encit 2020

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

18th Brazilian Congress of Thermal Sciences and Engineering
November 16–20, 2020 (Online)

ENC-2020-0401
MACHINE LEARNING PREDICTIONS OF THE TURBULENT FLOW IN THE SQUARE-DUCT EMPLOYING A TRANSPORT EQUATION FOR THE REYNOLDS STRESS TENSOR

Matheus de Souza Santos Macedo [1]
Roney Leon Thompson [2]
Matheus Altomare Cruz [3]
[1,2,3] Universidade Federal do Rio de Janeiro
Centro de Tecnologia - Rua Horácio Macedo, Bloco G - Cidade Universitária da Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ
[1] matheus.macedo@mecanica.coppe.ufrj.br, [2] rthompson@mecanica.coppe.ufrj.br, [3] matheusaltomare@ufrj.br

Abstract. *The use of Machine Learning (ML) techniques to correct RANS simulations has been recently explored by a number of works. In the present paper a novel approach for these corrections is introduced. For the first time, the ML target was not a turbulent quantity to be directly injected into the mean-momentum balance. A transport equation for the Reynolds Stress Tensor* $\mathbf{R}$*, fueled by a ML predicted source-term, is presented and employed to correct the turbulent flow in a square-duct. DNS data was used to train Neural Networks (NN) which were employed to predict the source-term. Predictions from the NN were then injected into the RANS environment through a data-driven turbulence model, which coupled the proposed Reynolds Stress' transport equation with the mean-momentum equations and a pressure-velocity correlation. The data-driven model consistently corrected the baseline RANS velocity field and turbulent stresses of the square-duct flow.*

Keywords: *Turbulence, Machine Learning, Square-duct, RSTE, OpenFOAM*

1. INTRODUCTION

The numeric simulation of turbulent flows has been a limitation of Computational Fluid Dynamics (CFD) for many decades. The prohibitively high cost of simulating flows in this regime via Direct Numerical Simulations (DNS) is the main reason why industry still resorts to lower-fidelity simulations. An alternative to DNS, Large-Eddy Simulations (LES) require a fraction of its computational effort while still providing considerably accurate results. LES however is still much more computationally expensive than the popular Reynolds Averaged Navier-Stokes (RANS) models.

RANS models simplify the physics of simulations through a time-averaging procedure of flow's quantities and of the Navier-Stokes equations. This mathematical maneuver greatly diminishes the cost of CFD simulations, but also considerably impairs the quality of results. A promising strategy of coupling high-fidelity simulations data with the computational low-cost of RANS models is the use of Machine Learning (ML) techniques to correct RANS simulations. This becomes possible due to the increasing availability of *DNS*, *LES* and experimental data, which has occurred due to the increasing computational prowess available, not only for simulating problems but also for data storage and sharing purposes.

1.1 Machine Learning Assisted Turbulence

One of the first works to evaluate the possibility of data-driven improvements to RANS simulations was performed by Tracey *et al.* (2015). There, Neural Networks (NN) capabilities in predicting the analytical source-term of the Spallart-Allmaras (SA) one-equation turbulence model are demonstrated.

Most works of this promising field focus on correcting the RANS velocity fields through the direct use of the Reynolds Stress Tensor $\mathbf{R}$ or some type of representation of $\mathbf{R}$, which is later reconstructed and propagated by the momentum equations in a CFD environment. Deep Neural Networks were used by Ling *et al.* (2016) to predict the deviatoric part of $\mathbf{R}$, based on nine different flows' datasets. The predicted deviatoric part of $\mathbf{R}$ was then propagated to the mean flow on a square-duct. The predicted tensor succeeded in correcting the secondary motion on the duct, which is not captured by the popular linear eddy-viscosity RANS models.

In the same context of reconstructing the turbulent stresses by ML techniques, Wang *et al.* (2017) employed Random Forests (RF) to predict the discrepancy $\mathbf{\Delta R}$ between DNS and RANS Reynolds Stresses. Wu *et al.* (2017) explored other

possibilities in the description of the discrepancy $\mathbf{\Delta R}$ and its use in ML predictions. However, both Wang *et al.* (2017) and Wu *et al.* (2017) did not propagate their predicted quantities in a suitable CFD environment for flow correction, due to inconsistencies between targets and predictions. Another perspective of targeting higher-fidelity $\mathbf{R}$ data is explored by Wu *et al.* (2018), which decomposed the tensor into a proportional and an orthogonal part with respect to the mean straining-rate tensor $\mathbf{S}$. The proportionality constant is an optimal eddy-viscosity ν_t.

Methodologies that directly target the turbulent stresses, such as the mentioned above, can be particularly troublesome. As demonstrated by Thompson *et al.* (2016), the numerical convergence of $\mathbf{R}$ on high-fidelity simulations may not be as well observed as the mean flow quantities. Therefore, using $\mathbf{R}$ as the targeted quantity can lead to misinformed ML techniques and a propagation of error to the corrected mean flow quantities.

In an effort to overcome this limitation Cruz *et al.* (2019) indirectly computed the Reynolds Force Vector $\mathbf{r} \equiv \nabla \cdot \mathbf{R}$ through the momentum equations to use it as the ML target. Through this strategy, Cruz *et al.* (2019) avoided using the DNS tensor $\mathbf{R}$, which leads to reasonably poorer results when used as the ML target, as demonstrated by the authors.

To this moment, all efforts to correct RANS simulations through data-driven predictive strategies used quantities directly injected as source-terms into the momentum equations, be it the stress tensor $\mathbf{R}$ or the force vector $\mathbf{r}$. An unexplored strategy is to employ the transport equation for the Reynolds Stress tensor coupled with the mean momentum equations (MME) in a data-driven turbulence model to correct the mean flow field. This new approach is presented in this work and employed to correct the turbulent flow in a square-duct.

2. METHODOLOGY

Departing from the Reynolds Stress Transport Equation (RSTE) in Eq. (1), where $\mathbf{u}$ is the mean velocity field and the tensor $\mathbf{\Gamma}$ takes into account all the terms that require modelling (Thompson *et al.*, 2019).

$$\mathbf{u} \cdot \nabla \mathbf{R} = -\nabla^T \mathbf{u} \cdot \mathbf{R} - \mathbf{R} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{R} + \mathbf{\Gamma} \tag{1}$$

To ensure numerical stability and convergence of Eq. (1), two modifications were required. First, the production terms $-\nabla^T \mathbf{u} \cdot \mathbf{R} - \mathbf{R} \cdot \nabla \mathbf{u}$ were incorporated to the equation's source-term $\mathbf{\Gamma}$. Second, a turbulent viscosity ν_t is introduced into the diffusive term. These modifications transform Eq. (1) into Eq. (2). The symmetric tensor $\hat{\mathbf{\Gamma}}$ in Eq. (2) is the source-term responsible for correcting the RANS simulations. Equation (3) shows the terms that compose the source-term $\hat{\mathbf{\Gamma}}$.

$$\mathbf{u} \cdot \nabla \mathbf{R} = \nabla \cdot ((\nu + \nu_t) \nabla \mathbf{R}) + \hat{\mathbf{\Gamma}} \tag{2}$$

$$\hat{\mathbf{\Gamma}} = \mathbf{\Gamma} - \nabla^T \mathbf{u} \cdot \mathbf{R} - \mathbf{R} \cdot \nabla \mathbf{u} - \nabla \cdot (\nu_t \nabla \mathbf{R}) \tag{3}$$

The modifications done to Eq. (1) were necessary to ensure the functionality of the RSTE data-driven turbulence model. When coupling Eq. (1) with the MME and the pressure equation, the system of partial differential equations (PDEs) diverged in few iterations. The MME coupled with Eq. (2) is presented in Eq. (4), where $\mathbf{B}$, defined in Eq. (5), is the deviatoric part of $\mathbf{R}$, and $Tr(\cdot)$ is the trace of a tensor.

$$\mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} - \nabla \cdot \mathbf{B} \tag{4}$$

$$\mathbf{B} = \mathbf{R} - \frac{1}{3} Tr(\mathbf{R}) \mathbf{I} \tag{5}$$

The set of 10 coupled PDEs composed by Eq. (2), Eq. (4) and a pressure-velocity correlation compose the proposed RSTE data-driven turbulence model. The RANS mean flow quantities of the square-duct flow were corrected by the ML predicted source-term $\hat{\mathbf{\Gamma}}_{ML}$. Since most RANS turbulence models use a turbulent viscosity in their calculations, even the ones that do not employ the linear eddy-viscosity assumption, the ν_t in Eq. (2) can be taken from the baseline RANS simulations.

2.1 Application of the RSTE model on the square-duct flow

The proposed data-driven model was used to correct the flow on a square-duct. This problem is specially interesting to turbulence modelling since one of the most popular category of models, the linear eddy-viscosity, is not able to capture the secondary motion on the duct. Other more complex RANS models like the direct Reynolds Stress Tensor Modeling (RSTM) are able to attain this phenomena to some extent (Wu *et al.*, 2018), but not without a considerably higher computational effort.

The dataset used to assess the RSTE model were the DNS simulations by Pinelli *et al.* (2010), post-processed by Rangel (2019). The DNS data by Pinelli *et al.* (2010) was used to calculate the source-term $\hat{\mathbf{\Gamma}}$ through the manipulation of Eq. (2), as described by Eq. (6). The chosen ML technique to perform the predictions of $\hat{\mathbf{\Gamma}}$ were Neural Networks.

NN have been widely used in many areas due to their simplicity, effectiveness and large amount of available open-source programming libraries.

$$\hat{\mathbf{\Gamma}}_{DNS} = \mathbf{u}_{DNS} \cdot \nabla \mathbf{R}_{DNS} - \nabla \cdot ((\nu + \nu_t) \nabla \mathbf{R}_{DNS}) \tag{6}$$

Before training the networks, six corresponding $\kappa - \epsilon$ RANS simulations were conducted on the same Reynolds numbers as the ones available in Pinelli *et al.* (2010). The six simulations used the same geometries, boundary and initial conditions, and were performed on a single quadrant of the duct, taking advantage of the flow's symmetry while also reducing computational cost. The simulations' Reynolds numbers are based on the same cross-sectional bulk velocity $u_{BULK} = 0.4819\, m/s$, the duct's half wall length $h = 1m$ and the varying kinematic viscosity ν. Table (1) displays the viscosities values for the corresponding simulations. The Standard κ - ϵ model's coefficients used were the ones proposed by Launder and Spalding (1974).

Table 1: Simulated Re and corresponding kinematic viscosities

Re	$\nu[m^2/s]$
2200	2.1858×10^{-4}
2400	2.0082×10^{-4}
2600	1.8537×10^{-4}
2900	1.6619×10^{-4}
3200	1.5061×10^{-4}
3500	1.3770×10^{-4}

2.2 Neural Networks setup

2.2.1 Data splitting and cross validation

To properly train and evaluate the NN, the dataset by Pinelli *et al.* (2010) was divided into three separate groups: training, validation and testing datasets. The training data is used to train the network, it is based on this data that the network updates its parameters. Validation data is used to measure the evolution of the network learning throughout the iterative process, it is based on the network's performance on predicting this data that the learning process is interrupted. The test group is a fragment of the dataset that the network does not get in contact with, and is used to evaluate its performance on an never before seen case.

Since six simulations are available, the division of the dataset was done in a way that ensures an entire simulation for validation and another one for testing. As a consequence, four simulations remained to be used as the training data by the NN. Therefore, the predicted testing dataset is the one that was later injected into the CFD environment to correct the corresponding RANS simulation.

Separating two entire simulations for validation and testing implied that 30 different combinations of training, validation and test groups were available. To evaluate each of the groups in terms of the quality of their testing predictions a cross validation analysis was performed. Cross validation is the analysis of testing accuracy on different subsets of training, validation and testing configurations of the same dataset. It is useful to determine the best data splitting configuration and also evaluate if the applied ML technique and architectures performs similarly on different data splitting setups. A reasonable overall performance on all of the groups is a requirement to determine if the selected ML technique is adequate for the problem.

2.2.2 Network architecture and hyper-parameters

Numerous network architectures were evaluated, but optimal results were obtained by using the same architecture as Cruz *et al.* (2019), with two hidden-layers and 100 neurons per layer. The activation function of each neuron was the hyperbolic tangent function $tanh(\upsilon)$. An adaptive learning-rate α was used, starting at 1×10^{-3}, α was reduced by 0.6 whenever 5 consecutive iterations resulted in no decrease of the loss function on the validation dataset. Training was interrupted when 20 consecutive iterations resulted in no improvement of the validation performance.

Using this architecture, cross validation was done by training 5 separate networks for each of the 30 data splitting configurations, resulting in a total of 150 networks. The average coefficient of determination R^2 of the 150 networks on their testing predictions was of 0.984 ± 0.008. The reasonable R^2 range indicates that the Neural Networks are correctly comprehending and generalizing the employed dataset.

The networks whose predictions were injected into the CFD solvers had the $Re = 2400$ as the validation dataset, and the $Re = 2900$ as the testing dataset. This configuration presented an averaged R^2 score of 0.9908 ± 0.0001 and was selected because it combined a reasonable average score with considerably small standard deviation, indicating that most predictions from networks trained within this group are consistently close to the DNS target.

2.2.3 Network's inputs

Network's inputs were selected based on a *a priori* group of inputs proposed by Cruz *et al.* (2019). These *a priori* inputs are derived by combinations of the baseline straining-rate tensor $\mathbf{S} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^T\mathbf{u})$, the Reynolds stress tensor $\mathbf{R}$ and the non-persistence-of-straining tensor $\mathbf{P}$ (Thompson and Mendes, 2005). Along with the tensors obtained by combining $\mathbf{S}$, $\mathbf{R}$ and $\mathbf{P}$, according to Cruz *et al.* (2019) using the divergence of each of the tensors is also important to increase stability on the predictions. The *a priori* set of inputs is presented in Tab. (2), with the inclusion of the tensor $\hat{\mathbf{\Gamma}}$ and its divergence $\nabla \cdot \hat{\mathbf{\Gamma}}$.

Table 2: A priori inputs

Tensors	Vectors
$\mathbf{S}$	$\nabla \cdot \mathbf{S}$
$\mathbf{P}$	$\nabla \cdot \mathbf{P}$
$\mathbf{S}^2$	$\nabla \cdot \mathbf{S}^2$
$\mathbf{P}^2$	$\nabla \cdot \mathbf{P}^2$
$\mathbf{S}\cdot\mathbf{P} + \mathbf{P}\cdot\mathbf{S}$	$\nabla \cdot (\mathbf{S}\cdot\mathbf{P} + \mathbf{P}\cdot\mathbf{S})$
$\mathbf{S}^2\cdot\mathbf{P} + \mathbf{P}\cdot\mathbf{S}^2$	$\nabla \cdot (\mathbf{S}^2\cdot\mathbf{P} + \mathbf{P}\cdot\mathbf{S}^2)$
$\mathbf{P}^2\cdot\mathbf{S} + \mathbf{S}\cdot\mathbf{P}^2$	$\nabla \cdot (\mathbf{P}^2\cdot\mathbf{S} + \mathbf{S}\cdot\mathbf{P}^2)$
$\mathbf{R}$	$\mathbf{r}$
$\hat{\mathbf{\Gamma}}$	$\nabla \cdot \hat{\mathbf{\Gamma}}$

Each component of the tensors and vectors in Tab. (2) corresponds to an argument, or feature, of the function constructed by the Neural Networks. Therefore, the NN builds a functional relationship between the 81 components of all tensors and vectors in Tab. (2) and the 6 components of the source-term $\hat{\mathbf{\Gamma}}$. The considerable difference between the number of features and targets impairs the predictive capabilities of Neural Networks in two ways: first, with more features more parameters need to be trained by the network, therefore requiring more data to do so. Second, many of the components of the given tensors and vectors from RANS simulations are null, this implies that a significant amount of the data that is provided to the network for computing predictions is basically numerical noise.

To avoid these problems, the *a priori* input list was filtered by three criteria. The first one removed any component that was null in the whole flow geometry. Secondly, any feature which was spatially discontinuous in any segment of the duct's geometry was discarded. Lastly, features whose mean absolute values over the flow geometry were much higher than the remaining ones were also excluded.

Each of the filtering criteria mentioned above was evaluated individually, and all three represented a significant improvement in predictive capabilities. The remaining components of tensors and vectors composed the final features' set that were used by the network to predict $\hat{\mathbf{\Gamma}}$. Table (3) presents the final set of NN inputs.

Table 3: Final list of inputs

Tensors Components	Vectors Components
$\mathbf{S}$: $[S_{xy}, S_{xz}]$	$\nabla \cdot \mathbf{S}$: $(\nabla \cdot \mathbf{S})_x$
$\mathbf{P}$: $[P_{xx}, P_{yy}, P_{yz}, P_{zz}]$	-
$\mathbf{S}^2$: $[S^2_{xx}, S^2_{yy}, S^2_{yz}, S^2_{zz}]$	-
$\mathbf{R}$: $[R_{xx}, R_{xy}, R_{xz}, R_{yy}, R_{zz}]$	$\mathbf{r}$: $[r_x, r_y, r_z]$
$\hat{\mathbf{\Gamma}}$: $[\hat{\mathbf{\Gamma}}_{xx}, \hat{\mathbf{\Gamma}}_{xy}, \hat{\mathbf{\Gamma}}_{xz}, \hat{\mathbf{\Gamma}}_{yy}, \hat{\mathbf{\Gamma}}_{zz}]$	$\nabla \cdot \hat{\mathbf{\Gamma}}$: $[(\nabla \cdot \hat{\mathbf{\Gamma}})_x, (\nabla \cdot \hat{\mathbf{\Gamma}})_y, (\nabla \cdot \hat{\mathbf{\Gamma}})_z]$

3. RESULTS

20 networks were trained using the $Re = 2400$ simulation as the validation data and $Re = 2900$ as the testing data. The network which provided the highest R^2 coefficient in its test predictions with respect to $\hat{\mathbf{\Gamma}}_{DNS}$ was the one selected to fuel the data-driven turbulence model. Once training was completed, the predicted $\hat{\mathbf{\Gamma}}_{NN}$ was injected into the CFD environment to correct the RANS mean quantities $\mathbf{R}$ and $\mathbf{u}$. Both baseline RANS and data-driven corrections were carried on the open-source platform OpenFOAM. In order to use this data-driven framework, the proposed turbulence model was programmed in OpenFOAM, enabling the use of the software's available solvers and pressure coupling schemes. The solver *simpleFOAM*, which uses the SIMPLE algorithm for pressure calculation (Patankar and Spalding, 1972), was the one employed.

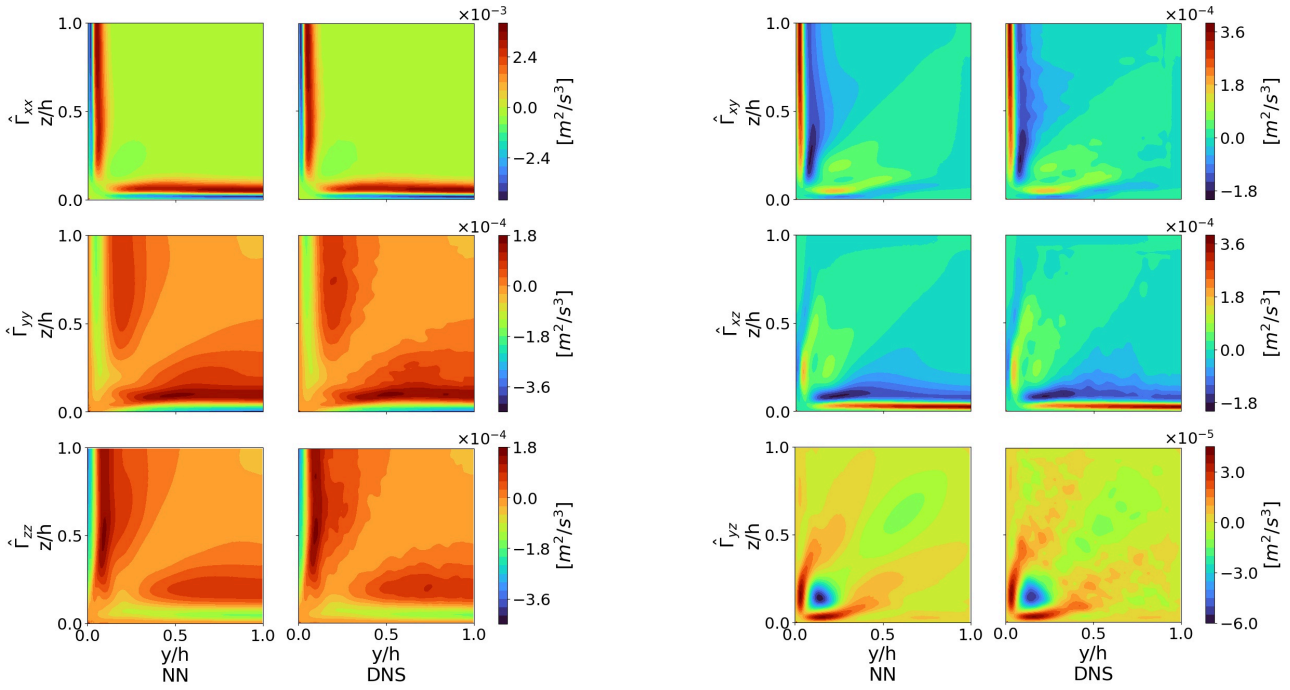
The numerical setup for the corrections used the RANS fields as initial conditions (IC) for $\mathbf{u}$ and p, but null IC were used for solving $\mathbf{R}$. Although possible to use $\mathbf{R}_{RANS}$ as IC, results when using null conditions were more accurate, while

also being computationally faster to solve.

Results are presented separately, divided into *a priori* and *a posteriori* segments. *A priori* results were the six components of the source-term $\hat{\mathbf{\Gamma}}$ predicted by the NN. *A posteriori* results were the Reynolds stress tensor and the mean flow corrected by the injection of $\hat{\mathbf{\Gamma}}_{NN}$ into the RSTE data-driven turbulence model. All presented results correspond to the testing dataset of $Re = 2900$.

3.1 A priori results

The six components of the predicted $\hat{\mathbf{\Gamma}}$ are shown in Fig. (1), in comparison with the corresponding DNS components. For comparison, the baseline $\hat{\mathbf{\Gamma}}_{RANS}$ field from the $Re = 2900$ simulation is shown in Fig. (2). The RANS tensor is presented separately because the magnitude of its components differs too widely from the DNS and NN tensors, impairing the scale of the plots in case they were presented together.



(a) Predicted and DNS components $\hat{\Gamma}_{xx}$, $\hat{\Gamma}_{yy}$ and $\hat{\Gamma}_{yz}$

(b) Predicted and DNS components $\hat{\Gamma}_{xy}$, $\hat{\Gamma}_{xz}$ and $\hat{\Gamma}_{yz}$

Figure 1: Predicted and DNS $\hat{\mathbf{\Gamma}}$ ($Re = 2900$)

Figure (1) demonstrates the exact symmetry in the predicted components $\hat{\Gamma}_{xx}$ and $\hat{\Gamma}_{yz}$ with respect to the duct's quadrant diagonal line. This characteristic is ensured by a post-processing imposed on the NN predictions. Since training is done in batches, that is, network's parameters are updated after multiple data points are passed through it, exact symmetry is not observed by predictions. The imposed post-processing also ensures that the components which are mirrored with respect to the duct's diagonal, *e.g.* $\hat{\Gamma}_{xy}$ and $\hat{\Gamma}_{xz}$, will be numerically identical.

The smooth nature of the NN predictions can be noticed in Fig. (1). This occurs due to the employment of smooth activation functions in each of the network's neurons. In fact, the predicted quantities are even spatially smoother than their DNS targets.

3.2 A posteriori results

The quality of the predicted source-term $\hat{\mathbf{\Gamma}}_{NN}$ can be truly evaluated only when injected into the RSTE data-driven turbulence model. The corrected normal turbulent stresses are shown in Fig. (3), for comparison both the baseline RANS and the DNS normal components of $\mathbf{R}$ are also depicted. The corrected shear turbulent stresses are also compared with its RANS and DNS counterparts in Fig. (4).

As can be seen in Fig. (3) and Fig. (4), corrections on both normal and shear components of $\mathbf{R}$ are reasonably closer to DNS values than the baseline RANS stresses. This demonstrates that the proposed data-driven model efficiently corrects the turbulent stresses. Discrepancies between DNS and corrected stresses are more evident in the duct's center.

The corrected velocity field is displayed in comparison with the baseline RANS and the DNS $\mathbf{u}$ in Fig. (5). The efficiency in correcting the velocity field by using the RSTE data-driven turbulence model is demonstrated by the agreement between DNS and corrected data. Once again, the corrected components are much closer to DNS than the baseline RANS.

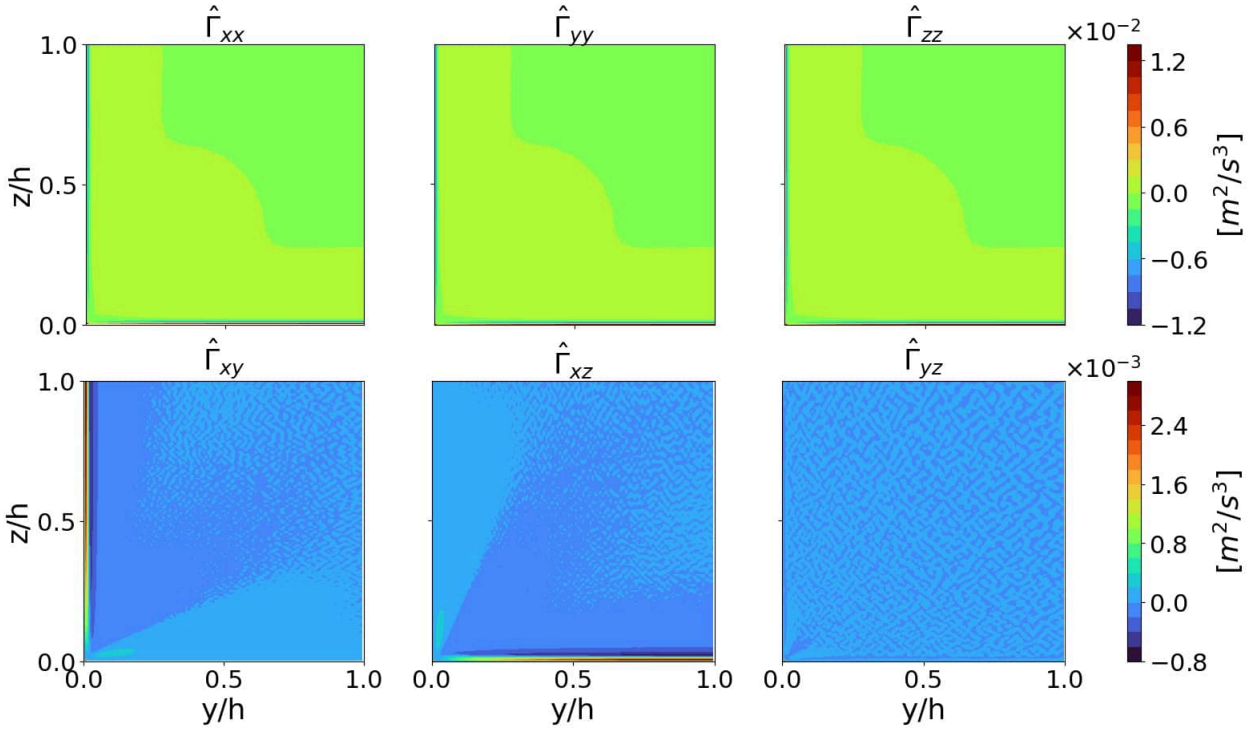


Figure 2: RANS source-term $\hat{\mathbf{\Gamma}}$ ($Re = 2900$)

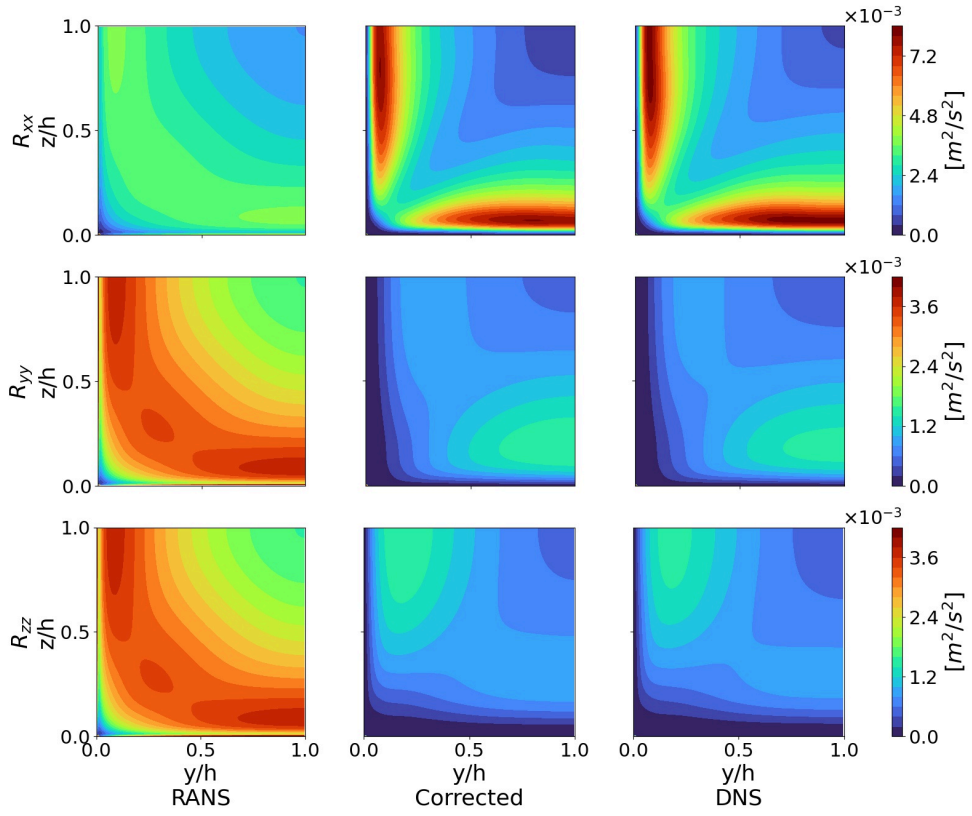


Figure 3: RANS, corrected and DNS normal turbulent stresses on the square duct ($Re = 2900$)

The duct's recirculation, which was nonexistent on the RANS simulations, is effectively captured by the corrections based on the predicted source-term $\hat{\mathbf{\Gamma}}_{NN}$.

The small discrepancies between DNS and corrected flows can be examined in more detail in Fig. (6). It presents samples of the three components of the mean velocities on different coordinates of the square-duct, normalized by the bulk velocity u_{BULK}. For comparison, Fig. (6) also depicts $\mathbf{u}_{RANS}$ components, demonstrating its significant discrepancy

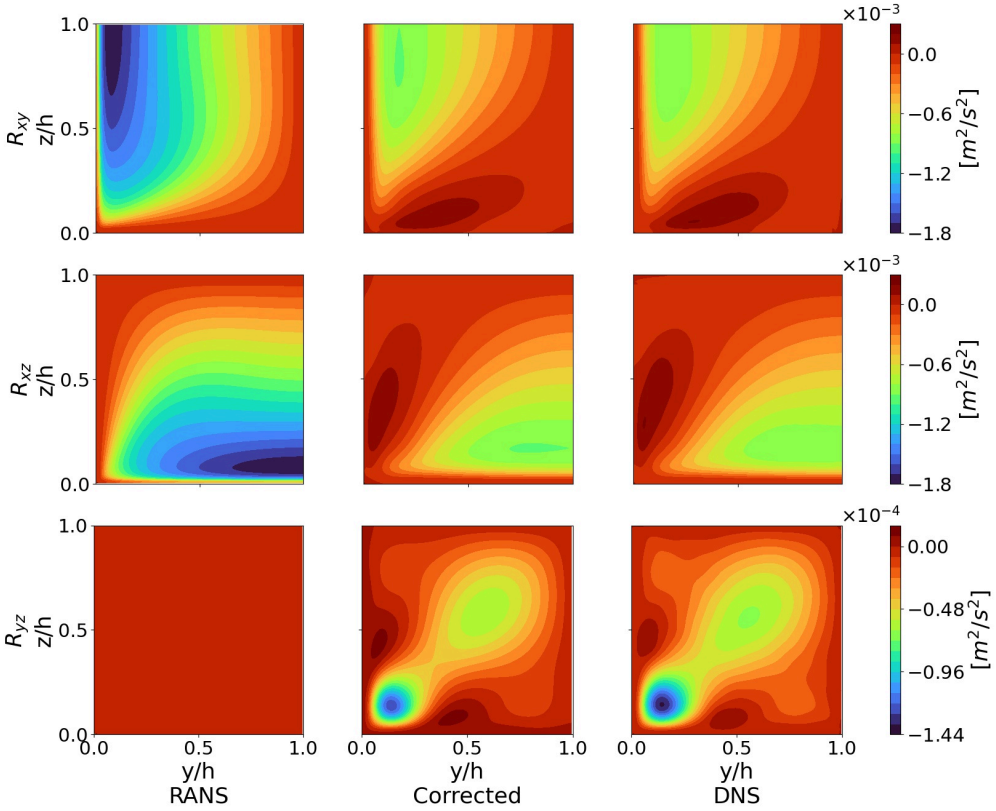


Figure 4: RANS, corrected and DNS normal turbulent stresses on the square duct ($Re = 2900$)

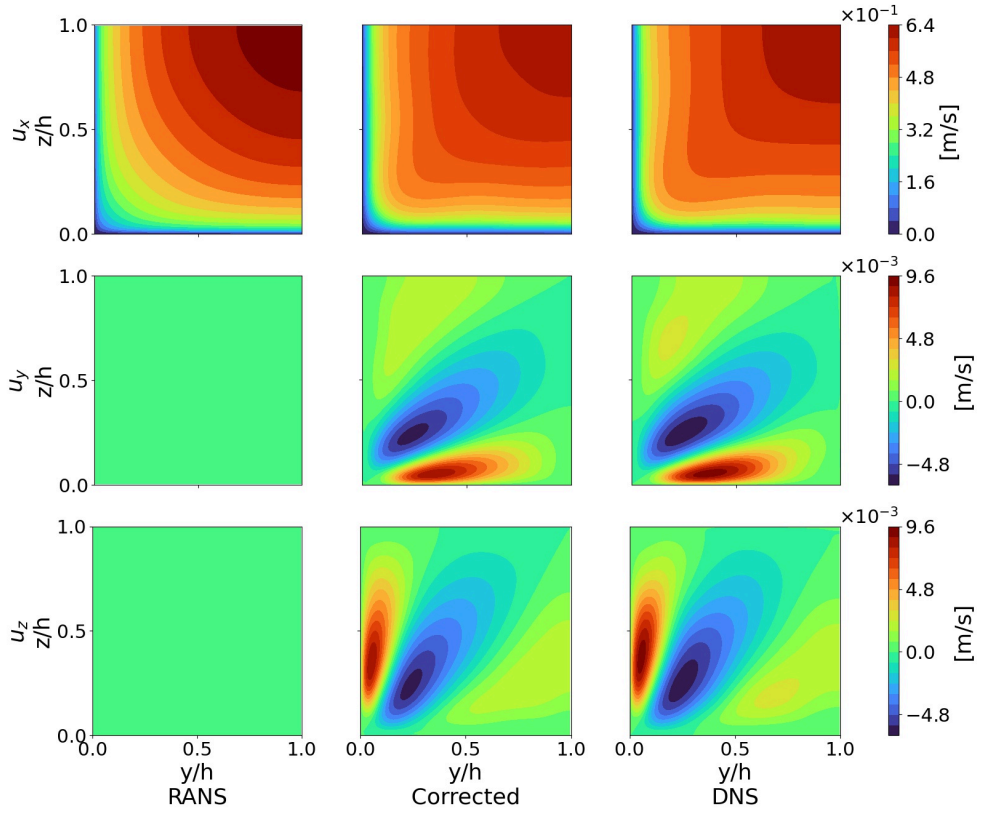
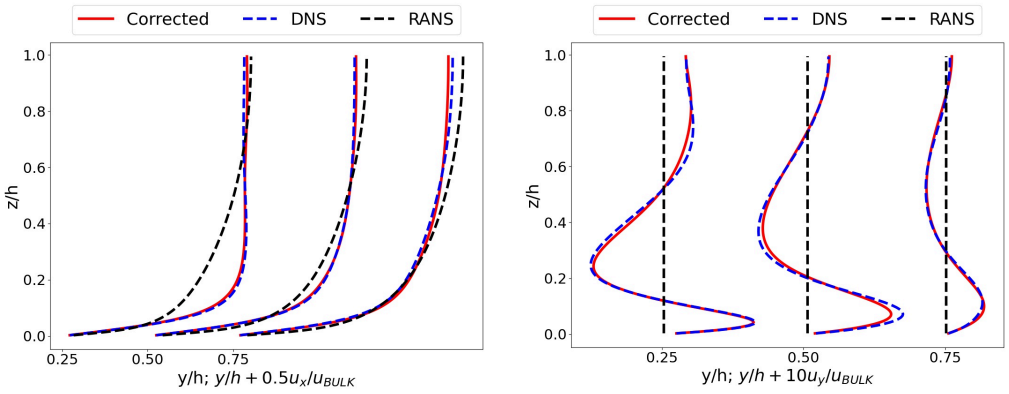


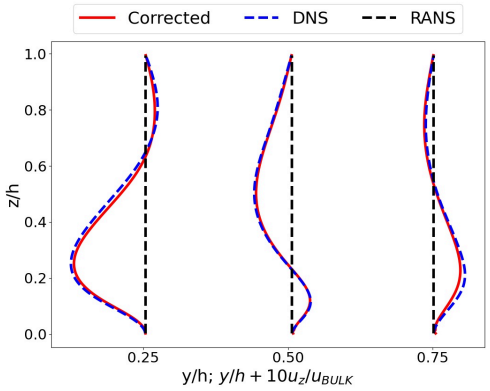
Figure 5: RANS, corrected and DNS velocity components on the square duct ($Re = 2900$)

with respect to the high-fidelity DNS data, and to the corrections as well.



(a) Primary flow component u_x samples

(b) Secondary flow component u_y samples



(c) Secondary flow component u_z samples

Figure 6: Corrected, DNS and RANS velocity components on three locations of the square duct ($Re = 2900$)

4. CONCLUSIONS

A novel approach for improving simulated turbulent flows based on Machine Learning predictions was presented and evaluated on the square-duct flow. This methodology is the first to correct both the velocity field and the turbulent stresses. Corrections were done by the use of a transport equation for $\mathbf{R}$, which contains the source-term $\hat{\mathbf{\Gamma}}$. The proposed RSTE was coupled with the mean momentum equations and a pressure coupling scheme, resulting in an ensemble of 10 coupled PDEs. The set of 10 PDEs formed a data-driven turbulence model.

Neural Networks were employed to predict the symmetric tensor $\hat{\Gamma}$, based on DNS data for the square-duct flow by Pinelli *et al.* (2010). A total of four different simulations were used for training, while other two were used as validation and testing datasets. Once training was finished, the predicted $\hat{\mathbf{\Gamma}}_{NN}$ was injected into the RANS environment and successfully corrected the mean velocity field and the turbulent stresses.

The reasonable agreement between $\mathbf{u}_{DNS}$ and $\mathbf{u}_{Corrected}$ indicates that the present work's methodology is capable of consistently achieving results comparable to high-fidelity data, departing from data that considerably deviated from it, such as the employed baseline $\kappa - \epsilon$ simulations. The corrected $\mathbf{R}$ was also considerably close to $\mathbf{R}_{DNS}$.

Most notably, the nonexistent recirculation on the popular linear eddy-viscosity models was efficiently corrected. The reasonable proximity between DNS and corrected recirculations is especially significant, since the main concern of applying ML techniques to correct RANS simulations of the square-duct is recovering this aspect of the flow. Good agreement between the target and the corrected primary flows was also achieved, indicating that the RSTE methodology is capable of correcting with reasonable efficiency all of the model's quantities of interest.

In order to further evaluate the present work's approach, its application on other sets of flows is required. Extending it to different Machine Learning techniques is also recommended.

5. ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the support provided by the National Council for Scientific and Technological Development - CNPq, Brazil.

6. REFERENCES

Cruz, M.A., Thompson, R.L., Sampaio, L.E. and Bacchi, R.D., 2019. "The use of the reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling". *Computers & Fluids*, Vol. 192, p. 104258.

Launder, B.E. and Spalding, D.B., 1974. "The numerical computation of turbulent flows". *Computer methods in applied mechanics and engineering*, pp. 269–289.

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

Patankar, S.V. and Spalding, D.B., 1972. "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows". *International Journal of Heat and Mass Transfer*, Vol. 15.

Pinelli, A., Uhlmann, M., Sekimoto, A. and Kawahara, G., 2010. "Reynolds number dependence of mean flow structure in square duct turbulence". *Journal of fluid mechanics*, Vol. 644, pp. 107–122.

Rangel, V.B., 2019. *Influência do Tratamento da Base de Dados DNS na Aplicação de Técninas de Aprendizagem de Máquina para Melhorar Acurácia de Simulações RANS*. Master's thesis, Universidade Federal do Rio de Janeiro.

Thompson, R.L. and Mendes, P.R.S., 2005. "Persistence of straining and flow classification". *International journal of engineering science*, Vol. 43, No. 1-2, pp. 79–105.

Thompson, R.L., Mishra, A.A., Iaccarino, G., Edeling, W. and Sampaio, L., 2019. "Eigenvector perturbation methodology for uncertainty quantification of turbulence models". *Physical Review Fluids*, Vol. 4, No. 4, p. 044603.

Thompson, R.L., Sampaio, L.E.B., de Bragança Alves, F.A., Thais, L. and Mompean, G., 2016. "A methodology to evaluate statistical errors in dns data of plane channel flows". *Computers & Fluids*, Vol. 130, pp. 1–7.

Tracey, B.D., Duraisamy, K. and Alonso, J.J., 2015. "A machine learning strategy to assist turbulence model development". In *53rd AIAA aerospace sciences meeting*. p. 1287.

Wang, J.X., Wu, J.L. and Xiao, H., 2017. "Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data". *Phys. Rev. Fluids*, Vol. 2, p. 034603. doi: 10.1103/PhysRevFluids.2.034603. URL `https://link.aps.org/doi/10.1103/PhysRevFluids.2.034603`.

Wu, J.L., Wang, J.X., Xiao, H. and Ling, J., 2017. "A priori assessment of prediction confidence for data-driven turbulence modeling". *Flow, Turbulence and Combustion*, Vol. 99, No. 1, pp. 25–46.

Wu, J.L., Xiao, H. and Paterson, E., 2018. "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework". *Phys. Rev. Fluids*, Vol. 3, p. 074602. doi:10.1103/PhysRevFluids.3.074602. URL `https://link.aps.org/doi/10.1103/PhysRevFluids.3.074602`.

7. RESPONSIBILITY NOTICE

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