Neural network-based wall models for large-eddy simulation

Michael Whitmore

March 19 2021

1 Introduction

The problem of wall-bounded turbulent flows is ubiquitous in nature and engineering. In particular high Reynolds number wall-bounded flows are of interest in the transportation and aerospace industries. Computational fluid dynamics (CFD) is an important tool for understanding these flows. While CFD approaches are powerful in the richness of information they can provide, they face a significant challenge because the large separation of scales (due to high Reynolds number) often leads to intractably large grid point requirements for direct simulations [Choi and Moin, 2012]. A popular approach to simulating turbulent flows is large eddy simulation (LES) where only the large energy-containing scales are resolved on the computational grid and the effect of small scales is handled by a model, often referred to as the subgrid-scale (SGS) model. The power of LES relative to other CFD methods is in its predictive nature, since it can provide high-fidelity information at a low cost with few or no empirically tuned constants. However, the simulation of wall-bounded turbulence faces an issue in the region near the wall where the energy-containing eddies scale with the distance from the wall. In order to resolve this region additional grid points are required which quickly leads to very expensive simulations. Instead, the approach of wall-modeled large-eddy simulation (WMLES) keeps the coarse LES grid in the near-wall region and introduces a wall model to handle the effect of the wall on the flow.

A common approach to wall modeling is the use of wall stress models where information from the LES is used to predict the correct value of the wall shear stress. This wall shear stress is then applied back on the flow as a boundary condition. Wall models of this type are often physics-based. A common example is the equilibrium wall model which assumes the flow between the wall and some wall-normal height is in equilibrium and in turn solves a Reynolds-averaged Navier-Stokes (RANS) equation to find the wall stress. While physics-based approaches have been widely studied, data-based approaches to wall modeling are less well understood. One reason for this is because the data needed to train such a model is not widely available and can be difficult to process. In particular, experimental data often cannot provide the level of detail needed to train a data-based wall model. This leaves only direct numerical simulation (DNS) data. DNS data can be hard to come by because simulating high Reynolds number flows is expensive and leads to extremely large data sets. Further DNS data is often only available for simple flow geometries.

Despite these challenges, in the past decade or two, with the advent of more powerful computational resources, DNS data is becoming increasingly available. This has led to some work on data-based approaches to wall modeling. Yang et al. [2019] developed a neural network model for estimating the wall shear stress based on the LES velocities. Physically motivated scalings were used to normalize the inputs in order to overcome the limited range of DNS data sets to give the model improved extrapolation behavior. However this model implementation relied on inputs that are not typically known in a realistic simulation and which would require an iterative approach. Additionally the training data and inputs would limit this model to channels or similar flows. An alternative approach was taken by Lozano-Duran and Bae [2020] to try to incorporate training data from more complex flows. One draw back of this model compared to the previous model is that its predictions are limited to flow conditions available in its training data. To address this, the regression model was augmented with a classifier. The benefit of the classifier was such that the model could provide a confidence level in its prediction by determining whether the flow state was similar to any of its training data. In this way the model was self-critical and could highlight areas of the flow where it was underperforming.

Data-based approaches have seen success in other areas of LES, notably data-based SGS models. Early work by Sarghini et al. [2003] showed the predictive capability of neural networks in turbulent flows. Further, considerable work has been done using data-based approaches for the analogous problem of turbulence modeling in RANS formulations. Work in this area provides important insights that are also relevant to the problem of wall modeling in LES, particularly the idea of embedding invariance by choosing an appropriate set of inputs or applying transformations to the inputs in order to train invariance [Ling et al., 2016a,b].

2 Methodology

2.1 Physical framework

In fluid mechanics length and velocity scales are chosen in order to non-dimensionalize quantities and provide scalings. For incompressible wall-bounded turbulence the viscous scaling is given by the kinematic viscosity ν and the wall shear stress τ_w. The relevant velocity and length scales become respectively $u_\tau = \sqrt{\tau_w}$ and $\delta_\nu = \nu/u_\tau$, where u_τ is the friction velocity and δ_ν is the viscous length scale. Non-dimensionalization in viscous units is denoted with $(\cdot)^+$. From these scales a friction Reynolds number is defined

$$\mathrm{Re}_\tau = \frac{u_\tau \delta}{\nu} \tag{1}$$

where δ is typically some measure of the boundary layer thickness or outer flow length scale. This Reynolds number becomes important in governing the behavior of turbulent velocities in wall-bounded flows. Ideally any wall-model should be invariant to Reynolds number in order to work in a broader range of flows.

In wall-bounded flows it is commonly found that the mean velocity profile follows a universal log-law, given by

$$\langle u_{||}^+ \rangle = \frac{1}{\kappa} \ln\left(\frac{y}{y_0}\right) \tag{2}$$

where $u_{||}$ is the wall-parallel velocity, y is the wall-normal coordinate, $\langle\cdot\rangle$ denotes the average in homogeneous directions, and κ and y_0 are empirical parameters. A common value for κ is 0.41 while the value of y_0 depends on the viscous scaling but can be calculated by $y_0^+ = \exp(-\kappa B)$ where B is commonly ~ 5.0. Following the work of Yang et al. [2019], this scaling for the velocity will be assumed when formulating the wall model in order to provide Reynolds number in variance. More explanation of the physical motivations for this scaling is provided in that work.

2.2 Model formulation

In this work densely-connected neural networks are implemented. The models are designed to solve a regression problem where the inputs are information from the outer LES and the output is the wall shear stress. For application to WMLES it is important that the model be relatively inexpensive to evaluate. The reason for this is that the model must be evaluated at every point on the wall boundary for every time step in a simulation. This means the model will likely need to be evaluated anywhere from millions to billions of times in a simulation. Thus, its computational cost can quickly become the bounding operation in a simulation.

The model is implemented in python with tensorflow using subclassing. This, among other features, allows for customization of the loss function. The hidden layers use a hyperbolic tangent activation and the output layer is linear. The model is optimized using the L-BFGS-B optimizer from the scipy package. The code to use this optimizer is adapted from the starter code for homework 5 (`github.com/EricDarve/me343-cme216-winter-2021`). The details of the models are given in Table 1 including structure, inputs, and outputs.

For the model optimization a mean squared error loss function is used. However, an additional term is included in the loss function. This term is a penalty term used to enforce invariance of the mean wall stress to filter size. This term is computed by taking the mean of the predicted stress across the whole channel

Name	Layers	Nodes	Inputs	Outputs
DNN1	3	(4,3,3)	$\frac{\lvert u_{\lvert\lvert}^+\rvert}{h_{wm}^+}$, $\frac{\ln(h_{wm}/y_0)}{\lvert u_{\lvert\lvert}^+\rvert}$	$\lvert\tau_w^+\rvert$
DNN2	4	(6,4,3,3)	$\frac{\lvert u_{\lvert\lvert}^+\rvert}{h_{wm}^+}$, $\frac{\ln(h_{wm}/y_0)}{\lvert u_{\lvert\lvert}^+\rvert}$, Δ_R	$\lvert\tau_w^+\rvert$

Table 1: Details of the neural networks implemented. All inputs and outputs are dimensionless. These models are designed following the work of Yang et al. [2019]. Δ_R is the ratio of filter width to h_{wm}.

wall, then computing the variance of the predicted mean wall stress for each data set with different Δ_R or h_{wm}. Mathematically the problem can be expressed as follows

$$\tau_w = f_\theta(\mathbf{x}) \tag{3}$$

where $\mathbf{x}$ is the set of training features. The optimization problem is

$$\min_\theta \left[\sum_j \sum_i (f_\theta(\mathbf{x}_i^j) - \tau_{w,i}^j)^2 + \lambda \sum_j (\text{mean}_i(f_\theta(\mathbf{x}_i^j)) - \text{mean}_{i,j}(f_\theta(\mathbf{x}_i^j)))^2 \right] \tag{4}$$

where θ are the neural network weights and biases, the first term in the sum is mean squared error relative to the training data output, the second term is the variance of the mean stress across transformations of the data set, and λ is a weighting parameter.

3 Training data

The data set used is a turbulent channel DNS at Re_τ of 180 [Del Alamo and Jimenez, 2001, Del Álamo and Jiménez, 2003]. From the DNS data, the components of the wall shear stress are computed using finite differences and the velocities are sampled at a few wall normal heights, referred to as h_{wm}. The range of wall normal heights are chosen such that the velocities are inside or near the log-layer. The channel data is filtered in order to approximate LES data. A two-dimensional box filter is used in the wall-parallel directions with a range of different filter sizes and aspect ratios. The ratio of the filter size in the streamwise direction over the height of the matching location is denoted by Δ_R. This parameter can be thought of as a measure of grid anisotropy in the context of an LES grid. The cases used for training correspond to $h_{wm}^+ = \{30, 30, 50, 50\}$ and $\Delta_R = \{4.4, 9.0, 2.7, 5.6\}$. Additionally the cases used for testing correspond to $h_{wm}^+ = \{30, 30, 50, 50\}$ and $\Delta_R = \{2.0, 18.4, 1.3, 11.5\}$. The Δ_R values in increasing order correspond to filters of width 8, 16, 32, and 64 grid points on the channel grid. By training with the data set under multiple different filtering operations and at different wall-normal sampling heights, the model can in essence learn to be invariant to these changes [Ling et al., 2016a]. Using this data, the model is trained a few times until convergence is achieved in order to avoid any sub-optimal local minima.

4 Results

The results for models DNN1 and DNN2 are given in Table 2. Overall it is seen that the model can achieve around $\sim 5\%$ mean absolute error for data with a large filter size. For the smaller filter size the model is less accurate achieving $\sim 20\%$ mean absolute error.

(a)

		Δ_R			
		8	16	32	64
h_{wm}^+	30	22.3%	15.2%	9.1%	5.1%
	50	23.7%	16.1%	9.4%	5.3%

(b)

		Δ_R			
		8	16	32	64
h_{wm}^+	30	21.5%	15.3%	9.0%	10.6%
	50	22.7%	16.2%	9.4%	6.9%

Table 2: (a) DNN1 and (b) DNN2: Mean absolute error of wall stress prediction as a percentage of τ_w target. Δ_R values are given in # of grid points from channel DNS.

Looking at the plots of τ_w in space in Figure 1 shows why the model becomes less accurate overall as the filter size decreases. It is seen that the model is not capable of resolving the "peak" events where the wall stress reaches extreme values on the wall. However, while the model cannot resolve these extreme values of the wall stress, it is doing a good job in estimating the mean value of the wall stress, which for this normalization should be 1.0. The result for a larger filter size is shown in Figure 2. In contrast, for the larger filter there are fewer extreme stress events in the domain so the model is better able to capture the behavior.

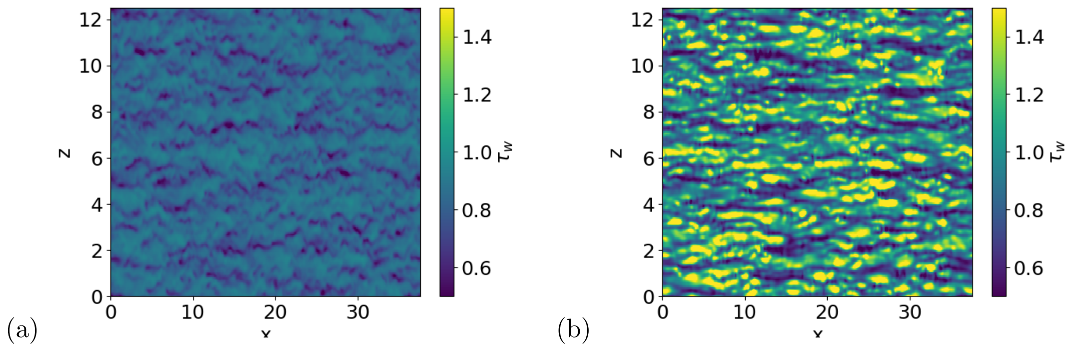


Figure 1: (a) DNN2 predicted wall shear stress and (b) true wall shear stress as a function of spatial coordinates. $h_{wm}^{+} = 50$ and Δ_R is the width of 8 grid points.

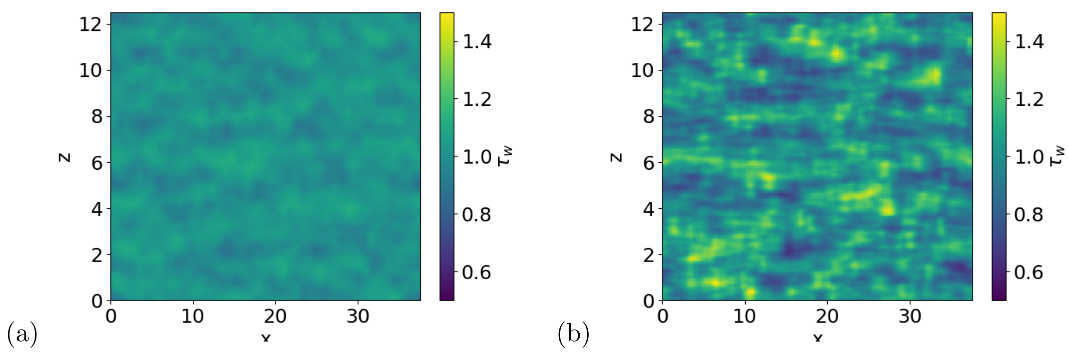
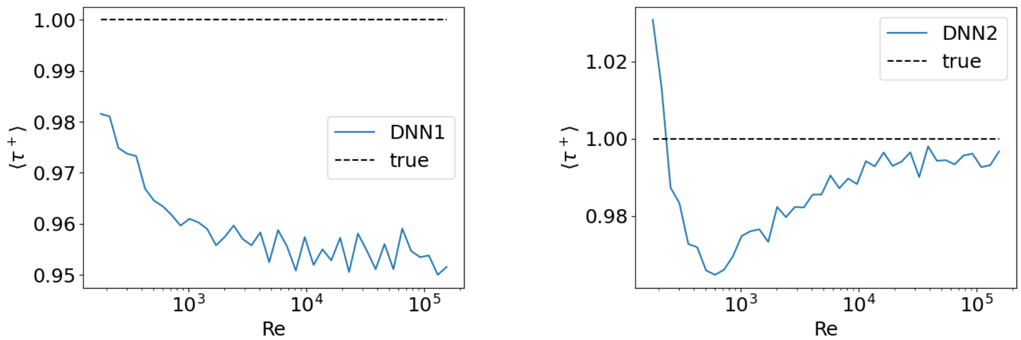


Figure 2: (a) DNN2 predicted wall shear stress and (b) true wall shear stress as a function of spatial coordinates. $h_{wm}^{+} = 50$ and Δ_R is the width of 32 grid points.

Another question to answer regarding the performance of the model is how it would behave at different Reynolds numbers. In order to assess the behavior of the model at higher Reynolds numbers, random data for $u_{||}$ was generated for higher Re_τ by assuming the log-law scaling was valid. This scaling was then used to generate pseudo-data at a corresponding wall normal height of 0.1 times the boundary layer thickness. By passing this data through the models it is possible to assess the sensitivity of the model to Reynolds number. In Figure 3 it is seen that while the predicted value of the wall stress is varying fairly significantly at low Reynolds number, both models eventually show a Reynolds number invariant behavior. This is a positive result suggesting that the model would be able to extrapolate to higher Reynolds numbers even though it was trained using data at a relatively low Reynolds number of 180. This is further supported by the fact that, while this Reynolds number pseudo-data is not real, it is expected that high Reynolds number flows will more strongly follow a log-law behavior.



(a) (b)

Figure 3: (a) DNN1 and (b) DNN2 performance of model at a range of Re_τ using random $u_{||}$ generated assuming a log-law scaling.

5 Discussion

In general, it is difficult to train a data-based wall model due to the limited amount of high-fidelity high Reynolds number data available, as well as the complications in accessing and processing it. A model that directly uses the LES velocities as inputs would then be limited in its ability to function in high Reynolds number flows because it would not be able to extrapolate outside of its training set. By training a model with data that has been scaled based on an assumed Reynolds number scaling of turbulent wall-bounded flows, it is seen here that it is possible to train a model that can accurately extrapolate to higher Reynolds number flows.

Due to the small size of the networks and the limited inputs used, it is seen that the models are not able to recover the extreme values of the wall stress. This is, however, a trade-off for computational cost and complexity. Since a wall model must be called many times during a simulation, the reduced computational cost would be highly beneficial in practice, so long as the model can reproduce the mean wall stress values.

By training the model with multiple transformations of the data set, particularly at different wall normal heights and at different filter scales, it is possible to allow the model to learn to be invariant to these changes. This is consistent with the recommendations in Ling et al. [2016a]. In addition to this a penalty term in the loss function forced the model to keep the mean wall stress roughly constant when the filter width or sampling height varied. This helped to improve the invariance of the model potentially at the expense of resolving some of the finer detail spatial structures in the wall stress.

One important limitation of this model implementation is that the input features used are not typically known in a real simulation. The result of this is that this model would likely have to be used iteratively to solve for the wall stress. However this iteration is also commonly needed in implementations of the equilibrium wall model so this is not a drawback over existing models. Another important limitation of this model is that, while it is capable of achieving good extrapolation behavior by assuming a log-law scaling, this log-law scaling is not valid in more complex flow cases. As a result this model would likely only be useful for simple flow geometries such as the channel used here, a flat plate boundary layer, or other flows that are similar in nature. In order to overcome this difficulty, the model could include more input features and also could be trained using some non-equilibrium flows.

References

H. Choi and P. Moin. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of Fluids*, 24(1):30–35, 2012.

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

J. C. Del Álamo and J. Jiménez. Spectra of the very large anisotropic scales in turbulent channels. *Physics of Fluids*, 15(6), 2003.

J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016a.

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

A. Lozano-Duran and H. J. Bae. Self-critical machine-learning wall-modeled LES. *Annual Research Briefs, Center for Turbulence Research*, 2020.

F. Sarghini, G. Felice, and S. Santini. Neural networks based subgrid scale modeling in large eddy simulations. *Computers and Fluids*, 32(1):97–108, jan 2003.

X. I. Yang, S. Zafar, J. X. Wang, and H. Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 4(3):34602, 2019.