COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0079

Using Neural Networks to Compute the Divergence of the Reynolds Stress Tensor with Fundamental Mean Flow Properties

Thales Arantes de Castelo Branco e Souza
thalescastelobranco@gmail.com, t148997@g.unicamp.br
William Roberto Wolf
wolf@fem.unicamp.br
University of Campinas

Luiz Augusto Camargo Aranha Schiavo
luiz.schiavo@northumbria.ac.uk
Northumbria University

Abstract. *Numerical simulations of complex flows commonly rely on the solution of the Reynolds-Averaged Navier-Stokes (RANS) equations due to the high computational cost of high-fidelity simulations. However, RANS simulations model the entire range of turbulent scales, requiring closure models for the Reynolds stresses. This paper introduces an alternative approach that avoids the use of turbulence models for correcting the Reynolds stress field. Instead, it employs machine learning techniques based on neural networks (NNs) to directly calculate the divergence of the Reynolds stress tensor using mean flow properties (pressure, velocity, and their gradients) which are readily available from a RANS calculation. The proposed NN model successfully reproduces the divergence of the Reynolds stress tensor in a turbulent flow within a convergent-divergent channel, where separation and reattachment occur downstream of a smooth bump. The strong physical correlation between the mean properties and the divergence of the Reynolds stress tensor enables the machine learning solution to be data-efficient and computationally inexpensive. Accurate reconstructions of the divergence of the Reynolds stress tensor are achieved with only 20% of the training data. Training strategies are employed to ensure rotational invariance of the solutions. Finally, the model's interpolation capacity is tested using different Reynolds numbers for training and testing, demonstrating its ability to learn the physics of turbulent flows from fundamental properties from a RANS formulation.*

Keywords: *Machine learning, artificial neural networks, turbulence modeling, CFD*

1. INTRODUCTION

The physics of complex turbulent flows can be investigated using high-fidelity computational fluid dynamics (CFD) simulations. In these cases, direct numerical simulations (DNS) or large-eddy simulations (LES) are typically employed. Although these methods are accurate, their computational cost is expensive, sometimes requiring millions of computing hours depending on the flow setup. Therefore, Reynolds-averaged Navier-Stokes (RANS) formulations with turbulence closure models are used in most industrial applications since they require lower computational effort compared to LES and DNS, making it possible to simulate complex geometries at high Reynolds numbers. However, RANS modeling has always lacked accuracy in complex flows, especially those with separation and reattachment, as described by Tracey *et al.* (2015), Ling *et al.* (2016b), Jesus *et al.* (2015) and de Jesus *et al.* (2016).

In recent years, with advances in data science, there has been an increasing interest in improving the accuracy of RANS turbulence models by employing machine learning techniques. Ling *et al.* (2016b) built a neural network (NN) model using databases of canonical flows, where the invariants of the velocity gradient tensor were used as input features for training the models aiming to predict the Reynolds stress anisotropy tensor. Applying the calculated stress tensor on two RANS models, they verified that the NN surrogates were able to perform better compared to the RANS linear and quadratic eddy viscosity models, LEVM and QEVM, respectively. Such results were obtained through comparison to the ground truth results provided by LES and DNS.

One of the key points debated in the literature of machine learning for turbulence closure addresses the capability of the NN models to maintain their Galilean invariance. Ling *et al.* (2016a) have previously studied this topic using either invariants of relevant fluid flow tensors computed from the turbulence databases, or using raw data combined with multiple transformations, e.g., rotation and translation of the domain, forcing the models to learn these respective invariances. This study showed that models based on invariant tensors are cheaper to compute than working with modifications of the raw data, basically because the databases used are smaller. The previous authors verified small errors when using invariant modeling, but the errors could be reduced as the networks were trained using more cases with different transformations.

Moreover, a comparison between neural networks and random forest models was also made, indicating that the former is better at learning invariant properties since deep learning is more efficient when applied to non-linear parameters.

Tracey *et al.* (2015) presented a similar study in terms of invariance analysis using a neural network associated with the Spalart-Allmaras turbulence model to predict the turbulent transport in flows over NACA airfoils and flatplates. They showed that machine learning methods could be successfully employed to improve RANS modeling calibration. Milani *et al.* (2017) used an analogous method applied to the study of film cooling with a jet in crossflow. They used a random forest approach to improve a two-equation RANS model associated with the gradient diffusion hypothesis (GDH) closure. These authors also used invariants of tensors as input features, and the main objective was to predict the turbulent thermal diffusivity and apply it in the GDH formulation. These authors also made a test with an extrapolation for an untrained flow. The results, despite inferior than those cases for which the networks were trained, demonstrated that the method can interpolate and extrapolate between different flow conditions, still providing better results than the usual RANS modeling.

Recently, Tang *et al.* (2023) proposed an approach for improving the Boussinesq assumption by adding an extra non-linear stress tensor in the equation. The linear part of the Reynolds stress tensor was resolved with an usual LEVM model, while the added non-linear tensor was learned by a Bayesian deep neural network (BNN). The database used to train the BNN consisted of four canonical flows: a square cylinder in a channel, a square duct channel, tandem cylinders in a channel, and a periodic hill. The authors applied the model to four other canonical flows: a backward-facing step, a convergent-divergent channel, a periodic hill, and a surface-mounted cube. They concluded that the BNN can perform better than usual RANS closure. However, as the test geometry differed from the trained cases, the quality of the results decreased.

In this work, we build NN models for predicting the divergence of the Reynolds stresses for a turbulent flow in a channel with an installed bump. The models are trained with databases from high-fidelity simulations from Schiavo *et al.* (2017), Marquillie *et al.* (2011) and Schiavo *et al.* (2015) for Reynolds numbers $Re_\tau = 180$, 617 and 950. Here, Re_τ denotes the Reynolds number based on the friction velocity. For the intermediate Reynolds number, flow data is computed from a DNS, while for the other cases, a wall-resolving LES was performed. The present setup is relevant for turbulence modeling because the flows are subjected to varying favorable and adverse pressure gradients, and display separation and reattachment. Similarly to previous studies, the main purpose here is to improve the accuracy of the RANS formulation using machine learning. However, differently from the previous authors, we work directly on the reconstruction of the divergence of the Reynolds stress tensor, instead of computing the constants from RANS turbulence models. Therefore, we avoid errors introduced by the Boussinesq approximation. An investigation in terms of the invariance of the models with respect to rotation is presented, and the capability of the models to interpolate results is assessed.

2. METHODOLOGY

2.1 Theoretical Formulation

The Reynolds-averaged Navier-Stokes equations for an incompressible flow can be written in a non-dimensional form as

$$\begin{aligned}
&\text{Continuity equation:} && \frac{\partial \overline{u}_j}{\partial x_j} = 0\,, \\
&\text{Momentum equations:} && \frac{\partial \rho \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{p}}{\partial x_i} - \frac{1}{Re}\frac{\partial^2 \overline{u}_i}{\partial x_j\, \partial x_j} + \frac{\partial \overline{u_i' u_j'}}{\partial x_j} = 0\,, \\
&\text{Poisson equation for pressure:} && \frac{\partial^2 \overline{p}}{\partial x_j\, \partial x_j} = -\frac{\partial}{\partial x_i}\left(\overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j}\right) - \frac{\partial^2 \overline{u_i' u_j'}}{\partial x_i\, \partial x_j}\,.
\end{aligned} \tag{1}$$

Here, u_i is the i-th component of the velocity vector, p is the static pressure, and Re is the reference Reynolds number. The $\overline{(\,)}$ represents time averaged quantities, while the $(\,)'$ denotes the fluctuation terms from the Reynolds-averaging procedure.

The equations above are solved in the RANS methodology for incompressible flows. In such case, the mean flow quantities are solved, while the terms of the Reynolds stress tensor, $\overline{u_i' u_j'}$, require additional modeling. The closure of these terms, also called turbulence modeling, seeks to approximate the Reynolds stresses, and it can be divided into two main approaches. The first is based on the Boussinesq hypothesis and consists of calculating an eddy (turbulent) viscosity, while the second relies on solving balance equations for the individual Reynolds stresses. In both cases, the closure models are used to find an approximation for the Reynolds stress tensor, and they often introduce intrinsic modeling errors. On one hand, the Boussinesq approximation assumes that the momentum transfer from the turbulence fluctuations is proportional to the mean strain rate tensor, which is incorrect for several practical flows. On the other hand, model transport equations are solved for the individual Reynolds stresses and for an additional quantity (e.g., the turbulent dissipation) in the Reynolds-stress closure approach. This methodology also requires additional modeling since several terms involving higher-order moments are unknown in the transport equations.

In the literature, several studies are concerned with applying machine learning to improve the predictive capability of

eddy viscosity models. In such cases, the several constants appearing in the classical turbulence models are fine-tuned using neural networks or regression trees. Although this is an ingenious procedure to enhance the turbulence models, it is important to remember that such models are still associated with intrinsic errors. Moreover, most turbulence models require the solutions of complex partial differential equations, which can have numerical stability issues besides several sets of constants to tune.

In the present work, we propose a simplified approach where a neural network (NN) model is trained to directly estimate the components of the divergence of the Reynolds stress tensor as they appear in the RANS equations. For this, we employ fundamental flow quantities as input for the NN training. Applying this procedure, we can reduce the number of unknowns from 6 to 2 in a spanwise-homogeneous flow. This procedure also avoids the Boussinesq hypothesis or any other closure model, and the following section shows the details of the NN model.

2.2 Neural Network Model

In this study, we employ a neural network to model the divergence of the Reynolds stresses as they appear in the Navier-Stokes equations. The flowchart shown in Figure 1 represents the entire procedure, including the initial post-processing of high-fidelity simulation data, the selection of input features, as well as the model architecture and its output. The first step consists in extracting relevant flow features from large-scale LES and DNS databases. Here, we work with mean flow quantities such as velocity components, pressure, and their derivatives. In order to improve the present predictive capability, data augmentation is performed so the models can learn rotational invariances of the flow.

The next step consists of selecting the input features for the model training. Several studies employ invariants of the velocity gradient tensor and higher-order terms composed of these invariants. This approach was tested, but the intrinsic noise associated with these quantities led to large associated errors. Hence, the mean flow quantities are used instead since these features have smoother features and will be shown in the next section to provide good results. The NN architecture consists of 6 hidden layers with 256 neurons per layer. More details about the parameters employed in the NN model training can also be seen in Table 1.

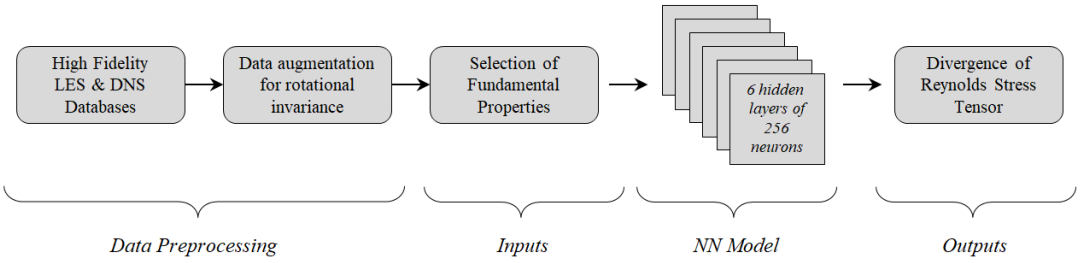

Figure 1. Schematic flowchart describing the data preprocessing and the neural networks modeling.

The proposed framework of supervised learning uses the NN model to infer the 2 components of the divergence of the Reynolds stress (outputs), utilizing 9 fundamental flow properties (input features) available in RANS solvers. The present technique consists in a calibration (training) with sample cases, in which the true values of the target and the features are given by LES or DNS data. Usually, a considerable part of the data (50-70%) has to be used in this stage. However, here, only 10-20% of the dataset is necessary to accurately predict all the outputs, ensuring rotational invariance. This accuracy is possible because a high physical correlation between features and targets. Therefore, it is easier for the model to learn the complex nonlinear behavior of the turbulent flow.

The second phase, called testing, consists of using the model to predict unseen values of the flow and comparing results with the true values. At that stage, it is already possible to measure the error of the model. Several metrics can be used to evaluate the performance of a model, and it is vital to choose one that better represents its accuracy, i.e., when the error metric diminishes, the solution quality improves. In this work, the best metric found in the tests is the Root Mean Squared Error (RMSE), and all results in the following sections will use such method to measure the performance of the model.

The machine learning algorithms employed in this work use the Keras and Tensorflow libraries described in Chollet *et al.* (2015) and Abadi *et al.* (2015). Several architectures were evaluated in the construction of the neural network, and the best one (with the smallest RMSE results) was a standard Multi-Layer Perceptron (MLP) feedforward artificial neural network. The parameters and model specifications were defined by sensitivity analysis after hundreds of experiments. The final specifications of the neural network that had the best RMSE performance are shown in Table 1.

Table 1. Main parameters of the Multi-Layer Perceptron Artificial Neural Network model.

Neural Network Parameter	**Specification**
Number of inputs	9
Number of outputs	2
Number of hidden layers	6
Number of neurons in each hidden layer	256
Number of epochs to convergence	30
Activation function	Rectified Linear Unit (ReLU)
Loss function	Mean Squared Error (MSE)
Optimizer	Adam

3. RESULTS

3.1 Computational Datasets

The analyses presented in this work employ the computational datasets from the high-fidelity simulations of Schiavo *et al.* (2017), Marquillie *et al.* (2011) and Schiavo *et al.* (2015) for Reynolds numbers $Re_\tau = 180$, 617 and 950. After some tests, we select 9 parameters in the RANS equations as inputs to the neural network (NN) model. Figure 2(a) shows the streamlines with velocity vectors for the $Re_\tau = 180$ flow to highlight the details of the present channel flow including the smooth bump. Figures 2(b – j) show the NN inputs, which consist of the mean velocity components, pressure, and their respective gradients.

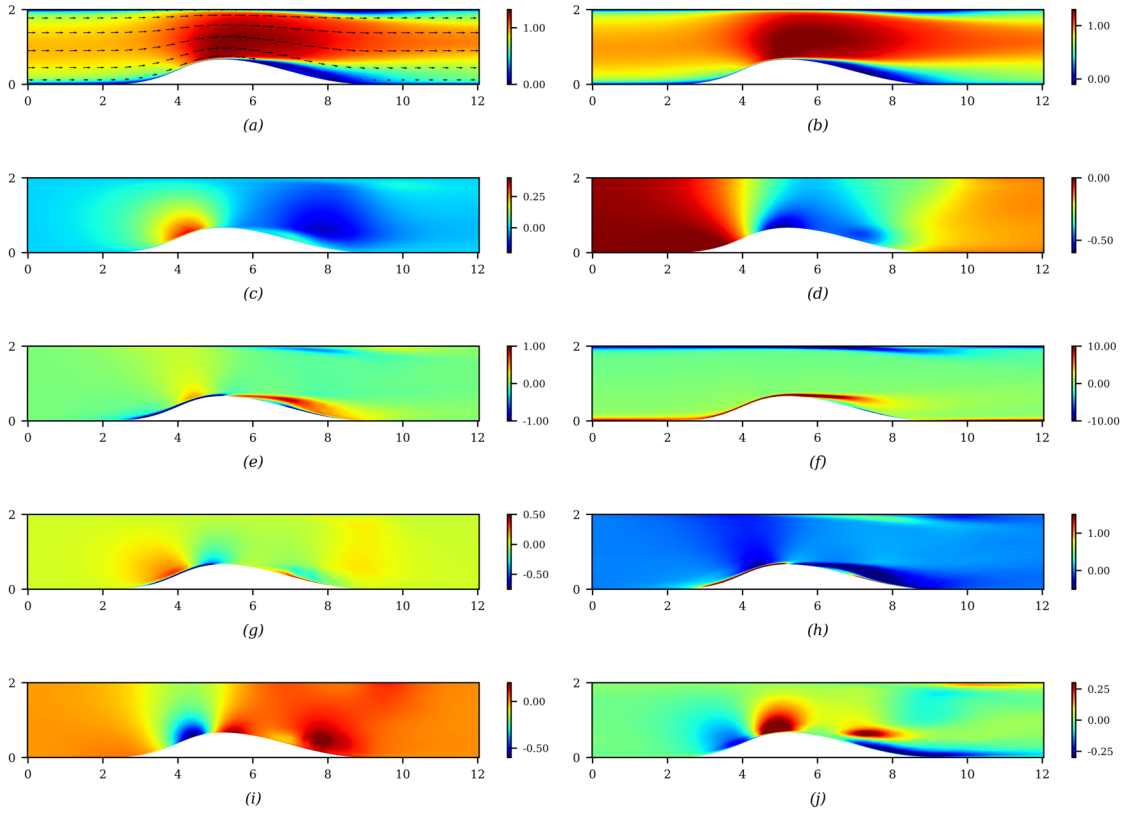

Figure 2. (a) Streamlimes and velocity vectors with contours of velocity magnitude, (b) $\overline{u_1}$, (c) $\overline{u_2}$, (d) $\overline{p}$, (e) $\frac{\partial \overline{u}_1}{\partial x_1}$, (f) $\frac{\partial \overline{u}_1}{\partial x_2}$, (g) $\frac{\partial \overline{u}_2}{\partial x_1}$, (h) $\frac{\partial \overline{u}_2}{\partial x_2}$, (i) $\frac{\partial \overline{p}}{\partial x_1}$,(j) $\frac{\partial \overline{p}}{\partial x_2}$.

The inputs shown in Figure 2 are consolidated in a tabular form, i.e., the features are disposed as columns where each grid point is represented by a line of 9 features, to create the main database. We also apply a data-augmentation process, where the training data is increased by adding copies of the original data in a rotated coordinate system as discussed in previous sections. The data-augmentation process is a widespread technique in data science to improve issues with overfitting and also ensure rotational invariance. The channel geometry is rotated, and new vectors and tensors are computed applying a rotation matrix to generate new inputs.

Machine learning models always require a split of the database in train and test sets since that the model should learn with the train dataset and be used to predict with the test dataset. Traditionally, 70% to 90% of the database is used to train the model, and the remaining data is used for testing. However, the DNS and LES employ large meshes, and we verify that the proposed method requires substantially less data in training to achieve convergence compared to traditional machine learning approaches. This is due to the high physical correlation between the input flow properties and their outputs. Hence, the tests conducted in this paper employ 1%, 5%, 20%, and 40% of the data, as indicated in Figure 3.

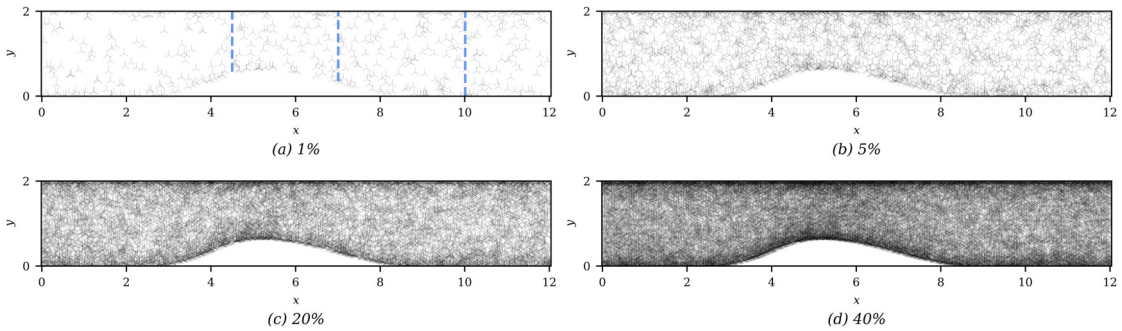

Figure 3. (a) Schematic diagram of the 3 sections used for comparison of results at x=[4.5, 7.0 10.0], and sample distribution of 1% of the database, (b) schematic diagrams showing 5% of the database, (c) 20% of the database, and (d) 40% of the database.

The results presented in the following sections are divided into two parts: the first test is presented in section 3.2, where the data is separated into 3 datasets, one for each Re_τ. The training and predictions are performed for the same Reynolds number of each database. This study evaluates the predicting capability of the NN to obtain results in a rotated coordinate system which is not provided in the training dataset. In the second part, described in section 3.3, the $Re_\tau = 180$ and $Re_\tau = 950$ flows are merged into a single database, which test the NN capability to predict results at $Re_\tau = 617$, at a rotation angle that is not contained in the dataset, i.e., an entirely different flow condition than those used in the training.

3.2 Interpolation of Rotation Angle

In this case, for each Re_τ, three rotation subsets (8, 15 and 35 equally spaced rotations) and four data ratio subsets (1%, 5%, 20% and 40% of the CFD database) are combined, giving 12 different training datasets, for a performance evaluation. The model comparison consists in measuring the associated error using the presented NN methodology to predict the entire outputs at all rotations. This procedure allows an accurate evaluation of the rotational invariance. The angle $\pi/3$ is set as the benchmark for all comparisons in this work since it is a common interpolation for all subsets. As previously described, the used performance metric is the Root Mean Squared Error (RMSE).

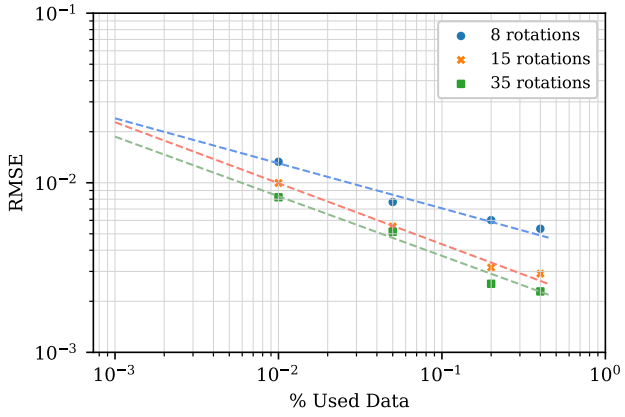

Figure 4. RMSE variation per used data ratio for $Re_\tau = 180$.

Machine learning models usually improve as more data are provided in the training stage. Then, one should expect that the RMSE decreases for the model as more rotations are added. Figure 4 shows the RMSE as a function of the percentage of data used in the NN training. We can notice that the error decays linearly in the logarithmic plot, showing that the RMSE values follow a power law. We can also observe a clear improvement in the NN from increasing the number of rotations from 8 to 15. However, the error decay from 15 to 35 rotations is significantly smaller, indicating convergence of the model. As more data and rotations are provided, the rotational invariance is achieved, and the models are able to

predict the full set of rotations. For an RMSE < 1%, the proposed methodology employing 8 rotations converges with 5% of the database, while using 15 and 35 rotations leads to similar results with 1% of the data. As will be shown, excellent results are obtained when the model is trained with 20% of the data using 15 rotations.

Figure 5 presents contours of the components of divergence of the Reynolds stress tensor for each direction, and Fig. 6 shows results computed for the vertical slices in the channel geometry at x =4.5, 7, and 10, as shown in Fig. 3. It is possible to observe that the model can reconstruct the two outputs with good accuracy upstream and downstream of the bump. Some fine oscillations observed in the CFD results are smoothed by the NN model prediction, but the overall magnitudes and shapes have good agreement.

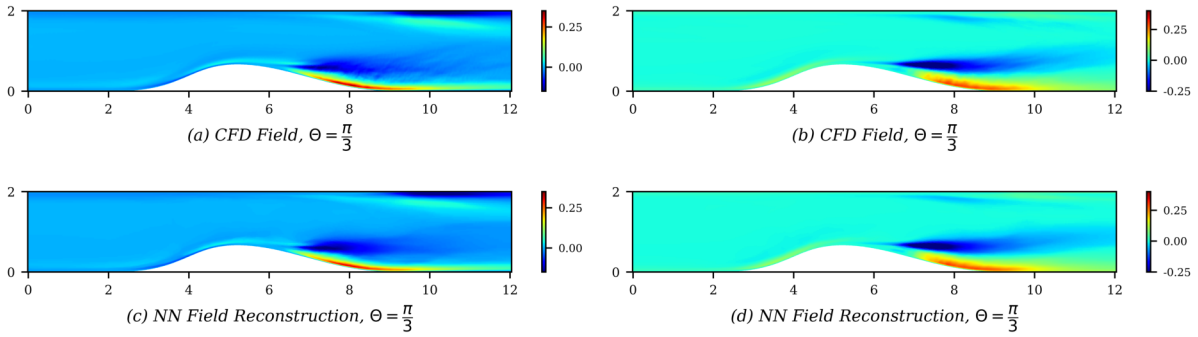

Figure 5. Reconstruction of the divergence of the Reynolds stress tensor for $Re_\tau = 180$ model. The NN model is trained with 20% of the data and 35 rotations. Figures (a) and (c) refer to the $i = 1$ direction, while figures (b) and (d) refer to $i = 2$.

Neural networks usually have more difficulty in predicting the more oscillatory behavior of the data. Therefore, in Figure 6, Model 1 computed with 5% of the data, using 8 rotations, has a poorer agreement with the true results close to the walls, where the gradients are stepper. However, as more data is provided in Model 2, which uses 20% of the dataset information and 35 rotations, it is possible to notice an improvement in the quality of the results. For this case, the NN model can predict all the fine scale variations of the flow adequately.

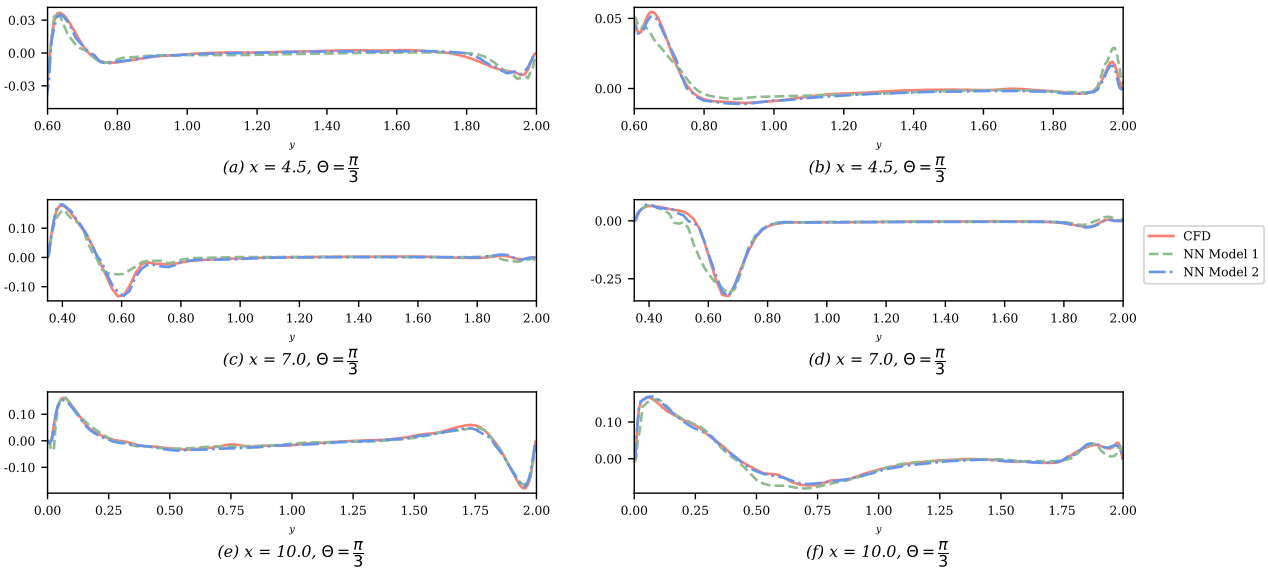

Figure 6. Comparison of the divergence of the Reynolds stress tensor for each direction for $Re_\tau = 180$. Models 1 and 2 use 5% and 20% of the data with 8 and 35 rotations, respectively. Figures (a), (c) and (e) refer to the $i = 1$ direction, while figures (b), (d) and (f) refer to $i = 2$.

3.3 Interpolation of the Reynolds Number

In this section, the neural network model is trained with data from the $Re_\tau = 180$ and $Re_\tau = 950$ simulations to predict the divergence of the Reynolds stress tensor for the flow with $Re_\tau = 617$. This case is particularly more complex than the one presented in the previous section, as it involves interpolating physical dynamic properties, besides

the rotational interpolation. The flow simulations with a higher Reynolds number require a finer CFD mesh, and hence, generate a base with more elements than a simulation with a lower Reynolds number. Thus, if we use 10% of the points for each flow, we would have a larger amount of data from the $Re_\tau = 950$ flow than that for $Re_\tau = 180$. Therefore, when combining results from different simulations, it is necessary to carry out a data-balancing process. Basically, this technique ensures that the training set has the same number of inputs for both flows.

Figure 7 shows divergence of the Reynolds stress tensor components for $Re_\tau = 617$, reconstructed with the proposed methodology. Here, we use 20% of the total training base and 35 rotations to generate the model. It can be noted that the model is successful in reconstructing the convergent-divergent channel outputs, both upstream and downstream the bump, getting close to the solution computed from CFD. As expected, the separation region is more complex to calculate because due to the variations in the Reynolds stresses. Figure 8 shows the comparison between the NN model and CFD solutions at the 3 sections described in Figure 3(a). It is possible to observe that the model can adequately solve this more complex case where an interpolation in terms of the Reynolds interpolation is sought. The RMSE error of this case is $7.80\text{e}{-3}$. Therefore, it is possible to notice that the proposed methodology of using a neural network model with only fundamental mean flow properties is capable of calculating the divergence of the Reynolds stress tensor in convergent-divergent channels with different flow conditions.

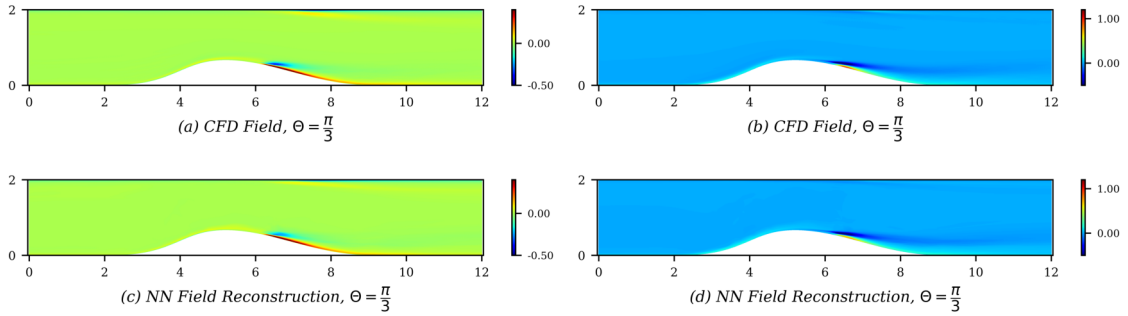

Figure 7. Reconstruction of the divergence of the Reynolds stress tensor for $Re_\tau = 617$. The NN model is trained with 20% of the data and 35 rotations. Figures (a) and (c) refer to the $i = 1$ direction, while figures (b) and (d) refer to $i = 2$.

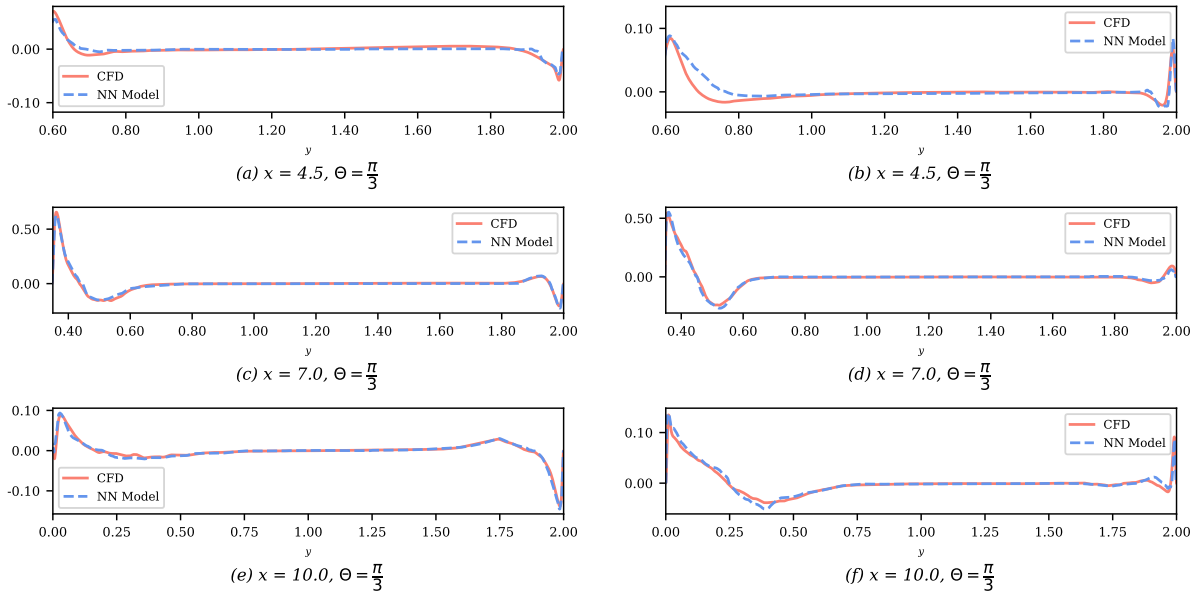

Figure 8. Comparison of the divergence of the Reynolds stress tensor for each direction $Re_\tau = 617$. The NN Model uses 20% of the data with 35 rotations. Figures (a), (c) and (e) refer to the $i = 1$ direction, while figures (b), (d) and (f) refer to $i = 2$.

4. CONCLUSIONS

In this work, a novel machine learning approach is presented for the closure problem typical of RANS equations. Differently from other studies that employ machine learning for improving turbulence models, we develop a model that

uses the fundamental flow properties to directly compute the divergence of the Reynolds stress tensor. The proposed methodology is tested for a convergent-divergent channel flow, being applied in two case studies: (1) interpolation of a rotated flow at an angle which is unknown to the model, and (2) interpolation of a Reynolds number for which the model is not trained.

In the first case study, it is possible to observe the excellent solution obtained by the NN model, since the divergence of the Reynolds stress tensor is accurately reconstructed, capturing the major effects of separation and reattachment of the turbulent flow. The second case study has an additional difficulty, which consists of interpolating between different Reynolds numbers. Thus, a model is trained with data from $Re_\tau = 180$ and 950 flows, and it is used to calculate the divergence of the Reynolds stress tensor for a flow with $Re_\tau = 617$. For this case, data balancing is employed as the simulations with different Reynolds have different grid sizes, and this procedure is necessary to provide a similar quantity of inputs to the neural network. The results obtained are also in good agreement with those obtained by CFD and the model is able to completely reconstruct the outputs, capturing fine wall turbulence effects, with a root mean square error of the order of 10^{-3}.

5. ACKNOWLEDGEMENTS

The authors acknowledge Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, for supporting the present work under research grants No. 2013/08293-7 and 2021/06448-0.

6. REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. *et al.*, 2015. "TensorFlow: Large-scale machine learning on heterogeneous systems". URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Chollet, F. *et al.*, 2015. "Keras". `https://keras.io`.

de Jesus, A.B., Schiavo, L.A.C.A., Azevedo, J.L. and Laval, J.P., 2016. "Adverse pressure gradients and curvature effects in turbulent channel flows". In M. Stanislas, J. Jimenez and I. Marusic, eds., *Progress in Wall Turbulence 2*. Springer International Publishing, Cham, pp. 295–306.

Jesus, A.B., Schiavo, L.A.C.A., Azevedo, J.L.F. and Laval, J.P., 2015. "Further studies of adverse pressure gradient effects in turbulent channel flows". In AIAA Paper No. 2015-2619*, Proceedings of the 22nd AIAA Computational Fluid Dynamics Conference*. Dallas, TX. doi:http://dx.doi.org/10.2514/6.2015-2619.

Ling, J., Jones, R. and Templeton, J., 2016a. "Machine learning strategies for systems with invariance properties". *Journal of Computational Physics*, Vol. 318. doi:10.1016/j.jcp.2016.05.003.

Ling, J., Kurzawski, A. and Templeton, J., 2016b. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". *Journal of Fluid Mechanics*, Vol. 807, pp. 155–166. doi:10.1017/jfm.2016.615.

Marquillie, M., Ehrenstein, U. and Laval, J.P., 2011. "Instability of streaks in wall turbulence with adverse pressure gradient". *Journal of Fluid Mechanics*, Vol. 681, pp. 205–240.

Milani, P., Ling, J., Sáez-Mischlich, G., Bodart, J. and Eaton, J., 2017. "A machine learning approach for determining the turbulent diffusivity in film cooling flows". *Journal of Turbomachinery*, Vol. 140. doi:10.1115/1.4038275.

Schiavo, L.A.C.A., Jesus, A.B., Azevedo, J.L.F. and Wolf, W.R., 2015. "Large eddy simulations of convergent-divergent channel flows at moderate Reynolds numbers". *International Journal of Heat and Fluid Flow*, Vol. 56, pp. 137–151. ISSN 0142-727X. doi:http://dx.doi.org/10.1016/j.ijheatfluidflow.2015.07.006. URL `http://www.sciencedirect.com/science/article/pii/S0142727X15000880`.

Schiavo, L.A.C.A., Wolf, W.R. and Azevedo, J.L.F., 2017. "Turbulent kinetic energy budgets in wall bounded flows with pressure gradients and separation". *Physics of Fluids*, Vol. 29, No. 11, p. 115108. doi:10.1063/1.4992793. URL `https://doi.org/10.1063/1.4992793`.

Tang, H., Wang, Y., Wang, T., Tian, L. and Qian, Y., 2023. "Data-driven Reynolds-averaged turbulence modeling with generalizable non-linear correction and uncertainty quantification using Bayesian deep learning". *Physics of Fluids*, Vol. 35, No. 5. ISSN 1070-6631. doi:10.1063/5.0149547. URL `https://doi.org/10.1063/5.0149547`. 055119.

Tracey, B., Duraisamy, K. and Alonso, J., 2015. "A machine learning strategy to assist turbulence model development". In AIAA Paper No. 2015-3642*, Proceedings of the 46th AIAA Fluid Dynamics Conference*. doi:10.2514/6.2016-3642.

7. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.