24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1241 MACHINE LEARNING TECHNIQUES FOR ACCURACY IMPROVEMENT OF RANS SIMULATIONS

Cruz, M. A.
Thompson, R. L.
Bacchi, R. D. A.
Universidade Federal do Rio de Janeiro, Department of Mechanical Engineering
matheusaltomare@ufrj.br

Sampaio, L. E. B.
Stanford University, Department of Mechanical Engineering

Abstract. *There is a wide number of applications where the flow is turbulent. Since Direct Numerical Simulation (DNS) and experiments are prohibitively expensive, the use of Reynolds Average Navier-Stokes (RANS) models is a necessity. However, the obtained models from this approach have low accuracy. This fact justifies the high demand for better models. In this work, a technique that uses machine learning, by means of neural networks, is used to correct the* κ*-*ω*SST RANS model considering the DNS data as ideal. The methodologies available in the literature employ the Reynolds stress tensor as the main quantity to be corrected. Once this entity is corrected, the velocity field is recalculated by the RANS transport equations. Consequently, the obtained velocity field gets closer to DNS results. It is proposed, as a new methodology, the correction of the divergent of the Reynolds stress tensor, because it is the only part that is computed in the mean linear momentum balance. This divergence can be calculated from the mean velocity and pressure fields, which are well converged, using the mean linear momentum equation. The results obtained so far have demonstrated that the divergent correction of the RANS turbulent stress field is able to reconstruct mean velocity fields.*

Keywords: *Turbulence flows, Machine learning, OpenFOAM*

1. INTRODUCTION

There is a wide number of applications where the flow is turbulent. Since Direct Numerical Simulation (DNS) ans experiments are prohibitively expensive, the use of Reynolds Average Navier-Stokes (RANS) models is a necessity. However, the obtained models from this approach have low accuracy. This fact justifies the high demand for better models. In this work, a technique that uses machine learning, by means of neural networks, will be used to correct turbulence κ-ωSST RANS model with the aim of this approaching DNS results. The methodologies present in the literature corrects the RANS Reynolds stress tensor with machine learning techniques. With this corrected, the velocity field is recalculated by the RANS transport equations. Consequently, the obtained velocity field gets closer to DNS results. It is proposed, as a new methodology, the correction of the divergent of the Reynolds stress tensor, because it is the only part that is computed in the mean linear momentum balance. This divergence can be calculated from the mean velocity and pressure fields using the mean linear momentum equation.

Reynolds-everaged Navier-Stokes (RANS) simulations are widely employed in industrial turbulent fluid dynamics applications. The two-equations models, the most popular ones, assumes the Boussinesq hypothesis J.Boussinesq (1877), a linear relation between Reynolds stress tensor ($\mathbf{R}$) and the mean strain rate tensor ($\mathbf{D}$) defined by a turbulent viscosity. But, these models do not capture all the physics of some relevant flows, compromising their results reliability. Hoffman *et al.* (1985) shows that those models do not provide satisfactory results in curvature flows, e.g., secondary flows in ducts. Flows with strong gradients and separation are poorly predicted when analysed by RANS models Craft *et al.* (1996) as well.

For the past few years, some studies have been made to apply machine learning (ML) techniques into turbulent fluid mechanics problems in order to develop Reynolds stress closures. Milano and Koumoutsakos (2002) used a multiple hidden layer neural network (NN) to replicate a near-wall channel flow, however that NN doesn't predict the turbulence in forward flows. Tracey *et al.* (2013) used kernel regression to model turbulent stress anisotropy eigenvalues. They reported some difficulties in generalizing the results to new flows and to scale to a large amount of data. After that, Tracey *et al.* (2015) proposed a single hidden layer NN to model the source terms from the Spalart Allmaras RANS model. This attempt has shown the potential of NN for turbulence modeling. Zhang and Duraisamy (2015) has used NN to correct the production term, only affecting the magnitude, but not the anisotropy, of $\mathbf{R}$. Ling *et al.* (2016c) used a random forest regression to model the deviatoric part of $\mathbf{R}$. This technique presented a poor ability to predict this tensor because of difficulty to ensure Galilean invariance. Later, Ling *et al.* (2016a) developed a NN to predict the anisotropy Reynolds stress tensor eigenvalues using a set of Galilean invariants inputs. That technique showed significant performance gains when compared to Ling *et al.* (2016c). Those results evidenced the importance of the Galilean invariance in ML turbulence modelling. More recently, Ling *et al.* (2016b) developed a deep specialized NN architecture witch ensured Galilean invariant anisotropic turbulent tensor. Those results showed that deep learning and the specialized architecture is able to provide a significant performance improvement.

In the present work, a different ML strategy is proposed to improve turbulence modeling. Instead of predict the anisotropy Reynolds stress tensor, a NN will correct its divergent. The main idea is to correct the term that really affects the mean linear momentum equation (MLME), $(\langle \mathbf{v}\rangle \cdot \boldsymbol{\nabla})\mathbf{v} - \nu\boldsymbol{\nabla}^2\langle \mathbf{v}\rangle + 1/\rho\boldsymbol{\nabla}\langle p\rangle = \mathbf{t}$. The proposed NN predicts the discrepancy between the high-fidelity $\mathbf{t}$=-$\boldsymbol{\nabla} \cdot \mathbf{R}$ (obtained from DNS or well-solved LES) and from RANS, e.g., $\Delta\mathbf{t} = \mathbf{t}_{\text{DNS}} - \mathbf{t}_{\text{RANS}}$. Once obtained that discrepancy, the turbulent flow obtained from a new RANS simulation will be corrected as, $\mathbf{t}_{\text{corrected}} = \Delta\mathbf{t}_{\text{predicted}} + \mathbf{t}_{\text{RANS}}$.

2. SQUARE DUCT FLOW

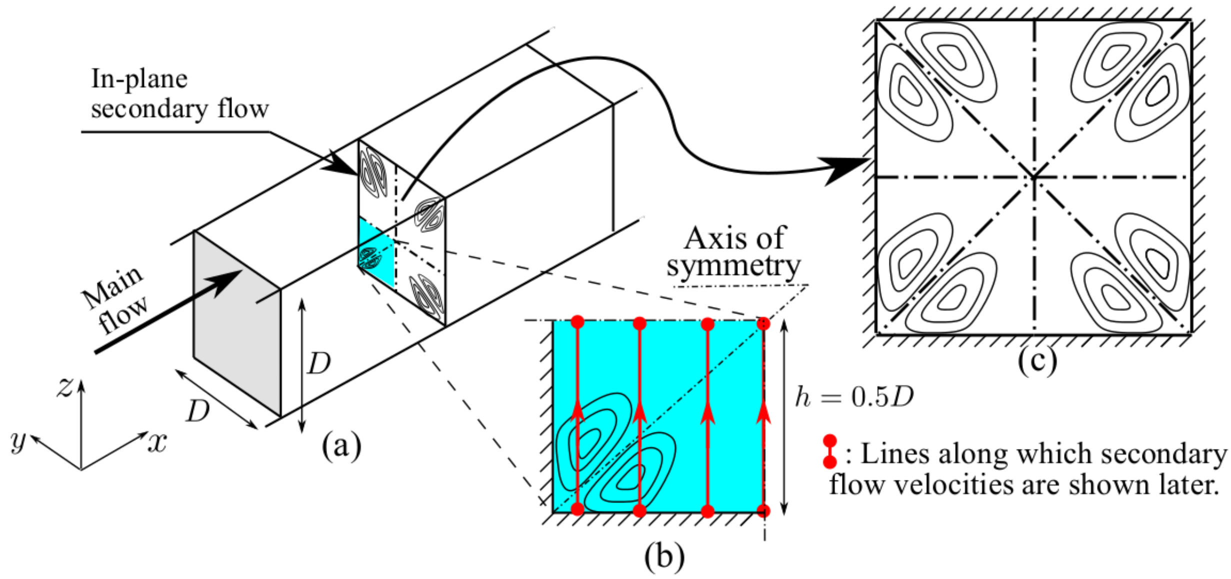


Figure 1. Schematic drawing of the square duct flow Wang *et al.* (2016).

The geometry of the square duct flow is presented in the Fig. (1). The Reynolds number that rules the problem is computed with respect to the hydraulic diameter D and the bulk velocity. As it can be seen, only a quarter of the duct section is numerically analyzed (the colored one). This flow pattern was chosen due to the existence of a secondary flow located in the cross-plane. In each one of the four edges of this cross-plane, there is a counter-rotating recirculation pair. It is known that RANS models of two equations have difficulty on capture this type of phenomenon because of the linear eddy viscosity hypothesis Hoffman *et al.* (1985). The inability to describe this secondary flow will be overcome by the correction of turbulent stresses via machine learning.

The chosen RANS model was the κ-ωSST implemented in OpenFOAM-4.x (OF). The employed boundary conditions are the non-slip condition on the solid faces, symmetry in the cut-off regions of the quarter section. In the direction of the main flow the periodic condition and the constant section average velocity are defined. The turbulent kinetic energy in the walls is zero, and for turbulent dissipation the OF wall function epsilonWallFunction was used. It was used the simpleFoam finite volume OF solver.

In order to create a neural network capable of correcting the turbulence calculated by the κ-ωSST, it is necessary to provide a reference database. The machine, to carry out its learning process, will use as reference the turbulence coming from a high fidelity and very computationally expensive database, DNS simulations. For this work, it will be used a set of flow cases corresponding to the following values for the Reynolds number:2200, 2400, 2600, 2900, 3200 and 3500. The square duct DNS made available by Pinelli *et al.* (2010). For each one of those six DNS, a corresponding RANS simulation is performed. And, with this DNS and RANS data, a NN will be trained in order to correct the turbulent flow obtained from a RANS model using the results provided by the DNS as targets.

For all the simulations, it was used a bulk velocity of $0.4819\mathrm{m\,s^{-1}}$. The solved domain has $1\mathrm{m}x1\mathrm{m}x10\mathrm{m}$, that is

$D = 1\text{m}$. Each simulations is differentiated by the kinematic viscosities listed in tab. 1.

Table 1. Simulations viscosities.

Re	$\nu[\text{m}^2\,\text{s}^{-1}]$
2200	0.00021858
2400	0.00020082
2600	0.00018537
2900	0.00016619
3200	0.00015061
3500	0.00013770

The cross-section has $40x40$ nodes, where the mesh is more refined near the walls in such a way to ensure $y^+ < 0.2$. In the main flow direction, the mesh is equally divided in 20 parts. Then the mesh has a total of 32000 centroids.

3. METHODOLOGY

At this work, the square duct turbulent flow will be corrected by the turbulent stress divergent term present in the mean linear momentum equation,

$$(\langle \mathbf{v} \rangle \cdot \boldsymbol{\nabla})\langle \mathbf{v} \rangle = -\frac{1}{\rho}\boldsymbol{\nabla}\langle p \rangle + \nu \boldsymbol{\nabla}^2 \langle \mathbf{v} \rangle - \boldsymbol{\nabla} \cdot \mathbf{R}.$$

To create these NN, the training stage will be performed with the square duct flows $Re = 2200$, 2400, 2600, 2900 and 3500 data. And to test the NN, a $Re = 3200$ square duct flow will be corrected by both NN.

The neural networks will do a non-linear mapping between same collection of inputs related to the RANS kinematics and turbulence entities, tab. 2.

Table 2. Inputs.

$\mathbf{D}$	$\mathbf{W}^2 \cdot \mathbf{D} + \mathbf{D} \cdot \mathbf{W}^2$
$\mathbf{D} \cdot \mathbf{W} - \mathbf{W} \cdot \mathbf{D}$	$\mathbf{W} \cdot \mathbf{D} \cdot \mathbf{W}^2 - \mathbf{W}^2 \cdot \mathbf{D} \cdot \mathbf{W}$
$\mathbf{D}^2$	$\mathbf{D} \cdot \mathbf{W} \cdot \mathbf{D}^2 - \mathbf{D}^2 \cdot \mathbf{W} \cdot \mathbf{D}$
$\mathbf{W}^2$	$\mathbf{W}^2 \cdot \mathbf{D}^2 + \mathbf{D}^2 \cdot \mathbf{W}^2$
$\mathbf{W} \cdot \mathbf{D}^2 - \mathbf{D}^2 \cdot \mathbf{W}$	$\mathbf{W} \cdot \mathbf{D}^2 \cdot \mathbf{W}^2 - \mathbf{W}^2 \cdot \mathbf{D}^2 \cdot \mathbf{W}$

Summing all the components of the tensorial and vectorial entities with the scalar ones presented in tab. 2, it gives a total of 60 inputs that will be used in the NN. The samples used for training the net are taken from each RANS centroids, the same was done for the output. The DNS data used as target for both nets were interpolated for those centroids as well. The chosen square duct flows for the training stage were the $Re = 2200$, 2400, 2600, 2900 and 3500. The $Re = 3200$ were separated for the NN test.

The NN has two hidden layers, each one of them with 200 neurons. The activation function of all neurons was the sigmoid function, with the exception of the output ones. These are usually chosen as linear for regressions tasks. The minimized error function was the mean square error (MSE), and the optimizer was ADAM. The stopping condition was the 'early stopping criteria', it was randomly separated 20% of the train samples to compose a validation group. During the training, the error function is calculated for the two groups, training and validation. Both errors must decrease, but when the validation error starts to increase, the training stage stops. All the neural network analyses were done with a Python's library called Keras Chollet *et al.* (2015).

3.1 Correcting $\boldsymbol{\nabla} \cdot \mathbf{R}$

To corrects the Reynolds stress tensor's divergent,

$$\boldsymbol{\nabla} \cdot \mathbf{R} = -(\langle \mathbf{v} \rangle \cdot \boldsymbol{\nabla})\langle \mathbf{v} \rangle - \frac{1}{\rho}\boldsymbol{\nabla}\langle p \rangle + \nu \boldsymbol{\nabla}^2 \langle \mathbf{v} \rangle.$$

But, for this specific DNS database Pinelli *et al.* (2010) the mean pressure field wasn't given. So, the vector $\mathbf{t}$ is defined,

$$\mathbf{t} \equiv \boldsymbol{\nabla} \cdot \mathbf{R} + \frac{1}{\rho}\boldsymbol{\nabla}\langle p \rangle, \tag{1}$$

and, this can be calculated only in terms of the velocity fields,

$$\mathbf{t} = -(\langle \mathbf{v} \rangle \cdot \boldsymbol{\nabla})\langle \mathbf{v} \rangle + \nu \boldsymbol{\nabla}^2 \langle \mathbf{v} \rangle. \tag{2}$$

For this methodology, the output will be the discrepancy between the DNS and RANS $\mathbf{t}$,

$$\Delta \mathbf{t} = \mathbf{t}_{\text{DNS}} - \mathbf{t}_{\text{RANS}}. \tag{3}$$

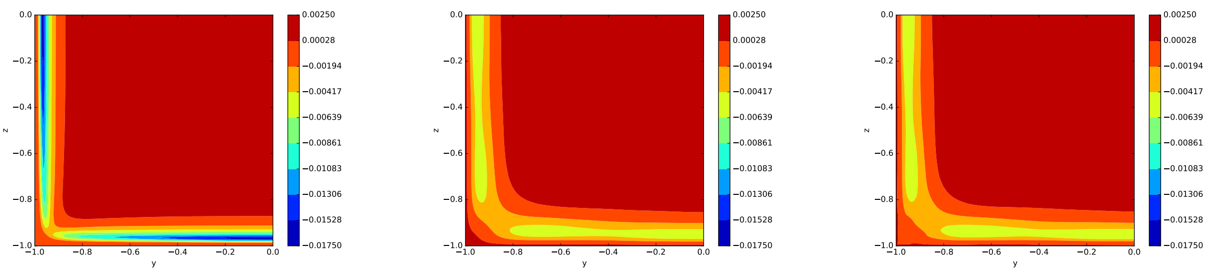
Once the NN is trained, the methodology to correct the mean velocity and pressure fields of the $Re = 3200$ square duct flow test case is

- Run the RANS;
- Post process the converged results in order to get the NN inputs described in tab. 2 from each RANS centroid;
- Run the NN with the extracted inputs before, and predicts the Reynolds stress tensor discrepancy in relation with DNS, $\mathbf{t}|_{\text{pred}}$;
- Corrects $\mathbf{t}_{\text{RANS}}$ with the estimated discrepancy before, $\mathbf{t}_{\text{cor}} = \Delta \mathbf{t}|_{\text{pred}} + \mathbf{t}_{\text{RANS}}$;
- With corrected $\mathbf{t}$, solve $(\langle \mathbf{v} \rangle \cdot \boldsymbol{\nabla})\langle \mathbf{v} \rangle = -\mathbf{t} + \nu \boldsymbol{\nabla}^2 \langle \mathbf{v} \rangle$ in order to obtain the mean corrected velocity field.

4. RESULTS

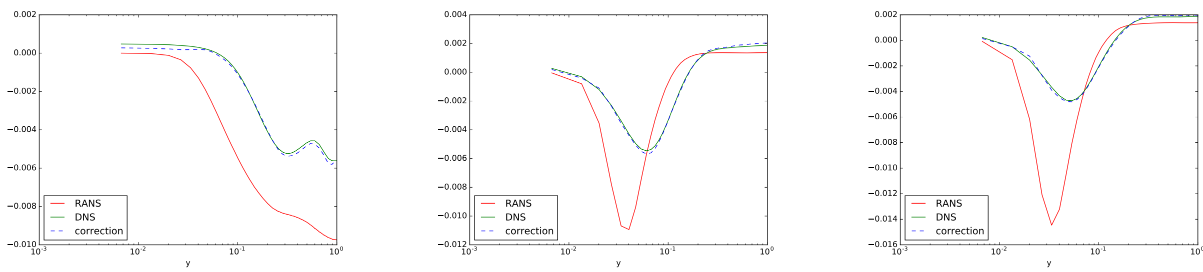
4.1 t correction

The field is well corrected in the main direction of the flow, as can be seen in Figs. (2), (3),



(a) $\mathbf{t} \cdot \hat{\mathbf{e}}_x$ RANS. (b) $\mathbf{t} \cdot \hat{\mathbf{e}}_x$ DNS. (c) $\mathbf{t} \cdot \hat{\mathbf{e}}_x$ corrected.

Figure 2. $\mathbf{t} \cdot \hat{\mathbf{e}}_x$.



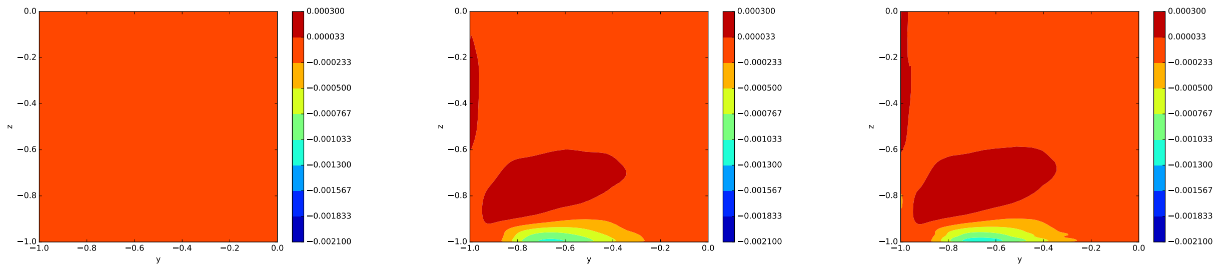
(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

Figure 3. $\mathbf{t} \cdot \hat{\mathbf{e}}_x$ component samples.

In the y direction, the field is also well corrected. Rebuilding the existing structures near the bottom solid wall from a null RANS field, Fig. (4).

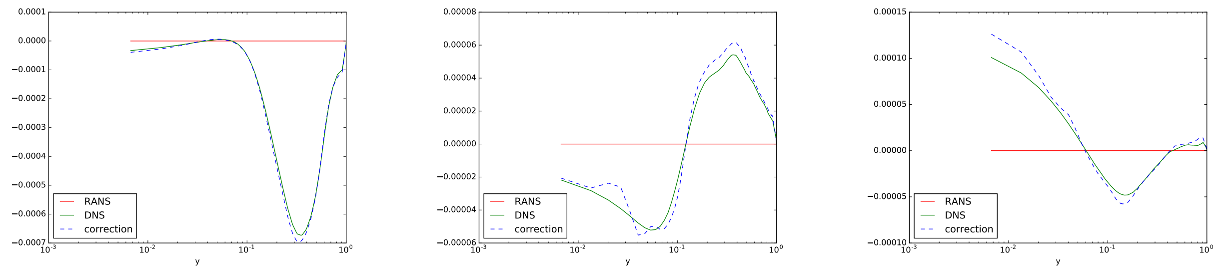
However, in the Fig. (5) it is observed that the NN generates a result as it moves away from the solid wall, towards the center of the flow.

Due to the symmetry of the flow, a behavior similar to that observed in y is seen in the z-direction. The field is reconstructed from a null RANS field, and the correction worsens in regions far from the wall, Figs. (6) and (7).



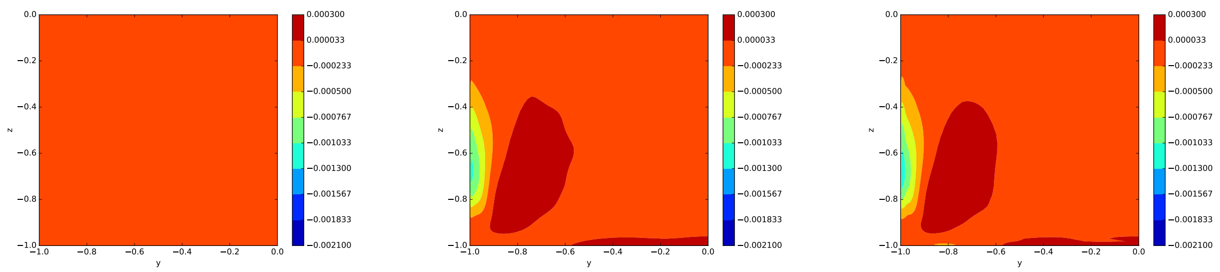
(a) $\mathbf{t} \cdot \hat{\mathbf{e}}_y$ RANS. (b) $\mathbf{t} \cdot \hat{\mathbf{e}}_y$ DNS. (c) $\mathbf{t} \cdot \hat{\mathbf{e}}_y$ corrected.

Figure 4. $\mathbf{t} \cdot \hat{\mathbf{e}}_y$ component.



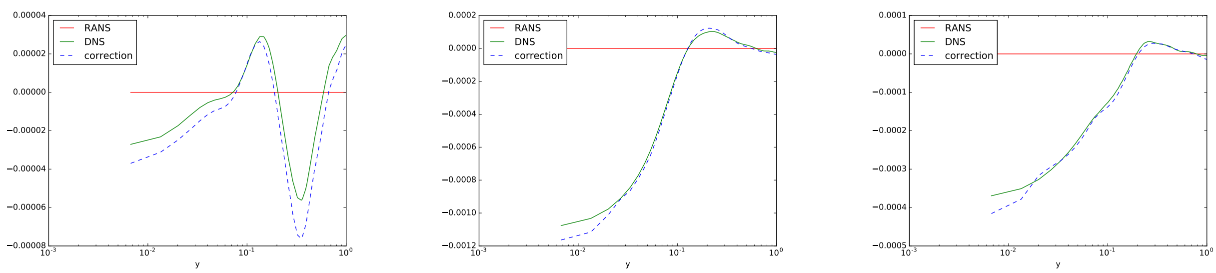
(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

Figure 5. $\mathbf{t} \cdot \hat{\mathbf{e}}_y$ component samples.



(a) $\mathbf{t} \cdot \hat{\mathbf{e}}_z$ RANS. (b) $\mathbf{t} \cdot \hat{\mathbf{e}}_z$ DNS. (c) $\mathbf{t} \cdot \hat{\mathbf{e}}_z$ corrected.

Figure 6. $\mathbf{t} \cdot \hat{\mathbf{e}}_z$ component.

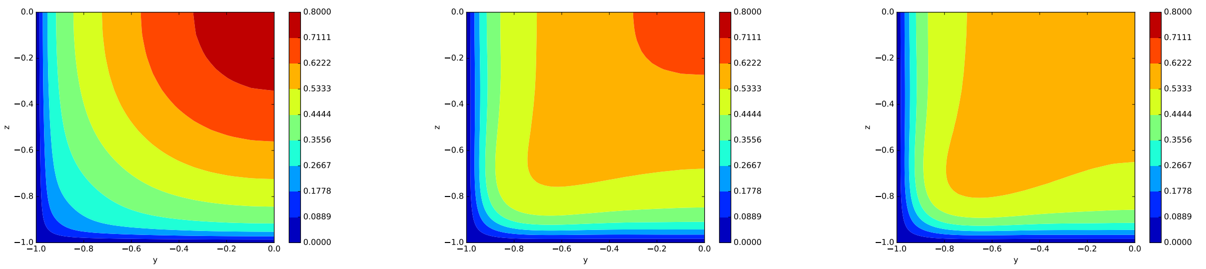


(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

Figure 7. $\mathbf{t} \cdot \hat{\mathbf{e}}_z$ component samples.

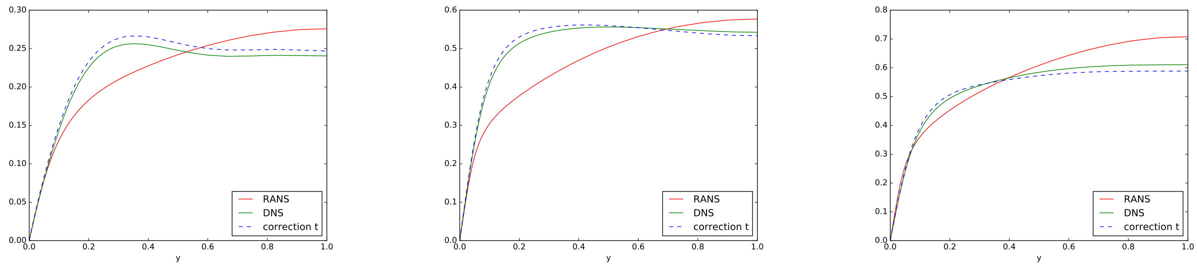
4.2 Velocity and pressure corrections

The main flow is well corrected by $\mathbf{t}$, Figs. (8) and (9).



(a) U_x RANS. (b) U_x DNS. (c) U_x corrected by $\mathbf{t}$.

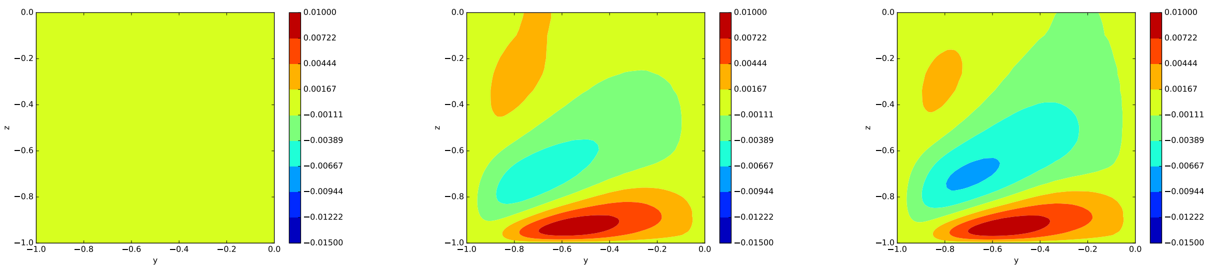
Figure 8. U_x component.



(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

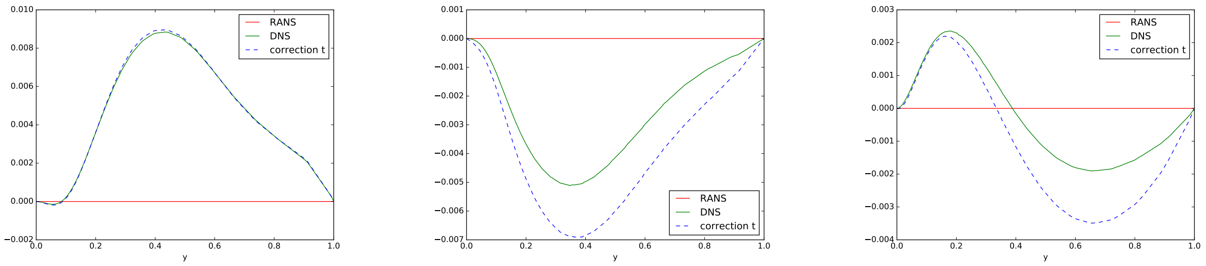
Figure 9. U_x component samples.

The technique reconstruct the structures observed in the DNS field, from a null RANS field, Fig. (10), in the y-direction. The same is seen in the z-direction, Figs. (12) and (13).



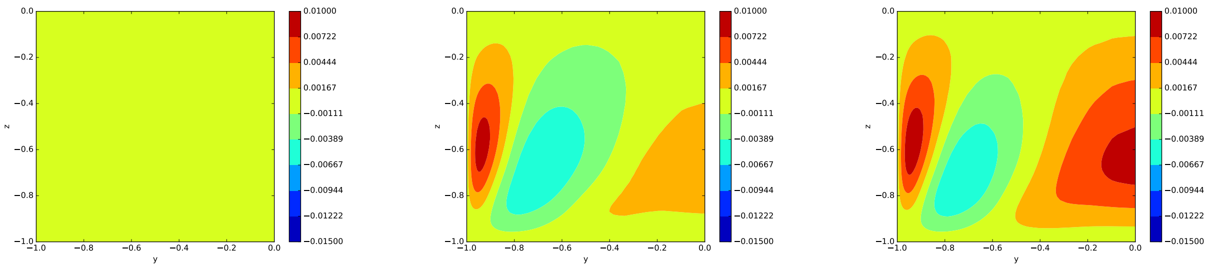
(a) U_y RANS. (b) U_y DNS. (c) U_y corrected by $\mathbf{t}$.

Figure 10. U_y component.



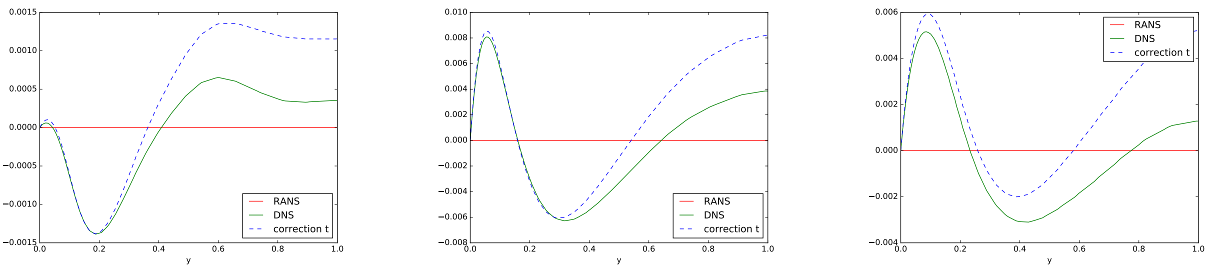
(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

Figure 11. U_y component samples.



(a) U_z RANS. (b) U_z DNS. (c) U_z corrected by $\mathbf{t}$.

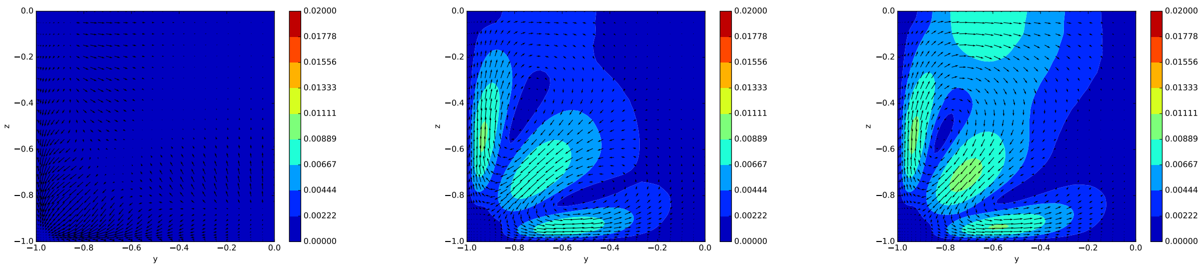
Figure 12. U_z component.



(a) $z = -0.95$. (b) $z = -0.65$. (c) $z = -0.35$.

Figure 13. U_z component samples.

The secondary flow is plotted in Fig. (14), the correction reconstruct the recirculation cells that are not seen in the RANS result.



(a) RANS. (b) DNS. (c) Corrected by$\mathbf{t}$.

Figure 14. Secondary flow magnitude contour levels ($\sqrt{U_y^2 + U_z^2}$), and stream lines.

The converged pressure gradient that feeds the flow is also corrected via $\mathbf{t}$, Fig. (15).

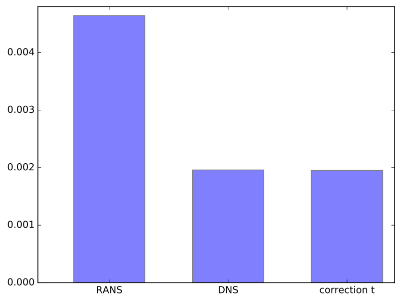


Figure 15. $-\frac{1}{\rho}\frac{\partial p}{\partial x}$.

5. CONCLUSIONS

The flow of the main direction is very well corrected by $\mathbf{t}$.
The secondary flow that is not captured by the RANS model, is recovered.
The pressure gradient that feeds the flow is corrected by $\mathbf{t}$.

6. REFERENCES

Chollet, F. *et al.*, 2015. "Keras". `https://github.com/fchollet/keras`.

Craft, T., Launder, B. and Suga, K., 1996. "Development and application of a cubic eddy-viscosity model of turbulence". *International Journal of Heat and Fluid Flow*, Vol. 17, pp. 108–115.

Hoffman, P.H., Muck, K.C. and Bradshaw, P., 1985. "The effect of a concave surface curvature on turbulent boundary layers". *J.F.M*, Vol. 161, p. 371.

J.Boussinesq, 1877. "Essai sur la theorie des eaux curantes". *Mem. Pres. Acad. Sci.*, , No. XXIII, p. 46.

Launder, B. and Spalding, D., 1974. "The numerical computation of turbulent flow computer methods". Vol. 3, pp. 269–289.

Ling, J., Jones, R. and Templeton, J., 2016a. "Machine learning strategies for systems with invariance properties." *J. Comput. Phys.*, Vol. 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.*, Vol. 807, pp. 155–166.

Ling, J., Ruiz, A., Lacanze, G. and Oefelein, J., 2016c. "Uncertainty analysis and data-driven model advances for a jet-in-crossflow." *ASME Turbo Expo*.

Milano, M. and Koumoutsakos, P., 2002. "Neural network modeling for near wall turbulent flow". *J. Comput. Phys*, Vol. 182, pp. 1–26.

Pinelli, A., Uhlmann, M., Sekimoto, A. and Kawahara, G., 2010. "Reynolds number dependence of mean flow structure in square duct turbulence." *J. Fluid Mech*, Vol. 644, p. 107âĂŞ122.

Pope, S.B., 1975. “A more general effective-viscosity hypothesis.” *J. Fluid Mech.*, Vol. 72, p. 311âĂŞ340.

Thompson, R.L. and Mendes, P.R.S., 2005. “Persistence of straining and flow classification”. *International Jornal of Engeneering Sciense*, , No. 43, pp. 79–75.

Thompson, R.L., Sampaio, L.E.B. and Alves, F.A.V.B., 2016. “A methodology to evaluate statistical errors in dns data of plane channel flows.” *Computers and Fluids*, Vol. 130, p. 1âĂŞ7.

Tracey, B., Duraisamy, K. and Alonso, J.J., 2013. “Application of supervised learning to quantify uncertainties in turbulence and combustion modeling.” *AIAA Aerospace Sciences Meeting*, Vol. 0259.

Tracey, B., Duraisamy, K. and Alonso, J.J., 2015. “A machine learning strategy to assist turbulence model development.” *AIAA Aerospace Sciences Meeting*, Vol. 1287.

Wang, J.X., Wu, J.L., Iaccarino, G. and Xiao, H., 2016. “Physics-informed machine learning for predictive turbulence modeling:towards a complete framework.”

Zhang, Z.J. and Duraisamy, K., 2015. “Machine learning methods for data-driven turbulence modeling.” *AIAA Computational Fluid Dynamics*, Vol. 2460.

7. RESPONSIBILITY NOTICE

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