Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVIII, págs. 905-914 (artículo completo)
H.G. Castro, J.L. Mroginski, R.R. Paz, M.A. Storti (Eds.)
Resistencia, 1-5 Noviembre 2021

REYNOLDS STRESSES PREDICTION USING DEEP NEURAL NETWORKS

Hugo D. Pasinato[a], Fernando D. Gerosa[a] and Ezequiel A. Krumrick[b]

[a]*Departamento de Ing. Electromecánica, FRP-Universidad Tecnológica Nacional, Avda. Almafuerte 1033, 3100 Paraná, Argentina; hpasinato@frp.utn.edu.ar*

[b]*Departamento de Ing. Química, FRN-Universidad Tecnológica Nacional, Avda. Rotter s/n, 8318 Plaza Huincul*

Keywords: Deep Learning; RANS simulation; Bounded flow

Abstract. After approximately five decades since the first proposed model, Reynolds-averaged Navier-Stokes, RANS, simulations remain the most used technique for engineering applications in turbulent flows. In the last few years, after a stagnant period in RANS model development, there has been a resurgence of research on RANS techniques. Based on a large amount of high-quality data of turbulent flows (from direct numerical simulation, DNS, and large eddy simulation, LES), researchers have begun to systematically use this information in turbulence to quantify and reduce model uncertainties and to incorporate more physics of turbulence into RANS models or the eddy viscosity model. This study presents the results of using of deep neural networks to directly calculate the Reynolds stresses of a turbulent flow, without using the linear eddy viscosity model. Based on features of turbulent flows from DNS data, deep neural networks are adjusted to predict the relevant Reynolds stresses. First, an a priori comparison for perturbed turbulent flows, and second, the propagation of the Reynolds stresses through the velocity field of a developed channel flow are made. In both cases, there is a significant improvement versus results from the standard kappa-epsilon model.

1 INTRODUCTION

After a long period of stagnation, RANS modeling is entering a new era due to using machine learning techniques. Machine learning encompasses a broad set of data-driven algorithms, including known methods such as linear regression and more advanced concepts such as neural networks. With high-fidelity data available, machine-learning use includes capturing potentially complex relationships between turbulence mean-flow characteristics (features) and modeling terms (predictions).

The machine learning techniques have been used in RANS models in a wide range of problems and with a broad range of methods. One problem can be to study RANS uncertainty as it was done by Ling and Templeton (2015). Also, Ling with another group (Ling et al. (2015)) used highly resolved LES jet-in-crossflow results and the corresponding RANS results for the same flow, intending to determining in which regions of this flow the various RANS eddy viscosity assumptions were violated and to explore the potential of machine learning techniques to provide improved models.

Deep learning, DL, is a branch of machine learning that has gained attention owing to its flexibility and precision. In DL input features get transformed through multiple layers of non-linear interaction. In one of the first applications to turbulent flow, Ling et al. (2016b) presented a method of using deep neural networks, DNN, to learn a model for the Reynolds stress anisotropy tensor from high-fidelity simulation data. They proposed a DNN with an invariant tensor basis to embed Galilean invariance into the predicted anisotropy tensor. They propagated the Reynolds stress anisotropy prediction in two flows and found significant improvements (by propagation is meant to use these predicted values in the RANS equations and solve it for the mean flow).

Wang et al. (2017) used a machine learning technique based on random forests, to predict the discrepancies between RANS and the true Reynolds stresses formulated as functions of the mean flow features. Using this technique, predicted these discrepancy functions based on existing DNS databases. Then, they used the predicted Reynolds stress discrepancies in a developed flow in a square duct and flows with massive separation with clear improvement.

Based on the fact that the Reynolds stresses are the principal source of model-form uncertainty in RANS simulations, the majority of the recent work aim to predict the Reynolds stress tensor or the anisotropy of the Reynolds stress tensor. However, predictions of the correct Reynolds stresses from a machine learning model adjusted to DNS databases cannot necessarily guarantee obtaining improved mean flow fields, as is commented by Wu et al. (2019). Even solving RANS equations with Reynolds stresses from DNS data for high Reynolds numbers can lead to large errors in the velocities Poroseva et al. (2016); Thompson et al. (2016). Wu and co-workers addressed the possible ill-conditioning problem of the RANS equations when explicit Reynolds stresses get propagated for the mean velocity. They have shown that the global matrix condition number of the discretized RANS equations cannot explain the ill-conditioning. As they have shown, the local metric can explain deteriorated model conditioning with increasing Reynolds number. They considered an ideal scenario in which the Reynolds stress is directly computed from the DNS database at various friction Reynolds numbers of 180, 550, 1000, 2000, and 5200, showing the increasing errors for higher Reynolds numbers.

In an attempt to circumvent the amplification of errors in the propagation of DNS data through RANS equations, Cruz et al. (2019) employed the Reynolds force vector rather than the Reynolds stress tensor as the target for learning and prediction. They used to predict with machine learning the Reynolds force vector correction, and have shown to be able to reconstruct the mean velocity field with a higher fidelity concerning the DNS data.

Other aspects in machine learning apart from the ill-conditioning of the RANS equations are input-output invariance and model feasibility. When choosing the input features to adjust a data-driven model, a question is: how should the domain knowledge be incorporated into the machine learning process? For example, the classical laws of motion are known to obey Galilean invariance. Therefore, the input and thus the obtained regression functions should be in principle Galilean-invariant (Pope (2000)). There are two ways to encompass the invariance of the input data: a)entering data that are Galilean-invariant, and; b)training the model with different data so that it learns the property of invariance from these data. Although it is expected that the model trained on the invariant basis will be more effective at enforcing this invariance, it is also possible to let the model learn these properties (Milano and Koumoutsakos (2002), Cruz et al. (2019)).

This study presents the first results predicted with DL applied to some simple turbulent flows. As a novelty, in this work the input features are from the DNS data rather than the RANS data. A frame-independent formulation is unnecessary for problems where training and testing cases have the same geometry and the same coordinate system, as in the flows used here. A minimum of 12 features is adjusted through a DNN to predict the Reynolds stresses from a DNS database. The Reynolds stresses are propagated for the mean velocity in a developed flow. These a posteriori results are presented together with some a priori results for perturbed flow. In the following section, the methodology is presented, together with a description of the input and output data to the DNN. Then in section 3, results are commented on and discussed. In 4 some conclusions are listed.

2 METHODOLOGY

2.1 Deep Neural Networks

DL is a branch of machine learning in which input features get transformed through multiple layers of nonlinear interaction. DNN is compound by multiple interconnected nodes (neurons), two of them modeling a perceptron (Lecun et al. (2015)). Each node takes an input (the output of a previous node) and outputs the result through an activation function; e.g., Relu and Sigmoid are the most common. In this study, the Sigmoid activation function is used. All interconnected nodes conform to a mesh with a different number of neurons and layers depending on the problem: a) one input layer (in this case the layer with the input features of the flow); b) one or multiple hidden layers (called hidden layers because the physical interpretation of their activation is not always clear); and c) one output layer (with the predicted values, in the present work the Reynolds stress). DNN with multiple layers can capture very complex and nonlinear interactions among the different input features, that get transformed through the final activation function.

Since the idea was to propagate the predicted Reynolds stress through the RANS equations for the mean flow, only 12 features were used as input.

Neural networks have three tunable hyperparameters: the number of hidden layers, the number of neurons per hidden layer, and the learning rate coefficient. For the multilayer perceptron and the invariant neural network used in their work, Ling et al. (2016b) used a Bayesian optimization for optimizing DNN for these hyperparameters, and the optimal number of hidden layers was 10 and 8, respectively.

In the present study, only one hidden layer was used to make a simple and faster prediction with the DNN in the RANS solver. The number of neurons in the hidden layer and the learning rate were chosen by try and failure, with 12 neurons in the hidden layer (the input number of features), and the initial learning rate was 0.05. After the initial training, the net was saved, and then subsequent training with decreasing learning rate was used. A DNN was considered adjusted when the accuracy of the prediction of the testing data was higher than 95%.

In this work, a numerical code in Fortran developed by M. Curcic from the University of Miami, was used to train and test the DNN (Curcic (2016)). These efforts in developing tools for machine learning techniques are invaluable for the turbulence modeling community.

The following is a summary of the approach:

1. Compute DNS for the base flows;
2. Compute the features from these DNS data;
3. Train, and test the DNN;
4. Use the trained and tested DNN in a RANS solver to compute for the mean flow.

2.2 Input and output data

To have a minimum computational cost, only twelve simple features of the flow were used. Table 1 shows a list of these features, which are dimensionless using the friction velocity, u_τ, and half the distance between walls, δ.

When adjusting a machine learning model like a DNN, it should be trained based on features of a different turbulent flow configuration, upon which it is then tested, looking for the ability of the model to generalize to new flows for which high fidelity results are not available. The flows used to train the DNN are named training flows, and the flow to be predicted test flow.

Description	Symbol	Description	Symbol
Kinetic energy	κ	Dissipation of κ	ϵ
Longitudinal velocity	U	Wall-normal velocity	V
Longitudinal U gradient	$\partial U/\partial x$	Wall-normal U gradient	$\partial U/\partial y$
Longitudinal V gradient	$\partial V/\partial x$	Wall-normal V gradient	$\partial V/\partial y$
Longitudinal pressure gradient	$\partial P/\partial x$	Wall-normal pressure gradient	$\partial P/\partial y$
Deformation tensor moduli	S	Wall distance	y

Table 1: Input or features used to feed the DNN. All these variables are dimensionless using the friction velocity, u_τ, and half of the channel height, δ (the dimensionless symbol has been dropped for simplicity).

The usual practice in the literature is that since the true mean flow (DNS) in the test flows is not available, the inputs features to the DNN should be information of the mean flow produced by the RANS simulations (Ling and Templeton (2015); Ling et al. (2016b); Wu et al. (2018)). Therefore, the adopted features are usually based on RANS-predicted pressure P, mean velocity U, turbulent kinetic energy κ, and so on. Hence, it is usual that RANS results (not DNS results), are inputs to the neural network. High fidelity data is used only to provide true (output) anisotropy, or the true Reynolds stress, during model training and testing. The reason for this choice is that the DNN must be able to make predictions in the absence of DNS data as input. In this situation, the input features are computed using a RANS model; e.g., $\kappa - \epsilon$, or $\kappa - \omega$.

Then, the output or final prediction are the Reynolds stresses, which for an optimal prediction, are equal to the true (DNS) Reynolds stresses.

Ideally, the machine learning prediction would be identical to the DNS data; to the true anisotropy of Reynolds stresses, or to the total Reynolds stress. In this case, the RANS mean flow should approach the Reynolds-averaged mean flow extracted from DNS data. The errors or differences between these two mean flows should be attributed only to the small errors associated with the RANS techniques themselves (Pasinato and Krumrick (2021)). Hence, in this case the RANS mean flow should be close to the Reynolds-averaged DNS mean flow. Consequently, if these optimal Reynolds stresses from the DNN get introduced in an a posteriori RANS simulation, the mean features will substantially differ from those of the solution computed with a RANS model. For this reason in the present study the input features are extracted from the DNS database.

Another aspect of machine learning techniques is the Galilean invariance requirement of the input and output. In the present work, vector components and scalars feed the DNN. A frame-independent formulation is unnecessary for problems where training and testing cases have the same geometry and the same coordinate system, as in the flows presented here.

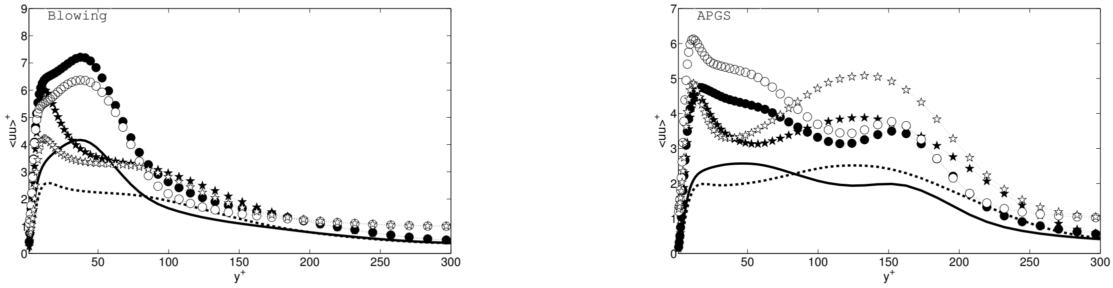
2.3 Training and testing set of data

The usual practice in the literature is to train the network in turbulent flows with different characteristics, to achieve an extensive spectrum of turbulence phenomena. In this case, the DNN must adjust to a large number of non-linear phenomena, requiring a large number of hidden layers and neurons per layer. This would be a versatile DNN, but would also require more processing time to propagate the predicted Reynolds stresses. In this study, following a different path, a DNN as simple as possible with only one hidden layer with 12 neurons was adjusted to developed turbulent flow, with the goal to predict developing turbulent flows with a different Reynolds number.

The DNN was trained with a developed flow with $Re_\tau = 149$, and then used to predict the Reynolds stress of two developing turbulent flows, one perturbed with blowing and the other perturbed with an adverse pressure gradient step (APGS) in the buffer region; both with $Re_\tau = 302$. Furthermore this DNN was used to propagate the Reynolds stress in a developed flow with $Re_\tau = 300$.

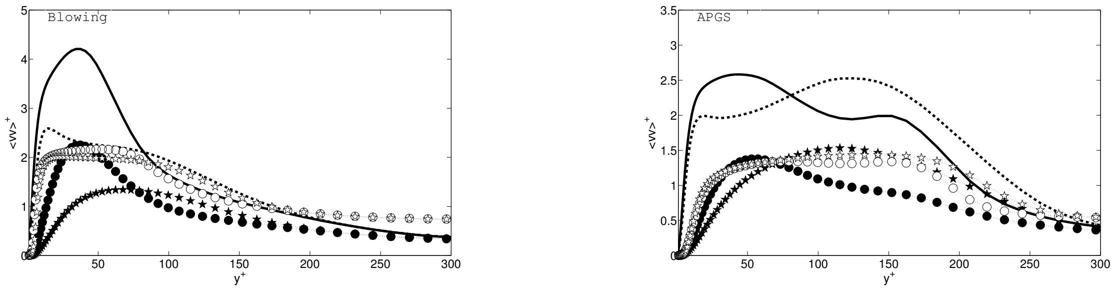
The DNS data belong to previous works. Here only a summary description of these flows is commented; more details are elsewhere (Pasinato (2013). Every DNS case is run with two simulations in parallel. Both simulations are for a channel flow with the same Re, the same physical domain, and the same grid. The first simulation is implemented with periodic boundary conditions to give inlet boundary conditions for the second simulation. Thus, the simulation with the perturbed (developing) flow runs with the inlet boundary condition of a developed turbulent flow and convection boundary conditions at the outlet. The width of the region for perturbation or slot is $W^+ = 220$, the injection dimensionless velocity is $v^+ = 0.60$, and the adverse pressure gradient step, $\partial P^+/\partial x^+ = 0.25$. Furthermore, the distance from the entrance to the slot or perturbation region is $X^+ = 600$. All previous parameters made dimensionless with the friction velocity, u_τ, and half the distance between walls, δ. These perturbed turbulent flows are 2D statistical stationary. The statistics (e.g. mean velocities, Reynolds stresses, and so on) are evaluated along z and time.

To propagate the predicted Reynolds stress in a developed flow, the grid used in the RANS simulations was $64 \times 64 \times 64$, and the time step $0.15\delta/u_\tau^2$. In the wall-normal direction, a non-uniform mesh is used and the expansion ratio is adjusted to ensure that the y^+ of the first



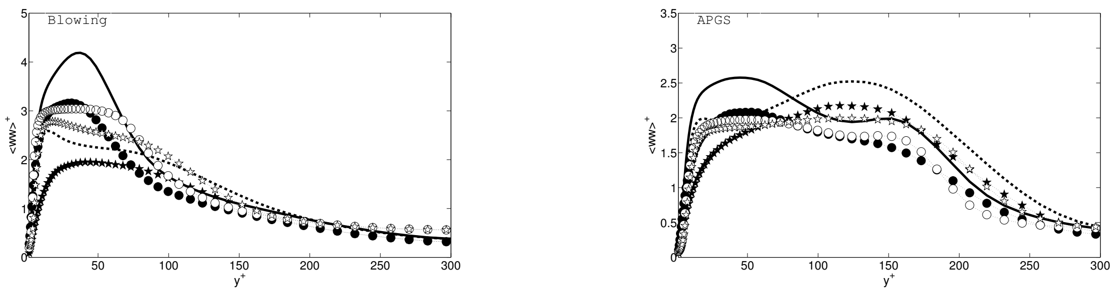
(a) (b)

Figure 1: Comparison of the $\langle uu\rangle^+$ prediction with DNN (unfilled symbols), with DNS data (filled symbols), and a priori results from the $\kappa - \epsilon$ model (lines), at two positions from the perturbation slot start. (a) blowing, (b) APGS. circle, $6W^+$; ——, $\kappa - \epsilon$; star, $10W^+$, $- \ - \ -$, $\kappa - \epsilon$ (where $W^+ = 220$ is the dimensionless width of the blowing/pressure step perturbation region).



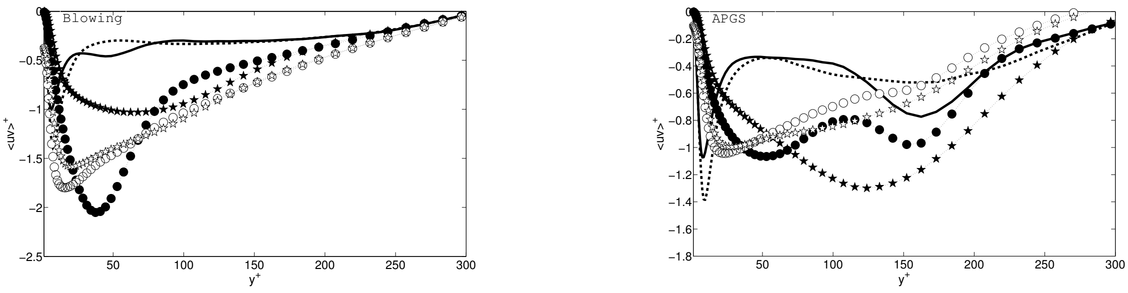
(a) (b)

Figure 2: Idem to Figures 1(a)-1(b) for $\langle vv\rangle^+$.



(a) (b)

Figure 3: Idem to Figures 1(a)-1(b) for $\langle ww\rangle^+$.



(a) (b)

Figure 4: Idem to Figures 1(a)-1(b) for $\langle uv \rangle^+$.

cell center is equal to 1. A van Driest function near the wall is used in those cases with viscosity from the $\kappa - \epsilon$ model.

3 RESULTS AND DISCUSSION

The main goal of the present work is to show the methodology used to predict the Reynolds stresses in some simple turbulent flows using DNN, together with details of some problems related to the propagation of the Reynolds stresses. In this sense, only the first results are considered, since more work to improve these aspects should be done.

A priori results (or DL predictions as they are named here) for perturbed flows with blowing and with an adverse pressure gradient are presented and a posteriori (DL-RANS) results of the propagation of the Reynolds stress for a developed turbulent flow; all these turbulent flows for a friction Reynolds number of about 300. The $\kappa - \epsilon$ model is used for comparison because its results have become a reference for researchers in RANS models.

3.1 Results without propagation (a priori comparison)

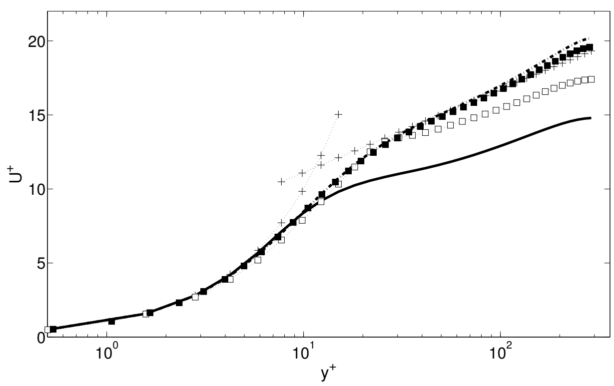


Figure 5: Mean velocity U^+ comparison for developed turbulent channel flow with $Re_\tau = 300$. Filled squares, DNS; ——, ke; $- \cdot - \cdot -$, dnsrs; unfilled squares, DL-RANS; $+ \cdot + \cdot +$, $U^+ = (1/0.41) ln(y^+) + 5.5$.

Two a priori results are presented for turbulent channel flow perturbed with blowing from a thin slot ($W^+ = 220$) and two more for channel flow perturbed with a pressure gradient step through the buffer layer in the same region. Details of the flow are elsewhere Pasinato (2013);

Pasinato and Krumrick (2021).

Figures 1(a)-3(b) show a comparison of the normal Reynolds stresses and 4(a)-4(b) show a comparison of the shear Reynolds stresses, of the DL prediction with DNS data and with a priori results of the $\kappa - \epsilon$ model. These comparisons are at two positions from the slot start (6 and 10 W^+).

All these figures show an improvement of the DL results in comparison with the $\kappa - \epsilon$ model. However, the optimism should be moderate since these are only a priori results. As found in the literature (Wu et al. (2018, 2019,b)), RANS solvers are unstable and difficult to drive to convergence with explicit Reynolds (See also Thompson et al. (2016); Poroseva et al. (2016)). This clearly shows the difference between a priori and a posterior performance in the assessment of turbulence models.

Having said that, it is also true that the agreement between DNS data and DL adjusted with only 12 features of two developed turbulent flows is reasonably good. Blowing and APGS are perturbed turbulent flows with significant modifications from the developed conditions. The best agreement between DL and DNS data is for $\langle uu \rangle^+$. The reason for the good behavior of $\langle uu \rangle^+$ from DL is the information entered to the DNN through κ. As is well known, in a boundary layer, the turbulent production enters energy from the mean flow to the $\langle uu \rangle^+$, and then this energy in $\langle uu \rangle^+$ is redistributed into $\langle vv \rangle^+$, and $\langle ww \rangle^+$, through the pressure gradient (Tennekes and Lumley (1972)).

The DL prediction for the shear Reynolds stress $\langle uv \rangle^+$ is substantially better than the results from the $\kappa - \epsilon$ model, but it is a poor prediction in comparison with DNS data. It is important to remark that generally, the role of $\langle uv \rangle^+$ in turbulent channel flows is substantially more relevant than the role of the normal Reynolds stresses. In fact, for developed turbulent flows the normal Reynolds stresses are irrelevant (their effects in the momentum equations are zero).

3.2 Predicted results propagated in developed flow (a posteriori comparison)

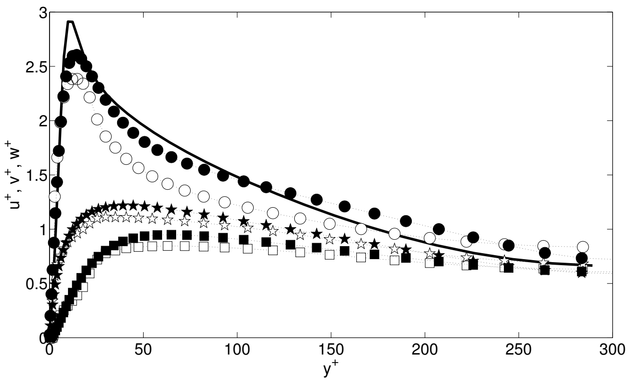


Figure 6: Normal Reynolds stresses comparison of DL-RANS (unfilled symbols) for developed turbulent flow with $Re_\tau = 300$, with DNS data (filled symbols), and $\kappa - \epsilon$ model (solid line). $\circ \cdot \circ \cdot \circ$, u^+; $\star \cdot \star \cdot \star$; w^+; $\square \cdot \square \cdot \square$, v^+.

Figures 5-7 show the comparison of the predicted results from the DNN, then propagated in a developed turbulent flow (DL-RANS), with data from DNS, the $\kappa - \epsilon$ model, and DNS Reynolds stress propagated through the RANS equations (dnsrs) (Pasinato and Krumrick (2021)).

Figure 5 shows a comparison of the distribution of the mean longitudinal velocity U. The profile of U for DL-RANS is a pseudo converged value since the equation for κ (κ and ϵ from

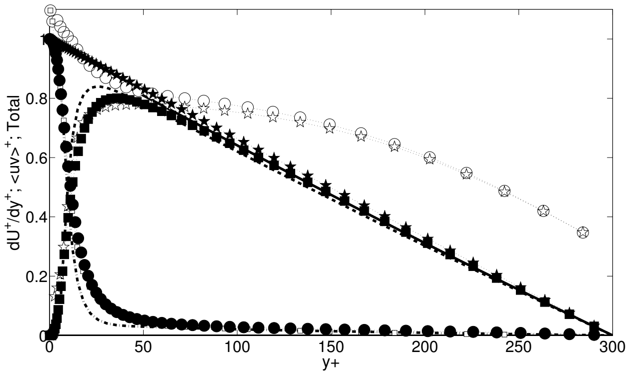


Figure 7: Shear Reynolds stresses comparison of DL-RANS (unfilled symbols) for a developed turbulent channel flow with $Re_\tau = 300$, with DNS data (filled symbols), and $\kappa - \epsilon$ results (lines). ; $\star \cdot \star \cdot \star$, total; $\square \cdot \square \cdot \square$, turbulent; $\circ \cdot \circ \cdot \circ$, molecular. $\kappa - \epsilon$, solid line, total; $-\ \ -\ \ -$, turbulent; $- \cdot - \cdot -$, molecular.

the $\kappa - \epsilon$ model) occasionally presented a low increase generating a wrong value of $\langle uv \rangle^+$, as is shown in Figure 7. Figure 6 shows the performance of the Reynolds normal stresses from the DNN. Since normal Reynolds stresses do not affect the mean flow, there is not a non-linear interaction between the input features and this prediction. Thus this prediction of the normal Reynolds stresses is irrelevant in developed flow.

Figure 7 shows the comparison of the molecular, the turbulent, and the total stress for DNS, $\kappa - \epsilon$, and DL-RANS. It became clear in this figure that the prediction of $\langle uv \rangle^+$ for DL-RANS is wrong. Note that this value of the shear Reynolds stress is the result from the propagation through the RANS equations for a developed flow. It is the result of the nonlinear interaction between the RANS features with the predicted Reynolds stresses. As commented above, instability problems have been found in the literature when Reynolds stresses are propagated in RANS equations. These problems are due to amplification of errors of the Reynolds stresses (Poroseva et al. (2016); Thompson et al. (2016). Here the production term in the κ equation is evaluated with the predicted $\langle uv \rangle^+$, generating this wrong value in the center of the channel.

4 CONCLUSIONS

This study presented results of the use of DL to predict the Reynolds stresses of turbulent flow. Based on features of turbulent flows from DNS data, a DNN was adjusted to predict the relevant Reynolds stresses. First, an a priori comparison for perturbed turbulent flows was made, and second, the predicted Reynolds stresses were propagated through the velocity field of a developed channel flow. In both cases there were improvements versus results from the standard $\kappa - \epsilon$ model, although some instability problems were found in the κ equation, resulting in the poor final prediction of the shear Reynolds stress.

The following are the main conclusions of this study:

1. In this study, to adjust the DNN, input features taken from DNS were used rather than RANS features.

2. A DNN with only 12 simple features of the flow has been efficient enough in capturing the Reynolds stress-flow characteristics dependence. Machine learning has proved to be a powerful tool for turbulence modeling research.

3. The propagation of the Reynolds stresses in a developed flow has shown instability problems in the κ equation, leading to the wrong prediction of the shear Reynolds stress.

4. Is it necessary to develop universal ML turbulence models or specialized models for specific turbulent flows? In this study, a DNN was adjusted to a developed flow and used in perturbed flow with reasonable a priori comparison with DNS data.

REFERENCES

Cruz M.A., R.L. Thompson, L. B. Sampaio, and R.D. Bacchi 2019 The use of the Reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling, *Comp. and Fluids*, Vol. 192, DOI: http://doi.org/10.1016/j.compfluid/2019.104258.

Curcic, M. 2019 A parallel Fortran framework for neural networks and deep learning, modern-fortran/neural-fortran, arXiv:1902.06714; GitHub.

Lecun, Y. Bengio, and G. Hinton 2015 Deep Learning. *Nature*, Vol. 521, DOI: 10.1038/nature14539.

Ling J, and Templeton J. 2015. Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Phys. Fluids*, 27:085103

Ling J, A. Ruiz, G. Lacaze, and J. Oefelein 2017. Uncertainty analysis and data-driven model advances for a jet-in-crossflow. *Journal of Turbomachinery*, **139**, Transaction of the ASME, pp. 1-9.

Ling J, Jones R, and Templeton J. 2016a. Machine learning strategies for systems with invariance properties, *J. Comput. Phys*, **318**, pp. 22–35.

Ling J, Kurzawski A, and Templeton J. 2016b. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.*, **807**.

Milano M. and P. Koumoutsakos 2002 Neural Network Modeling for Near Wall Turbulent Flow, *J. Comp. Physics*, DOI:10.1006/jcph.2002.7146.

Pasinato, H.D. 2013 Dissimilarity of Turbulent Fluxes of Momentum and Heat in Perturbed Turbulent Flows, *J. Heat Transfer*, ASME, Vol. 135, pp. 1-12.

Pasinato, H.D, and E. A. Krumrick 2021 Errors characterization of a RANS simulation, *J. V. V. Unc. Quant.*, ASME Transaction, Vol. **6**, pp. 1-9.

Pope, S. 2000 Turbulent Flows, Cambridge University Press, Cambridge.

Poroseva, S., J. D. Colmenares F., and S. M. Murman 2016 On the accuracy of RANS simulations with DNS data. *Physics of Fluids*, **28**, 115102, 1-22.

Tennekes H. and J. L. Lumley 1990 *A First Course in Turbulence*, MIT Press, Cambridge.

Thompson, R. L., L. B. Sampaio, F. B. Alves, L. Thais, and G. Mompean 2016 A methodology to evaluate statistical errors in DNS data of plane channel flows, *Comp. Fluids*, 130, pp. 1-7.

Xiao, H. and P. Cinnella 2018 Quantification of Model Uncertainty in RANS Simulations: A Review, *Progress in Aerospace Sciences*, arXiv:1806.10434 [physics.flu-dyn].

Wang, J.X., J.L. Wu, and H. Xiao 2017 Physics−informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on DNS data, Physical Review Fluids **2**(3) 034603.

Wu, J., H. Xiao, R. Sun and Q. Wang 2019 Reynolds-averaged Navier-Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. *J. Fluid Mech.*, **869**, pp 553 - 586.

Wu, J.L., Sun, R., Laizet, S., and H. Xiao, 2019 Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling. Comp. Meth. Appl. Mech. Eng. 346, 707-726.

Wu, J.L, H. Xiao, and E. Paterson 2018 Physics-informed machine learning approach for augmenting turbulence models. Physical R. Fluids **3**(7) (2018) 074602.