CS230

Galilean Invariance Preserving Deep Learning for Canonical Fluid Flows

Carlos A. Gonzalez
Department of Mechanical Engineering
Stanford University
`cagonzal@stanford.edu`

Kimberly Liu
Department of Mechanical Engineering
Stanford University
`kimliu@stanford.edu`

1 Introduction

Computational fluid dynamics is a branch of fluid mechanics that combines physics, numerical analysis, and computer science to develop algorithms, techniques, and physical models to analyze and solve problems involving fluid flow. One of the most widespread techniques used in industry today are the Reynolds Averaged Navier-Stokes equations (RANS), a simplified version of the full Navier-Stokes (NS) equations. In the development of the RANS equations, an unclosed term, known as the Reynolds stress tensor, appears. This term must be modelled in order to close the system of equations so that it can be solved. Many physics-based RANS turbulence models have been proposed to model the Reynolds stress tensor, but these models are generally highly specialized to their application and suffer from poor performance in generalized settings. Data-driven turbulence models have recently garnered attention from the turbulence modeling community[1,2]. In this project, we will be focusing on the Tensor Basis Neural Network[3] (TBNN) approach for deriving the modeled Reynolds stress.

2 Background

The Navier-Stokes equations are the set of partial differential equations that describe the motion of viscous fluids. For this project, we will consider the incompressible Navier-Stokes equations,

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u}. \tag{2}$$

These equations represent conservation of mass and conservation of momentum for a three-dimensional velocity field $\mathbf{u} = (u, v, w)$ and a pressure field p as a function of time. The parameters ν and ρ are the kinematic viscosity and the density of the fluid, respectively. One critical observation that can be made from this set of equations is that they can be nondimensionalized by the Reynolds number $Re = UL/\nu$, where U and L are characteristic velocity and length scales inherent to the problem one is interested in studying.

$$\nabla \cdot \mathbf{u}^* = 0 \tag{3}$$

$$\frac{\partial \mathbf{u}^*}{\partial t} + \mathbf{u}^* \cdot \nabla \mathbf{u}^* = -\nabla p^* + \frac{1}{Re}\nabla^2 \mathbf{u}^* \tag{4}$$

,

The Reynolds number can be interpreted as a ratio between inertial and viscous forces. At high Reynolds number, which is the case for most engineering problems of interest, the computational cost of solving the full NS equations is prohibitively expensive. This is because high Reynolds number characterizes turbulence, which introduces a wide range of temporal and spatial scales that must be resolved. For reference, the direct numerical simulation (DNS) of a segment of an airplane wing at $Re = 400{,}000$ took 35 million CPU hours in 2015[4]. Practical engineering problems frequently feature much higher Reynolds numbers: $Re \sim 10^7$ for automobiles and $Re \sim 10^9$ for aerospace applications.

3 RANS Modeling

The RANS equations are derived by decomposing the velocity and pressure fields into average and fluctuating components,

$$\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}' \tag{5}$$

$$p = \overline{p} + p' \tag{6}$$

where $\overline{(\cdot)}$ denotes an averaged quantity and $(\cdot)'$ denotes a fluctuating quantity. This decomposition is useful for engineering purposes because we often do not care about the time-dependent solution as much as we care about average quantities. Substituting this decomposition into the Navier-Stokes equations yield the following RANS equations which describe the evolution of the mean flow fields.

$$\nabla \cdot \overline{\mathbf{u}} = 0 \tag{7}$$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \overline{\mathbf{u}} \cdot \nabla \overline{\mathbf{u}} = -\frac{1}{\rho} \nabla \overline{p} + \nu \nabla^2 \overline{\mathbf{u}} - \nabla \cdot \mathcal{R} \tag{8}$$

Here,

$$\mathcal{R} = \overline{\mathbf{u'} \otimes \mathbf{u'}} = \overline{u_i' u_j'} \tag{9}$$

is the Reynolds stress tensor. The effect of turbulent fluctuations on the mean momentum acts through this tensor. That is, $\overline{u}_i$ cannot be solved for without prescribing $\overline{u_i' u_j'}$. The Reynolds stress term is most often modeled by using physics-based arguments or, more recently, through machine learning approaches.

When using machine learning approaches for modeling in physical problems, it is important that the output of the model obeys the physical laws that are being analyzed. For example, conservation of mass and energy should always hold. In fluid mechanics, one important physical property is Galilean invariance, which states that the laws of physics do not change in different inertial frames of reference. Specifically, any scalar flow variable such as pressure or velocity magnitude should remain unchanged when the frame of reference is rotated, reflected, or translated. Thus, any machine learning model used to learn the Reynolds stress tensor needs to obey this property as well.

Past work has focused on modeling the anisotropic portion of the Reynolds stress tensor[5]

$$a_{ij} = \mathcal{R} - \frac{2}{3} k \delta_{i_j} \tag{10}$$

where k, the turbulent kinetic energy, is given by

$$k = \frac{1}{2} u_i' u_i'. \tag{11}$$

The standard normalization for the anisotropic Reynolds stress tensor is

$$b_{ij} = \frac{a_{ij}}{2k} \tag{12}$$

It can be shown that the most general representation of the anisotropic Reynolds stress tenser can be written as

$$b_{ij}(S_{ij}, \Omega_{ij}) = \sum_{n=1}^{10} G^{(n)}(\lambda_1, ..., \lambda_5) T_{ij}^{(n)} \tag{13}$$

where $T_{ij}^{(n)}$ are tensors depending on the rate of strain tensor S_{ij} and the rotation rate tensor Ω_{ij}[6]. By writing the anisotropy tensor as a linear combination of the strain rate and rotation rate tensors, and using machine learning to calculate the Reynolds stress tensor through these quantities, we can guarantee that Galilean invariance is preserved.

4 Dataset

We are using data from the UT Austin Oden Institute[7,8,9]. This database contains high-fidelity statistics and data from direct numerical simulations (DNS) of periodic channel flows ($Re_\tau \approx 180 - Re_\tau \approx 5200$) and turbulent Couette flow ($Re_\tau \approx 93 - -Re_\tau \approx 500$). The provided data includes velocity and pressure fields, as well as statistics such as mean velocity, mean pressure, mean vorticity, turbulent kinetic energy, and the Reynolds stresses. The mean statistics encompass all the field quantities required as inputs to the TBNN.

We also used channel data with superhydrophobic surface (SHS) modeling from Kim's personal research.

See Appendix B for more detailed descriptions on the characteristics of these canonical fluid flows.

5 Methods

Details on the implementation of the tensor basis neural network can be found in Ling 2016[3] (Appendix A, Figure 4). We have written code to preprocess all of our data to generate the inputs required for the TBNN. We have generated synthetic RANS data by applying a simple moving average filter to the DNS data. This serves to reproduce the lower-fidelity, smoothed out data one would expect from a RANS solver[10].

Our y profiles of data are split into 80%, 10%, 10% training, dev, and test sets on cases with the same Reynolds number. When training and dev/testing on different Reynolds number, 100$ of one set of Reynolds number data is used for training and the other set of Reynolds number data is split 50% and 50% between dev and test.

Because the dynamics of the fluid flow are highly dependent on the the y location of the data, we randomly shuffle the data profiles so that each data set is equally likely to sample every region of the flow. We have currently trained and tested TBNN models for turbulent channel data, turbulent Couette flow, and slip boundary data (see Table 1).

Our network architecture uses 100 hidden layers with 25 neurons in each layer. The training batch size is 40 points per batch with 2000 epochs. The learning rate was 2.5E-5 and L1 regularization was used. These parameters were chosen by beginning with the values used in Fang[10] and performing a small grid search in the neighborhood.

	Re_τ	N_y
Channel	550	192
	1000	256
Couette	220	96
	500	128
SHS	180	142

Table 1: Turbulent flow dataset details

6 Results

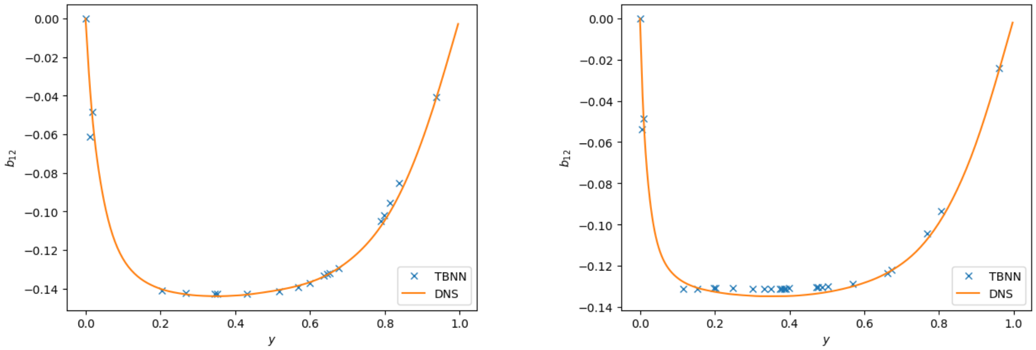
In all the canonical fluid flows we have chosen, the dominant term in the anisotropic Reynolds stress tensor is b_{12}, which represents the nonlinear effect of the fluctuating streamwise and wall-normal velocities. Below, we have plotted the TBNN predictions for this term against the exact DNS values.

Figure 1 displays the results for channel flow if our train/dev/test sets all come from the same Reynolds number. The performance of the TBNN is very good. Note that for channel flow, the Reynolds stress modeling near the wall ($y^+ = 0$) is most important.

Figure 2 shows the results if we train and dev/test on different Reynolds numbers. The b_{12} predictions for these cases are not as good as when we use the same Reynolds number, but are fairly accurate for a majority of the channel domain and maintain the correct order of magnitude and sign.

Appendix A, Figure 5-6 has the results for turbulent Couette flow, when the train/dev/test sets are of the same Reynolds number. Like in channel flow, the TBNN performs very well.

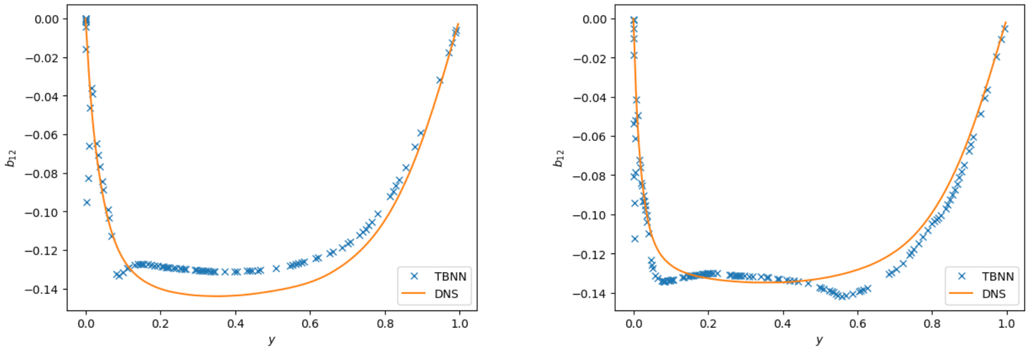
Training and testing turbulent Couette flow on different Reynolds numbers is shown in Appendix A, Figure 7-8. The performance here is significantly worse than when all data has the same Reynolds number. It is also a steeper dropoff than in the channel flow case.



(a) $Re_\tau = 550$

(b) $Re_\tau = 1000$

Figure 1: TBNN results for channel flow trained and tested on same Reynolds number.



(a) Trained on $Re_\tau = 550$, tested on $Re_\tau = 1000$

(b) Trained on $Re_\tau = 1000$, tested on $Re_\tau = 550$

Figure 2: TBNN results for channel flow trained and tested on different Reynolds numbers.

Analysis of the SHS model was attempted, but the training loss never converged. Figure 3 shows the evolution of loss over number of steps. The loss is an order of magnitude larger than the loss when training on channel flow and Couette flow (see Appendix A, Figure 9-16).

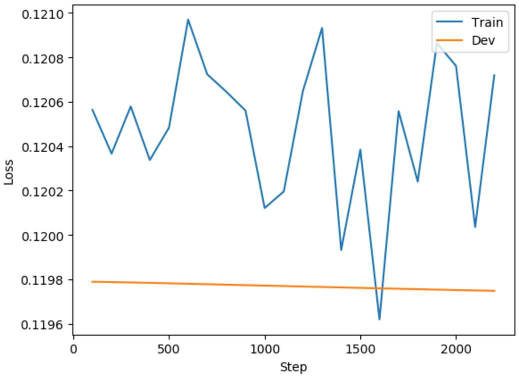


Figure 3: Training loss on SHS data

7 Discussion

We have obtained excellent results from the neural network model for both channel and Couette flow models that were trained and tested on only one Reynolds number. As expected, training a model on one Reynolds number and testing on another led to less satisfactory results. This is due to the extreme nonlinearities present in turbulent flows and is most likely a result of overfitting. However, how to resolve this dilemma is an open research question in the field.

It is likely that the SHS model failed to converge because the slip boundary condition is highly dependent on local velocity gradient, rather than the mean quantities that the TBNN is trained on (see Appendix B). It is also notable that the slip boundary condition is, itself, a model that has been shown to accurately represent mean drag reduction of a full multiphase superhydrophobic simulation. It is possible that the other mean quantities that are inputted to the TBNN are not as accurate. For example, the turbulent kinetic energy (Appendix A, Figure 18) is significantly larger than the TKE in any of the other flows.

A high priority for future work on this topic would be to use an true RANS solver to generate low-fidelity data, rather than smoothing DNS fields. This will allow us to use more realistic training data to evaluate the performance of the model. Additionally, a more thorough investigation of the TBNN would involve inputting the predicted Reynolds stress tensors to a RANS solver to create new low-fidelity data and compare flow statistics with DNS.

Another consideration for future work is that relatively small amounts of data were used to train the models. Due to the geometric simplicity of both channel and Couette flow, not many grid points are necessary in the computational domain. In effect, this means that the neural network model does not have a large volume of training data. It would be good to generate RANS and DNS profiles of channel flow and Couette flow at many more Reynolds numbers to generate more data.

Finally, it is of interest to extend the neural network model to more complex geometric configurations, such as flow over an airfoil or a backward facing step. However, these flows may suffer from the same locality issues as we encountered with the SHS model.

8 Contributions

Both group members contributed equally to the project, but members were more involved in certain areas than others. Carlos took the lead on the initial topic research, wrote much of the preprocessing code, and handled the hyperparameter turning. Kim debugged the preprocessing code, processed her research data for the case of the SHS flow, and took lead on making slides for the video and writing reports.

9 Appendix A: Figures

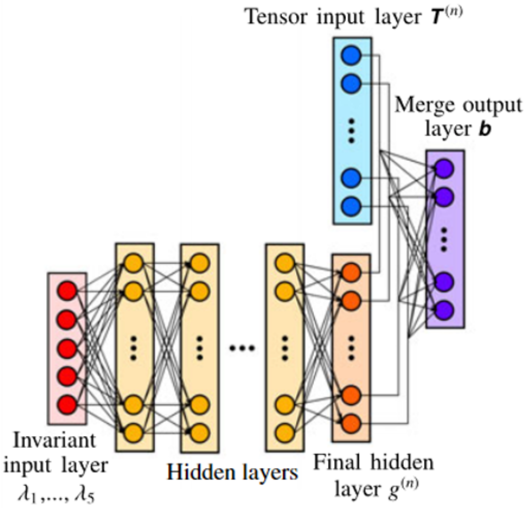


Figure 4: Schematic of the tensorial basis neural network architecture[3]

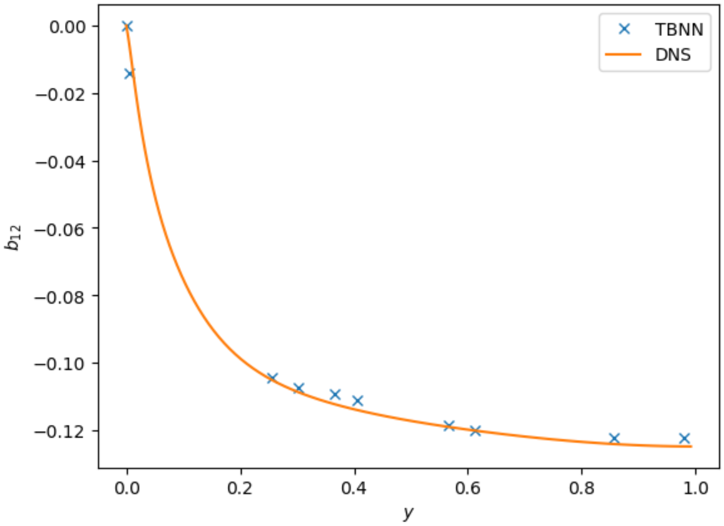


Figure 5: TBNN results for Couette flow trained and tested on $Re_\tau = 220$.

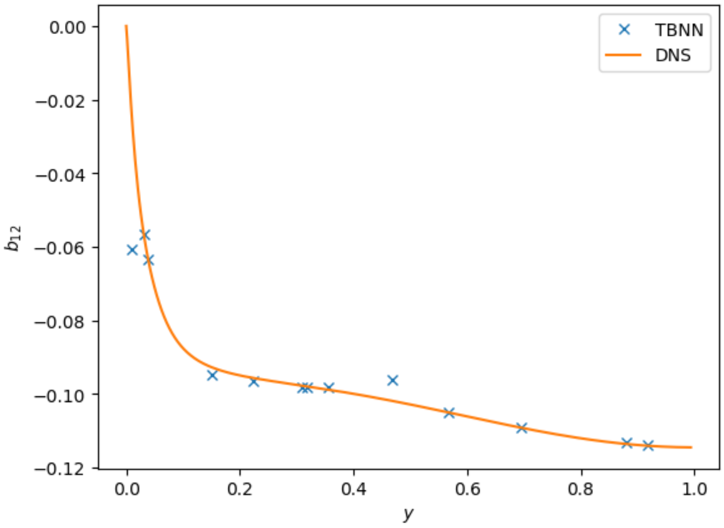


Figure 6: TBNN results for Couette flow trained and tested on $Re_\tau = 500$.

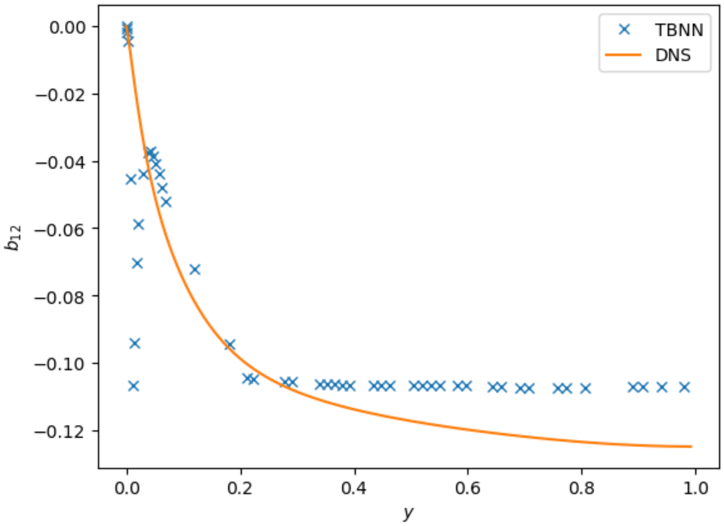


Figure 7: TBNN results for Couette flow trained on $Re_\tau = 220$ and tested on $Re_\tau = 500$.

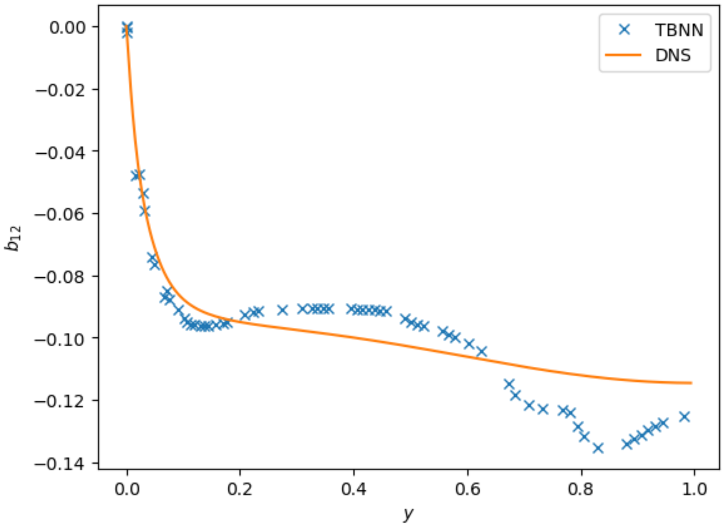


Figure 8: TBNN results for Couette flow trained on $Re_\tau = 500$ and tested on $Re_\tau = 220$.

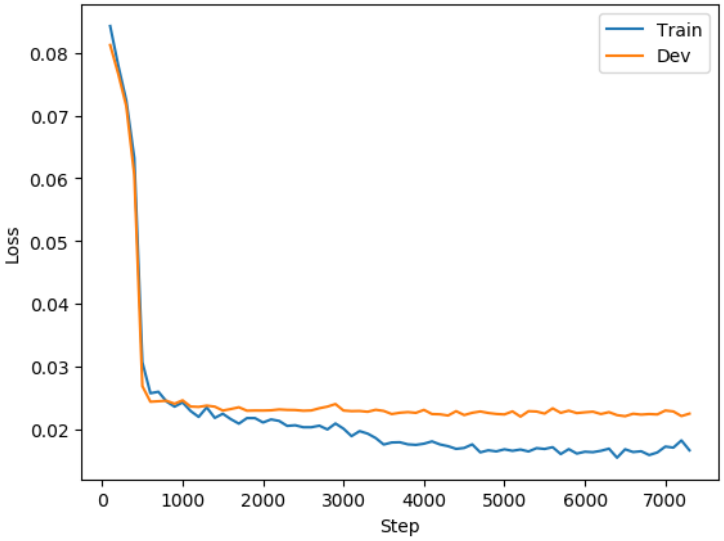


Figure 9: Loss for channel flow trained and tested $Re_\tau = 550$.

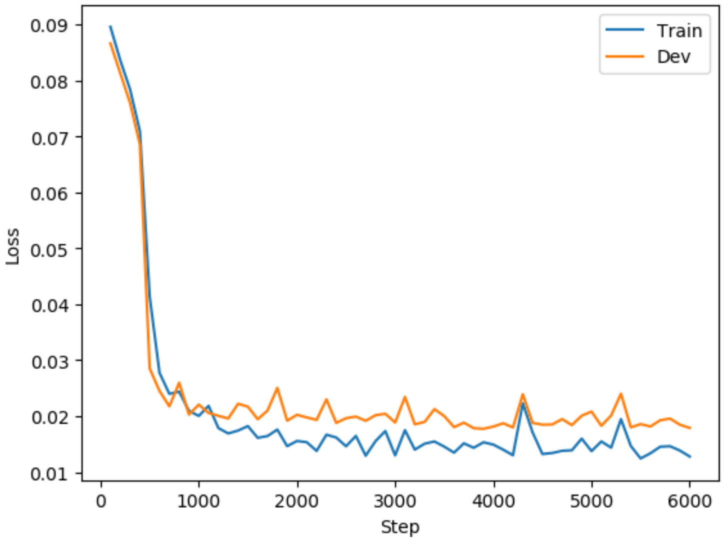


Figure 10: Loss for channel flow trained and tested on $Re_\tau = 1000$.

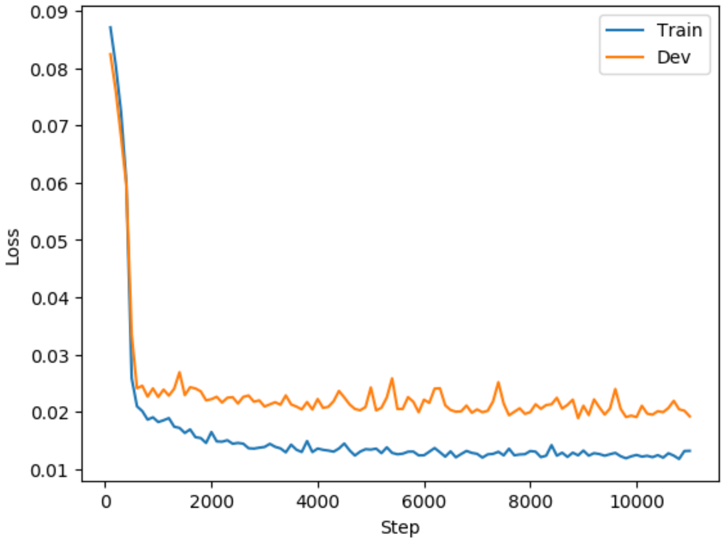


Figure 11: Loss for channel flow trained on $Re_\tau = 550$ and tested on $Re_\tau = 1000$.

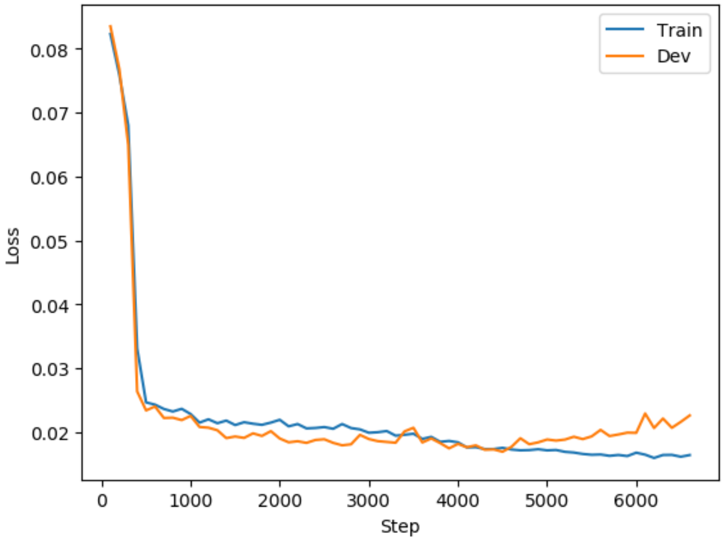


Figure 12: Loss for channel flow trained on $Re_\tau = 1000$ and tested on $Re_\tau = 550$.

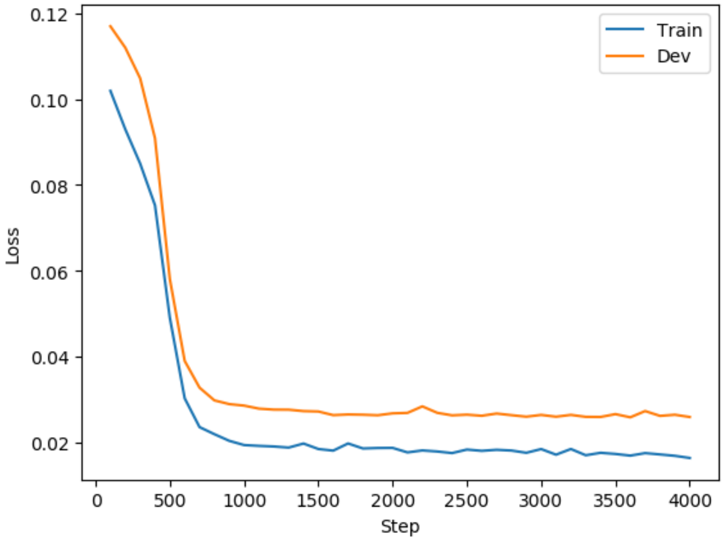


Figure 13: Loss for Couette flow trained and tested on $Re_\tau = 220$.

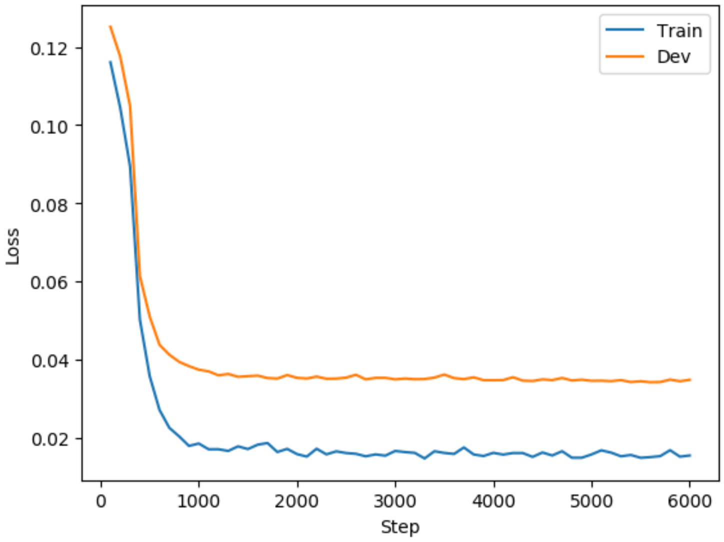


Figure 14: Loss for Couette flow trained and tested on $Re_\tau = 500$.

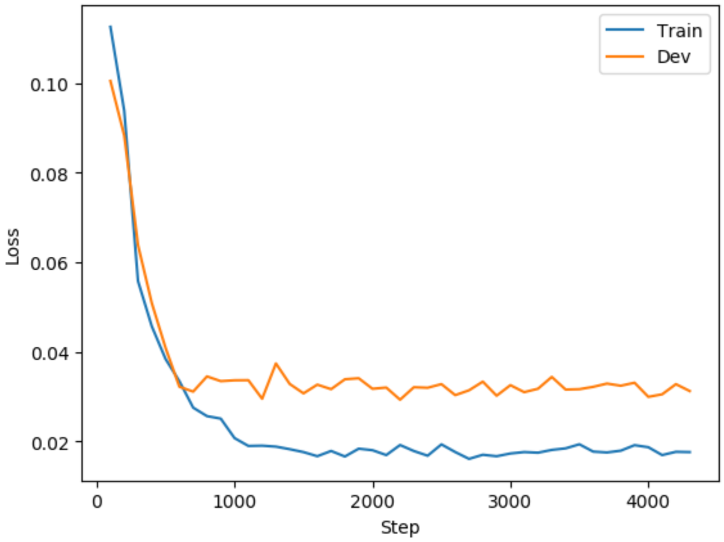


Figure 15: Loss for Couette flow trained on $Re_\tau = 220$ and tested on $Re_\tau = 500$.

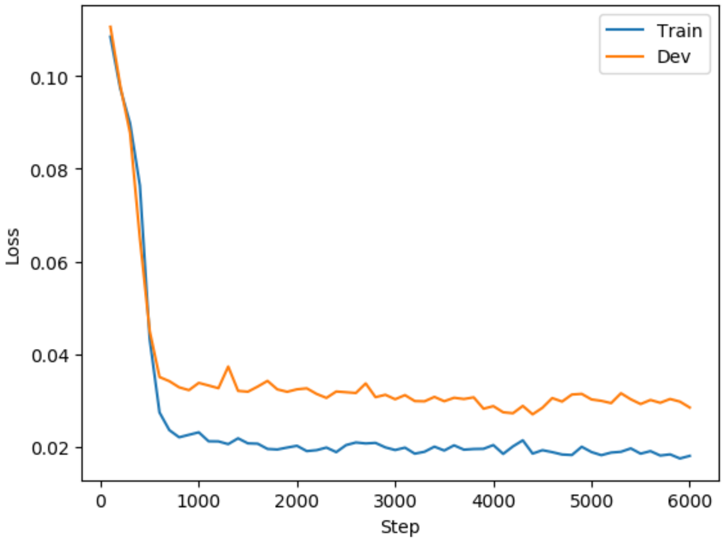


Figure 16: Loss for Couette flow trained on $Re_\tau = 500$ and tested on $Re_\tau = 220$.

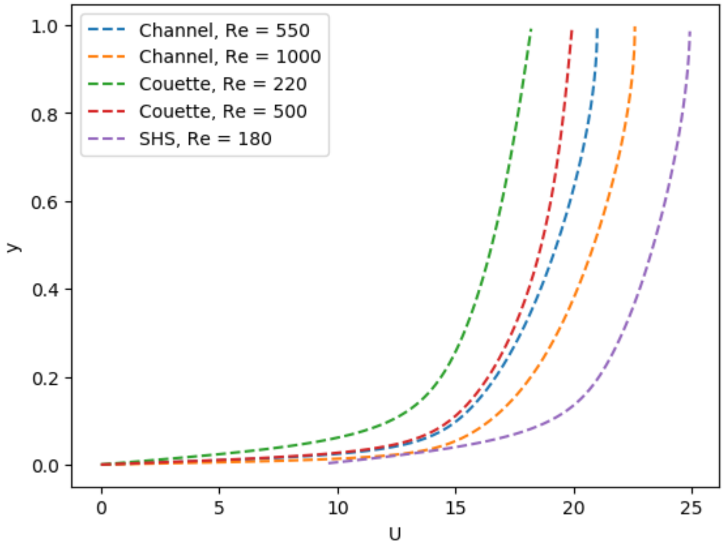


Figure 17: Mean velocity profiles of all flows studied.

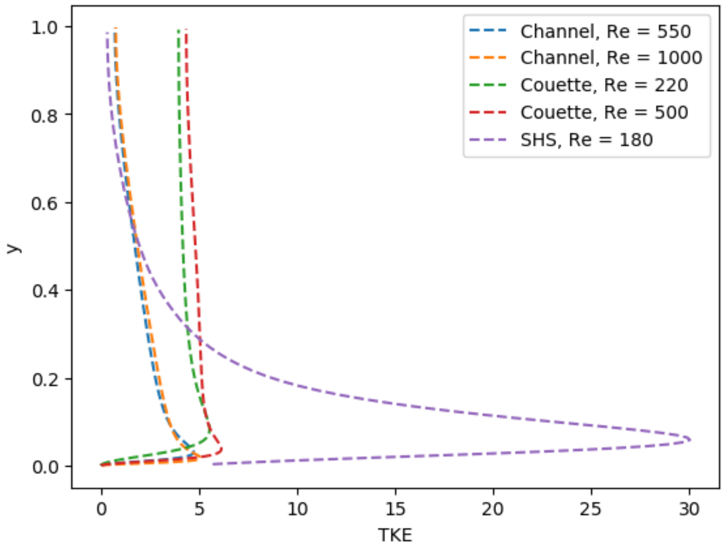


Figure 18: Mean turbulent kinetic energy of all flows studied.

10 Appendix B: Background on Fluid Flows

i. Channel flow

Channel flow is a pressure-driven flow of a fluid through a rectangular duct of height $2h$ in which the mean flow is predominantly in the axial direction (x), and the mean velocity is varying mainly in the wall-normal (y) direction. The extent of the channel in the spanwise (z) direction is large such that the flow is statistically independent of z. Near the entry of the channel ($x = 0$), there is a flow-development region in which the physics are complicated. However, sufficiently far away from the inlet, flow statistics will no longer vary with x. Computationally, the fully developed flow is represented by implementing periodic boundary conditions in the x and z directions.

Note that the convention in fluid dynamics is that $x = x_1$, $y = x_2$, and $z = x_3$ are used practically interchangeably. Similarly, the velocity in the x direction can be u or u_1, velocity in the y direction is v or u_2, and velocity in the z direction is w or u_3.

The governing equations for the mean flow are

$$\nu \frac{d\overline{u}}{dy} = \overline{u'v'} + \frac{\tau_w}{\rho}\left(1 - \frac{y}{h}\right) \tag{14}$$

where

$$\tau_w = \rho\nu \frac{d\overline{u}}{dy}\bigg|_{y=0}. \tag{15}$$

The boundary conditions at the wall are no-slip, meaning that the fluid velocity is equal to the velocity of the walls. In this case, the fluid velocity is zero at the walls.

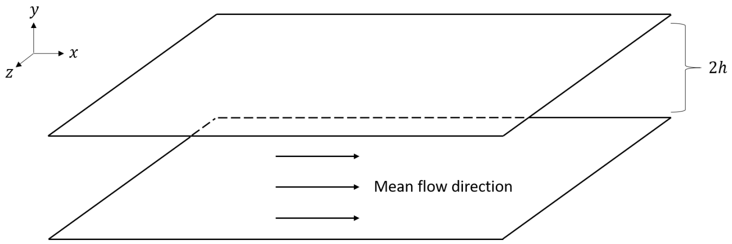


Figure 19: Channel flow schematic

ii. Couette flow

Couette flow is the flow of a fluid between two surfaces, one of which is impulsively started and moving tangentially relative to the other. The relative motion of the surfaces imposes a shear stress on the fluid and induces the flow. After the initial start up period, the flow becomes statistically stationary in time and independent of x and z.

$$\frac{d\overline{v}}{dy} = 0 \tag{16}$$

$$\nu \frac{d\overline{u}}{dy} = \overline{u'v'} + \tau_w \tag{17}$$

There are two key differences between channel flow and Couette flow. The first is that channel flow is driven by a pressure gradient, while Couette flow is driven by the shear induced by the velocity of the moving surface. The second difference is that, by the no-slip boundary condition, the velocity of the fluid on the upper wall of Couette flow is equal to the moving wall's velocity. This is in contrast to channel flow, where the upper wall is stationary.

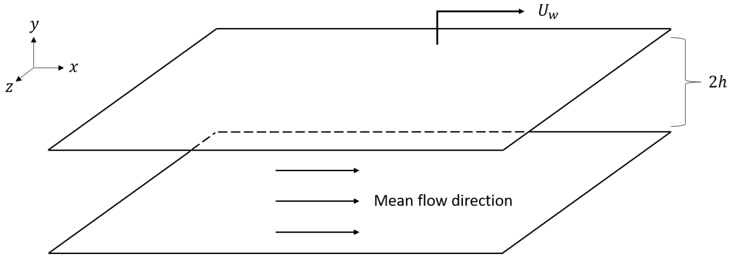


Figure 20: Couette flow schematic

iii. Superhydrophobic surface

A superhydrophobic surface is one in which a thin air film is formed between the solid surface and the fluid it is immersed in. This air film reduces the viscosity at the wall and leads to significant drag reduction at the wall.

This drag reduction is commonly modelled as a slip boundary condition[11,12,13,14]. Instead of the no-slip condition described in the channel and Couette flows, a slip boundary condition at the wall means that

$$u(y=0) = b\frac{du}{dy}\bigg|_{y=0} \tag{18}$$

$$w(y=0) = b\frac{dw}{dy}\bigg|_{y=0} \tag{19}$$

where b is some fixed slip length. Note that the velocity gradient involved in the boundary condition is a local velocity gradient, not a mean velocity gradient.

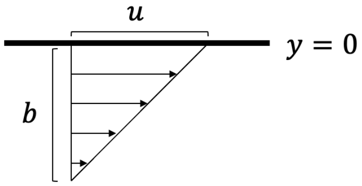


Figure 21: Slip boundary condition schematic

The SHS data used in our analysis follows the same governing equations as channel flow. The only difference is the boundary condition at $y = 0$ and $y = 2h$. In our data, the nondimensionalized slip length is $b^+ = 9$.

v. Tensor invariants

It is shown in Pope[6] that the most general representation of the anisotropic Reynolds stress tensor in terms of the mean strain-rate and rotation rate tensors is

$$b_{ij}(\hat{S}_{ij}, \hat{R}_{ij}) = \sum_{n=1}^{10} G^{(n)}(\lambda_1, ..., \lambda_5)T_{ij}^{(n)} \tag{20}$$

where λ_i are scalar invariants and T_{ij} are the tensor basis.

$$
\begin{aligned}
T^{(1)} &= \hat{S} & T^{(6)} &= \hat{R}^2\hat{S} + \hat{S}\hat{R}^2 - \frac{2}{3}\operatorname{Tr}\Big(\hat{S}\hat{R}^2\Big)I \\
T^{(2)} &= \hat{S}\hat{R} - \hat{R}\hat{S} & T^{(7)} &= \hat{R}\hat{S}\hat{R}^2 - \hat{R}^2\hat{S}\hat{R} \\
T^{(3)} &= \hat{S}^2 - \frac{1}{3}\operatorname{Tr}\Big(\hat{S}^2\Big)I & T^{(8)} &= \hat{S}\hat{R}\hat{S}^2 - \hat{S}^2\hat{R}\hat{S} \\
T^{(4)} &= \hat{R}^2 - \frac{1}{3}\operatorname{Tr}\Big(\hat{R}^2\Big)I & T^{(9)} &= \hat{R}^2\hat{S}^2 + \hat{S}^2\hat{R}^2 - \frac{2}{3}\operatorname{Tr}\Big(\hat{S}^2\hat{R}^2\Big)I \\
T^{(5)} &= \hat{R}\hat{S}^2 - \hat{S}^2\hat{R} & T^{(10)} &= \hat{R}\hat{S}^2\hat{R}^2 - \hat{R}^2\hat{S}^2\hat{R}
\end{aligned}
\tag{21}
$$

where

$$\hat{S} = \frac{k}{2\epsilon}\left(\nabla\overline{\mathbf{u}} + (\nabla\overline{\mathbf{u}})^T\right) \tag{22}$$

$$\hat{R} = \frac{k}{2\epsilon}\left(\nabla\overline{\mathbf{u}} - (\nabla\overline{\mathbf{u}})^T\right). \tag{23}$$

The invariants are

$$\lambda_1 = \operatorname{Tr}\Big(\hat{S}^2\Big), \quad \lambda_2 = \operatorname{Tr}\Big(\hat{R}^2\Big), \quad \lambda_3 = \operatorname{Tr}\Big(\hat{S}^3\Big), \quad \lambda_4 = \operatorname{Tr}\Big(\hat{R}^2\hat{S}\Big), \quad \lambda_5 = \operatorname{Tr}\Big(\hat{R}^2\hat{S}^2\Big). \tag{24}$$

For the case of channel flow and Couette flow, the structure of $\hat{S}$ and $\hat{R}$ are such that λ_3 and λ_4 are identically zero. It was important to remove these scalar invariants from the neural network model, otherwise division by zero was encountered. For the SHS channel, all five scalar invariants are active.

References

[1] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[2] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[3] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[4] Seyed Mohammad Hosseini, Ricardo Vinuesa, Philipp Schlatter, Ardeshir Hanifi, and Dan S Henningson. Direct numerical simulation of the flow around a wing section at moderate reynolds number. *International Journal of Heat and Fluid Flow*, 61:117–128, 2016.

[5] Stephen B Pope. Turbulent flows, 2001.

[6] SB Pope. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(2):331–340, 1975.

[7] JC Del Alamo and J Jimenez. Direct numerical simulation of the very large anisotropic scales in a turbulent channel. *Center for Turbulence Research Annual Research Briefs*, 2001.

[8] Juan C Del Alamo and Javier Jiménez. Spectra of the very large anisotropic scales in turbulent channels. *Physics of Fluids*, 15(6):L41–L44, 2003.

[9] Juan C Del Alamo, Javier Jiménez, Paulo Zandonade, and Robert D Moser. Scaling of the energy spectra of turbulent channels. *Journal of Fluid Mechanics*, 500:135, 2004.

[10] Rui Fang, David Sondak, Pavlos Protopapas, and Sauro Succi. Neural network models for the anisotropic reynolds stress tensor in turbulent channel flow. *Journal of Turbulence*, 21(9-10):525–543, 2020.

[11] Eric Lauga and Howard A Stone. Effective slip in pressure-driven stokes flow. *Journal of Fluid Mechanics*, 489:55, 2003.

[12] Robert J Daniello, Nicholas E Waterhouse, and Jonathan P Rothstein. Drag reduction in turbulent flows over superhydrophobic surfaces. *Physics of Fluids*, 21(8):085103, 2009.

[13] A Busse and ND Sandham. Influence of an anisotropic slip-length boundary condition on turbulent channel flow. *Physics of Fluids*, 24(5):055111, 2012.

[14] Jongmin Seo and Ali Mani. On the scaling of the slip velocity in turbulent flows over superhydrophobic surfaces. *Physics of Fluids*, 28(2):025110, 2016.