Exploring the potential and the practical usability of a Machine Learning approach for improving wall friction predictions of RANS wall functions in non-equilibrium turbulent flows

Erwan Rondeaux, Adèle Poubeau, Christian Angelberger, Miguel Munoz Zuniga, Damien Aubagnac-Karkar, Roberto Paoli

Exploring the potential and the practical usability of a Machine Learning approach for improving wall friction predictions of RANS wall functions in non-equilibrium turbulent flows

Erwan Rondeaux[1], Adèle Poubeau[1*], Christian Angelberger[1], Miguel Munoz Zuniga[1], Damien Aubagnac-Karkar[1], Roberto Paoli[2]

[1*]IFP Energies nouvelles, Institut Carnot Transport Energie, 1-4 Avenue de Bois Préau, Rueil-Malmaison, 92500, France.
[2]Department of Mechanical Engineering, Polytechnique Montréal, Montréal, QC H3T 1J4, Canada.

*Corresponding author(s). E-mail(s): adele.poubeau@ifpen.fr;

Abstract

A data-driven wall function estimation approach is proposed, aimed at accounting for non-equilibrium effects in turbulent boundary layers in RANS simulations of wall bounded flows. While keeping key simplifying hypothesis of standard wall functions and their general structure, the law-of-the-wall is replaced by a fully connected feed-forward neural network. The latter is trained to infer wall friction from the local flow state at the first of-wall nodes, described by an extended set of flow variables and gradients. For this purpose, the neural network is trained on high-fidelity wall resolved simulation data. It is then applied to formulate two different wall functions trained on high-fidelity data: a backward-facing step and a round jet impacting a flat wall. After integration into an industrial CFD code, they are applied to perform RANS simulations of the flow configurations they were trained for, and are shown to yield a largely improved prediction of wall friction as compared to standard wall functions. Finally, key issues related to the

practical usability in RANS applications of the proposed data-driven approach are critically discussed.

Keywords: deep learning, data-driven CFD modeling, wall functions, RANS

1 Introduction

It is well known that the numerical simulation of the near-wall region of turbulent flows is a challenging problem in computational fluid dynamics (Choi and Moin, 2012; Slotnick et al, 2014). At high Reynolds numbers, the full resolution of the near-wall region requires extremely fine grids to capture the intricate flow structures and the rapid evolution of the flow variables. This in turn calls for the prohibitive computational costs associated with Direct Numerical Simulations (DNS) or Wall-Resolved Large-Eddy Simulations (WRLES). Wall-modeled LES are feasible but their cost is very high especially in the presence of complex geometries (Choi and Moin, 2012; Slotnick et al, 2014). In RANS simulations, wall functions are commonly employed to reduce the computational cost. In their standard formulation they assume that a non-dimensional velocity profile called the *law-of-the-wall* exists within boundary layers. Resulting Law-of-the-Wall Wall Functions (LWWF) use analytical relations derived from the *law-of-the-wall* to estimate the wall shear stress from a limited number of flow variables at a certain distance away from the wall, including the velocity component tangential to the wall U_t and fluid properties. The *law-of-the-wall* is based on equilibrium assumptions, and is theoretically valid only under very specific conditions (Pope, 2000). In many engineering applications however, turbulent flows exhibit non-equilibrium effects, such as non-negligible velocities in the direction normal to the wall, variable pressure gradients, wall curvature, density gradients, recirculation,

etc. The validity of LWWF for most flows of practical interest is therefore questionable and can lead to poor or even erroneous predictions of the wall shear stress.

Different methodologies were proposed in the literature to extend the range of validity of wall functions, including models taking into account different non-equilibrium effects (Wilcox, 2006; Popovac and Hanjalic, 2007; Shih et al, 2002; Ong and Chan, 2018; Joshi et al, 2019). Those wall-function strategies however lack generality, due to the difficulty to reduce complex non-linear Navier-Stokes equations in the presence of turbulence and walls to analytical expressions or to numerically simple relationships between wall shear stress and flow conditions off the wall. This is also a reason why such approaches can usually render a limited number of isolated non-equilibrium effects. Other works do not rely on an analytical expression of the wall shear stress. Instead, the later is estimated by numerically solving a simplified turbulence model on a one-dimensional sub-grid embedded within the first near-wall cell (Craft et al, 2006). Finally, adaptative approaches have been developed to overcome the constraints on the near-wall mesh discretization when applying a *law-of-the-wall* (Billard et al, 2015).

Machine Learning (ML) algorithms have received a considerable attention in the field of fluid mechanics, as they offer the potential to replace physical models for different aspects of turbulent flows by an approach based on learning from dependencies extracted from High-Fidelity (HiFi) data obtained from experiments or simulations (Brenner et al, 2019; Duraisamy et al, 2019; Brunton et al, 2020; Duraisamy, 2021). Literature reports different applications, such as turbulence model parameterization (Jiang et al, 2020), correction of modelling errors (Parish and Duraisamy, 2015; Wu et al, 2016; Wang et al, 2017; Wu et al, 2018; Matai and Durbin, 2019), development of turbulence models (Ling et al, 2016b; Weatheritt and Sandberg, 2016;

Xie et al, 2020; Zhou et al, 2019; Beck et al, 2019; Schmelzer et al, 2020; Saïdi et al, 2022; Amarloo et al, 2023), reduced-order modelling (Guo and Hesthaven, 2018, 2019; Hasegawa et al, 2020; Cheng and Zhang, 2021), discovery of partial differential equations (Rudy et al, 2017; Vaddireddy et al, 2020) super-reconstruction (Ferdian et al, 2020; Güemes et al, 2021), etc. Fewer studies were devoted to near-wall flows or to the development of wall models. One of the first applications of ML to wall flows is the work of Milano & Koumoustakos (Milano and Koumoutsakos, 2002). They used neural networks to reconstruct the near-wall flow field from wall shear stresses and pressures. Wang et al. (Wang et al, 2020) trained a model to reconstruct the near-wall velocity field from PIV measurements. One of the first attempts to develop a data-driven wall function was the work of Thieghi et al. (Tieghi et al, 2020, 2021), who focused on turbulent production and dissipation close to the wall, but did not explicitly address the prediction of wall shear stresses. Yang et al. (Yang et al, 2019) developed a data-driven wall model for LES of equilibrium flows, based on channel flow data. They integrated knowledge of the flow by defining input features based on the *law-of-the-wall*. A limitation of their approach is that it is strictly valid for flows in which the *law-of-the-wall* is valid. This is not the case in flows exhibiting non-equilibrium boundary layers and, even worse, there is no general scaling for the flow field, making it impossible to define such input features. Huang et al. (Huang et al, 2019) later extended their approach to rotating channels. Recently, Zhou et al. (Zhou et al, 2021) developed a data-driven wall model for LES of channel flows with periodic hills. They used data from WRLES and mapped local flow features to the instantaneous wall shear stresses. While their approach yielded promising results, it focused on LES and was not tested on other flow configurations and regimes.

Albeit RANS still remains the most widely used approach for simulating turbulent flows of practical interest, research on using ML to formulate improved wall functions

in this framework appears scarce. The objective of the present work is to explore the potential a ML algorithm can offer to formulate RANS wall functions yielding an improved prediction of wall shear stresses $\boldsymbol{\tau_w}$ in turbulent adiabatic non-equilibrium boundary layers. This work focuses on the prediction of wall friction only, and does not propose new models for boundary conditions of turbulent quantities such as the turbulent kinetic energy or the dissipation rate. The rationale of the reported work is to propose a comprehensive approach based on a state-of-the-art and readily available ML technique, and to critically assess its performances in RANS simulations with an industrial CFD code. The question of generalisability of the proposed approach is not an objective of the presented work, which aimed at performing a first exploration of such an alternative to wall laws in RANS. Instead, key issues related to the practical usability of such approaches are critically discussed, indicating possible directions for future research in the domain.

First, Section 2 exposes the formulation of a Machine-Learning Wall Function (MLWF) approach for RANS. Hereby, the dependency of wall shear stresses from the mean flow conditions at the first off-wall nodes is not supposed anymore to follow the *law-of-the-wall*, but is predicted based on a neural network that is trained on HiFi flow data for specific near-wall flows. The approach is formulated in order to relax some of the major limitations of the LWWF approach, while keeping the same overall structure, thus facilitating its integration into a CFD code. Two different flow configurations are selected to train specific MLWF, as well as to serve as validation test cases for assessing the performances of the obtained RANS approach. They are selected based on the fact that they exhibit non-equilibrium boundary layers a priori not compatible with using standard LWWF, and that data from HiFi simulations were available. Section 3 presents the two configurations (a backward facing step and a round jet impinging on a flat plate), details the training of two specific MLWF, and

shows first *a priori* evaluations of their performances. The resulting specific MLWF are then integrated into an industrial CFD solver, and their *a posteriori* performances in RANS simulations of the two configurations are presented and discussed in Section 4, including key issues related to their practical usage. Section 5 wraps up the main achievements, and outlines concluding remarks and research perspectives.

2 Formulation of the MLWF approach

As with any wall function, the main assumption behind the formulation of a machine-learning wall function is the existence of a universal relationship between the wall shear stress and flow variables taken away from the wall, even in flows affected by non-equilibrium effects. The classic law-of-the-wall represents a limiting case of this universal relationship, and a machine learning algorithm could assimilate an extended relationship and account for non-equilibrium effects.

The construction of a specific MLWF starts with the selection of a HiFi dataset from which to extract knowledge of the relationships between flow variables at distances from the wall and the local friction velocity u_τ (see Fig. 1a). As will be detailed hereafter, this is achieved by training a ML algorithm to infer the latter from the former. This is performed off-line from the CFD code. The trained ML algorithm can then be integrated into a CFD code and used on-line, i.e. during CFD simulations, to predict the local friction velocity from the flow variables known at the first off-wall nodes p located at normal distances h from the wall, as illustrated in Fig. 1b. Finally, the wall shear stress $\boldsymbol{\tau_w}$ at wall cell faces or nodes are then injected into the time advancement of the solver. The treatment of turbulent quantities is not the focus of this work, and a classical approach, similar to that of the LWWF is used in the present article. Naturally, the off-line training only has to be performed once for a given HiFi dataset before being applied on-line in as much RANS simulations as required.

2.1 Selection of the input variables

The objective of the MLWF approach is to relax the limitations of LWWF approaches. They assume the validity of the *law-of-the-wall* that is strictly only valid under the equilibrium condition stating that the near-wall flow is statistically one-dimensional and steady, and that only the tangential flow velocity varies solely along the wall normal direction. As a result, to predict $\boldsymbol{\tau_w}$, LWWF approaches rely on a limited number of input variables at the first off-wall node:

$$\{h,\ U_t^p,\ \nu,\ k\} \tag{1}$$

with from left to right the wall normal distance, the tangential velocity, the kinematic viscosity and the turbulent kinetic energy. The MLWF approach proposes retaining the hypothesis that the flow state at the first off-wall node can be used to estimate the wall shear stress. However, in order to account for non-equilibrium effects, additional input variables should be considered. In addition to the tangential velocity and turbulent kinetic energy, the MLWF proposes to rely on an extended set of 12 flow variables $\boldsymbol{I^p}$:

$$\boldsymbol{I^p} = \left\{h^*,\ U_i^*,\ \left(\frac{\partial U_i}{\partial x_j}\right)^*,\ \left(\frac{\partial P}{\partial x_i}\right)^*,\ k^*,\ \left(\frac{\partial k}{\partial x_i}\right)^*\right\}, \tag{2}$$

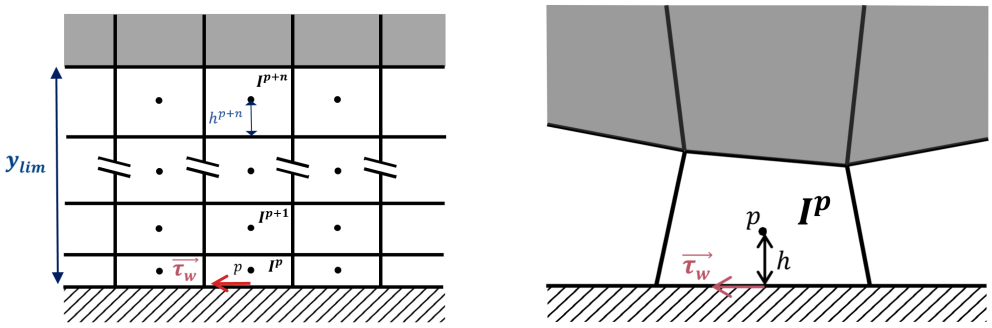
$$i = t, n, \quad j = t, n \tag{3}$$

where P is the static pressure, and the subscripts t, n stand respectively for the local tangential and normal direction to the wall. In the presence of non-equilibrium effects, the velocity normal to the wall may be non-negligible, and the boundary layer may exhibit strong spatial variations in the normal and tangential directions, as well as pressure gradients. For these reasons, the velocity normal to the wall was considered as an input, as well as the velocity and pressure gradients. As for the gradients of turbulent kinetic energy, they represent an attempt to account for the contribution

of the gradients of Reynolds stress appearing in the Navier momentum equation. It should be noted that the viscosity was not included to $\boldsymbol{I^p}$ in the present study, which necessarily limits the generalisability of the MLWF to other flows. Indeed, the two test cases considered here (presented in Sec. 3.1) have a quasi-uniform viscosity, meaning that as an input variable, it would be ignored by the neural network during training. The set $\boldsymbol{I^p}$ represents a first attempt, which could be improved in future work. It has the advantage to take into account only variables that are readily available in any CFD code, without requiring additional computational cost. Finally, sensitivity analyses were performed on the inputs (Sobol and SHAP), and they did not allow to remove any variable from $\boldsymbol{I^p}$.

As proposed in (Yang et al, 2019; Ling et al, 2016a,b), two treatments are applied to the flow variables to ensure Galilean and rotational invariance properties to the ML algorithm. First, they are projected in a local coordinate system defined by the wall normal and tangential directions. They are also normalized by a reference velocity U_{ref}, length scale y_{ref} and density ρ_{ref}, denoted by the subscript (*). The combination of those treatments ensures that the resulting MLWF will be agnostic to the case setup and functionnally applicable in a CFD solver. It does not, however, ensure that the yielded inferences are physically meaningful and generalizable to any flow configuration. Care must therefore be taken not to use a specific MLWF for flows or conditions it has not been specifically trained for, as it will then most probably yield physically erroneous inferences. Note that the definition of the reference scales may not be trivial in complex flows. In the present work, only configurations for which the reference scales can be easily defined are considered.

The retained set of 12 input variables should only be viewed as a first attempt to qualitatively include variables and their spatial gradients that can indicate the



(a) (b)

Fig. 1: Use of near-wall data for training and prediction: a) Input-output training datasets $(\boldsymbol{I^p}, u_\tau), (\boldsymbol{I^{p+1}}, u_\tau), .., (\boldsymbol{I^{p+n}}, u_\tau)$, extracted at a given longitudinal position in the flow from HiFi simulations, with $u_\tau = \sqrt{\tau_w \rho}$; b) on-line prediction of τ_w in a RANS simulation, based on the flow variables $\boldsymbol{I^p}$ at the first off-wall node p

presence of non-equilibrium phenomena in the near-wall region. In this sense, a non-negligible local normal velocity or tangential velocity gradients are viewed as characterizing departures from the hypothesis of purely tangential flow, and the possible existence of recirculations. Pressure gradients are retained as indicators for departures from a purely tangential pressure gradient resulting from wall friction, and for the presence of flow accelerations or decelerations and recirculations. Turbulent kinetic energy levels and gradients are used to indicate the existence of turbulence not only produced in the boundary layer but advected to the wall from the external flow. As for LWWF, the hypothesis of a steady-state mean near-wall flow is kept in MLWF and no attempt was made to consider time as an input variable or parameter. A detailed research work would be required to ascertain these hypotheses, and to ensure that the retained set of MLWF input variables is indeed an adapted metric accounting for non-equilibrium effects. This was not attempted in the reported work and will have to be the subject of future research.

2.2 Mapping the mean flow to the friction velocity via Machine Learning

The principle of the MLWF approach is to map the friction velocity to the local mean flow state at off-wall nodes represented by the 12 input features. To this purpose, a supervised ML algorithm is developed to extract this mapping from the near-wall flow data obtained from HiFi simulations. Training is performed *off-line* the CFD code, using input-output pairs extracted from HiFi data. In order to allow a physically reliable training, the latter must be based on a detailed resolution of near-wall flow features of specific flows, including for example flow detachments, curved walls or any other feature the resulting MLWF is supposed to be able to render. This can be provided by DNS or WRLES. The training database is built up from data points situated below a distance y_{lim} from the wall, as shown in Fig. 1a. The value of y_{lim} is chosen in order for the resulting training data to include sufficient data points in the region situated between the wall and a normal distance corresponding to twice the maximum normal distance of the first off-wall nodes in typical coarse RANS meshes. Consequently, y_{lim} may vary with the case considered (see details in Sec. 3.1). As illustrated in Fig. 1a, for a given longitudinal position, the wall shear stress $\boldsymbol{\tau_w}$ is associated to all cells normal to the wall. This means that for this position, $n+1$ input-output pairs are considered for training: $(\boldsymbol{I^p}, u_\tau), (\boldsymbol{I^{p+1}}, u_\tau), .., (\boldsymbol{I^{p+n}}, u_\tau)$, where n is the number of cells within y_{lim}.

Fully connected feed-forward neural networks are used in the present work. This type of neural-network includes an input and an output layer connected via several hidden-layers composed of multiple neurons. A neuron performs a weighted sum of its inputs and feeds it to a non-linear activation function. The output of each neuron in a layer serves as input for the neurons of the next one. The first layer $l = 0$ comprises the input features gathered in $\mathbf{A}^0$, then for $l > 0$, the output of all the neurons in the

layer l is denoted $\mathbf{A}^l$ and given recursively by:

$$\mathbf{Z}^l = \mathbf{W}^l \mathbf{A}^{l-1} + \mathbf{b}^l, \tag{4}$$

$$\mathbf{A}^l = g\left(\mathbf{Z}^l\right), \tag{5}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias matrices associated with the layer, g is the activation function, and $\mathbf{A}^{l-1}$ are the outputs from the neurons of the previous layer. In the present work, the Rectified Linear Unit (ReLU) activation function is used. The training consists in optimising the weight and bias matrices to minimize the Mean Squared Error (MSE) loss function J, measuring the error between target value $\mathbf{y}$ and the value predicted by the current state of the neural network $\hat{\mathbf{y}}$:

$$J(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N_e} \sum_{i=1}^{N_e} (y_i - \hat{y_i})^2, \tag{6}$$

where N_e is the size of the data batch fed to the network during training. A *mini-batch* gradient descent is used for an effective training: the training dataset is split into small batches (a *mini-batch*) and for each *epoch*, every *mini-batch* is fed to the neural network to compute the gradient of J and update the weights, following the Adam optimization algorithm (Kingma and Ba, 2015). After a fixed number of epochs t_{ed}, the learning rate α used in the gradient descent is decreased exponentially at each *epoch* of the training. An early stopping criterion is used to reduce overfitting: during training, the loss function J is evaluated on a training set and on a validation set composed respectively of 60% and 20% of the total data. Training is ended when the validation loss stops decreasing over a predefined number of *epochs*, called *patience*. The remaining 20% of the total data is dedicated to a test dataset, which is used at the end of the training to identify potential over- or underfitting of the model. The architecture of the network (number of hidden-layer and number of neurons per layer),

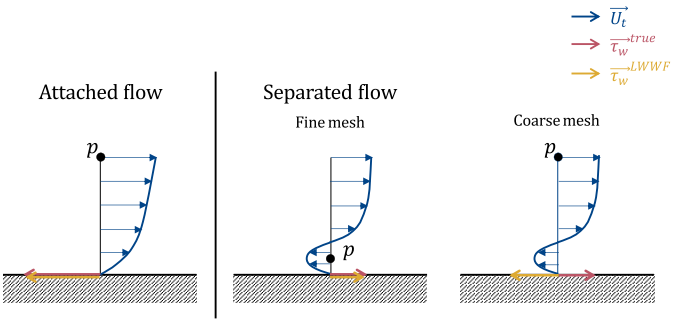


Fig. 2: Discrepancy between the orientation of the wall shear stress and the tangential velocity under the assumptions of a LWWF approach, depending on the location of the first off-wall node p in the CFD mesh.

the activation function, the *mini-batch* size, the initial learning rate α_0, the number of *epochs* t_{ed} and the *patience* are hyperparameters to be optimized (see Sec. 3.2).

2.3 Estimation of the wall shear stress vector

In a CFD code, the variable to be predicted by a wall function is the wall friction stress vector. LWWF approaches assume that it is parallel and oriented in the direction opposite to the mean tangential velocity vector at the first off-wall node U_t^p as illustrated in Fig. 2:

$$\boldsymbol{\tau_w}^{LWWF} = -\frac{\boldsymbol{U_t^p}}{\|\boldsymbol{U_t^p}\|}\rho u_\tau^2. \tag{7}$$

As shown in Fig. 2, in the presence of separation in the near-wall flow, the orientation between the wall shear stress tensor and the mean tangential velocity at the first off-wall node depends on the position of the latter: in the case of a sufficiently fine CFD mesh, both are indeed oriented in opposite directions. However, in the case of a coarse mesh with p situated outside of the recirculation zone, both are oriented in the same direction. In that case, the wall stress vector estimated by Eq. 7 would be mis-oriented. To alleviate this issue, the MLWF approach predicts a signed friction velocity such that:

$$sgn(u_\tau) = -sgn\left(\boldsymbol{\tau_w} \cdot \boldsymbol{U_t^p}\right), \tag{8}$$

where $\cdot$ denotes the scalar product. For an attached flow, where $\boldsymbol{U_t^p}$ and $\boldsymbol{\tau_w}$ are in opposite directions, the friction velocity is positive, and negative otherwise. The wall shear stress vector is then estimated as

$$\boldsymbol{\tau_w} = -\frac{\boldsymbol{U_t^p}}{\|\boldsymbol{U_t^p}\|}\rho u_\tau |u_\tau|. \tag{9}$$

Note that this process implies that the MLWF approach shares with the LWWF approach the hypothesis of alignment between the wall shear stress vector and the tangential velocity at the center of the first off-wall cell.

3 Formulation of two specific MLWFs

The developed MLWF approach is applied to formulate wall functions specific to two configurations: a first one named MLWF$_{BFS}$ trained on HiFi data available for turbulent flow over a backward-facing step (BFS), and a second one named MLWF$_{IJ}$ trained on HiFi data for a turbulent round impinging jet (IJ). These flows were selected as they exhibit strong non-equilibrium effects: in the BFS case, the flow detachment generated by the sudden expansion yields a strong recirculation zone, while the IJ case is characterized by high wall-normal velocities in the impact zone and a strong radial acceleration outside the jet.

3.1 Presentation of the reference cases

High-fidelity data for the BFS case were taken from DNS of the turbulent flow over a non-symmetrical BFS published in (Bin et al, 2022). In this configuration, a turbulent axial flow in a duct with a rectangular cross-section and a spanwise periodicity reaches a downward-oriented sudden expansion as shown in Fig. 3. The DNS database comprises time- and space-resolved flow data computed with a wall resolution of $y^+ < 1$. The simulated Reynolds number is $Re = \frac{U_{ref} y_{ref}}{\nu} = 5600$, where U_{ref} is

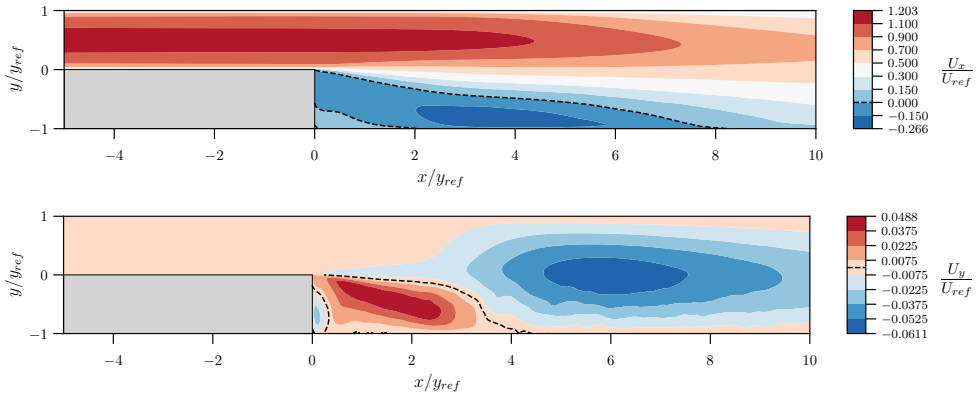


Fig. 3: Training data for MLWF$_{BFS}$: Steady mean axial and normal velocity components obtained by averaging DNS data. The black dashed line corresponds to the zero velocity contour.

the inlet bulk velocity and y_{ref} is the step height. The Reynolds number based on friction velocity at inlet and step height is $Re_\tau = 360$.

The data used for the training of MLWF$_{BFS}$ is obtained by a temporal averaging of the time-resolved DNS data followed by a spatial averaging in the spanwise direction. Figure 3 shows the obtained two-dimensional steady-state mean velocity fields. A massive recirculation region appears downstream of the step, and the average flow reattachment is located at $x/y_{ref} \sim 8.2$. Figure 4a shows the corresponding mean steady wall friction coefficient profiles along the top and bottom walls. The main recirculation is characterized by a negative C_f on the bottom wall, except immediately downstream the lower corner, where a secondary recirculation zone is present for $x/y_{ref} < 2$, with a slightly positive C_f. The top wall does not exhibit any detachment, with only a decrease in C_f resulting from the flow deceleration generated by the sudden expansion.

HiFi data for the formulation of MLWF$_{IJ}$ are based on data similar to those published in (Grenouilloux et al, 2021, 2022) for the flow resulting from injecting a single round jet normally onto a flat plate situated at a distance twice the nozzle

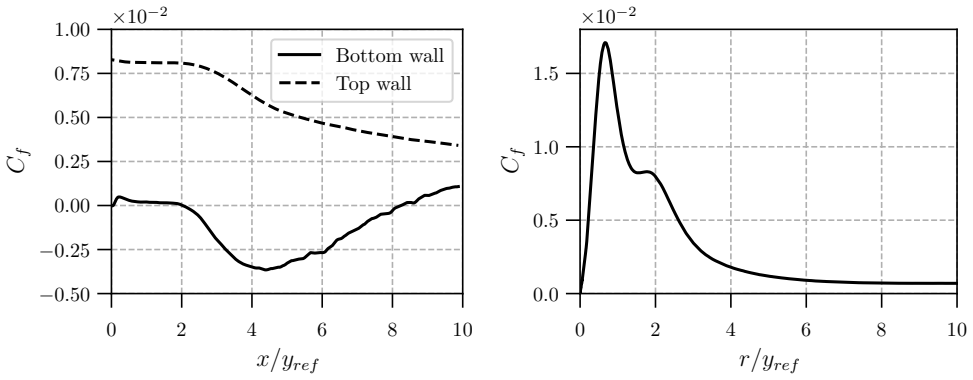


Fig. 4: Steady mean wall friction coefficient $C_f = 2\tau_w/(\rho U_{ref}^2)$ profiles obtained from averaging HiFi data: a) BFS, where the step is situated at $x/y_{ref} = 0$ - b) IJ, where the centre of the round jet is located at $r/y_{ref} = 0$.

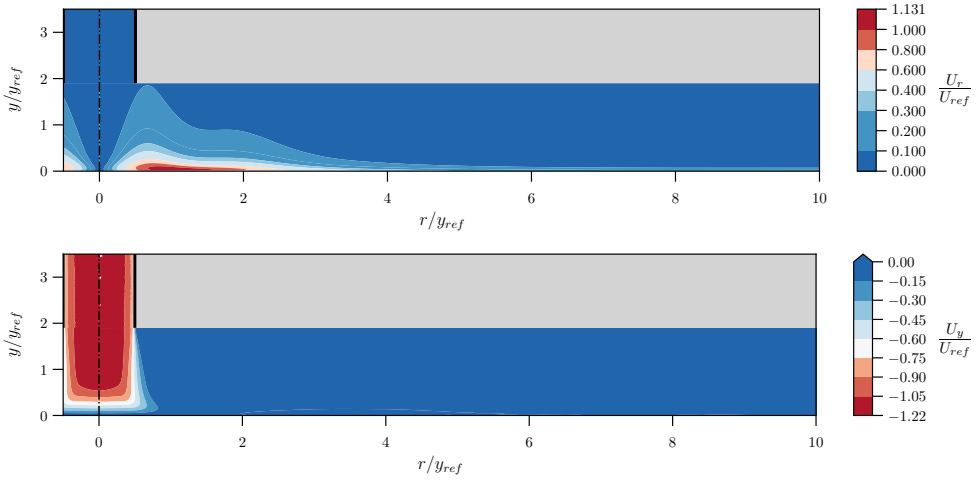


Fig. 5: Training data for MLWF$_{IJ}$: Steady mean normal and radial velocity components obtained by averaging LES data.

diameter y_{ref}. Time- and space-resolved data are available using a WRLES with a wall resolution of $y+ = 5$ in the jet impact region. The simulated Reynolds number based on bulk jet velocity U_{ref} and nozzle diameter is 23000.

Training data for MLWF_{IJ} is obtained by a temporal averaging of the instantaneous WRLES data followed by an azimuthal spatial averaging. Figure 5 shows the obtained steady mean two-dimensional velocity components in the wall normal and radial directions. The jet is deviated radially by the wall, creating a zone of high radial velocity in the region surrounding the jet. As shown in Fig. 4b, this results in C_f peaking at $r/y_{ref} = 0.9$. Far radially from the jet at $r/y_{ref} > 6$, the flow relaxes to a radially expanding boundary layer with a zero normal velocity and a power-law decay for C_f (Launder and Rodi, 1983).

Training data for each of the specific MLWF include only near-wall data from the obtained averaged 2D fields. As shown in Fig. 1a and detailed in Sec. 2.2, only data points situated below a distance y_{lim} from the wall are extracted and pre-processed, with $y_{lim} = 0.25y_{ref}$ for BFS and $y_{lim} = 0.5y_{ref}$ for IJ. The resulting typical number of datapoints along a wall normal included in the MLWF training database is between 50 and 100, which is estimated to be sufficiently large for ensuring an accurate training of the neural networks. The total number of points in the datasets are $38,627$ for BFS and $44,734$ for IJ.

No further spatial filtering of the training data aimed at mapping it from the highly resolved HiFi mesh to the coarse RANS meshes is applied. As it will be shown in Section 4.4, such a RANS pre-filtering did not yield any improvement of the predictive capability of the MLWF.

3.2 Training performances

Given the relatively limited size of the datasets, all trainings are performed on a laptop (CPU is Intel Core i7-10850H and GPU is NVIDIA Quadro T1000), in a Python environment, using the Tensorflow (Abadi et al, 2015) and Scikit-learn (Pedregosa et al, 2011) libraries. On average, training a neural network takes 400 s for BFS

and 240 s for IJ (average computed from 100 trainings). To optimize hyperparameters, systematic frameworks such as grid-search or random-search have been tested, in particular to determine the numbers of hidden-layer and of neurons per layer, but they tend to promote overly complex neural network architectures with a marginal gain in score compared to simpler architectures. For the training of the present models, excellent scores are obtained even for simple architectures (5 hidden-layers and 20 neurons per layer), and these scores are not very sensitive to the increasing complexity of the neural-network architecture. This limits the interest of these advanced hyperparameter-tuning techniques, which is why a manual trial and error procedure was used to select the final sets of hyperparameters (values are listed in Table 1), yielding the simplest architecture with the best training score, without overfitting. Table 2 gives for the two models, MLWF_{BFS} and MLWF_{IJ}, the R^2 score and the Mean Absolute Percentage Error (MAPE), computed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{10}$$

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right| \tag{11}$$

where $\mathbf{y}$ is the true value, $\overline{y}$ the average of vector $\mathbf{y}$, $\hat{\mathbf{y}}$ the value predicted by the neural network and N the size of the dataset. For training, validation and test, R^2 is higher than 0.99 and MAPE lower than 1.9% for both models, showing a good accuracy without obvious overfitting. Figure 6 depicts the evolution of the error function J, computed from the training and validation datasets, during training. Both errors globally follow a decreasing trend before stabilizing. The absence of deviation between training and validation errors also tends to rule out any overfitting.

The obtained MLWF_{BFS} and MLWF_{IJ} are first validated *a priori* by comparing their C_f predictions, when being informed with flow variables taken from the HiFi database they were respectively trained from, with the one found in HiFi data.

Hyperparameters	MLWF$_{BFS}$	MLWF$_{IJ}$
Architecture (N_l, N_n)	(5, 20)	(5, 20)
Activation func.	ReLU	ReLU
α_0, t_{ed}	(10^{-3}, 20)	(10^{-3}, 30)
Mini-batch size	8	16
Patience	25	20

Table 1: Values of hyperparameters.

Model	Metric	Dataset Training	Validation	Test
MLWF$_{BFS}$	R_2	0.9986	0.9983	0.9986
	MAPE (%)	1.11	1.20	1.10
MLWF$_{IJ}$	R_2	0.9997	0.9998	0.9997
	MAPE (%)	1.83	1.86	1.80

Table 2: Values of R^2 and MAPE for both models MLWF$_{BFS}$ and MLWF$_{IJ}$.

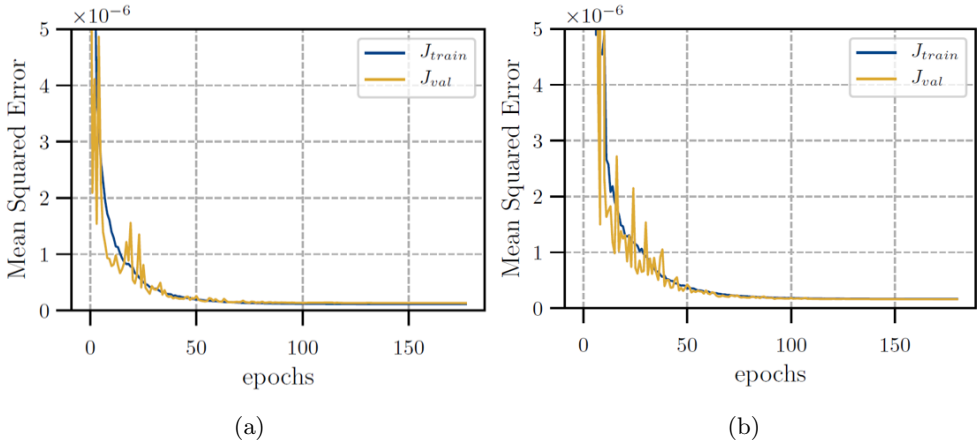


Fig. 6: Evolution of the MSE error function J on the training and validation datasets during training for a) MLWF$_{BFS}$ and b) MLWF$_{IJ}$.

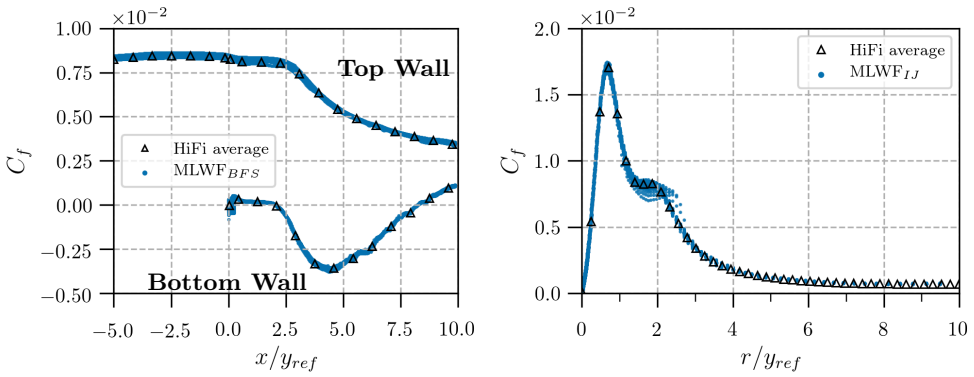
Fig. 7a shows that for BFS the evolution of C_f along the top wall is well captured, with a limited scattering depending on the local distance from the wall. This also holds for most of the bottom wall, where MLWF$_{BFS}$ yields a good reproduction of the magnitude and sign of the friction coefficient in the main recirculation zone (negative C_f) up to the reattachment point with $C_f = 0$ and beyond. This is also

verified in most of the secondary recirculation zone adjacent to the sudden expansion (positive C_f). Larger discrepancies are apparent for $x/y_{ref} < 0.3$, where MLWF$_{BFS}$ fails to precisely reproduce the magnitude and sign of C_f. The flow in this region is complex, and is only represented in the HiFi data by a relatively small number of points. As a result, the ML algorithm appears not to be able to learn with sufficient accuracy the mapping between mean flow and friction coefficient in this region.

Fig. 7b shows that MLWF$_{IJ}$ accurately captures the peak of friction coefficient of the impinging jet at $r/y_{ref} = 0.7$, as well as its evolution in the centre of the jet and in the surrounding flow for $r/y_{ref} > 3.0$. However, discrepancies are apparent in the intermediate region $1.5 < r/y_{ref} < 3$, where HiFi data exhibit a plateau with a relatively constant wall friction, while MLWF$_{IJ}$ yields a variation of level depending on the distance from the wall. The related maximum prediction error is less than 20%. As will be shown below, this relatively large error is however much smaller than the one associated with using LWWF, so that no attempt to further optimize the training and reduce it was undertaken.
In summary, both models yield an overall satisfactory, even if not fully exact, *a priori* reproduction of the learned data.

4 Application in RANS simulations

In order to evaluate the performances of MLWF$_{BFS}$ and MLWF$_{IJ}$ *a posteriori*, they are integrated into the industrial CFD code CONVERGE (Richards et al, 2022) and applied in RANS simulations of the two flow configurations for which HiFi training data was used. The resulting predictions are compared with HiFi data and RANS predictions using a LWWF (Launder and Spalding, 1974). Details about the implementation of the LWWF can be found in Appendix A.



(a) (b)

Fig. 7: Comparison of *a priori* friction coefficient predictions by MLWF, informed with HiFi flow conditions they were trained for, with HiFi findings: a) BFS - b) IJ.

4.1 Integration of a MLWF into a CFD code

A MLWF is functionally similar to a standard wall function in the sense that it predicts wall friction based on the mean flow state at the first off-wall node. As a result, its integration into a CFD code is quite straightforward. It simply consists of feeding the required flow variables as inputs, and to inject the inferred wall friction into the CFD code's method for closing the momentum and turbulence transport equations at the wall.

The trained model for a specific MLWF is imported at the start of a simulation from files containing the architecture and optimized parameters saved at the end of the off-line training. The trained MLWF is then called at each timestep and in every wall cell to infer the wall friction, instead of using the *law-of-the-wall* as in LWWF. The overall structure of the CFD solver is hereby unchanged. The import and inference of the MLWF are handled by the open-source library NNICE (Aubagnac-Karkar and Mehl, 2023), previously applied in (Mehl and Aubagnac-Karkar, 2023). This tool leverages the EIGEN library (Guennebaud et al, 2010) to efficiently infer neural networks. The

impact of the integration of the MLWF on the computational cost is detailed in the next section.

4.2 Computational set-up

RANS simulations of the two reference configurations are performed using a $k-\omega$ SST turbulence model (Menter et al, 2003). An unsteady incompressible PISO solver is used with an implicit time advancement and a centred second order scheme for spatial discretization of the convective terms. A cut-cell approach is used to handle non-conformal mesh boundaries.

The computational domains and boundary conditions reflect those reported in the HiFi simulations from which the training data was extracted. Since the mean flow in the BFS case is two-dimensional, the computational domain is taken as the central plane of the original 3D domain, as illustrated in Fig. 8a. Dirichlet conditions are used to impose the steady mean inlet velocity profiles extracted from the averaged HiFi data, and the outlet pressure. All solid walls are handled with MLWF$_{BFS}$.

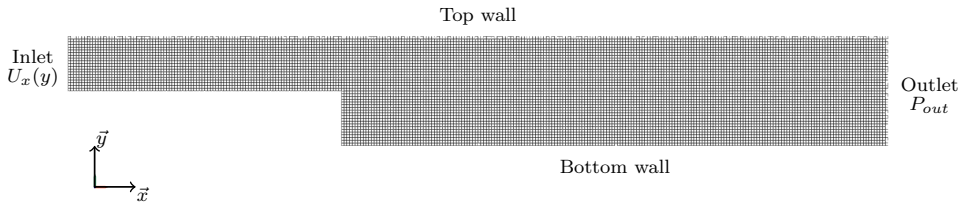
The flow in the IJ case is statistically axisymmetric, hence a 30-degrees portion is simulated using a 3D mesh, as illustrated in Fig. 8b. The lateral boundaries in the azimuthal direction are taken as periodic. The mean velocity profile at the nozzle inlet is imposed based on the HiFi data. A coflow corresponding to 5% of the nozzle bulk velocity is imposed on the top boundaries using a Dirichlet method (Grenouilloux et al, 2021, 2022). A constant pressure is imposed at the lateral outlet. MLWF$_{IJ}$ is only applied at the bottom wall, the lateral nozzle walls are handled with a classical LWWF.

For both BFS and IJ cases, the wall boundary conditions for k and ω are a zero-gradient Neumann condition and a wall function such as $\omega = \frac{u_\tau}{C_\mu^{1/2}\kappa h}$, respectively. Note that the prediction of u_τ given by the MLWF is taken into account in the calculation of ω. Two different sets of meshes are simulated. The fine M1 meshes shown in Fig. 8

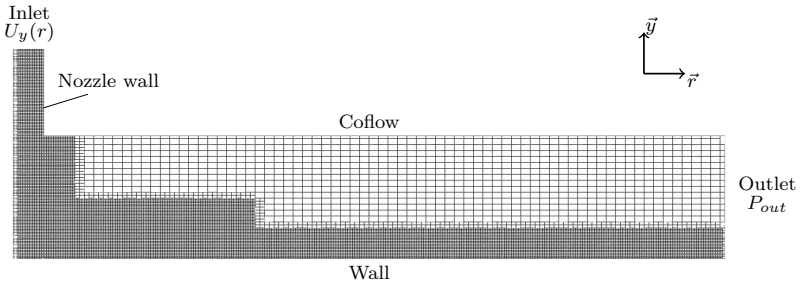
ensure an overall wall resolution of $y^+ \sim 15$ (estimated at the cell center). The coarser M2 meshes are obtained by doubling all cell sizes as compared to the M1 meshes. All meshes are structured, using rectangular (BFS) and hexahedral (IJ) cells.

All simulations start from simple initial conditions of a fluid at rest, and first use LWWF on all walls to cover a physical time of around $t = 14\,t_c$ in order to establish converged steady flow fields, where the flow-through time t_c corresponds to the time for a fluid particle to cross the entire computational domain at the bulk velocity imposed at the inlet. The so obtained converged flow fields then serve as initial conditions for the simulations with MLWF, which are run for 7 additional flow-through times to reach a final converged steady state. This procedure proved necessary as applying MLWF starting from initial conditions of a fluid at rest leads to a divergence of the simulations (not shown here). This is a direct consequence of the fact that the MLWF are trained only on the average steady state HiFi data, which ideally corresponds to the RANS prediction. Therefore, when fed with input variables lying outside of the training range, the inference becomes de facto an extrapolation which is well known to lead to large errors (see also Section 4.4). The necessity to initialize the MLWF simulation with a flow closer to the final result instead of a flow at rest doubles the computational time in the present cases. This was not a limitation for the present exploratory study, but an improvement to reduce the impact on computational time could easily be implemented in future work: the simulation could start with LWWF from a flow at rest, and as soon as the a certain level of convergence is reached, the CFD solver could automatically switch to the MLWF, until final convergence is reached. An optimization work would need to be carried out to determine the level of convergence needed before switching to the MLWF to minimize the simulation time.

The additional computational cost per iteration associated to the MLWF is found to be limited. For the IJ case, applying MLWF_{IJ} entails an additional computational time of 3.9% as compared to LWWF simulations. Approximately half of this is due to



(a)



(b)

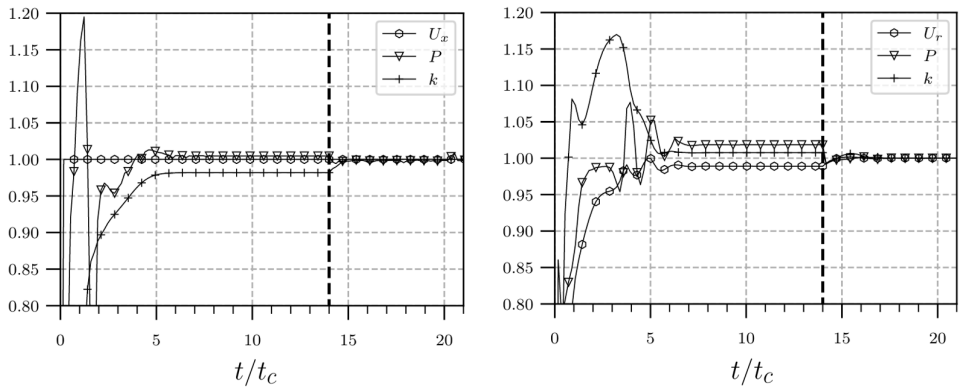
Fig. 8: Computational domain, $M1$ mesh and boundary conditions for the RANS simulations : a) BFS and b) IJ (only a central plane of the 30-degree domain in the azimuthal direction is shown.

the cost related to the inference of the model, and the other half results from additional computational time required to converge the resolution of the transport equations as compared to LWWF simulations.

4.3 Results

Figure 9 shows for both cases and the fine M1 meshes the time evolution of the major velocity component, pressure and turbulent kinetic energy spatially averaged over the whole computational domain. For the simulations with LWWF ($t \leq 14\,t_c$), they reach a steady state after approximately $7\,t_c$, ensuring that simulations using MLWF are initialized by a converged flow field. After applying MLWF (for $t \geq 14\,t_c$), global variables are slightly modified but do ultimately reach a new steady state within

$1.5\,t_c$ for the BFS case and $3\,t_c$ for the IJ case. A similar behavior was observed for the simulations with the M2 meshes.



(a) (b)

Fig. 9: Evolution of spatially averaged variables during RANS simulations on M1 mesh with LWWF ($t \leq 14\,t_c$) and MLWF ($t \geq 14\,t_c$) : a) BFS, b) IJ. The profiles are normalized by the value of the corresponding averaged variable at $t = 21\,t_c$. The vertical dashed line denotes the start of the simulation with the MLWF model at $t = 14\,t_c$.

Figure 10 shows the converged steady-state wall friction coefficient profiles predicted by RANS simulations with MLWF$_{BFS}$ and MLWF$_{IJ}$ for the respective configurations they were developed for, and compares them with averaged HiFi data as well as predictions using LWWF. For the BFS case, MLWF$_{BFS}$ strongly improves the C_f predictions as compared to LWWF. While on the top wall LWWF qualitatively renders the decrease in C_f as a result of the flow acceleration caused by the sudden expansion, it largely over-predicts the C_f magnitude by a factor between 2 and 3 for the coarse M2 mesh, compared to HiFi results. In contrast, MLWF$_{BFS}$ yields a quite satisfactory reproduction of HiFi data, and has in addition the advantage of being

remarkably less dependent on the mesh resolution. It is interesting to see that the findings are somehow less contrasted on the bottom wall where the flow is very strongly affected by non-equilibrium phenomena. LWWF predictions, especially on the fine M1 mesh, are globally close to average HiFi findings. However, a very strong discontinuity of the C_f profile is predicted by LWWF for both mesh resolutions towards the end of the main recirculation around $x/y_{\text{ref}} \approx 8$, which is not at all apparent in HiFi data. The reattachment is also very poorly predicted by LWWF. Results for MLWF$_{BFS}$ are much more accurate and less dependent on the mesh resolution, yielding a quite accurate prediction of the length of the main recirculation zone. However, it noticeably under-estimates the axial extend of the secondary recirculation close to the lower corner of the sudden expansion, characterized by positive C_f. While for MLWF$_{BFS}$ this shortcoming could be related to the above reported lack of a priori accuracy close to the corner, it is also apparent in the LWWF results, indicating that it probably is more a result of the difficulty of RANS to capture high-strain regions of the flow-field.

For the IJ case, LWWF fails to predict the radial position and over-estimates by up to 40% the peak wall friction found in averaged HiFi data. The results very close to the symmetry axis also exhibit a non-physical disruption of the C_f profile as opposed to the smooth increase seen in HiFi data. As the wall resolution of the M1 mesh is too fine for the wall-law to be valid, it exhibits higher error levels than the ones for the coarse M2 mesh. Using MLWF$_{IJ}$ strongly improves the prediction of both the location and magnitude of the C_f peak, which are much closer to HiFi findings. However, a dependency to mesh resolution is apparent: while on the fine mesh the zone with a C_f plateau apparent in the HiFi data is more accurately predicted, the magnitude of the C_f peak is under-estimated by around 17%. The coarse mesh slightly reduces this error, but is unable to render the C_f plateau.

Overall, using LWWF can lead to large overpredictions of the magnitude of wall friction, and exhibits a strong dependency on the mesh resolution. The two evaluated

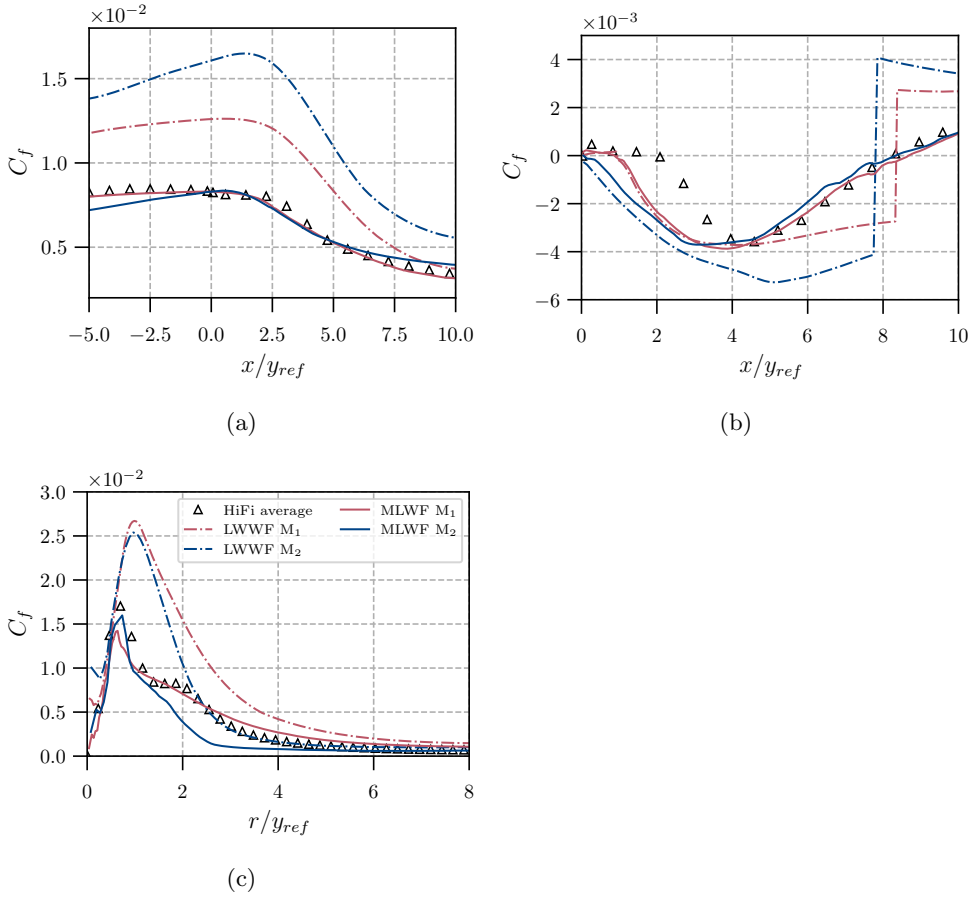


Fig. 10: RANS predictions of wall friction coefficients with specific MLWF, and comparison with LWWF findings and HiFi data : a) BFS top wall, b) BFS bottom wall, c) IJ.

specific MLWF yield generally more accurate reproductions of HiFi findings when applied to the cases they were trained for, and show a comparatively smaller mesh dependency.

Converged velocity profiles along the vertical axis for different axial positions are reported for the BFS case in Figure 11, comparing the predictions yielded by MLWF$_{BFS}$ with averaged HiFi data as well as with results using LWWF. For the sake

of conciseness, only the results for the M1 mesh are shown, those obtained with the M2 mesh being very similar. Despite a slight overprediction of the velocity peak in the jet on the upper side of the channel (for $y/y_{ref} > 0$), the simulation with MLWF$_{BFS}$ reproduces the velocity profiles reasonably well compared to the reference HiFi data. Interestingly, the velocity profiles obtained with LWWF are very similar to those with MLWF$_{BFS}$: profiles at $x/y_{ref} = 0.5$ and $x/y_{ref} = 2$ exhibit no discernible differences, and the ones at $x/y_{ref} = 3.5$ and $x/y_{ref} = 6$ are limited to the first off-wall nodes, which seems consistent with the differences in wall friction coefficients between both approaches.

Figure 12 reports vertical profiles of radial velocity at different radial positions of the IJ case obtained with MLWF$_{IJ}$ and LWWF. Both MLWF$_{IJ}$ and LWWF predict very similar results for the first radial position $r/y_{ref} = 1$, both under-estimating the radial velocity maximum as compared to averaged HiFi data. With increasing radial distance, this under-estimation reduces (as seen for $r/y_{ref} = 2$), and both approaches over-estimate the velocity peak found in HiFi data for the last two shown radial positions. For the latter, results predicted by MLWF$_{IJ}$ are slightly less accurate than the ones yielded by LWWF, and differences between the predictions of both approaches do not only concern the first off-wall nodes but are apparent up to 3 cells away in the wall-normal direction. For the IJ case, it appears less clear to link the differences in wall friction predictions predicted by both approaches to the overall reproduction of HiFi data.

Overall, the obtained results show the inability of a standard LWWF approach to predict accurate wall friction for the two presented cases exhibiting strong non-equilibrium effects. The proposed MLWF approach largely improves these predictions, and generally yields a quite satisfactory reproduction of wall friction as compared to HiFi data. These improvements, obtained with specifically trained MLWF models,

however did not fundamentally improve the prediction of general flow features as the velocity fields are very similarly predicted with LWWF despite its shortcomings. Differences between both models and HiFi data are probably more dependent from the underlying RANS approximations and from numerical approximations, than from the local estimation of wall friction. Using the MLWF in the RANS CFD code proved to be numerically robust and stable, under the restriction that they were applied starting from accurately initiated flow conditions. Reported findings are specific for the two simulated cases and would have to be confirmed in the general case. The practical general usability of MLWF approaches raises in addition specific questions as discussed hereafter. It should be noted that both MLWF models were tested on a 2D rectangular duct with good predictions, showing their ability to predict shear stress under the equilibrium condition. These results are not shown in the paper, but a similar flow is obtained at the entrance of the BFS case: the flow before the step ($x/y_{ref} < 0$) verifies the hypotheses of a flat boundary layer, and was correctly predicted by the trained MLWF (good comparison with the HiFi data for an *a priori* application of the MLFW in Fig. 7a, and in the CFD code in Fig. 10a). It is worth noting that the LWWF should be able to correctly predict the wall shear stress in this area, at least for mesh M2, which is not the case according to Fig. 10a. This should be further investigated in future work, but it was not the scope of the present study.

4.4 A critical discussion of the practical usability of a MLWF

It is important to underline that the improved prediction of wall shear stresses by the MLWF could only be achieved when applying a MLWF trained on HiFi data that is specific to the case simulated in RANS. Indeed, and not shown here, applying MLWF$_{BFS}$ to the IJ case, or inversely MLWF$_{IJ}$ to the BFS case yields largely erroneous results, that are even way less accurate than the ones obtained with LWWF. This is a direct consequence of two factors. First, the two reference flows chosen

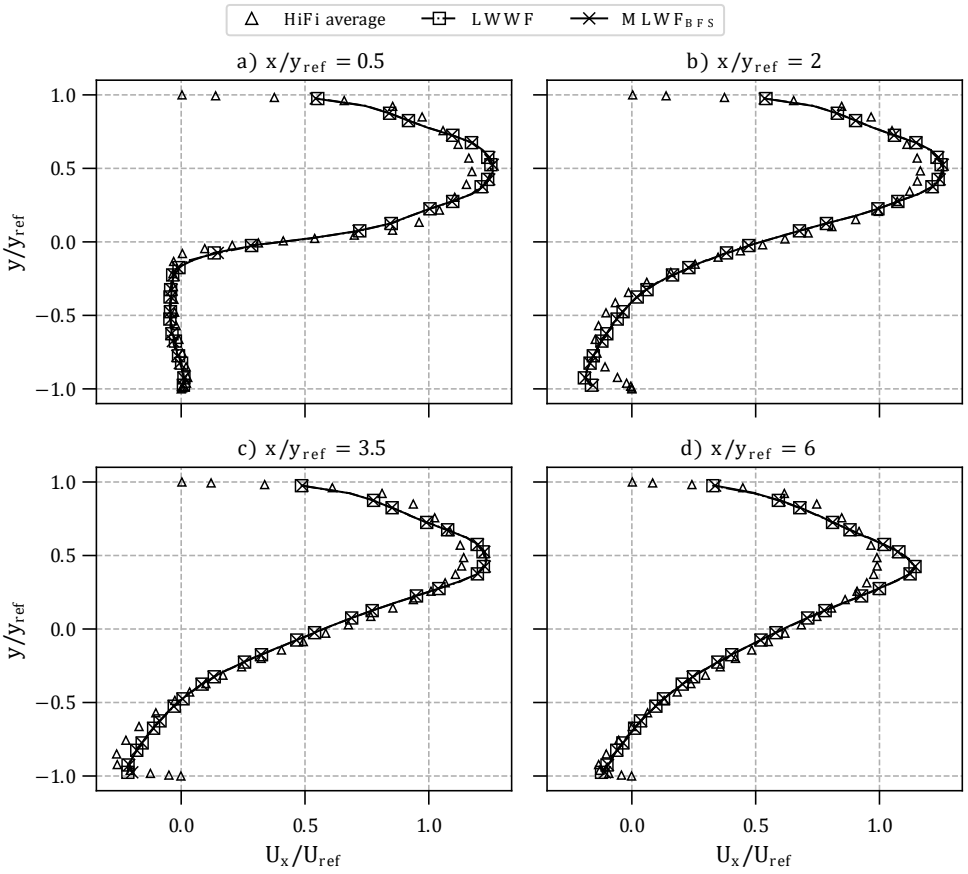


Fig. 11: Vertical profiles of streamwise average velocity at four radial locations of BFS: RANS predictions with MLWF$_{BFS}$ compared to LWWF predictions (M1 mesh) and to average HiFi data

in this study are very different, which means each MLWF has learned a particular physics, from a particular range of input variables. When used on a case it has not been explicitely trained for, a specific MLWF is fed with input variables lying outside that range, it will act as an extrapolator and yield unpredictable, possibly erroneous predictions. This strongly limits the general usability of a MLWF, and of many comparable ML approaches. An attempt consisting in formulating a MLWF$_{BFS+IJ}$ model by training it on lumped HiFi data from both BFS and IJ yielded very poor predictions (not shown here) and cannot be considered as achieving the goal to

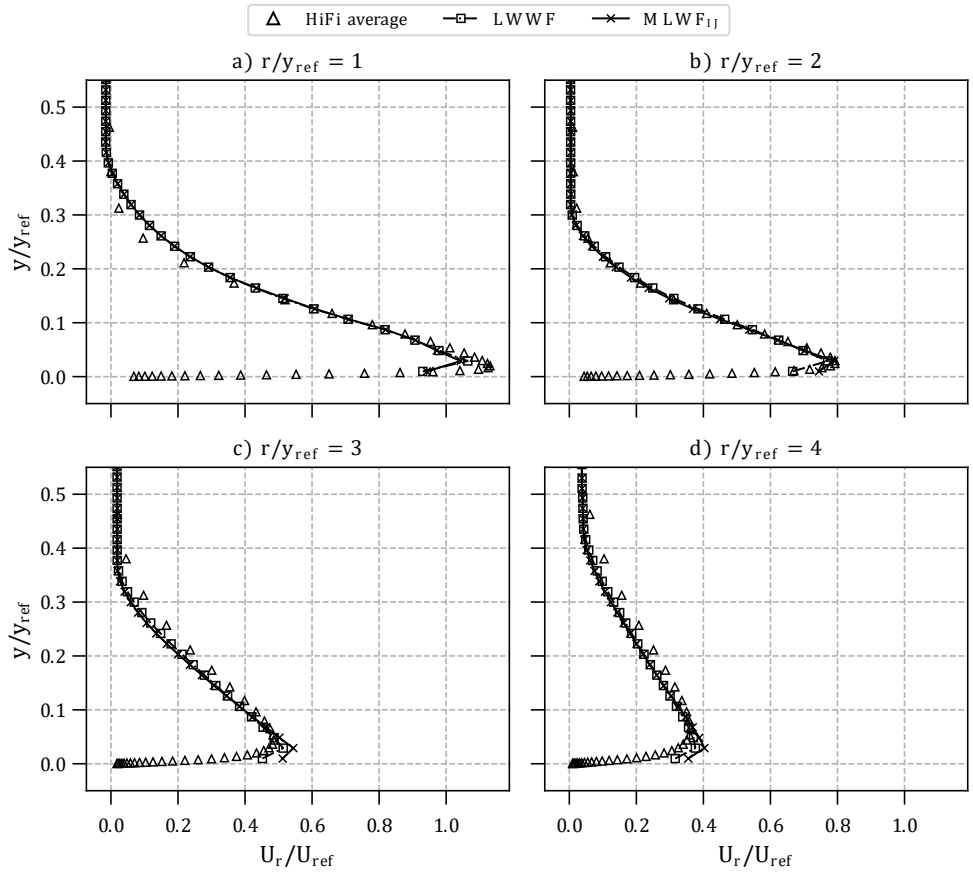


Fig. 12: Vertical profiles of radial average velocity at four radial locations of IJ: RANS predictions with MLWF$_{BFS}$ compared to LWWF predictions (M1 mesh) and to average HiFi data

formulate a general MLWF approach applicable on any non-equilibrium flow. Work as the one published in (Lozano-Durán et al, 2020) on mixture of experts approaches could open promising perspectives in this context. Secondly, the viscosity was not introduced in the set of input variables, as both cases have a constant viscosity. This prevents the generalisability of the MLWF at different Reynolds numbers. A perspective would be to add the viscosity as an input, and train a MLWF on a series of flows with varying Reynolds numbers.

Another key factor affecting the predictivity of a MLWF, even when it is used in a case it has been specifically trained for, is the necessary requirement that the neural network has to be informed with input data from the CFD code that are consistent with the training data in order to yield physically meaningful inferences in a RANS simulation (Duraisamy, 2021). Figure 13 illustrates this aspect for the BFS case. It compares three steady-state flow variables at the first off-wall nodes of the M1 mesh, with which MLWF$_{BFS}$ is informed during RANS simulations in order to predict τ_w, with the corresponding variables extracted from the BFS HiFi training data. The latter are extracted from all HiFi data points situated in the range $0.9h_{M1} < y < 1.1h_{M1}$ where h_{M1} is the distance of the first off-wall nodes in the M1 mesh of BFS. The RANS input data for axial velocity and pressure gradient are mostly lying within the range of the training data or close to it, with some local discrepancies especially on the bottom wall. However, this is hardly the case for turbulent kinetic energy, where CFD input data locally lie far outside the range of the training data for most points. This illustrates that HiFi training data and CFD data can at least partly be inconsistent in practice, as a consequence of unavoidable differences between the HiFi simulations having generated the former and RANS simulations providing the latter. For such outlyers, the neural network is informed with conditions for which it has not been explicitly trained, and has to perform an extrapolation task, which can lead to erroneous inferences. The impact of inconsistency possibly explains part of the a posteriori C_f prediction errors reported above.

This inconsistency is particularly marked for all flow variables when starting RANS simulations with MLWF from approximate initial conditions. This explains why the proposed MLWF models could not be used to start RANS simulations from such initial conditions and could only be used after an initiation phase with LWWF. The latter approach, despite its reported inability to predict accurate wall friction in non-equilibrium flows, nevertheless has the advantage of yielding bounded predictions as a

consequence of being based on analytical relations. Ensuring such a characteristic for the presently used neural networks is not straightforward and would require dedicated research in order to make MLWF more adapted for usage in practical simulations.

A specific feature of the proposed MLWF approach is that no pre-filtering of the HiFi data to map it from a high resolution mesh to a RANS-type coarse mesh is performed when generating the training data. Other authors made a different choice, either explicitly introducing the wall resolution as an additional input parameter (Zhou et al, 2019), or implicitly through the database content (Ling et al, 2016a). In order to estimate whether such a RANS-pre-filtering would improve the consistency of input data, Fig. 13 also shows training data resulting from filtering the HiFi data, as detailed in Appendix B. Clearly, the filtering is not found to improve the consistency of the input data, and even appears to globally deteriorate it, with larger differences between the filtered HiFi data and the CFD input data than without filtering. While not illustrated here, these findings for BFS on the M1 mesh are also found to be similar for the M2 mesh, as well as for IJ.

An alternative to using HiFi-data as in the presented work for training a MLWF would be to base it on near-wall flow data generated by wall resolving RANS simulations. This could be obtained using meshes with a high resolution of the near-wall regions, combined with an adapted turbulence model as $k-\omega$, or with other low-Reynolds number wall modelling approaches. This could have a number of advantages, and in particular the ability of having access to a larger quantity of training data generated at a lower cost than with HiFi-approaches based on LES or DNS. Plus, it would enable a wide range of Reynolds number of the same flow to be considered, and to include the viscosity parameter as an input variable. Furthermore, it could possibly contribute overcoming the inconsistency between training and input data, as the

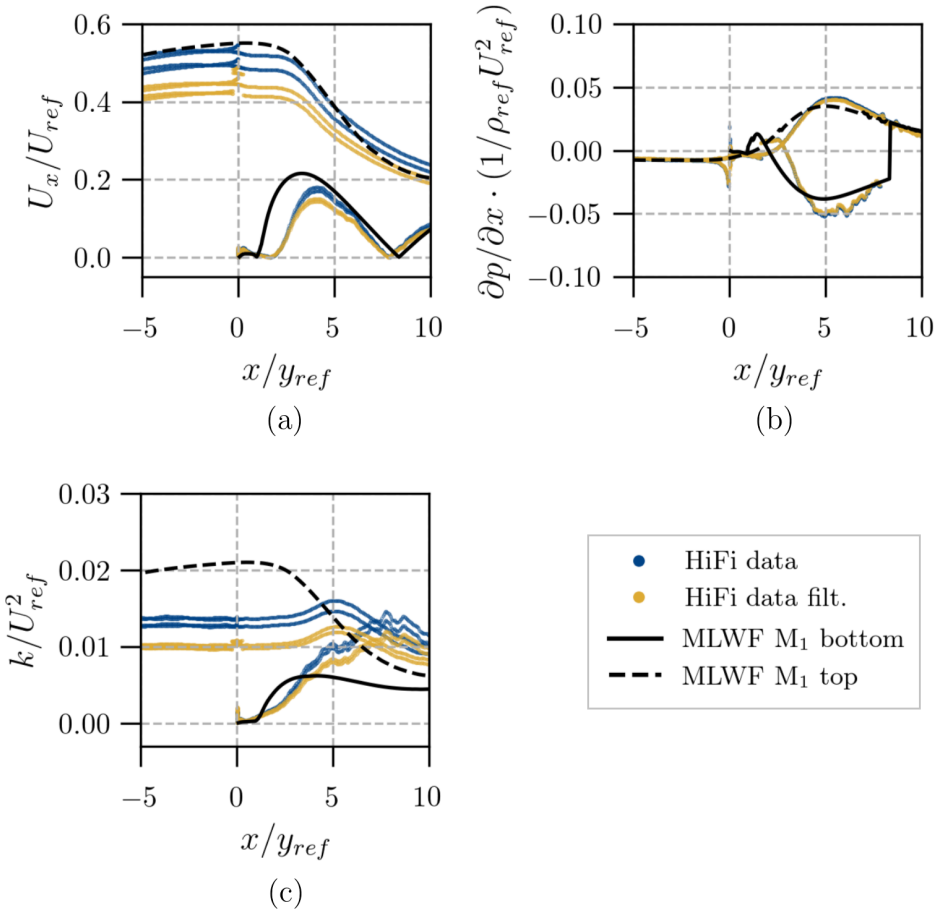


Fig. 13: Consistency of input data for BFS on the M1 mesh : comparison between three flow variables - a) streamwise velocity, b) streamwise pressure gradient, c) turbulent kinetic energy - taken in wall cells and informing MLWF$_{BFS}$ in the RANS simulations, and HiFi training data as well as RANS pre-filtered HiFi data (both taken at a wall distance range around the RANS wall cell resolution).

former would also be based on a RANS simulation and possibly have been been generated with the same CFD code. However, it could also have disadvantages related to inherent limitations of RANS as compared to HiFi simulations, possibly limiting the physical accuracy of the resulting training data. This could nevertheless be a key element in the development of future MLWF approaches.

5 Conclusions and perspectives

The present paper proposed a data-driven approach for formulating RANS wall functions able to improve the prediction of wall friction in turbulent wall-bounded flows exhibiting non-equilibrium effects, validated its predictions in RANS simulations by comparing them with those from standard wall functions, as well as with high-fidelity CFD data it was trained with, and provided a critical discussion of its practical usability. The proposed MLWF approach relies on central key simplifying assumptions used in standard wall functions, such as the quasi-steadiness of the flow, the alignment of wall friction and mean velocity at the first of-wall node, or the closure of turbulent transport equations at walls. However, it replaces the analytical law-of-the-wall by a fully connected feed-forward neural network built to infer wall friction from the flow state at the first off-wall nodes based on an off-line training from high-fidelity flow data. An extended set of input flow variables was selected in order to account for non-equilibrium effects that are not addressed by the law-of-the-wall.

This approach was then applied to formulate two different wall functions trained for two different configurations: the turbulent flow over a backward-facing-step (BFS) and a turbulent round jet orthogonally impacting a planar wall (IJ). *A priori* tests showed that the two specific MLWF accurately reproduced wall friction as compared to HiFi data, with limited discrepancies in some flow areas, but which were considered acceptable. The resulting predictive capabilities were further explored *a posteriori* by integrating the trained specific MLWF into an industrial CFD code, and using them in RANS simulations of the two flow configurations having served to generate the HiFi data. The wall friction predictions yielded by the two specific MLWF on the configuration they were trained for showed to be (both qualitatively and quantitatively) largely improved as compared to simulations using standard wall functions.

In terms of overall flow field predictions, the MLWF was found to yield predictions very similar to those of standard wall functions with only minor differences especially for the IJ case. Using a MLWF in the RANS CFD code was found to be numerically robust and stable, under the restriction that they were applied starting from accurately initiated flow conditions. The latter major limitation for a practical usage was attributed to the fact that the HiFi training data for MLWF and the input data provided to it by a CFD code can be at least partly inconsistent. This might also explain some discrepancies observed in the predictions of the RANS simulations with MLWF. Finally, issues related to the lack of generalizability of neural networks when applied to conditions not included in the training process were also critically discussed.

In summary, and despite the reported highly improved level of predictivity over standard RANS wall functions, an extensive usage of MLWF or similar ML approaches to predict wall friction will require a wealth of dedicated research. This would have to include work on improving the generalizability of ML approaches, via the inclusion of a much wider range of flow configurations and conditions in the training set, and by providing an adapted normalization of input and output variables. In this context, work could explore more advanced approaches such as the mixture of experts (Lozano-Durán et al, 2020) that selectively uses as inference subsets of the training data that are most pertinent to the case under scrutiny. Another aspect requiring dedicated research are means to ensure the consistency between input data furnished by the CFD code and the training data. In this context, providing means for bounding the inferences of ML approaches, as by construction provided by analytical wall functions, could be a highly desirable feature to limit highly erroneous predictions and contribute to a numerical robustness and stable approach. Here, training data generated by wall-resolved RANS simulations and using CFD codes similar to the one in which a resulting MLWF would be implemented could have the combined advantage of reducing the inconsistency issues while giving access to a larger volume of training

data for a higher number of flow configurations of interest but at a much lower effort and cost than possible with HiFi-simulations.

Finally, extending the approach to account for wall heat fluxes, and relaxing simplifying hypothesis of RANS wall functions, as in particular those related to turbulence closure at walls or to the alignment of wall stress and off-wall tangential velocity could be other interesting avenues of future research.

Appendix A Formulation of the LWWF

The LWWF is based on the Launder and Spalding wall model (Launder and Spalding, 1974). The balance between production and dissipation rates of turbulent kinetic energy in the wall bounbary layer leads to the following estimation of the friction velocity:

$$u_k = C_\mu^{1/4} k_1^{1/2} \tag{A1}$$

with $C_\mu = 0.09$ and k_1 the value of the turbulent kinetic energy at the centroid of the first off-wall cell p. This friction velocity u_k is used to compute the dimensionless wall distance h^+:

$$h^+ = \frac{h u_k}{\nu} \tag{A2}$$

with h the wall normal distance to the centroid of the first off-wall cell. Combining the logarithmic *law-of-the-wall* with Eq. A2 yields a second estimation of the friction velocity u_τ:

$$u_\tau = \begin{cases} \frac{\|\boldsymbol{U_t^P}\|}{\frac{1}{\kappa} ln(E h^+)} & \text{if } h^+ > 11.2 \\ \\ \frac{\|\boldsymbol{U_t^P}\|}{h^+} & \text{if } h^+ <= 11.2 \end{cases} \tag{A3}$$

with $E = e^{\kappa B}$, $\kappa = 0.4187$ and $B = 5.5$. The wall shear stress is then computed as follows:

$$\boldsymbol{\tau_w}^{LWWF} = -\frac{\boldsymbol{U_t^P}}{\|\boldsymbol{U_t^P}\|} \rho u_k u_\tau. \tag{A4}$$

Regarding the wall boundary conditions applied to the turbulence quantities, a zero-gradient Neumann condition is used for k and a wall law for the specific dissipation rate ω:

$$\omega = \frac{u_\tau}{C_\mu^{1/2} \kappa h} \tag{A5}$$

Appendix B Filtering high-fidelity data to coarse RANS meshes

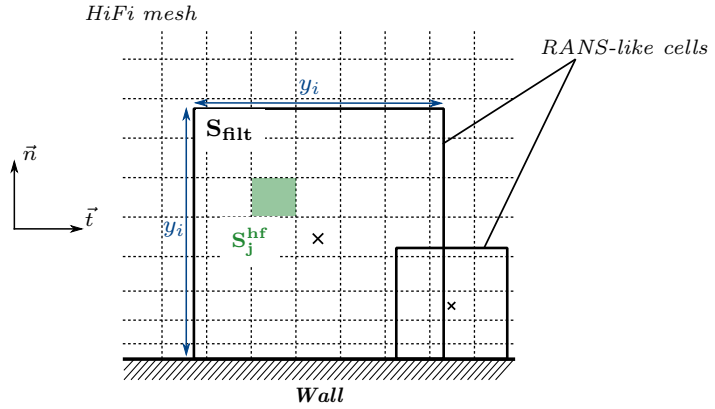


Fig. B1: Schematic principle of a pre-filtering on coarse RANS meshes along the wall normal distance of HiFi data available on a highly resolved mesh.

A two-dimensional filtering along the local normal distance to the wall is applied to HiFi data in order to map the mean flow variables from the highly resolved mesh having served to generate it, to the coarser mesh resolution typical of practical RANS simulations.

As illustrated in Fig. B1, the unfiltered value $I^{hf}(y_i)$ of any mean flow variable I in the initial HiFi data is replaced by an area-averaged value $I^{filt}(y_i)$ over all cells of the

HiFi mesh intersecting with the coarse RANS cell of area $S_{filt} = y_i^2$, as :

$$I^{filt}(y_i) = \frac{1}{S_{filt}} \sum_{j=1}^{N^{hf}} I_j^{hf}(S_j^{hf} \cap S_{filt}), \tag{B6}$$

where N^{hf} is the number of cells in the HiFi mesh.

Ethical statement

- Funding: N/A
- Conflict of Interest: N/A
- Ethical approval: N/A
- Informed consent: N/A
- Author contribution: E.R. developed the methodology, performed the computations and wrote the original draft of the manuscript text. A.P., C.A., M.M.Z and R.P. contributed to the development of the methodology, the interpretation of the results and the drafting of the manuscript. D.A.K. contributed to the technical implementation of the methodology. All authors reviewed the manuscript.
- Data Availability Statement: Data available on request from the authors

References

Abadi M, et al (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/, software available from tensorflow.org

Amarloo A, Cinnella P, Iosifidis A, et al (2023) Data-driven Reynolds stress models based on the frozen treatment of Reynolds stress tensor and Reynolds force vector. Physics of Fluids 35(7):075154. https://doi.org/10.1063/5.0160977

Aubagnac-Karkar D, Mehl C (2023) NNICE: Neural Network Inference in C made Easy. https://doi.org/10.5281/zenodo.7645515, URL https://github.com/aubagnacd/NNICE

Beck A, Flad D, Munz CD (2019) Deep neural networks for data-driven les closure models. Journal of Computational Physics 398:108910

Billard F, Laurence D, Osman K (2015) Adaptive wall functions for an elliptic blending eddy viscosity model applicable to any mesh topology. Flow, Turbulence and Combustion 94:817–842. https://doi.org/https://doi.org/10.1007/s10494-015-9600-x

Bin Y, Park G, Lv Y, et al (2022) LES on RANS-quality grids. submitted to AIAA J

Brenner MP, Eldredge JD, Freund JB (2019) Perspective on machine learning for advancing fluid mechanics. Physical Review Fluids 4(10):1

Brunton SL, Noack BR, Koumoutsakos P (2020) Machine learning for fluid mechanics. Annual Review of Fluid Mechanics 52(1):477–508

Cheng C, Zhang GT (2021) Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. Water 13

Choi H, Moin P (2012) Grid-point requirements for Large Eddy Simulation: Chapman's estimates revisited. Physics of Fluids 24(1):011702

Craft TJ, Gant SE, Gerasimov AV, et al (2006) Development and application of wall-function treatments for turbulent forced and mixed convection flows. Fluid Dynamics Research 38(2-3):127. https://doi.org/10.1016/j.fluiddyn.2004.11.002, URL https://dx.doi.org/10.1016/j.fluiddyn.2004.11.002

Duraisamy K (2021) Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. Physical Review Fluids 6(5):050504.

https://doi.org/10.1103/PhysRevFluids.6.050504

Duraisamy K, Iaccarino G, Xiao H (2019) Turbulence modeling in the age of data. Annual Review of Fluid Mechanics 51(1):357–377

Ferdian E, Suinesiaputra A, Dubowitz DJ, et al (2020) 4DFlowNet: Super-resolution 4D flow MRI using deep learning and Computational Fluid Dynamics. Frontiers in Physics 8:486

Grenouilloux A, Moureau V, Lartigue G, et al (2021) Feature-based mesh adaptation applied to the Large Eddy Simulation of multiple jets impinging on a surface. In: Proceedings UK Heat Transfer Conference

Grenouilloux A, Balarac G, Leparoux J, et al (2022) On the use of kinetic-energy balance for the feature-based mesh adaptation applied to Large Eddy Simulation in complex geometries. In: Proceedings ASME Turbo Expo

Guennebaud G, Jacob B, et al (2010) Eigen v3. http://eigen.tuxfamily.org

Guo M, Hesthaven JS (2018) Reduced order modeling for nonlinear structural analysis using gaussian process regression. Computer Methods in Applied Mechanics and Engineering 341:807–826

Guo M, Hesthaven JS (2019) Data-driven reduced order modeling for time-dependent problems. Computer Methods in Applied Mechanics and Engineering 345:75–99

Güemes A, Discetti S, Ianiro A, et al (2021) From coarse wall measurements to turbulent velocity fields through deep learning. Physics of Fluids 33(7):075121

Hasegawa K, Fukami K, Murata T, et al (2020) Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. Theoretical and Computational Fluid Dynamics 55(12):4013

Huang XLD, Yang XIA, Kunz RF (2019) Wall-modeled Large Eddy Simulation of spanwise rotating turbulent channels—comparing a physics-based approach and a data-based approach. Physics of Fluids 31(12):125105

Jiang C, Mi J, Laima S, et al (2020) A novel algebraic stress model with machine-learning-assisted parameterization. Energies 13(1):258

Joshi A, Assam A, Nived MR, et al (2019) A generalised wall function including compressibility and pressure-gradient terms for the Spalart-Allmaras turbulence model. Journal of Turbulence 20(10):626–660

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: 3rd International Conference for Learning Representations,, San Diego, p 15

Launder B, Spalding D (1974) The numerical computation of turbulent flows. Computer Methods in Applied Mechanics and Engineering 3(2):269–289. https://doi.org/https://doi.org/10.1016/0045-7825(74)90029-2

Launder BE, Rodi W (1983) The Turbulent Wall Jet Measurements and Modeling. Annual Review of Fluid Mechanics 15(1):429–459. https://doi.org/10.1146/annurev.fl.15.010183.002241

Ling J, Jones R, Templeton J (2016a) Machine learning strategies for systems with invariance properties. Journal of Computational Physics 318:22–35

Ling J, Kurzawski A, Templeton J (2016b) Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. Journal of Fluid Mechanics 807:155–166

Lozano-Durán A, Giometto MG, Park GI, et al (2020) Non-equilibrium three-dimensional boundary layers at moderate reynolds numbers. Journal of Fluid

Mechanics 883:A20. https://doi.org/10.1017/jfm.2019.869

Matai R, Durbin PA (2019) Zonal eddy viscosity models based on machine learning. Flow, Turbulence and Combustion 103(1):93–109

Mehl C, Aubagnac-Karkar D (2023) On-the-fly accuracy evaluation of artificial neural networks and hybrid method to improve the robustness of neural network accelerated chemistry solving. Physics of Fluids 35(6). https://doi.org/10.1063/5.0151026

Menter F, Kuntz M, Langtry R (2003) Ten years of industrial experience with the SST turbulence model. In: Turbulence, Heat and Mass Transfer 4

Milano M, Koumoutsakos P (2002) Neural network modeling for near wall turbulent flow. Journal of Computational Physics 182(1):1–26

Ong KC, Chan A (2018) A unified wall function for compressible turbulence modelling. Journal of Turbulence 19(5):414–430

Parish E, Duraisamy K (2015) A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of Computational Physics

Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in python. the Journal of Machine Learning Research 12:2825–2830

Pope SB (2000) Turbulent Flows

Popovac M, Hanjalic K (2007) Compound wall treatment for RANS computation of complex turbulent flows and heat transfer. Flow, Turbulence and Combustion 78(2):177–202

Richards K, Senecal P, Pomraning E (2022) CONVERGE 3.0.19

Rudy SH, Brunton SL, Proctor JL, et al (2017) Data-driven discovery of partial differential equations. Science Advances 3(4):e1602614

Saïdi IBH, Schmelzer M, Cinnella P, et al (2022) CFD-driven symbolic identification of algebraic Reynolds-stress models. Journal of Computational Physics 457:111037. https://doi.org/https://doi.org/10.1016/j.jcp.2022.111037, URL https://www.sciencedirect.com/science/article/pii/S0021999122000997

Schmelzer M, Dwight RP, Cinnella P (2020) Discovery of algebraic Reynolds-stress models using sparse symbolic regression. Flow, Turbulence and Combustion 104:579–603

Shih TH, Povinelli LA, Liu NS (2002) Application of generalized wall function for complex turbulent flows. In: Rodi W, Fueyo N (eds) Engineering turbulence modelling and experiments 5. Elsevier, Amsterdam and Boston and London, p 177–186

Slotnick J, Khodadoust A, Alonso J, et al (2014) CFD vision 2030 study: A path to revolutionary computational aerosciences. URL https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140003093.pdf

Tieghi L, Corsini A, Delibra G, et al (2020) Assessment of a machine-learnt adaptive wall-function in a compressor cascade with sinusoidal leading edge. Journal of Engineering for Gas Turbines and Power 142

Tieghi L, Corsini A, Delibra G, et al (2021) A Machine-Learnt Wall Function for Rotating Diffusers. Journal of Turbomachinery 143(8). 081012

Vaddireddy H, Rasheed A, Staples AE, et al (2020) Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. Physics of Fluids 32(1)

Wang H, Yang Z, Li B, et al (2020) Predicting the near-wall velocity of wall turbulence using a neural network for particle image velocimetry. Physics of Fluids 32(11):115105

Wang JX, Wu JL, Xiao H (2017) Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on DNS data. Physical Review Fluids 2(3)

Weatheritt J, Sandberg R (2016) A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. Journal of Computational Physics 325:22–37

Wilcox D (2006) Turbulence Modeling for CFD (Third Edition)

Wu JL, Wang JX, Xiao H (2016) A bayesian calibration–prediction method for reducing model-form uncertainties with application in RANS simulations. Flow, Turbulence and Combustion 97(3):761–786

Wu JL, Xiao H, Paterson E (2018) Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Physical Review Fluids 3(7):18

Xie C, Wang J, Li H, et al (2020) Spatially multi-scale artificial neural network model for Large Eddy Simulation of compressible isotropic turbulence. AIP Advances 10(1):015044

Yang XIA, Zafar S, Wang JX, et al (2019) Predictive Large Eddy Simulation wall modeling via physics-informed neural networks. Phys Rev Fluids 4:034602

Zhou Z, He G, Wang S, et al (2019) Subgrid-scale model for Large Eddy Simulation of isotropic turbulent flows using an artificial neural network. Computers & Fluids 195:104319

Zhou Z, He G, Yang X (2021) Wall model based on neural networks for les of turbulent flows over periodic hills. Phys Rev Fluids 6:054610