A Data-driven Wall-shear Stress Model for LES using Gradient Boosted Decision Trees

Sarath Radhakrishnan[1][0000−0003−1560−8812], Lawrence Adu Gyamfi[0000−0002−4100−8717], Arnau Miró[1][0000−0002−2772−6050], Bernat Font[1][0000−0002−2136−3068], Joan Calafell[1][0000−0002−2333−7314], and Oriol Lehmkuhl[1][0000−0002−2670−1871]

Barcelona Supercomputing Center, Barcelona, Spain

Abstract. With the recent advances in Machine Learning, strategies based on data could be used to augment wall modeling in Large Eddy Simulation(LES). In this work, a wall model based on gradient boosted decision trees is presented. The model is trained to learn the boundary layer of a turbulent channel flow so that it can be used to make predictions for significantly different flows where the equilibrium assumptions are valid.The methodology of building the model is presented in detail. The experiments conducted to choose the data for training the model, as well as to choose the model input features are described. The trained model is tested a posteriori on a Turbulent channel flow and the flow over a wall mounted hump. The results from the tests are compared with that of an algebraic equilibrium wall model and the performance is evaluated. The results show that the model has succeeded in learning the boundary layer and performs as good as an algebraic wall stress model.

Keywords: Machine Learning · XGBoost · Turbulence · Wall Models

1 Introduction

As a consequence of the recent developments in High-Performance Computing(HPC), Large Eddy Simulation (LES) is getting increased attention as a predictive tool in turbulence research. LES essentially resolves the dynamically significant, flow-dependent, energy-containing larger scales and models the smaller ones. For LES to give accurate results these integral scales of motion need to be completely resolved. This becomes prohibitively expensive at high Reynolds Number(Re). When it comes to wall-bounded flows, LES may become computationally unaffordable even at moderate Reynolds numbers, given the complex flow structure at the boundary layer.The estimated cost of calculation is of the order of $Re^{\frac{13}{7}}$ for Wall-Resolved LES (WRLES) [1] and it can be even more expensive when the temporal integration cost is considered [2]. Interestingly, over 50% of the computational resources are used to resolve only 10% of the flow [3] even at moderate Re. Therefore, the only economical way to perform LES of wall-bounded high Re flows is by resolving the outer layer alone. Since the grid sizes to resolve the outer layer are too coarse to resolve the viscous sub-layer, this needs to be modeled. The cost of Wall Modeled LES (WMLES) depends loosely on the Re and the type of the model. Wall models can be classified into two categories: zonal models and equilibrium law based models. In zonal approach, a different set of equations are solved closer to the wall to determine the boundary condition. Equilibrium law based models assume that the inner layer dynamics are universal hence they can be represented by a general law, at least in an average sense. A detailed review of different wall modeling strategies is presented in [3–6].

Conventional wall modeling strategies perform reasonably well in both low and high speed flows [7–11]. However, their industrial utility is still limited because of their poor performance on flows involving separation and/or heat transfer [12]. With the advances in Machine Learning, strategies based on data could be used to augment wall modeling in LES. Data-based approaches have already been used in Computational Fluid Dynamics(CFD) for turbulence modeling [13–25] and for the simulation of multi-phase flows [26, 27]. These approaches are reviewed in detail by [28] and [29]. However, wall-modeling using data-based approaches has just started getting attention from the CFD community. The previous works [30, 31] have used neural networks for modeling. In this study, a wall model with XGBoost [32] is presented. The performance of the model is evaluated on two cases, one of which is significantly different from the trained data and the results are compared with that of an algebraic equilibrium wall model(EQWM [33]).

The paper is organised as follows: Section 2 provides the background information. In section 3 results are presented. Concluding remarks are in section 4.

2 Background

XGBoost is a non-parametric method built using several “base learners” based on the statistical principle of boosting [34]. The base learners used by XGBoost are decision trees. Methods using decision trees have already been used in the context of turbulence modeling [19, 20]. Being a non-parametric model, XGBoost does not assume anything about the data and the distribution of the data. The number of parameters for such a model are not limited. This also increases the chance of over-fitting to the trained data. However, XGBoost also provides hyper-parameters to tune such that over-fitting can be effectively controlled. The complete methodology followed for generating the model is discussed below:

2.1 Data-sets

The data-sets for the model development are used from WRLES of turbulent channel flows. A turbulent channel flow is a typical case where Law of the Wall (LoW), given by Equation(1) holds. u in Equation(1) is the free-stream velocity and y the normal distance from the wall. $\kappa \approx 0.4$, is the Von Kármán constant, and ν is the dynamic viscosity of the fluid. B is a constant that has a value approximately equal to 5. u_τ is the characteristic velocity of the flow called friction velocity. Most of the turbulent flows exhibit this velocity distribution inside the thin wall layer and so it is considered universal [35]. We intend to train the model to learn the Boundary Layer(BL) of the channel flow so that it can be used for turbulent flows that are significantly different from the channel flow.

$$\frac{u}{u_\tau} = \frac{1}{\kappa}\log(\frac{yu_\tau}{\nu}) + B \tag{1}$$

WRLES is performed using the in-house multi-physics code Alya [36]. Alya is a low-dissipation finite-element code which solves the incompressible Navier-Stokes equation. Two simulations were run using Alya to generate the datasets, viz., $Re_\tau = 1000$ and $Re_\tau = 180$. Re_τ is the friction Reynolds number given by,

$$Re_\tau = \frac{u_\tau \delta}{\nu}$$

where δ is the half-channel height. Given below are the details of the simulation:

$\boldsymbol{Re_\tau}$ **= 180:** The computational domain for this case is $4\pi\delta \times 2\delta \times \frac{4}{3}\pi\delta$ in the stream-wise, wall-normal and span-wise directions respectively. A mesh of 64^3 elements is used to discretize the domain. The mesh is uniform in the stream-wise and span-wise directions, corresponding to $\mathit{\Delta} x^+ \approx 35$ and $\mathit{\Delta} z^+ \approx 12$ in wall-units respectively. The mesh is stretched in the wall normal direction using a hyperbolic tangent function given by,

$$y(i) = \frac{\tanh(\gamma(\frac{2(i-1)}{N_y} - 1)}{\tanh(\gamma)} \tag{2}$$

where N_y is the number of elements in the wall normal direction, with i ranging from 1 to N_y. γ is the factor which controls the stretching. The value of γ is chosen such that the $\mathit{\Delta} y^+ = 1$. Periodic boundary conditions are applied on the stream-wise and span-wise directions, while a no-slip boundary condition is imposed on the wall boundaries. Flow is driven by a stream-wise constant pressure gradient and Vreman [37] Sub-Grid Scale(SGS) model is used for turbulence closure.

$\boldsymbol{Re_\tau}$ **= 1000:** The computational domain for this case is $6\pi\delta \times 2\delta \times 3\pi\delta$ in the stream-wise, wall-normal and span-wise directions respectively. A mesh of 128^3 elements is used here. Just as before, The mesh is uniform in the stream-wise and span-wise directions, corresponding to $\mathit{\Delta} x^+ \approx 140$ and $\mathit{\Delta} z^+ \approx 70$ in wall-units respectively . The hyperbolic tangent function of Equation(2) is used to stretch the mesh in the wall normal direction. The $\mathit{\Delta} y^+ = 1$ in this simulation as well. Identical boundary conditions are applied with periodic stream-wise and span-wise boundaries and a no-slip boundary on the walls. A stream-wise constant pressure gradient drives the flow and an Integral Length-scale Approximation(ILSA) [38] SGS model used for turbulence closure.

Table 1: Distribution of data for Training, Validation and Testing of the model

	Training Data		Validation Data		Testing Data	
	sample size	simulation	sample size	simulation	sample size	simulation
Case Study 1	70%	$Re_\tau = 180$	30% 10%	$Re_\tau = 1000$ $Re_\tau = 180$	90%	$Re_\tau = 180$
Case Study 2	70%	$Re_\tau = 1000$	30% 10%	$Re_\tau = 180$ $Re_\tau = 1000$	90%	$Re_\tau = 1000$

Data from these two simulations are distributed for training, validation and testing as given in Table 1. Details of the case studies mentioned in the table are explained later in section 2.3. Choosing the validation data from two distributions is to prevent over-fitting of the model to the training data so that the generality is preserved.

It is made sure that the data chosen for training are statistically not correlated in time. In addition to these data-sets, a third set of data is synthetically generated from the above data-sets by varying the viscosity(ν) of the flow and the half-height of the channel (δ) and correspondingly scaling the velocity (u) and friction velocities (u_τ). This data-set is an augmentation to the Testing data to assess how well the model is able to generalize to new flow cases that are not included in the training. Below are the relations used for the scaling viscosity and the channel heights. The scaling ratios are computed such that the bulk and the friction Reynolds numbers of the flow are unaltered. Four sets of synthetic data are generated, two of which, by modifying ν and the other two by modifying δ. For viscosity modification from ν_{old} to ν_{new}, instantaneous velocity (u_{new}) and frictional velocity ($u_{\tau_{new}}$) are calculated as:

$$u_{new} = u_{old}\frac{\nu_{new}}{\nu_{old}}$$

$$u_{\tau_{new}} = u_{\tau_{old}}\frac{\nu_{new}}{\nu_{old}}$$

For the channel height modification from δ_{old} to δ_{new}, scaled instantaneous velocities (u_{new}) and friction velocity ($u_{\tau_{new}}$) are calculated as:

$$u_{new} = u_{old}\frac{\delta_{old}}{\delta_{new}}$$

$$u_{\tau_{new}} = u_{\tau_{old}}\frac{\delta_{old}}{\delta_{new}}$$

The synthetic data is with $\nu = \{10^{-5}, 10\}$ and $\delta = \{2, 10\}$.

2.2 Model Inputs and output

The output from the model is u^+, i.e., u/u_τ, from which τ_ω can be easily computed. τ_ω is the output from a typical EQWM, which acts as a boundary condition in WMLES [33]. However, choosing an input is not straight-forward. The choice of inputs should be such that, the model generated based on these inputs should be extrapolatable to higher Re flows. So the best set of inputs for the model is derived from a series of trial and error experiments. From the analysis of the results after each experiment, more knowledge w.r.t the physics and/or the domain of the flow field is supplemented as inputs. In this section we discuss the three set of inputs considered for training the model. For each of the three experiments detailed below, a separate XGBoost model is built. The comparison of their performance w.r.t each other is in section2.3. Based on their performance, a set of inputs is chosen for training a new model for a posteriori testing.

Primitive features (*xgb_1* model): For the first experiment, the features selected as inputs to the model xgb_1 are the raw outputs (primitives) from WRLES. The relevant parameters chosen are the stream-wise flow velocity (u) and the height (y) at each grid point in the discretized domain. Preliminary tests had shown that the span-wise and wall-normal velocities (v, w) did not impact significantly the performance of the models. These velocities as well as the pressure are not included in the training process of xgb_1. Since the model inputs are not normalized, generality is not expected from this model.

Scale-invariant features (*xgb_2* model): For the second experiment, the inadequacy of xgb_1 is improved by selecting features which carry some information regarding the physics of the flow. These are primarily the 'Local Reynolds Number (Re^*)' and its natural logarithm. Re^* is computed based on the formulae $Re^* = \frac{uL}{\nu}$ where, L is the wall distance, and u is the free-stream velocity at each grid point. In addition to that, a local u^+ variable was included, defined as $u_*^+ = \frac{u}{u_\tau^*}$, where $u_\tau^* = \sqrt{\tau}$. τ is the shear stress at each of the grid points. The model trained with these features is named as xgb_2. Since the inputs to the model includes normalized features, we expect the model to learn the generality of the flow.

Dimensionless features (*xgb_3* model): The third and final model, xgb_3, was built considering the features which can give some information regarding the flow domain. These new features are the non-dimensional velocity (u^*) and height (h^*). The scalar used for scaling the height is called off-wall grid height (h_{ow}), which is the height of a chosen grid point. In this experiment we choose the first off-wall grid point. The scalar used for scaling the velocity is the velocity value at h_{ow}. u^* and h^* are calculated at each grid point as follows:

$$h^* = \frac{h}{h_{ow}}$$
$$u^* = \frac{u}{u_{ow}}$$

The Local Reynolds number from xbg_2 is retained in this model. Table 2 shows the input variables used in each category of experiments as discussed above.

Table 2: Inputs and Outputs for the three experimental models.

Model name	Input Type	Inputs	Output
xgb_1	Flow Primitives	u, h	u^+
xgb_2	Scale-invariant	u, h, Re^*, $\ln(Re^*), u_*^+$	u^+
xgb_3	Dimensionless	u^*, h^*, Re^*	u^+

2.3 Model Training and Testing

We use the XGBoost package from scikit-learn [39] for generating the model. Each of the model described in the previous section is subjected to different data-sets for testing and training. Table. 3 shows the summary of the model training and testing. Case studies 1 and 2 decides which is the better data out of the two for training the models. Case studies 3 to 6 are done to evaluate the performance of the models on synthesized general flows. The distribution of the data for training validation and testing are as detailed in the Table 1 in section 2.1 The following sections describe the results of the above mentioned case studies and the performance of each of the models.

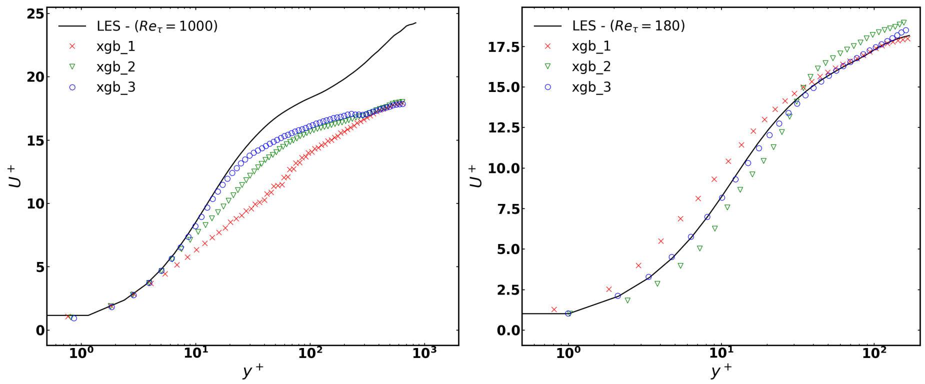
Table 3: Case studies done for each of the model.

Case Study	1	2	3-6
Training data	$Re_\tau = 180$	$Re_\tau = 1000$	$Re_\tau = 1000$
Testing data	$Re_\tau = 1000$	$Re_\tau = 180$	Synthesized data from $Re_\tau = 1000$

Case study 1: Training data from $Re_\tau = 180$, Testing data from $Re_\tau 1000$ The different models were trained on the data from the WRLES of $Re_\tau = 180$ and their performance tested on the data from WRLES of $Re_\tau = 1000$.

Fig. 1a shows the mean of the u^+ predicted by each of the models compared with the results of the WRLES simulations. The velocity profiles are underestimated all the way from the inner region (close to the wall) to the outer region of the boundary layer. The mean of the predictions of the model built on the non-dimensional values tends to be the closest to the results of the LES, but it is still not as accurate.

Case study 2: Training data from $Re_\tau = 1000$, Testing data from $Re_\tau = 180$ For this test case, the models were trained on the data from the flow case of $Re_\tau = 1000$ and their performance is evaluated data from $Re_\tau = 180$. The results of the test are shown in Figs. 1b. It is observed that the mean u^+ from the predictions of all the models follow quite well with the results of the



(a)

(b)

Fig. 1: Model performance: (a) Training data from $Re_\tau = 180$ and Testing data from $Re_\tau = 1000$. (b) Training data from $Re_\tau = 1000$ and Testing data from $Re_\tau = 180$.

LES simulations in the outer region of the boundary layer. The results from the model of dimensionless inputs gives the closest predictions. Closer to the wall, in the inner region of the boundary layer, the model built on primitive flow values tends to deviate from the LES results.

Case studies 3 to 6: Tests on scaled data The performance of the models are further evaluated on the synthetic data. Fig. 2 shows the mean u^+ generated by the predictions of the models when tested on the modified and scaled data sets. The results are compared to the mean velocity profiles of the corresponding data sets. When the height of the channel is increased (assuming the Reynolds number of the flow remains unchanged), the models trained on primitive and scale-invariant variables tend to underestimate the u^+, due to the corresponding decrease in the velocity. However when dimensionless variables are used, the model predictions follow the mean velocity profile of the LES simulations as shown in Fig. 2a and 2b.

Similar observations are made when the viscosity is modified and the data set scaled accordingly. The mean profile is underestimated by the models built on primitive and scale-invariant variables, when a lower viscosity is imposed. Using dimensionless inputs helps the model to identify the physics of the flow data irrespective of the modification or scaling that is applied. The model built on these inputs consistently predicts u^+ values that follow the mean velocity profile of the LES simulations throughout the boundary layer (inner and outer regions). This highlights the importance of including some domain information. The model built only on scale-invariant inputs, though includes some normalized

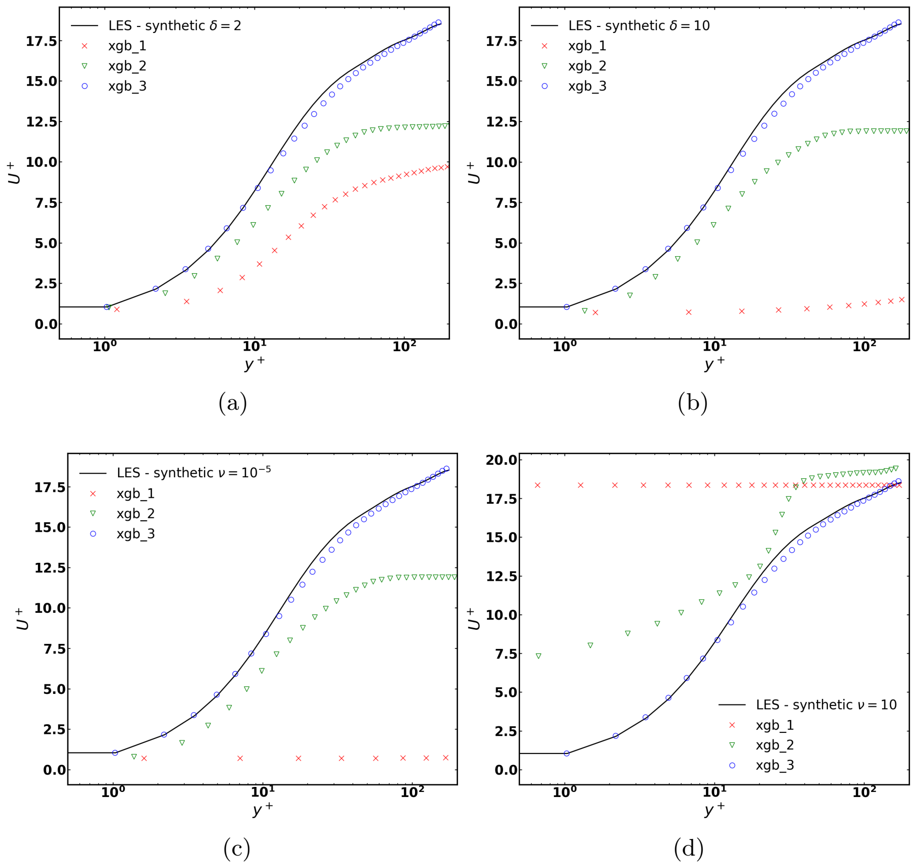


Fig. 2: Test of models on the synthetic data. (a) $\delta = 2$ (b) $\delta = 10$ (c) $\nu = 10^{-5}$ (d) $\nu = 10$

input variables such as Re^* and u_*^+, it seems to be inadequate when modified data is used for prediction.

Although xgb_3 performs better overall, it is not tested for its capability to extrapolate to a higher Re_τ flow. This will be comprehensively done in the a posteriori tests in the next section. In the sections to follow xgb_3 will be referred as Machine Learning Wall-Model(MLWM).

3 Results and discussion

The performance of MLWM is evaluated a posteriori on two cases: a) Turbulent channel flow of $Re_\tau = 2005$ and b) Flow over a wall mounted hump. The first case will test the model's capability to extrapolate to higher Re_τ flows. The second case will test the model's performance on complex geometries. Details of the cases are discussed below:

3.1 Turbulent channel flow

The model is tested on a channel flow of $Re_\tau = 2005$ to show that it can make the right predictions when coupled with an LES solver. The computational domain for the case is $6\pi\delta \times 2\delta \times \times 2\pi\delta$ in the stream-wise, wall normal and span-wise directions respectively. Three different meshes were used to produce converged results. All the meshes are uniform in the stream-wise, wall-normal and span-wise directions. Details of the meshes are given in Table. 4. Periodic boundary conditions are applied on the stream-wise and span-wise directions, while a no-penetration condition is imposed on the boundaries. Flow is driven by a stream-wise constant pressure gradient and the Vreman [37] SGS model is used for turbulence closure. LES velocity from the third-grid point and the corresponding wall-distance is used to compute Re^*. This is consistent with the direction of LES inputs for wall-models as explained in [40]. For computing u^* and h^*, velocity from the lower grid point and the corresponding wall-distance are used respectively. The case is run long enough to make sure it is statistically steady. The results are averaged for 20 flow-through times, where one flow-through time is the time taken by the center-line stream-wise velocity to cover the domain length. Subsequently, spatial averaging is also done and normalised with the friction velocity of the flow.

Table 4: Details of the meshes used for the simulation of the Turbulent channel flow, $Re_\tau = 2005$. N_x, N_y and N_z represent the number of elements in the stream-wise, wall-normal and span-wise directions respectively.

	size($N_x \times N_y \times N_z$)	Δx^+	Δy^+	Δz^+
M1	$64 \times 64 \times 64$	≈ 591	≈ 63	≈ 196
M2	$128 \times 96 \times 96$	≈ 295	≈ 42	≈ 131
M3	$256 \times 128 \times 128$	≈ 147	≈ 31	≈ 98

Numerical results Fig. 3 shows the mean-stream velocity for M1, M2 and M3 compared with the results from Direct Numeric Simulation(DNS) [41]. The mean-stream velocity profile tend to approach the DNS as the mesh density increases. For all the simulations, the error in u_τ predicted by the model is less than 0.5%. This ability of the model to extrapolate to higher Re_τ channel flows shows that the model has learnt the LoW without explicitly giving any information about LoW as input features. Now the performance of the model is compared with that of the algebraic wall-model, EQWM. The mesh used for the simulations is M3. The results are shown in figure 4. The mean-stream velocity

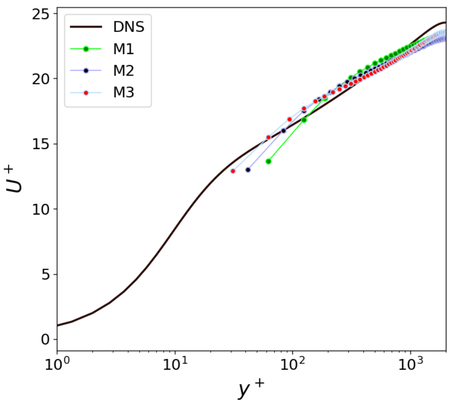


Fig. 3: Mean stream-wise velocity profiles for M1, M2 and M3 compare with DNS.

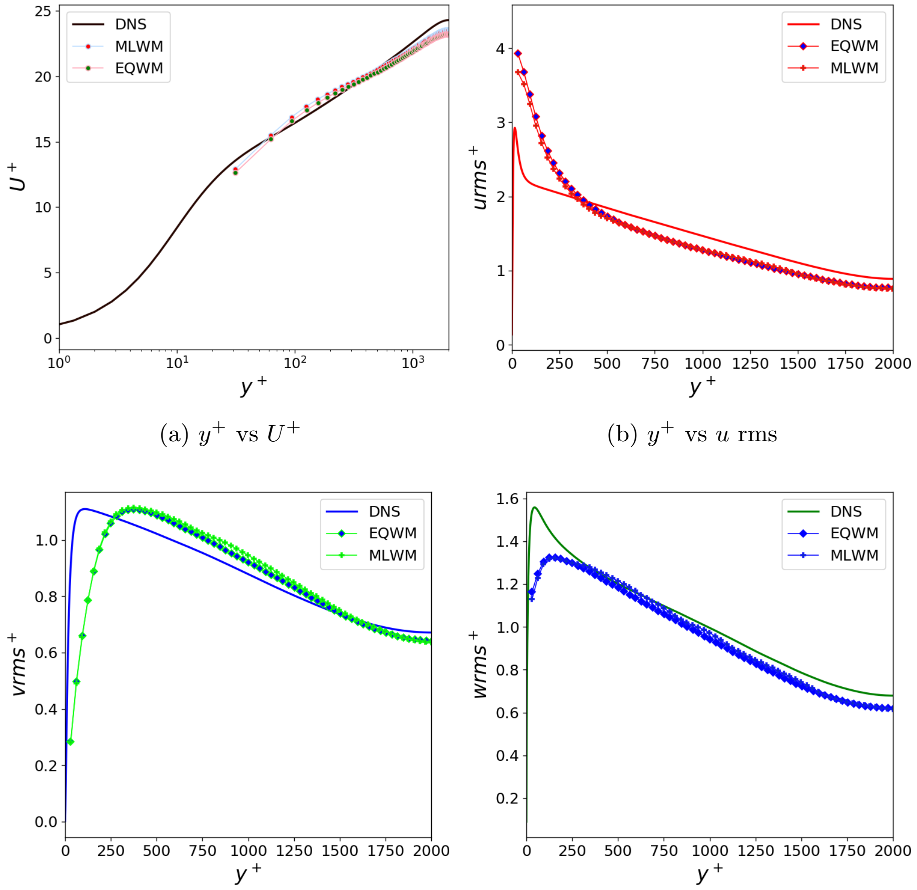
profiles as well as the mean fluctuations for both the models are very close to each other. This is an indication that our non-parametric model trained with the data predicts as good as a typical EQWM.

3.2 Wall-mounted hump

In this section, the model is used in the simulation of the flow over a wall-mounted hump. This is considered as a benchmark case to test turbulence models as it involves separation, re-attachment, and recovery of the boundary layer which are the hallmarks of many industrial flows. The geometry is defined following the guidelines of the NASA CFDVAL2004 workshop [42] and the inflow configuration is based on the work done by Park [43]. The results are compared with the experimental data of Greenblatt et.al. [44]. The performance of the model is also evaluated by comparing the results with that of the EQWM [33].

The computational domain is $4.64c$, $0.909c$ and $0.3c$ in the stream-wise, normal and span-wise directions respectively, where c is the chord-length of the hump. The hump leading edge is set at $x/c = 0$ so that inlet and outlet planes are at $x/c = -2.14$ and $x/c = 2.5$ respectively. In order to account for the effects of the end-plates used in the experiment, the top wall of the domain has a contour from $x/c = -0.5$ to $x/c = 1.5$ as shown in Fig 5.

The simulations were conducted on two different grids. The coarse mesh(G1) consists of approximately 3.1 million linear elements, with $742 \times 70 \times 60$ elements in the stream-wise, normal and span-wise directions respectively. The fine mesh(G2) consists of approximately 8 million linear elements with $900 \times 110 \times 80$ elements in the stream-wise, wall-normal and span-wise directions respectively. Fig 6 shows the grid spacing in wall units between $x/c = -0.5$ $x/c = 2.0$. G2 has more refinement in the tangential directions compared to G1. In the normal direction, the mesh growth-rate is changed from 1.06 in G1 to 1.03. However, the



(a) y^+ vs U^+

(b) y^+ vs u rms

(c) y^+ vs v rms

(d) y^+ vs w rms

Fig. 4: Comparison of the model with an algebraic equilibrium law based wall model: (a) shows the Mean stream-wise velocity and (b), (c), (d) show the mean fluctuations.

Δy^+ is maintained the same. The Reynolds number of the flow is $Re = 936000$, based on the hump cord length, c, and the free stream velocity, U_∞ at the inlet. Periodic boundary conditions are imposed on the span-wise direction, while a slip boundary condition is applied at the top boundary. At the bottom wall where the wall stress is predicted, a no-penetration condition is imposed.Vreman [37] SGS model is used for turbulence closure and synthetic turbulence as described in [45] generated as inflow data such that realistic turbulence evolves before the flow reaches the hump. The MLWM is fed using instantaneous LES data at the

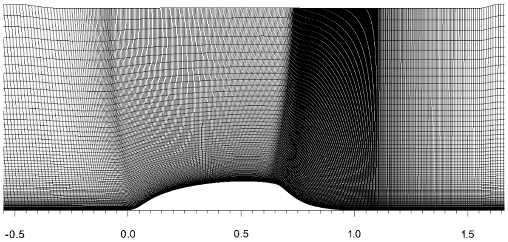
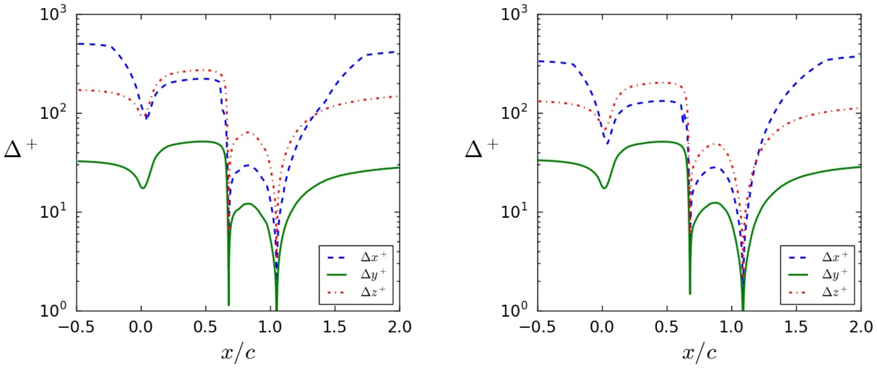


Fig. 5: Computational mesh around the hump. The contour between x/c = -0.5 and x/c = 1.5 is visible.

12.5% of the boundary layer thickness and is normalised using the first off-grid wall distance data.

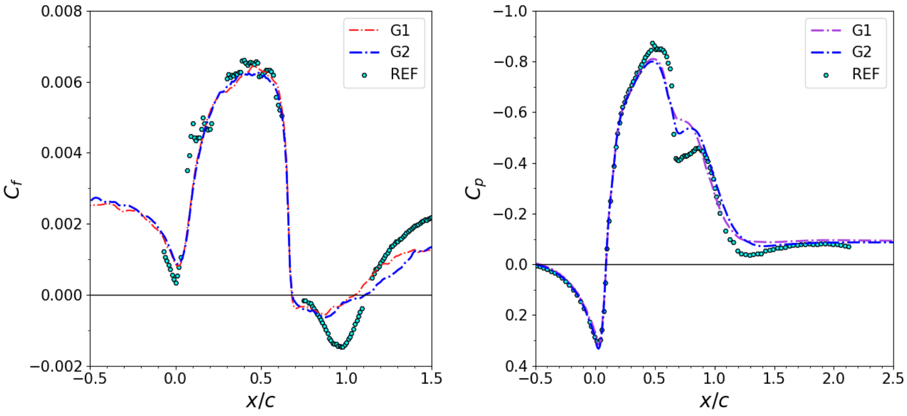


(a) *Coarse mesh(G1)*

(b) *Fine mesh(G2)*

Fig. 6: Grid spacing in wall units for (a) G1 and (b) G2 between x/c = -0.5 and x=̧ 2.0.

Numerical Results Fig. 7 shows the Skin friction(C_f) and Pressure coefficients (C_p) for the meshes G1 and G2. The results from both simulations are close to each other except at the re-circulation region. The effect of mesh refinement is reflected in the prediction of C_p. Both simulation fail to capture the primary suction peak before the re-circulation bubble and the secondary suction peak within the bubble, G2 is qualitatively better. Table 5 shows the details about the re-circulation bubble from the simulations. The length of the bubble as well as the re-attachment region are significantly better for G2.



(a) *Skin Friction Coefficient*

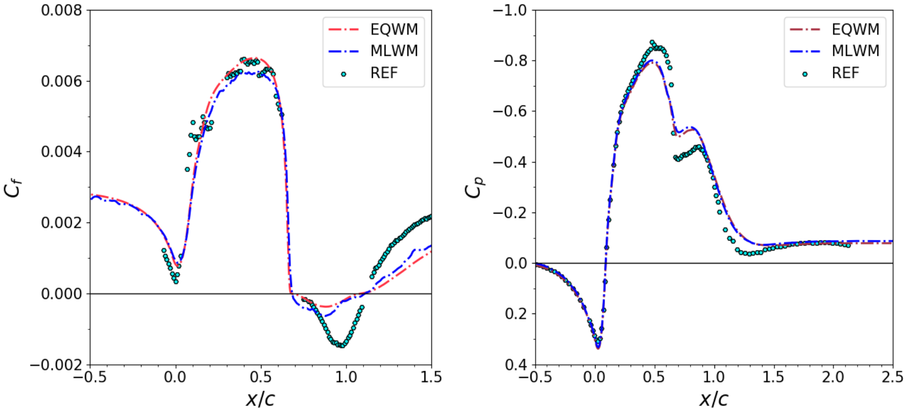
(b) *Pressure Coefficient*

Fig. 7: Skin friction and Pressure Coefficients for G1 and G2 compared with experimental results.

Next, we compare the results of the simulation done on mesh G2 with that of the EQWM performed on the same mesh. Figures 8a and 8b show the Skin friction and Pressure coefficients respectively. There are only minor differences in the prediction of C_f by both the models. Both models fail at the re-circulation region. Table 5 shows that both models predict the re-attachment location within 1% error, with EQWM predicting 0.01 units behind the actual location from the experiments and MLWM predicting 0.01 units ahead. This is due to the slightly lower wall-friction predicted by the MLWM(cf. Fig 7b) compared to EQWM. As a result the boundary layer carries more momentum in the case of MLWM. Due to the higher momentum, the point of separation as well as the point of re-attachment are slightly ahead of what is expected(cf. Table 5). The effect of the higher momentum is also visible in the mean-stream velocity profiles($\overline{u}/U_\infty$) shown in 9. The mean-stream velocity profiles of MLWM are slightly ahead of the ones from EQWM with a decrease in the longitudinal curvature throughout the re-circulation bubble. Fig 9 also shows that the Reynolds stresses are also higher for MLWM compared to EQWM. However, when compared to experimental data both models fail to capture the stream-wise velocity and stress profiles.

4 Conclusions

In this study, a wall model based on machine learning is presented. The model is based on the non-parametric method XGBoost. The model is trained to learn the boundary layer of a channel flow. For training, the data from a Wall resolved LES of channel flow $Re\tau = 1000$ is used. The performance of the model is



(a) *Skin Friction Coefficient*

(b) *Pressure Coefficient*

Fig. 8: Skin friction and Pressure Coefficients using the machine learning wall model(MLWM) and algebraic wall model(EQWM) compared with experimental results.

evaluated by testing the Turbulent channel flow $Re\tau = 2005$ and on the flow over a wall mounted hump. On the channel flow, the model performs as good as an algebraic equilibrium wall stress model, by predicting a wall stress within 0.5% error. On the wall mounted hump, the model predicts the coefficient of pressure as good as an EQWM. The skin-friction coefficient is very close to the values predicted by the EQWM. The mean-stream velocity profiles predicted by the MLWM is very close to the experimental data. The Reynolds stress profiles predicted by the model is slightly different EQWM. from the experimental data. However, as the predictions of the stress profiles by both models are significantly different compared to the experimental data we summarize that the MLWM is on par with EQWM and has succeeded in learning an equilibrium boundary layer.

Table 5: Details of the re-circulation bubble for MLWM and EQWM on both G1 and G2. $x/c|_{sep}$ is the location of separtion and $x/c|_{reatt}$ is the location of reattachment. $error|x_{reatt}$ is the error in the location of reattachment.

	mesh	$/x/c|_{sep}$	$x/c|_{reatt}$	bubble length	$error|x_{reatt}$
Experiment [44]	-	~ 0.665	~ 1.1	~ 0.435	-
EQWM	G1	0.67	1.05	0.38	4.5%
MLWM	G1	0.68	1.05	0.37	4.5%
EQWM	G2	0.67	1.09	0.42	0.9%
MLWM	G2	0.686	1.11	0.425	-0.9%

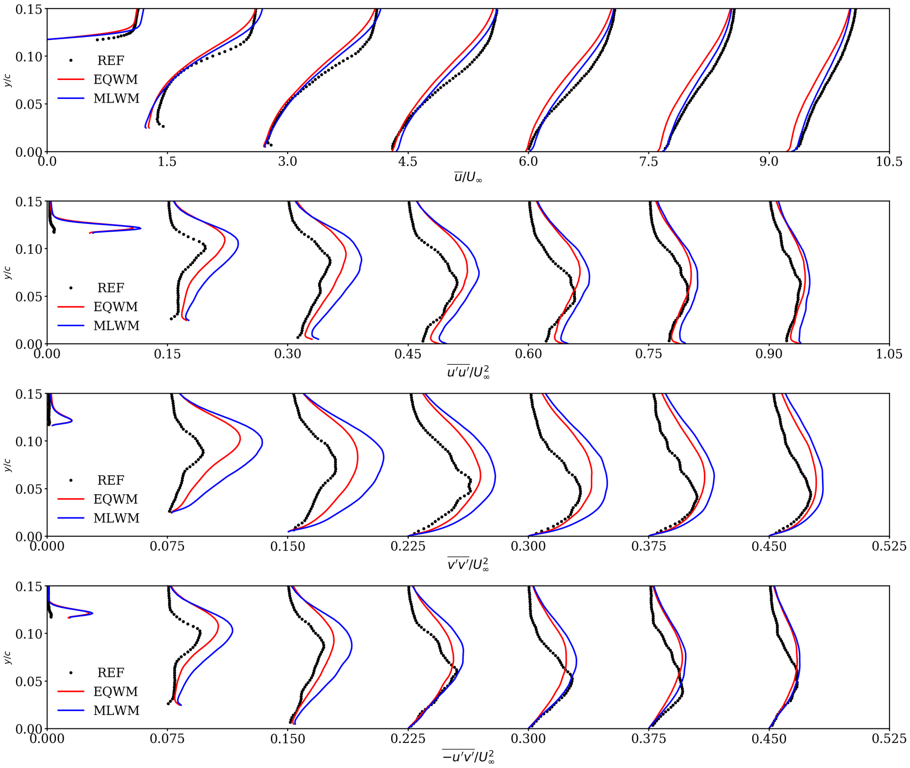


Fig. 9: Mean stream-wise velocity and Reynolds Stress at stream-wise positions: x/c = {0.65, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3}. The profiles are respectively shifted by $\varDelta = 1.5$, $\varDelta = 0.15$, $\varDelta = 0.075$ and $\varDelta = 0.075$ units.

Future works will be in the direction of teaching the model Non-Equilibrium phenomena.

5 Acknowledgment

This work was funded as part of the European Project HiFi-TURB which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814837. Sarath Radhakrishnan acknowledges the financial support by the Ministerio de Ciencia y Innovación y Universidades, for the grant, Ayudas para contratos predoctorales para la formación de doctores(Ref: BES-2017-081982). Oriol Lehmkuhl has been partially supported by a Ramon y Cajal postdoctoral contract (Ref: RYC2018-025949-I).We also acknowledge the Barcelona Supercomputing Center for awarding us access to the MareNostrum IV machine based in Barcelona, Spain.

Bibliography

[1] Haecheon Choi and Parviz Moin. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of fluids*, 24(1):011702, 2012.

[2] J Calafell, FX Trias, O Lehmkuhl, and A Oliva. A time-average filtering technique to improve the efficiency of two-layer wall models for large eddy simulation in complex geometries. *Computers & Fluids*, 188:44–59, 2019.

[3] Ugo Piomelli. Wall-layer models for large-eddy simulations. *Progress in Aerospace Sciences*, 44(6):437 – 446, 2008. ISSN 0376-0421. https://doi.org/https://doi.org/10.1016/j.paerosci.2008.06.001. URL http://www.sciencedirect.com/science/article/pii/S037604210800047X. Large Eddy Simulation - Current Capabilities and Areas of Needed Research.

[4] Ugo Piomelli and Elias Balaras. Wall-layer models for large-eddy simulations. *Annual review of fluid mechanics*, 34(1):349–374, 2002.

[5] Johan LARSSON, Soshi KAWAI, Julien BODART, and Ivan BERMEJO-MORENO. Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews*, 3(1):15–00418–15–00418, 2016. https://doi.org/10.1299/mer.15-00418.

[6] Sanjeeb T. Bose and George Ilhwan Park. Wall-modeled large-eddy simulation for complex turbulent flows. *Annual Review of Fluid Mechanics*, 50(1): 535–561, 2018. https://doi.org/10.1146/annurev-fluid-122316-045241. URL https://doi.org/10.1146/annurev-fluid-122316-045241.

[7] W. Cheng, D. I. Pullin, R. Samtaney, W. Zhang, and W. Gao. Large-eddy simulation of flow over a cylinder with re_D from 3.9×10^3 to 8.5×10^5 : a skin-friction perspective. *Journal of Fluid Mechanics*, 820:121–158, 2017. https://doi.org/10.1017/jfm.2017.172.

[8] J. Larsson, S. Laurence, I. Bermejo-Moreno, J. Bodart, S. Karl, and R. Vicquelin. Incipient thermal choking and stable shock-train formation in the heat-release region of a scramjet combustor. part ii: Large eddy simulations. *Combustion and Flame*, 162:907–920, 2015.

[9] Prahladh S. Iyer, George I. Park, and Mujeeb R. Malik. Application of Wall-modeled LES to Turbulent Separated Flows. In *APS Division of Fluid Dynamics Meeting Abstracts*, APS Meeting Abstracts, page G33.004, November 2016.

[10] X. I. A. Yang, J. Urzay, S. Bose, and P. Moin. Aerodynamic heating in wall-modeled large-eddy simulation of high-speed flows. *AIAA Journal*, 56(2):731–742, 2018. https://doi.org/10.2514/1.J056240. URL https://doi.org/10.2514/1.J056240.

[11] XIA Yang, J Urzay, and P Moin. Heat-transfer rates in equilibrium-wall-modeled les of supersonic turbulent flows. *Annual Research Briefs, Center for Turbulence Research, Stanford University*, pages 3–15, 2016.

[12] Jeffrey Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences, 2014.

[13] Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[14] Brendan Tracey, Karthik Duraisamy, and Juan Alonso. *Application of Supervised Learning to Quantify Uncertainties in Turbulence and Combustion Modeling*. https://doi.org/10.2514/6.2013-259. URL https://arc.aiaa.org/doi/abs/10.2514/6.2013-259.

[15] Brendan D Tracey, Karthikeyan Duraisamy, and Juan J Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA aerospace sciences meeting*, page 1287, 2015.

[16] Karthikeyan Duraisamy, Ze J Zhang, and Anand Pratap Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*, page 1284, 2015.

[17] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016. https://doi.org/10.1017/jfm.2016.615.

[18] Jin-Long Wu, Jian-Xun Wang, and Heng Xiao. A bayesian calibration–prediction method for reducing model-form uncertainties with application in rans simulations. *Flow, Turbulence and Combustion*, 97(3):761–786, 2016.

[19] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2(3):034603, 2017.

[20] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.

[21] Ryan N King, Peter E Hamlington, and Werner JA Dahm. Autonomic closure for turbulence simulations. *Physical Review E*, 93(3):031301, 2016.

[22] Masataka Gamahara and Yuji Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2(5):054604, 2017.

[23] Romit Maulik and Omer San. A neural network approach for the blind deconvolution of turbulent flows. *arXiv preprint arXiv:1706.00912*, 2017.

[24] Antoine Vollant, Guillaume Balarac, and C Corre. Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures. *Journal of Turbulence*, 18(9):854–878, 2017.

[25] Ze Jia Zhang and Karthikeyan Duraisamy. Machine learning methods for data-driven turbulence modeling. In *22nd AIAA Computational Fluid Dynamics Conference*, page 2460, 2015.

[26] Ming Ma, Jiacai Lu, and Gretar Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system. *Physics of Fluids*, 27(9):092101, 2015.

[27] Ming Ma, Jiacai Lu, and Gretar Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *International Journal of Multiphase Flow*, 85:336–347, 2016.

[28] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[29] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1): 477–508, 2020. https://doi.org/10.1146/annurev-fluid-010719-060214. URL https://doi.org/10.1146/annurev-fluid-010719-060214.

[30] X. I. A. Yang, S. Zafar, J.-X. Wang, and H. Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids*, 4:034602, Mar 2019. https://doi.org/10.1103/PhysRevFluids.4.034602. URL https://link.aps.org/doi/10.1103/PhysRevFluids.4.034602.

[31] Adrián Lozano-Durán and Hyunji Jane Bae. Self-critical machine-learning wall-modeled les for external aerodynamics. *arXiv preprint arXiv:2012.10005*, 2020.

[32] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. https://doi.org/10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[33] Herbert Owen, Georgios Chrysokentis, Matias Avila, Daniel Mira, Guillaume Houzeaux, Ricard Borrell, Juan Carlos Cajas, and Oriol Lehmkuhl. Wall-modeled large-eddy simulation in a finite element framework. *International Journal for Numerical Methods in Fluids*, 2019. https://doi.org/10.1002/fld.4770. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4770.

[34] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[35] Hermann Schlichting and Klaus Gersten. *Boundary-layer theory*. Springer, 2016.

[36] Mariano Vazquez, Guillaume Houzeaux, Seid Koric, Antoni Artigues, Jazmin Aguado-Sierra, Ruth Arís, Daniel Mira, Hadrien Calmet, Fernando Cucchietti, Herbert Owen, Ahmed Taha, Evan Dering Burness, José María Cela, and Mateo Valero. Alya: Multiphysics engineering simulation towards exascale. *Journal of Computational Science*, 14:15 – 27, 2016. ISSN 1877-7503. https://doi.org/https://doi.org/10.1016/j.jocs.2015.12.007. URL http://www.sciencedirect.com/science/article/pii/S1877750315300521. The Route to Exascale: Novel Mathematical Methods, Scalable Algorithms and Computational Science Skills.

[37] A. W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Phys. Fluids*, 16(10):3670–3681, 2004.

[38] Oriol Lehmkuhl, Ugo Piomelli, and Guillaume Houzeaux. On the extension of the integral length-scale approximation model to complex geometries. *International Journal of Heat and Fluid Flow*, 78:108422, 2019.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pas-

sos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[40] Soshi Kawai and Johan Larsson. Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy. *Physics of Fluids*, 24 (1):015105, 2012.

[41] Sergio Hoyas and Javier Jiménez. Scaling of the velocity fluctuations in turbulent channels up to re τ= 2003. *Physics of fluids*, 18(1):011702, 2006.

[42] C. L. Rumsey, T. B. Gatski, W. L. Sellers, V. N. Vasta, and S. A. Viken. Summary of the 2004 computational fluid dynamics validation workshop on synthetic jets. *AIAA Journal*, 44(2):194–207, 2006. https://doi.org/10.2514/1.12957. URL https://doi.org/10.2514/1.12957.

[43] George Ilhwan Park. Wall-modeled large-eddy simulation of a separated flow over the nasa wall-mounted hump by. 2015.

[44] J. W. Naughton, S. Viken, and D. Greenblatt. Skin friction measurements on the nasa hump model. *AIAA Journal*, 44(6):1255–1265, 2006. https://doi.org/10.2514/1.14192. URL https://doi.org/10.2514/1.14192.

[45] Andreas Kempf, Markus Klein, and Johannes Janicka. Efficient generation of initial-and inflow-conditions for transient turbulent flows in arbitrary geometries. *Flow, Turbulence and combustion*, 74(1):67–84, 2005.