac{3}{20}\phi_{i+2} + \frac{1}{60}\phi_{i+3} \right), \tag{7}$$

where i is the point where the gradient is computed. Additionally, the second derivative is computed as,

$$\frac{\partial^2 \phi}{\partial x_i^2} = \frac{1}{h^2} \cdot \left(\frac{1}{90}\phi_{i-3} + \frac{-3}{20}\phi_{i-2} + \frac{3}{2}\phi_{i-1} + \frac{-49}{18}\phi_i + \frac{3}{2}\phi_{i+1} + \frac{-3}{20}\phi_{i+2} + \frac{1}{90}\phi_{i+3} \right). \tag{8}$$

Here, h denotes the constant grid size and i denotes the points in X-direction where $2 < i <$ NX -2. For calculating gradients in the Y-direction, the same equations are applied on points j where $2 < j <$ NY-2. Employing sixth-order gradients leaves out three rows and three columns of grid points at the borders of the image. To compute the gradients near image boundaries, second-order central difference scheme is used. Finally, the gradient at the image boundary is calculated using a first order forward and backward difference. Consequently, the first and second derivatives of velocity components $\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 v}{\partial x^2}, \frac{\partial^2 v}{\partial y^2}\right)$ and first derivatives of pressure $\left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}\right)$ are calculated.

3.5. Governing equations

The RANS equations to be solved are non-dimensionalised into velocity $u^* = \frac{u}{U_{\text{in}}}$ and $v^* = \frac{v}{U_{\text{in}}}$ as well as pressure coefficient $p^* = \frac{p}{\rho U_{\text{in}}^2}$. This implies that the CNN outputs are the dimensionless variables. For brevity, the * notation is dropped for the remainder of this work. From the BC-encoded outputs and its spatial gradients, the residuals of continuity and momentum conservation equations are computed as,

$$L_{\text{con}} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \tag{9}$$

$$L_{\text{mx}} = u \cdot \frac{\partial u}{\partial x} + v \cdot \frac{\partial u}{\partial y} + \frac{1}{\rho} \cdot \frac{\partial p}{\partial x} - \nu_{eff} \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \tau_x, \tag{10}$$

$$L_{\text{my}} = u \cdot \frac{\partial v}{\partial y} + v \cdot \frac{\partial v}{\partial y} + \frac{1}{\rho} \cdot \frac{\partial p}{\partial y} - \nu_{eff} \cdot \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \tau_y, \tag{11}$$

where,

$$\tau_x = \frac{\partial \overline{u'u'}}{\partial x} + \frac{\partial \overline{u'v'}}{\partial y}, \tag{12}$$

$$\tau_y = \frac{\partial \overline{u'v'}}{\partial x} + \frac{\partial \overline{v'v'}}{\partial y}. \tag{13}$$

The density and effective viscosity of the fluid are given by ρ and ν_{eff}. The effective viscosity is a sum of kinematic and artificial viscosity for numerical stabilisation outlined in Section 3.6. The Reynolds stresses $\overline{u'u'}$, $\overline{u'v'}$, and $\overline{v'v'}$ are obtained from the Boussinesq's eddy viscosity hypothesis [30],

$$-\overline{u'u'} = \nu_T \cdot \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x}\right), \tag{14}$$

$$-\overline{v'v'} = \nu_T \cdot \left(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y}\right), \tag{15}$$

$$-\overline{u'v'} = \nu_T \cdot \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right). \tag{16}$$

Using a zero-equation mixing-length (l_{m}) based turbulence model [31], the eddy viscosity is obtained as

$$\nu_T = {l_{\text{m}}}^2 \cdot \sqrt{S}, \tag{17}$$

where the strain rate magnitude

$$S = 2 \cdot \frac{\partial u}{\partial x}^2 + 2 \cdot \frac{\partial v}{\partial y}^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2, \tag{18}$$

and the mixing length is defined as,

$$l_{\text{m}} = \min(0.41 \cdot d, 0.07 \cdot d_{\text{max}}). \tag{19}$$

The mixing-length is a function of the distance from the wall boundary (d) which is basically the DF-field. Along with the governing equations, the gradient-based Neumann BCs are also computed as,

$$L_{\text{nbc}} = \left.\frac{\partial u}{\partial x}\right|_{\text{outlet}}^2 + \left.\frac{\partial v}{\partial x}\right|_{\text{outlet}}^2 \tag{20}$$

3.6. Numerical stabilisation

It is known that higher-order gradient methods are susceptible to oscillations when the discretisation is coarse. In case of solving the RANS equations using a CNN, there is indeed a limitation to the grid size. To ensure stability during training and in the results, numerical or artificial diffusion ($\tilde{\nu}$) [32] is introduced as a function of the Peclet number $Pe = |\overline{u}| \cdot h/\nu$.

$$\overline{\nu} = \beta \cdot \nu \cdot Pe, \tag{21}$$

where,

$$\beta = \begin{cases} Pe/3, & \text{if } -3 \leq Pe \leq 3 \\ sign(Pe), & \text{if } |Pe| > 3 \end{cases}. \tag{22}$$

This numerical diffusion is added to the kinematic viscosity in Eqs. 10 and 11. It is possible to use more elaborate numerical stabilisation schemes however they are out of the scope of the current work.

3.7. Loss function

The total loss function is calculated as a sum of squared residuals of the governing equations and the Neumann BCs as,

$$L_{\text{total}} = \lambda \cdot \sum L_{\text{con}}^2 + \sum L_{\text{mx}}^2 + \sum L_{\text{my}}^2 + \sum L_{\text{nbc}}. \tag{23}$$

Notably, the continuity equation error is weighted (λ) to prioritise the incompressibility constraint and therefore mass conservation. The total loss is then sent to the Adam optimiser [33] for minimisation.

Combining the aforementioned sections completes the RANS-CNN training framework. A schematic outline of the model and training process is shown in Fig. 3.

4. Results

To evaluate the performance of this method across various duct geometries, it is implemented on four case studies. The first study considers a typical S-shaped duct with experimental and simulation data from literature as reference. S-shaped ducts are prevalent in the aerospace industry primarily as air intakes for engines and as cooling ducts. This case tests the model on geometries involving curvatures, favourable pressure gradients at the wall, and convergent-divergent sections. The second case study deals with flow through a ducted heat exchanger modelled as a body force. Here, the performance of the model on ducts with friction-inducing bodies as body forces and sharp redirection to the flow is assessed. The third case considers a forward facing step problem which is a standard benchmark problem for CFD codes. This geometry tests the proposed method on adverse pressure gradients, flow separation induced by sharp edges, and secondary flows. The results are compared with Particle-Image-Velocimetry (PIV) data from literature. Finally, the fourth case study includes the backward facing step wherein sparse data of a velocity profile is provided during training. Additionally, the effect of integrating a more elaborate turbulence model is studied.

4.1. S-shaped duct

The S-shaped duct geometry from Bruns et al. [34] is chosen for this case study. Fig. 4 displays the experiment setup and the sections where pressure measurements are carried out. The Reynolds number for the inlet width and velocity is 10^6. The binary mask and distance field as inputs and the supplementary inputs of boundary masks are shown in Fig. 1. For this study, a grid resolution of [256 x 512] is chosen.

The said geometry inputs are fed to the RANS-CNN framework. The model is then trained until the variation in the loss function is less than 10^{-5}. After an empirical evaluation, it is found that $\lambda = 40$ ensured the incompressibility constraint for the chosen grid resolution. The training time for 10,000 epochs is noted as 45 minutes on 18-Core Intel Xeon W-2295. After the unsupervised training, the predictions are compared with those from the experiment conducted by Bruns et al. and CFD simulation conducted by Garbaruk et al. [35] Fig. 5a shows the pressure coefficients,

$$C_P = 2 \cdot (p - p_{\text{in}})/(\rho \cdot U_{\text{in}})^2, \tag{24}$$

at 3 sections namely near upper wall (U), in the middle (M), and near bottom wall (D) of the duct as measured in Bruns et al. The streamwise pressures are plotted against X/H where H is the width of the duct at inlet.

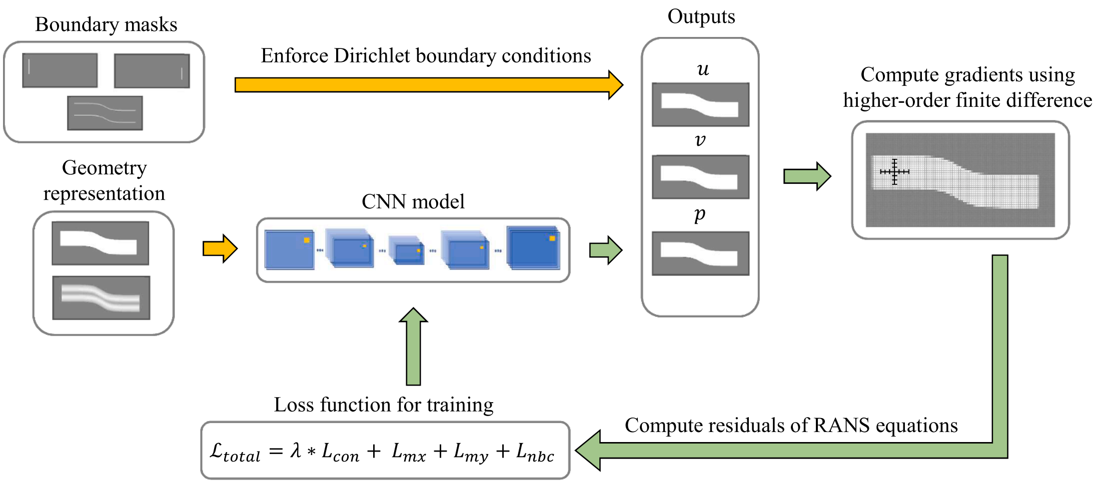


Fig. 3. Schematic description of the RANS-CNN framework.

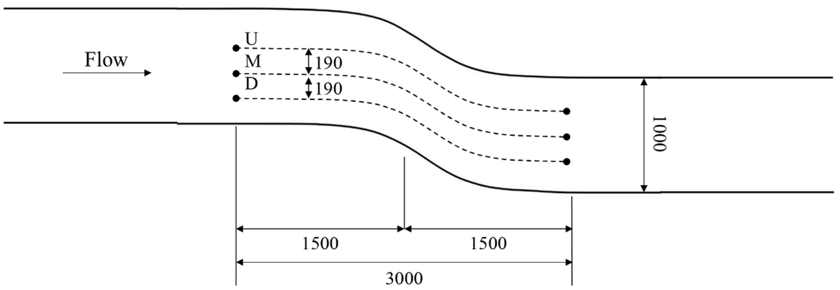
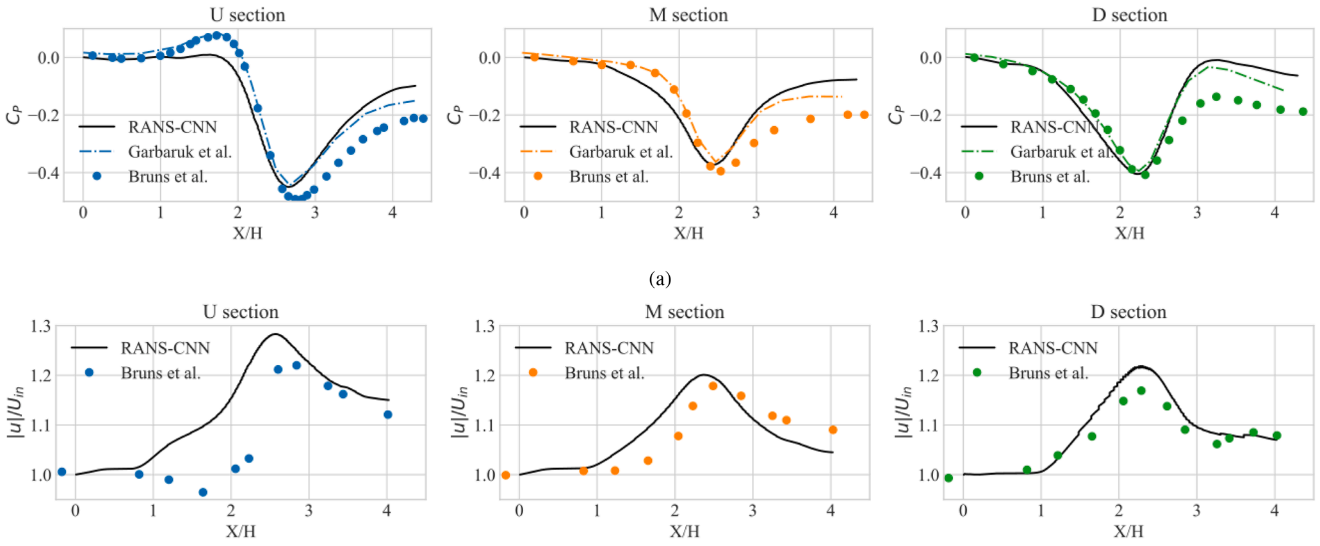


Fig. 4. Geometry based on experimental setup by Bruns et al. [34] with pressure measurement sections (U,M,D). All measurements are given in *mm*.

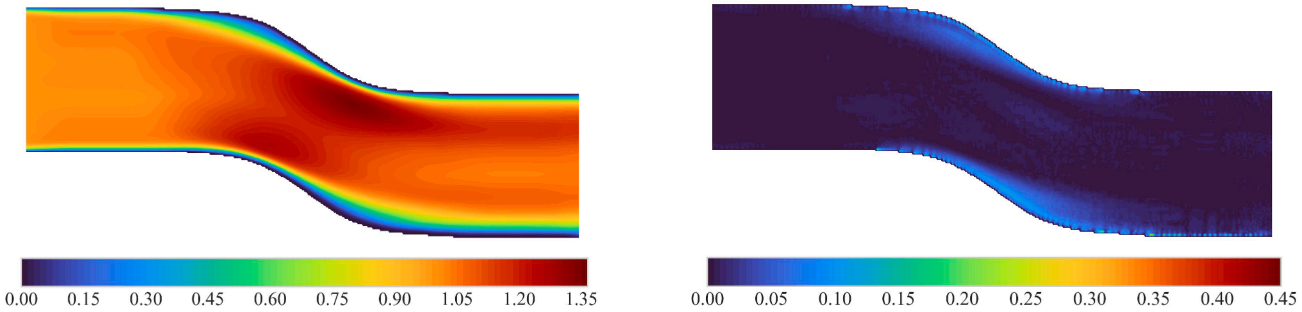


(a)

(b)

Fig. 5. Comparison of (a) pressure coefficients (C_p) and (b) dimensionless velocity magnitudes ($|\overline{u}|/U_{\mathrm{in}}$) at sections U, M, and D in the duct.

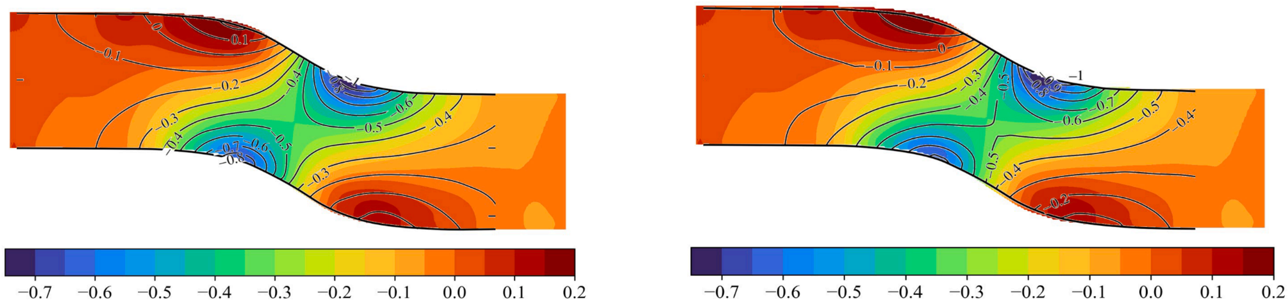
The PICNN model predicts the pressure coefficient at the sections with a mean absolute percentage error of 7 %. The solution at the D-section agrees very well with the simulation from Garbaruk et al. In case of the U and M sections, the predictions are also closer to the results from Garbaruk et al. than the experiment by Bruns et al. The same can be observed for the overall distribution of the pressure coefficient in Fig. 7. The high pressure on the walls upstream and downstream of the bend along with the low pressure zones in the centre are consistent with the reference results. Fig. 5b compares the velocity magnitude distribution along U, M, and D sections. Here, the streamwise velocity trend from RANS-CNN matches well with the experiment. However, some discrepancies are observed especially near the bend at the U-section. The velocities at the M and D sections are observed to be more accurate. These deviations are suspected to arise due the assumptions of the mixing-length model itself and the numerical errors caused by grid size. The orthogonal non-uniform computational grid used by Garbaruk et al.



(a) Dimensionless velocity magnitude ($|\overline{u}|/U_{\text{in}}$)

(b) Turbulence intensity in %

Fig. 6. Contour plots of mean velocity and turbulence intensity for the S-shaped duct.



(a) Overlaying contour with Garbaruk *et al.*

(b) Overlaying contour with Bruns *et al.*

Fig. 7. Comparison of pressure coefficients (C_p) with predictions depicted as colour contour and reference as contour lines.

consisted of 226,800 grid points. In contrast, the number of grid points which correspond to the duct area in the binary mask amount to only 37,722 points.

The velocity magnitude and turbulence intensity fields from PICNN are shown in Fig. 6. The turbulence intensity (TI) is calculated as,

$$\text{TI} = \frac{u'}{U_{\text{mean}}} = \frac{1}{||\overline{u}||} \cdot \sqrt{\frac{2}{3}k}. \tag{25}$$

The turbulent kinetic energy is given by,

$$k = \frac{1}{2} \cdot \left(\overline{u'u'} + \overline{v'v'} \right). \tag{26}$$

To see the effect of the order of discretisation scheme, the same case was simulated using second-order upwind method for calculating the gradients. Evidently, the simulation is unable to converge to the solution with this scheme. Instead, it leads to spurious oscillations in pressure and non-physical velocity field. The above result is obtained despite the inclusion of artificial diffusion. It is observed during training that the loss decreases steadily but diverges after certain epochs. This behaviour can be accounted to the accumulation of discretisation and gradient errors over iterations. This observation solidified the need of using higher-order spatial gradient schemes (Fig. 8).

4.2. Duct with body-force heat exchanger

As the popularity of electric aircraft propulsion system grows, so increases the need for efficient thermal management systems. Development of efficient cooling ducts and heat exchanger configurations is an important aspect in this area. In this context, a ducted heat exchanger (HEX) applied for cooling aircraft engines is considered [36,37]. Missirlis et al. conducted simulations for HEX with angle of attacks ranging from 0° to 60° and verified with the experiments conducted by Albanakis et al. By extracting the coordinates of the duct with a 60° angle of attack and the HEX placement, the geometry representation and supplementary boundary masks are generated similar to Fig. 1. The pressure gradient due to the HEX placed in the duct is imposed as a body force in the momentum conservation equations. Naturally, this source term is only applied on the HEX zone. To realise this in the physics-informed loss function, an additional input to the CNN had to be added. The new supplementary input is a binary mask (M_{hex}) with value one at the HEX location in the duct and zero otherwise. The source terms s_x and s_y are added to Eqs. 10 and 11 to mimic a pressure gradient at the HEX zone as,

$$L_{\text{mx}} = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \tau_x + \left(s_x \cdot M_{\text{hex}}\right), \tag{27}$$

$$L_{\text{my}} = u\frac{\partial v}{\partial y} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} - \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \tau_y + \left(s_y \cdot M_{\text{hex}}\right). \tag{28}$$

A single model is trained for Reynolds numbers of 6.2x10^4, 8.3x10^4, 10^5 and 1.25x10^5 as additional inputs at the duct inlet. This corresponds to a velocity at HEX inlet (u_x) of 3 m/s, 4 m/s, 5 m/s and 6 m/s respectively. The model is then trained on the RANS equations with the source term until convergence with a tolerance value of 10^{-4} which required 720 seconds of training time. Fig. 9 depicts the velocity magnitude, pressure coefficient and turbulence intensity for the case of $u_x = 5$ m/s.

To validate the results, the total pressure drop for all the Reynolds numbers is calculated by converting the non-dimensional predictions to dimensional quantity as,

$$\Delta P = p \cdot \rho \cdot U_{\text{in}}^2 + \frac{1}{2} \cdot \rho \cdot (|\overline{u}| \cdot U_{\text{in}})^2. \tag{29}$$

The pressure drop imposed by an HEX of length L is plotted against the bulk velocity at the HEX inlet and compared with Albanakis et al. in Fig. 10. It is observed that the RANS-CNN models the body force and predicts the total pressure drop with a mean absolute percentage error

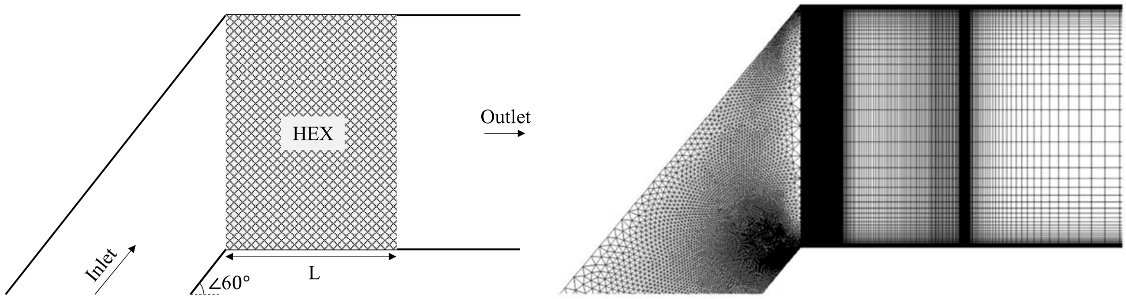
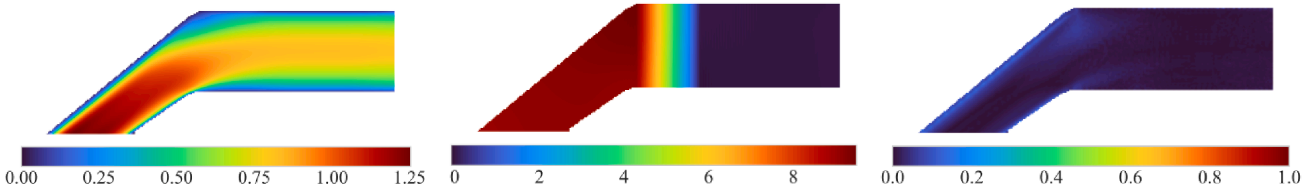


Fig. 8. Problem definition and mesh by Missirlis et al..



(a) Dimensionless velocity magnitude ($|\overline{u}|/U_{\text{in}}$)

(b) Dimensionless pressure

(c) Turbulence intensity in %

Fig. 9. Contour plots of RANS-CNN prediction for HEX with $u_x = 5\,\text{m/s}$ and $Re_{\text{in}} = 10^5$.

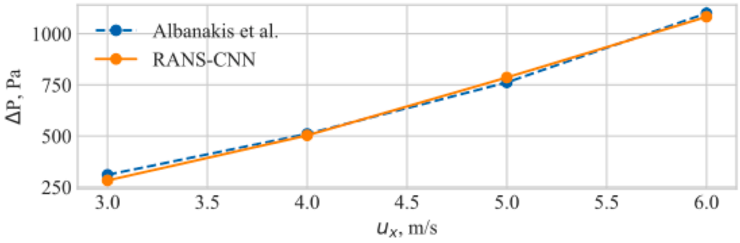
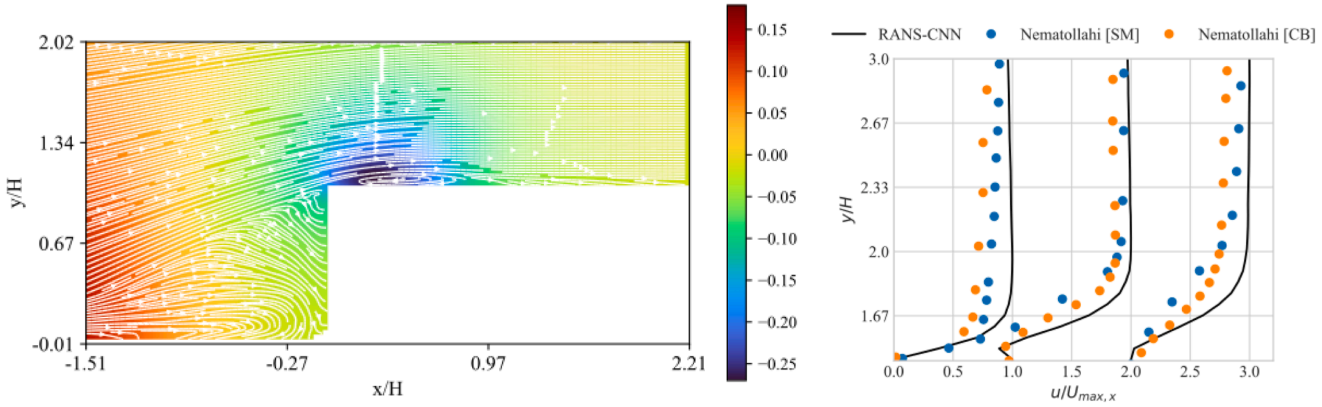


Fig. 10. Comparison of total pressure drop against HEX inlet velocity.

of only 2 %. Notably, the mesh used by Missirlis et al. for the simulation consisted of 60,000 cells while the grid points corresponding to the geometry in the [256 x 512] grid amount to only 25000. With lesser grid points the RANS-CNN is able to simulate a duct with a body force heat exchanger without losing accuracy.

4.3. Forward facing step

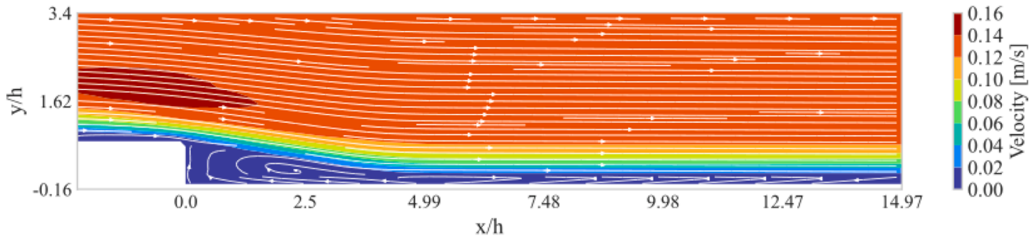
The forward facing step is one of the many benchmark problems in CFD. It is used here to test the framework on flows involving separation and recirculation. The geometry for this study is chosen from Nematollahi and Tachie [38]. Nematollahi and Tachie conducted the flow over a forward facing step experiment for smooth, sandpaper roughness, and cube roughness walls upstream of the step of height H. By following the RANS-CNN procedure, the model is trained on the said geometry for a Reynolds number of 7800 based on the step height without introducing any roughness effects. Instead of opting for the ReLU activation function as done in previous studies, the Leaky-ReLU function is used as it showed better performance. The weight on the continuity equation is tuned to a value of 3. The predictions from the RANS-CNN model are plotted in Fig. 11(a) as pressure contour and velocity streamlines. The flow passes over the step and separates, thus creating a low pressure zone. This in turn creates a recirculation bubble followed by reattachment on the step wall. This is referred to as primary circulation region. The pressure gradient $\partial p/\partial x$ tends to 0 at $x/H = 1.19$ and is considered as an indicator for reattachment. However, Nematollahi-Tachie noted the reattachment length for smooth wall as $x/H = 2.27$. The RANS-CNN prediction is not in good agreement with the smooth wall results. Instead, the reattachment length matches well with the case of cubed wall roughness wherein tiny cubes were introduced on the bottom wall upstream of the step as a rough surface. The secondary circulation region is found at the foot of the step. In case of smooth wall, a substantial secondary circulation was observed as opposed to the cube roughness wall. The secondary circulation region in the RANS-CNN, despite predicting a streamline reversal, does not completely capture the circulation bubble. Fig. 11b compares the velocity profiles of streamwise velocity at $x/L_r = 0, 0.5, 1$ in the primary circulation region. The predicted velocity field agrees closely with the experimental data. The inflection point at $x/L_r = 0$ is predicted slightly higher. Nevertheless, the presence of this inflection point indicates the prediction of onset of Kelvin-Helmholtz instabilities. The backflow observed in the profile at $x/L_r = 0.5$ is captured accurately by the CNN. Despite the differences in the reattachment lengths and heights, the ratios L_r/H_r and l_r/h_r are still in good agreement. Table 1 compares the reattachment lengths along with the height of the separation bubble for the primary and secondary circulations. Quantitatively, the RANS-CNN model predicts the velocity profiles with a mean relative error of 10 % with the rough wall while the same for smooth wall is noted to be 40 %. The reattachment lengths are estimated with 50 % relative error with respect to smooth wall and 96 % with respect to rough wall. This points out the challenges of the current implementation. Scheit et al. [39] noted the reattachment lengths from various experiments and direct numerical simulations and showed the range of variations that were observed. Cases involving separations are highly sensitive to the inflow boundary conditions, height of the boundary layer, and ratio of the step height to the channel height. Thus, the discrepancies observed can be either due to the simple turbulence model



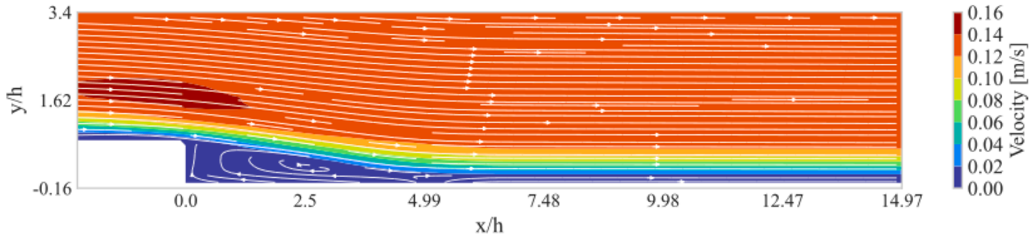
(a) Dimensionless pressure contour overlaid with velocity streamlines

(b) Comparison of velocity profiles downstream of the step

Fig. 11. Flow field at the step and comparison of velocity profiles with smooth (SM) and cube roughness (CB) wall experiments.



(a) RANS-CNN with mixing-length turbulence model



(b) RANS-CNN with Spalart-Allmaras turbulence model

Fig. 12. Velocity magnitude contour overlaid with streamlines for different turbulence models.

Table 1
Reattachment lengths and heights of primary (L_r, H_r) and secondary (l_r, h_r) recirculation regions.

Results	L_r/H	l_r/H	H_r/H	h_r/H	L_r/H_r	l_r/h_r
RANS-CNN	1.19	0.78	0.17	0.57	7	1.37
Nematollahi-Tachie [CB]	1.23	0.76	0.19	0.58	6.47	1.31
Nematollahi-Tachie [SM]	2.27	0.74	0.34	0.54	6.67	1.37

or due to the inflow boundary conditions. Here, increasing the grid size to [384, 768] was also investigated but yielded negligible improvement.

4.4. Data-guided backward facing step

In this case study, a backward facing step from the Direct Numerical Simulation (DNS) study by Le et al. [40] is considered with a step height Reynolds number of $\mathrm{Re}_h = 5100$ and an expansion ratio of 1.20. To test the method in presence of data, the streamwise velocity profile at x/h = 4 is provided as labels in addition to the RANS-based loss. A CNN is first trained with the mixing-length turbulence model, similar to previous cases. Additionally, the method is extended to incorporate the Spalart-Allmaras [41] turbulence model. Along with U-nets for u^*, v^*, and p^*, a third U-net is trained to predict the SA viscosity $\tilde{\nu}$ and the residual of the turbulent transport equation (Eq. 30) is added to the existing mass and momentum conservation losses (Eq. 23).

$$L_{SA} = \nabla \cdot (u\tilde{\nu}) - c_{b1} S\tilde{\nu} - \frac{1}{\rho}\left[\nabla \cdot (\nu + \tilde{\nu})\nabla\tilde{\nu} + c_{b2}(\nabla\tilde{\nu})^2\right] + c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 \quad (30)$$

where,

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}; f_w = \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{(1/6)}; g = r + c_{w2}(r^6 - r);$$

$$r = \min\left(\frac{\tilde{\nu}}{S\kappa d^2}\right) \quad (31)$$

are model parameters, S is the magnitude of shear strain tensor and

$$c_{b1} = 0.1355; c_{b2} = 0.622; \kappa = 0.41; \sigma = 2/3 \quad (32)$$

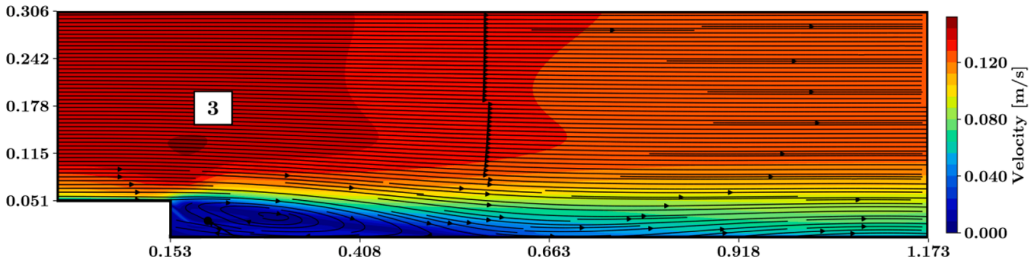
are model constants [41]. To complete the closure, eddy viscosity is computed from a damping function as,

$$\nu_T = \tilde{\nu} \cdot f_{v1} \quad (33)$$

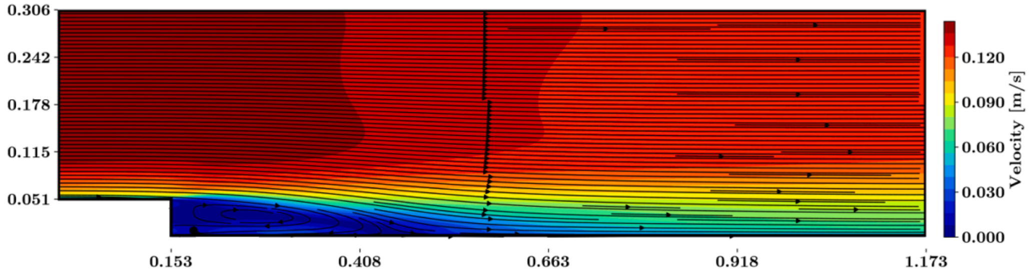
where,

$$f_{v1} = \frac{\zeta^3}{\zeta^3 + 7.1^3}; \xi = \tilde{\nu}/\nu \quad (34)$$

At wall boundaries, $\tilde{\nu}$ is set to zero as a Dirichlet boundary condition and at the outlet zero gradient Neumann boundary condition is



(a) PINN with eddy viscosity turbulence model



(b) PINN with $k - \omega$ turbulence model

Fig. 13. Figure reproduced from Pioch et al. [5] under Creative Commons Attribution (CC BY 4.0) license.

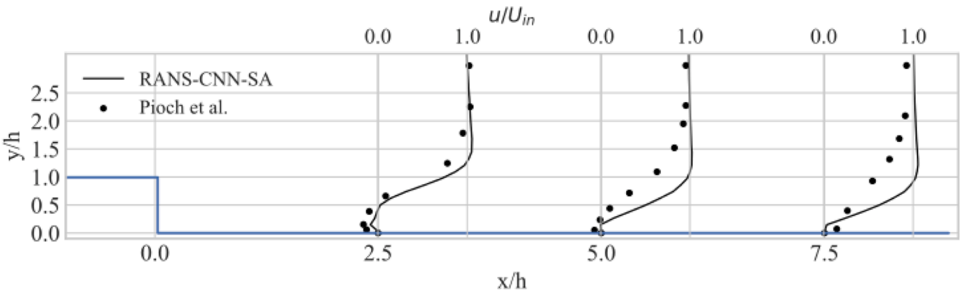


Fig. 14. Comparison of velocity profiles at x/h = 2.5, 5.0, 7.5 between RANS-CNN-SA and PINN.

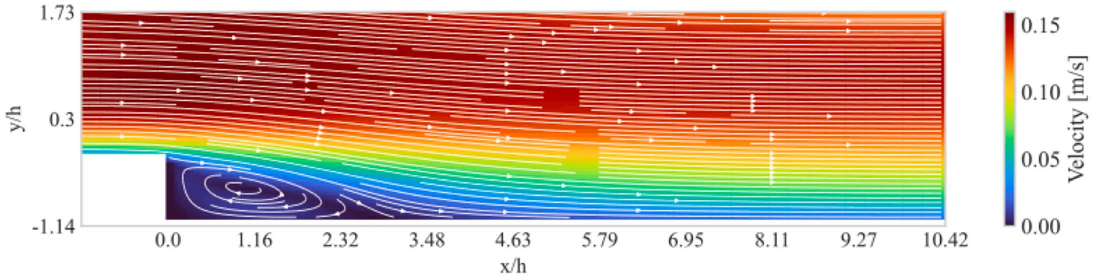


Fig. 15. Velocity magnitude and streamlines from CFD simulation.

applied. Both the RANS-CNN models, with mixing-length and SA turbulence models, are trained on this geometry with a resolution of [256 x 512] and Leaky-ReLU activation function. The results are compared with Pioch et al. wherein PINNs were trained on the backward facing step for various turbulence models with velocity profiles from Le et al. as labelled data. Fig. 12 compares the velocity magnitude distribution and streamlines from mixing-length and SA turbulence models. At a first observation, the mixing-length-based model captures the recirculation bubble albeit incomplete and fails to re-attach the flow. One the other hand, the SA-based model succeeds in predicting the circulation and reattachment.

This outcome is expected, as the zero-equation turbulence model does not account for turbulence transport but only dissipation and implicit production based on velocity gradients. The reattachment length from the RANS-CNN-SA model is observed at x/h = 5.3. This is in approximate agreement with x/h = 6.0 measured by Le et al. and the observation by [42] that the reattachment length varied between 5h and 7h for an expansion ratio between 1.1 to 1.3. Fig. 13 displays the velocity magnitude and streamlines from Pioch et al. for one-equation eddy viscosity and two-equation $k - \omega$ turbulence models among others.

By solving the same problem using PINNs, Pioch et al. were able to capture the flow behaviour with an accuracy greater than 90 %. In all the PINN models demonstrated in their work, flux through the step wall was observed. Notably, no such error is observed in the RANS-CNN approach as the boundary conditions are strongly enforced. However, adverse pressure gradient is poorly handled. The free shear layer aris-

Table 2
Quantitative summary of results.

Case study	Turbulence model	Reynolds no.	Relative % error	Runtime (s)
S-shaped duct	Mixing-length	10^6	7 %	700
Bodyforce heat exchanger	Mixing-length	$0.6 - 1.25\text{x}10^5$	2 %	720
Forward facing step	Mixing-length	7800	10 % (CB), 40 %(SM)	700
Backward facing step	Mixing-length	5100	40 %	750
Backward facing step	Spalart-Allmaras	5100	20 %	800

ing from the step is not captured and thus the flow attaches quickly to the wall downstream instead of dissipating. This behaviour, in both the RANS-CNN models, is believed to stem from low resolution background grid at the walls and numerical diffusion. Similar to Pioch et al., no corner vortices were reproduced. Fig. 14 plots the velocity profiles at three positions $x/h = 2.5$, 5.0, and 7.5 and compares with the PINN data. The model reproduces correct profile and inflection point at $x/h = 2.5$ but fails to do so downstream, as explained before. The RANS-CNN-SA model performed with a 33 % average relative percentage error, thus an accuracy of 77 %, on the velocity profiles and 91 % accuracy on the reattachment length.

In terms of computational cost, the grid points corresponding to the geometry amount to 69,290 while the PINNs by Pioch et al. were trained on 2000 collocation points. Despite the difference, the RANS-CNN required roughly 900 seconds to converge on an 18-Core Intel Xeon W-2295 CPU and PINNs took 2000 seconds on GeForce RTX 2070 GPU as per Pioch et al. To compare with CFD, a RANS simulation was carried out on the same CPU using the open source Stanford University Unstructured (SU2) code. It was conducted on a uniform unstructured grid with 70,000 mesh nodes (similar to the RANS-CNN domain points). Fig. 15 displays the velocity distribution and streamlines obtained from the CFD simulation. The reattachment length at $x/h = 3.4$ is underpredicted primarily due to the coarse grid and no inflation layers at the walls. The runtime was noted to be 800 seconds. It is evident here that the RANS-CNN training is as fast as the CFD simulation and achieves a better reattachment length prediction at the same level of resolution. Nevertheless, the overall velocity field especially downstream of the step is not reproduced accurately.

To summarise, the RANS-CNN approach was tested on several case studies including attached flows, separated flows, and a body force zone for Reynolds numbers ranging from 5100 - 10^6. Table 2 compares the quantitative results across cases with respect to flow regime, accuracy, and runtime. The cases with attached flows performed with over 93 % accuracy while those with separation performed with 60–80 %. The runtime does not vary across cases, except the backward facing step with Spalart-Allmaras turbulence model due to the additional transport equation to be solved.

5. Conclusion

This work advances the field of physics-informed convolutional neural networks (PICNN) for fluid dynamics through the following key contributions:

- the proposed RANS-CNN framework solves steady incompressible RANS equations, with and without sparse labelled data, at high Reynolds numbers,
- it incorporates mixing-length and one-equation turbulence models with scope for complex and data-driven models,
- higher-order spatial gradients with numerical stabilisation enable numerical accuracy without spurious oscillations,
- customised output layer imposes Dirichlet boundary conditions via boundary masks, improving convergence and stability.

The proposed method was tested on four different case studies. The first study dealt with flow through an S-shaped duct at a Reynolds number of 10^6 at the inlet. In this case, the model predicted the velocity and pressure distribution in the duct accurately. Especially, the pressure coefficients at three different sections in the duct strongly match those from the CFD simulations seen in the reference literature. In the second case study, an inclined body force heat exchanger is introduced in a duct. To solve this, an additional input of heat exchanger mask was provided as an input and was utilised to apply body force in momentum conservation. The trained models for different Reynolds numbers predicted the flow through the ducted HEX accurately and exhibited good agreement with the experiment data. This case study demonstrated the capability to simulate multi-zone problems with additional mask inputs. In the third and fourth case studies, geometries with separations and recirculation were considered. The forward facing step was trained without training data and the backward facing step with a single velocity profile data. Training with algebraic turbulence model captured the separation although not circulation. The backward facing step was also simulated with the one-equation Spalart-Allmaras (SA) turbulence model by adding a U-net for predicting the modified viscosity variable. With the inclusion of SA turbulence model, the CNN reconstructed the fluid flow with 80 % accuracy on average and comparable speed to the CFD simulation. The geometry representations of binary masks and distance function fields are easy to construct and thus relieves the effort of meshing the geometries. Currently, the value of divergence weight λ was determined heuristically per case and ranged from 5 to 50 times the momentum residual. Future work will deal with the implementation of adaptive and trainable weights.

Although this framework showed promising results, notable challenges were observed. In forward facing step, the results did not agree well with those from the experiments on smooth wall but instead matched with the rough wall. This implied that the flow behaved more damped than expected. In backward facing step, it was observed that this framework is sensitive to adverse pressure gradients and faced difficulties in constructing the velocity distribution downstream of the step. This calls for further investigation and testing for more sophisticated turbulence models like the k-ω SST [43]. In conclusion, the current PICNN methodology applied to RANS equations performs accurately on attached flows, but fails to reach the same performance in cases involving flow separation despite higher-order gradients and numerical stabilisation. To overcome the limitations, rather than numerical techniques, a better geometry representation free from the background grid must be incorporated.

In applications where only sparse data is available, this method would aid in reconstructing the complete fluid flow using the governing physics and measured data. An extension of this work also opens doors to 3D fluid flow modelling using 3D CNNs. The computational cost however is estimated to quadruple due to 3D convolutions and training. Prior work has been carried out in this direction, but only for low Reynolds numbers [44]. An extension to irregular geometries is possible by generating respective binary masks and boundary masks for applying Dirichlet boundary conditions. Moreover, it can also be adapted to solve different governing equations such as electromagnetics, hydrodynamics, or astrophysics. Combining this framework with a Recurrent Neural Network (RNN) can enable transient simulations with appropriate temporal discretisation. As demonstrated in Section 4.2, simulations consisting of additional components such as compressors, blockages or porous areas in the flow can also be addressed. Application of RANS-CNN to supersonic and transonic regimes involving shockwaves is also an interesting direction due to discontinuities. While standard turbulence models were employed in this work, data-driven turbulence model can also be coupled with the governing equations. With RANS-CNN as a base method, it is possible to generalise to various parametrized geometries by training a single model on multiple geometry representations at once. Additional variable inputs such as inlet velocity profile or Reynolds number can also be defined and trained for in the desired design space, which will be the focus of upcoming work.

This work is primarily carried out towards the development of cooling ducts for electric aircraft thermal management systems (TMS). The objective of this study was to develop a physics-informed CNN which can simulate turbulent fluid flow through cooling ducts at high Reynolds numbers. The results obtained from the case studies confirm the capability of achieving this goal provided the limitations are overcome. To this end, the next step is to tackle the aforementioned challenges and develop a RANS-based geometry-adaptive surrogate model. Such a surrogate model will then accelerate the design of optimal cooling ducts for electric aircraft.

CRediT authorship contribution statement

Gaurav Bokil: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization; **Sebastian Merbold:** Supervision, Project administration, Funding acquisition; **Stefanie De Graaf:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[2] Heger P, Hilger D, Full M, Hosters N. Investigation of physics-informed deep learning for the prediction of parametric, three-dimensional flow based on boundary data. Comput Fluids 2024;278:106302. https://doi.org/10.1016/j.compfluid.2024.106302

[3] Wassing S, Langer S, Bekemeyer P. Physics-informed neural networks for parametric compressible euler equations. Comput Fluids 2024;270:106164. https://doi.org/10.1016/j.compfluid.2023.106164

[4] Xiao M-J, Yu T-C, Zhang Y-S, Yong H. Physics-informed neural networks for the reynolds-averaged navier–stokes modeling of rayleigh–taylor turbulent mixing. Comput Fluids 2023;266:106025. https://doi.org/10.1016/j.compfluid.2023.106025

[5] Pioch F, Harmening JH, Müller AM, Peitzmann F-J, Schramm D, el Moctar O. Turbulence modeling for physics-informed neural networks: comparison of different RANS models for the backward-facing step flow. Fluids 2023;8(2):43. https://doi.org/10.3390/fluids8020043

[6] Wandel N, Weinmann M, Neidlin M, Klein R. Spline-PINN: approaching PDEs without data using fast, physics-informed hermite-spline CNNs. Proc AAAI Conf Artif Intell 2022;36(8):8529–38. https://doi.org/10.1609/aaai.v36i8.20830

[7] Eivazi H, Tahani M, Schlatter P, Vinuesa R. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. Physics of Fluids 2022;34(7). https://doi.org/10.1063/5.0095270

[8] Ghosh S, Chakraborty A, Brikis GO, Dey B. Rans-pinn based simulation surrogates for predicting turbulent flows. SSRN 2023; https://doi.org/10.2139/ssrn.4980383

[9] Guo X, Li W, Iorio F. Convolutional neural networks for steady flow approximation. In: Krishnapuram B, Shah M, Smola A, Aggarwal C, Shen D, Rastogi R, editors. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM. ISBN 9781450342322; 2016, p. 481–90. https://doi.org/10.1145/2939672.2939738

[10] Seo J, Yoon H-S, Kim M-I. Establishment of CNN and encoder–decoder models for the prediction of characteristics of flow and heat transfer around NACA sections. Energies 2022;15(23):9204. https://doi.org/10.3390/en15239204

[11] Bokil G, Geyer TF, Wolff S. Towards convolutional neural networks for heat exchangers in electrified aircraft. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth eV 2023; https://doi.org/10.25967/610419

[12] Thuerey N, Weissenow K, Prantl L, Hu X. Deep learning methods for reynolds-averaged navier-stokes simulations of airfoil flows. AIAA J 2020;58(1):25–36. https://doi.org/10.2514/1.j058291

[13] Wang Q, Zhou W, Yang L, Huang K. Comparison between conventional and deep learning-based surrogate models in predicting convective heat transfer performance of u-bend channels. Energy and AI 2022;8:100140. https://doi.org/10.1016/j.egyai.2022.100140

[14] Abucide-Armas A, Portal-Porras K, Fernandez-Gamiz U, Zulueta E, Teso-Fz-Betoño A. Convolutional neural network predictions for unsteady reynolds-averaged navier–stokes-based numerical simulations. J Mar Sci Eng 2023;11(2):239. https://doi.org/10.3390/jmse11020239

[15] Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. 2022. https://doi.org/10.1007/s10915-022-01939-z

[16] Faroughi SA, Pawar N, Fernandes C, Raissi M, Das S, Kalantari NK, et al. Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing. 2022. https://doi.org/10.1115/1.4064449

[17] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nature Rev Phys 2021;3(6):422–40. https://doi.org/10.1038/s42254-021-00314-5

[18] Zhang Z. A physics-informed deep convolutional neural network for simulating and predicting transient darcy flows in heterogeneous reservoirs without labeled data. Journal of Petroleum Science and Engineering 2022;211:110179. https://doi.org/10.1016/j.petrol.2022.110179

[19] Yuan B, Wang H, Heitor A, Chen X. F-PICNN: a physics-informed convolutional neural network for partial differential equations with space-time domain. J Comput Phys 2024;515:113284. https://doi.org/10.1016/j.jcp.2024.113284

[20] Zhao X, Gong Z, Zhang Y, Yao W, Chen X. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. 2021. https://doi.org/10.1016/j.engappai.2022.105516

[21] Sun L, Gao H, Pan S, Wang J-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput Methods Appl Mech Eng 2020;361(4):112732. https://doi.org/10.1016/j.cma.2019.112732

[22] Berrone S, Canuto C, Pintore M, Sukumar N. Enforcing dirichlet boundary conditions in physics-informed neural networks and variational physics-informed neural networks. Heliyon 2023;9(8):e18820. https://doi.org/10.1016/j.heliyon.2023.e18820

[23] Zhang R, Liu Y, Sun H. Physics-guided convolutional neural network (phyCNN) for data-driven seismic response modeling. 2019. https://doi.org/10.1016/j.engstruct.2020.110704

[24] Bokil GR, Geyer TF, Merbold S, Kazula S. Physics-guided convolutional neural network for flow prediction in heat exchangers in electrified aircraft. In: AIAA AVIATION FORUM AND ASCEND 2024. Reston, Virginia: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-716-0; 2024, https://doi.org/10.2514/6.2024-4108

[25] Grimm V, Heinlein A, Klawonn A. Learning the solution operator of two-dimensional incompressible navier-stokes equations using physics-aware convolutional neural networks. 2023. https://doi.org/10.48550/ARXIV.2308.02137

[26] Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. CoRR 2015;abs/1505.04597. https://doi.org/10.48550/arXiv.1505.04597

[27] Kazemi M, Takbiri-Borujeni A, Takbiri S, Kazemi A. Physics-informed data-driven model for fluid flow in porous media. Computers & Fluids 2023;264:105960. https://doi.org/10.1016/j.compfluid.2023.105960

[28] Fukami K, Fukagata K, Taira K. Super-resolution analysis via machine learning: a survey for fluid flows. Theor Comput Fluid Dyn 2023;37(4):421–44. https://doi.org/10.1007/s00162-023-00663-0

[29] Portal-Porras K, Fernandez-Gamiz U, Zulueta E, Ballesteros-Coll A, Zulueta A. Cnn-based flow control device modelling on aerodynamic airfoils. Sci Rep 2022;12(1):8205. https://doi.org/10.1038/s41598-022-12157-w

[30] Boussinesq J. Theorie de l'ecoulement tourbillant. Mem Acad Sci 1877;23:46.

[31] Hennigh O, Narasimhan S, Nabian MA, Subramaniam A, Tangsali K, Rietmann M, et al. Nvidia simnetˆTM: an ai-accelerated multi-physics simulation framework. 2020. https://doi.org/10.1007/978-3-030-77977-1_36

[32] Donea J, Huerta A. Finite Element Methods for Flow Problems. Wiley; 2003. ISBN 9780470013823. https://doi.org/10.1002/0470013826

[33] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017. https://doi.org/10.48550/arXiv.1412.6980

[34] BRUNS JM, FERNHOLZ HH, MONKEWITZ PA. An experimental investigation of a three-dimensional turbulent boundary layer in an s-shaped duct. J Fluid Mech 1999; https://doi.org/10.1017/S0022112099005522

[35] Garbaruk AV, Strelets MK, Shur ML. S-Duct CFD and exp. High Temp 2003;41(1):49–56. https://doi.org/10.1023/A:1022324620760

[36] Missirlis D, Donnerhack S, Seite O, Albanakis C, Sideridis A, Yakinthos K, et al. Numerical development of a heat transfer and pressure drop porosity model for a heat exchanger for aero engine applications. Appl Therm Eng 2010;30(11–12):1341–50. https://doi.org/10.1016/j.applthermaleng.2010.02.021

[37] Albanakis C, Yakinthos K, Kritikos K, Missirlis D, Goulas A, Storm P. The effect of heat transfer on the pressure drop through a heat exchanger for aero engine applications. Appl Therm Eng 2009;29(4):634–44. https://doi.org/10.1016/j.applthermaleng.2008.03.034

[38] Nematollahi A, Tachie MF. Time-resolved PIV measurement of influence of upstream roughness on separated and reattached turbulent flows over a forward-facing step. AIP Adv 2018;8(10). https://doi.org/10.1063/1.5063455

[39] Scheit C, Esmaeili A, Becker S. Direct numerical simulation of flow over a forward-facing step – flow structure and aeroacoustic source regions. Int J Heat Fluid Flow 2013;43:184–93. 7th International Symposium on Turbulence Heat and Mass Transfer (THMT-7), Palermo Conference on Modelling Fluid Flow (CMFF'12). https://doi.org/10.1016/j.ijheatfluidflow.2013.05.006

[40] Le H, Moin P, Kim J. Direct numerical simulation of turbulent flow over a backward-facing step. J Fluid Mech 1997;330:349–74.

[41] Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. AIAA 1992;439. https://doi.org/10.2514/6.1992-439

[42] Armaly BF, Durst F, Pereira J, Schönung B. Experimental and theoretical investigationof backward-facing step flow. J Fluid Mech 1983;127:473–96.

[43] Menter FR. Improved two-equation k-omega turbulence models for aerodynamic flows. Tech. Rep.; NASA; 1992.

[44] Wandel N, Weinmann M, Klein R. Teaching the incompressible navier–stokes equations to fast neural surrogate models in three dimensions. Phys Fluids 2021;33(4). https://doi.org/10.1063/5.0047428