Parameter Identification of RANS Turbulence Model using Physics-embedded Neural Network

Shirui Luo [1], Madhu Vellakal [1], Seid Koric [1], Volodymyr Kindratenko [1], Jiahuan Cui [2]

[1] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA,

[2] Zhejiang University – University of Illinois at Urbana-Champaign Institute, Haining, Zhejiang province 314400, China

Abstract. Identifying the appropriate parameters of a turbulence model for a class of flow usually requires extensive experimentation and numerical simulations, even a modest improvement of the turbulence model can significantly reduce the overall cost of a three-dimensional, time-dependent simulation. In this paper we demonstrate a novel method to find the optimal parameters in the Reynolds-averaged Navier–Stokes (RANS) turbulence model using high-fidelity direct numerical simulation (DNS) data. A physics informed neural network (PINN) that is embedded with the turbulent transport equations is studied, physical loss functions are proposed to explicitly impose information of the transport equations to deep learning networks. This approach solves an inverse problem by treating the five parameters in turbulence model as random variables, with the turbulent kinetic energy and dissipation rate as known quantities from DNS simulation. The objective is to optimize the five parameters in turbulence closures using the PINN leveraging limited data available from costly high-fidelity DNS data. We validated this method on two test cases of flow over bump. The recommended values were found to be $C_{\varepsilon 1}$ = 1.302, $C_{\varepsilon 2}$ = 1.862, C_{μ} = 0.09, σ_{κ} = 0.75, σ_{ε} = 0.273, the mean absolute error of the velocity profile between RANS and DNS decreased by 22% when used the neural network inferred parameters.

Keyword: Turbulence Modeling, Neural Network, Physics Informed, RANS

1. Introduction

Reynolds-averaged Navier–Stokes (RANS) simulation remains the workhorse computational fluid dynamics (CFD) method for industrial enterprises due to its computational efficiency and easy implementation. However, many flows are difficult to simulate accurately using RANS models, for example, flow with strong adverse pressure

and separation, jet-in-crossflow interactions. The inaccuracy could be due to RANS's inherent simplifications or, more often the case, the use of inappropriate RANS constants which usually estimated by fitting to experimental results of simple flows.[1] The closure model in RANS equations has traditionally evolved through a combined efforts of mathematics, flow theory, empiricism, and rudimentary data-driven techniques such as single or two variable curve-fitting. These tunable parameters have been determined from experiments with air and water for fundamental turbulent boundary layers and free shear flows.[2] Table 1 lists some of the most commonly ones used in many CFD solvers as the default parameter values (we will refer to them as the "default" values). They have been found to work well for a wide range of wall-bounded and free shear flows. However, there is ample empirical evidence that these parameters are far from being universal[3-5] and the optimal parameter values can vary substantially for different flow configurations. Thus, it is unlikely that the default values of κ-ε model parameters should yield accurate simulations for all cases, and calibration (either by experimental or high-fidelity simulation data) for each specific flow configuration should be a pre-requisite.[6]

Compared to experimental data, high-fidelity simulation data can provide a more comprehensive view of the flow than when used for RANS models calibration, as simulation data can provide flow quantities which cannot be measured in experiments. If high-fidelity simulation data is provided, the calibration process is then an inverse problem that given the abundant observable flow field data from simulation, we need to infer the unknown parameters in the RANS model transport equations. Solving these inverse problems with differential equations, however, is typically computationally prohibitive and often requires the solution of ill-posed problems. Early works on optimizing RANS closure parameters include schemes like adjoint-based method[7], ensemble Kalman filter[8], and evolution methods like the covariance matrix adaption evolution strategy. Instead of minimizing the error of the whole flow fields, the loss functions in these previous works usually are some aerodynamic coefficients, for example, the drag coefficient, lift coefficient, and the pitching moment coefficient when designing an airfoil.

In this work we used a physics-informed neural networks (PINN) to calibrate the five parameters in κ-ε turbulence model using the whole flow fields from high-fidelity simulations. The principal idea behind PINN is that: there are some principled physical laws that govern the time-dependent dynamics of a system (e.g. RANS turbulence model), this prior information can act as a regularization agent that constrains the space of admissible solutions to a manageable size. For example, in incompressible fluid dynamics, we can discard any non-realistic flow solutions that violate the conservation of mass principle. In return, encoding such structured information into a learning algorithm results in amplifying the information content of the data that the algorithm sees, enabling it to quickly steer itself towards the right solution and generalize well even when only a few training examples are available. The PINN is then grounded in a principled physics model yet offers the flexibility of learned representations.

It is worthy noted that our study is different with pure data-driven turbulence models, where researchers try to map a direct relationship between mean flow quantities with Reynolds stress. It is expected that a purely data-driven turbulence model is substantially more challenging than a physics-based model, because a turbulence model require to discovering both the model form and model parameters, a purely data-driven model will abandon the abundant physics in the RANS turbulence model by only relying on the neural networks to map the nonlinear relationships. As such, the drawback of purely data-driven models is that training datasets play pivoting roles and thus are lack of extrapolation ability, or poor generalization. Even though the model fits very well to training data, it cannot generalize well to unobserved test data. To overcome this drawback, the datasets are required to be sufficiently rich with great variability, which are still lacking. Despite recent efforts for efficient creation of datasets with encouraging results, generating the required training dataset size still requires substantial computational effort. Rather than improving the model-form error, our work only focuses on calibrating the uncertain parameters in the RANS model, that is, whether RANS simulations could be improved by using better parameters. This work has a similar concept of data assimilation, which involves combining observations (high-fidelity simulation data) with "prior knowledge" (mathematical representations of RANS turbulence model) to obtain an estimate of the true state (optimal parameters in RANS model). By exploiting inside the neural network training the underlying physical laws described by turbulence models, the PINN requires substantially less training data to achieve high accuracy. This work unlocks a range of opportunities in parameters tuning of fluid simulations.

The paper is organized as follows. Section 2 reviews the RANS κ-ε turbulence model, including two transport equations. Section 3 introduces the physics informed neural network and its specific application for RANS turbulence modeling. Section 4 runs a study case with the "flow over the bump" datasets. Section 5 concludes the paper.

2. RANS κ-ε turbulence model

The flow of a viscous incompressible fluid with constant properties is governed by the Navier-Stokes equations:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(u_i u_j\right) = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{1}$$

$$\frac{\partial u_i}{\partial x_j} = 0 \tag{2}$$

where u_i is the fluid velocity, p is the pressure (divided by the density ρ), ν is the fluid kinematic viscosity. In RANS, the Reynolds decomposition will decompose the dependent variables into mean and fluctuating parts:

$$u_i = \overline{u_\iota} + u_i', \quad p = \bar{p} + p' \tag{3}$$

where $\overline{u_\iota}$ and $\bar{p}$ are ensemble averages of u_i and p, respectively, and u_i' and p' are the random fluctuations about the mean field. By substituting the decomposed terms into NS equation and taking an ensemble average, one obtains the system of partial differential equations that governs the mean-velocity and pressure fields of incompressible turbulence flow:

$$\frac{\partial \overline{u_\iota}}{\partial t} + \overline{u_\jmath}\frac{\partial \overline{u_\iota}}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \nu\frac{\partial^2 \overline{u_\iota}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{4}$$

$$\frac{\partial \overline{u_\iota}}{\partial x_j} = 0 \tag{5}$$

where $\tau_{ij} = \overline{u_\iota' u_\jmath'}$ is the unclosed Reynolds-stress term that incorporates the effects of turbulence motions on the mean stresses. The Reynolds stress tensor contains six independent unknowns and solving equations requires approximating the Reynolds stress in terms of u, ∇u, or other computable quantities. In the RANS κ-ε turbulence model, this term is approximated by the eddy-viscosity model as:

$$\tau_{ij} = \overline{u_\iota' u_\jmath'} \approx \frac{2}{3}\kappa\delta_{ij} - 2\nu_T\overline{S_{\iota\jmath}} \tag{6}$$

where κ is the average kinetic energy of the velocity fluctuations:

$$\kappa = \frac{1}{2}\overline{u_\iota' u_\iota'} \tag{7}$$

$\overline{S_{\iota\jmath}}$ is the strain-rate tensor of the mean field:

$$\overline{S_{\iota\jmath}} = \frac{1}{2}\left(\frac{\partial \overline{u_\iota}}{\partial x_j} + \frac{\partial \overline{u_\jmath}}{\partial x_i}\right) \tag{8}$$

ν_T is the turbulent eddy viscosity, in the κ-ε turbulence model, this term is modeled as:

$$\nu_T = C_\mu \frac{\kappa^2}{\varepsilon} \tag{9}$$

where ε is the rate of dissipation of turbulent kinetic energy as:

$$\varepsilon = \nu \overline{\frac{\partial u_i'}{\partial x_j} \frac{\partial u_i'}{\partial x_j}} \tag{10}$$

The transport equations of turbulent kinetic energy and dissipation rate are:

$$\frac{\partial \kappa}{\partial t} + \overline{u}_\iota \frac{\partial \kappa}{\partial x_i} = -\tau_{ij} \frac{\partial \overline{u}_\iota}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_K} \frac{\partial \kappa}{\partial x_i}\right) + \nu \frac{\partial^2 \kappa}{\partial x_i \partial x_i} \tag{11}$$

$$\frac{\partial \varepsilon}{\partial t} + \overline{u}_\iota \frac{\partial \varepsilon}{\partial x_i} = -C_{\varepsilon 1} \frac{\varepsilon}{\kappa} \tau_{ij} \frac{\partial \overline{u}_\iota}{\partial x_j} + \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i}\right) - C_{\varepsilon 2} \frac{\varepsilon^2}{\kappa} + \nu \frac{\partial^2 \varepsilon}{\partial x_i \partial x_i} \tag{12}$$

The two transport equations represent the turbulent properties of flow, they can account for history effects like convection and diffusion of turbulent energy. The κ can be thought of as the variable that determines the energy in the turbulence and the ε determines the turbulence scale. The five tunable parameters in the above two transport equations are: $C_{\varepsilon 1}, C_{\varepsilon 2}, C_\mu, \sigma_\kappa, \sigma_\varepsilon$. Table 1 shows the various model constants that largely be used in CFD community. In previous efforts[9,10], the parameters are determined by requiring the turbulence model to satisfy experimental data for certain simple standard flow cases. In Launder and Sharma model, for example, the C_μ coefficient is obtained by considering the log-law region of a turbulent boundary layer. The $C_{\varepsilon 1}$ is usually fixed from calibrations with homogeneous shear flows, and $C_{\varepsilon 2}$ is usually determined from the decay rate of homogeneous, isotropic turbulence. The last two constants, σ_κ and σ_ε, are optimized by applying the model to various fundamental flows such as flow in channel, pipes, jets, wakes.[11]

Table 1. The various k-epsilon turbulence models[3,9,12,13] constants that largely be used in CFD community

	Launder & Sharma	Jones & Launder	Chien	Yakhot & Orszag
$\boldsymbol{C_{\varepsilon 1}}$	1.44	1.55	1.35	1.063
$\boldsymbol{C_{\varepsilon 2}}$	1.92	2.0	1.8	1.7215

C_μ	0.09	0.09	0.09	0.0837
σ_k	1.0	1.0	1.0	0.7179
σ_ε	1.3	1.3	1.3	0.7179

Previous efforts on clousure coefficient identification include schemes like adjoint-based method[7], ensemble Kalman filter[8], the Bayesian inference combined with some high dimensional model representation technique[14], and recently the Isogeometric Analysis for solving PDE-constrained optimization problems efficiently. The RANS model then requires case sensitive parameters in a sense that each category of flow should have their own most suitable parameters.[6,15] Once the high-fidelity data is available, a flexible and efficient and scheme that can easily identify the optimal parameters for a specific category of flow is thus imperative.

3. Physics informed neural network

The first glimpses of promise for exploiting structured prior information to construct data-efficient and physics-informed learning machines have already been showcased in the recent studies.[16-22] Based on these prior successes, the principal idea behind the PINN is that the principled physical laws (usually differential equations) that govern the time-dependent dynamic system are treated as the prior, this prior information are embedded in the network loss function and then can act as a regularization that constrains the space of admissible neural network solutions. The benefits of encoding such structured information is that it enables a learning algorithm to quickly steer itself towards the right solution. Such neural networks are constrained to respect any symmetries, invariances, or conservation principles originating from the physical laws that govern the observed data. In incompressible fluid dynamics problems, for example, we can constrain the solution space by discarding any non-realistic flow solutions that violate the conservation of mass principle. Figure 1 shows the architecture of physics informed neural network for turbulence modeling application. A feedforward neural network is constructed to map the relationship between the coordinates, velocity with the turbulent kinetic energy and dissipation rate. The five tunable parameters in the RANS models are unknowns that we wish the PINN to optimize, the network's loss function combined the mean squared error loss with physical constraints as:

$$L = \frac{1}{N}\sum(\kappa^{real} - \kappa^{pred})^2 + \frac{1}{N}\sum(\varepsilon^{real} - \varepsilon^{pred})^2 + \boldsymbol{w_f} * f + \boldsymbol{w_g} * g \quad (13)$$

where the MSE of κ and ε denote the mean squared error loss corresponding to the initial high fidelity data, the f and g enforces the physics by penalizing any deviations of the predicted physical law. They are defined based on the two transport equations as:

$$f = \frac{\partial \kappa}{\partial t} + \overline{u_\iota}\frac{\partial \kappa}{\partial x_i} + \tau_{ij}\frac{\partial \overline{u_\iota}}{\partial x_j} + \varepsilon - \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_K}\frac{\partial \kappa}{\partial x_i}\right) - \nu\frac{\partial^2 \kappa}{\partial x_i \partial x_i} \tag{14}$$

$$g = \frac{\partial \varepsilon}{\partial t} + \overline{u_\iota}\frac{\partial \varepsilon}{\partial x_i} + C_{\varepsilon 1}\frac{\varepsilon}{\kappa}\tau_{ij}\frac{\partial \overline{u_\iota}}{\partial x_j} - \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_{\varepsilon 2}\frac{\varepsilon^2}{\kappa} - \nu\frac{\partial^2 \varepsilon}{\partial x_i \partial x_i} \tag{15}$$

We did not need to specify the geometry or the boundary and initial conditions in the loss as other PINNs do. The parameters are calibrated not only by minimizing the squared residuals over specified collocation points, the two transport equations are also embedded to constrain possible neural network solutions. The model parameters can be calibrated according to:

$$C_{\varepsilon 1}, C_{\varepsilon 2}, C_\mu, \sigma_\kappa, \sigma_\varepsilon = \operatorname{argmin} L \tag{16}$$

Minimizing the loss function is usually performed using backpropagation in neural network models. In backpropagation, the gradients of an objective function with respect to the weights and biases of a deep neural network are calculated by starting off from the network output and propagating back towards the input layer using the chain rule. With the customized loss, the neural network is optimized under partial differential equation constraints. While it seems that the loss function (embedded with transport equations) is too sophisticated to quickly get the gradient, the truth is that the differential operations in the transport equations are easily adapted and implemented in the deep learning platform, as the backpropagation in *Tensorflow* and the derivatives computation are implemented in automatic differentiation. The differential operations in the transport equations can be easily embedded in the computational graph by taking advantages of the chain rule in automatic differentiation. The loss function is fully then differentiable yet enforced with PDE constraints.

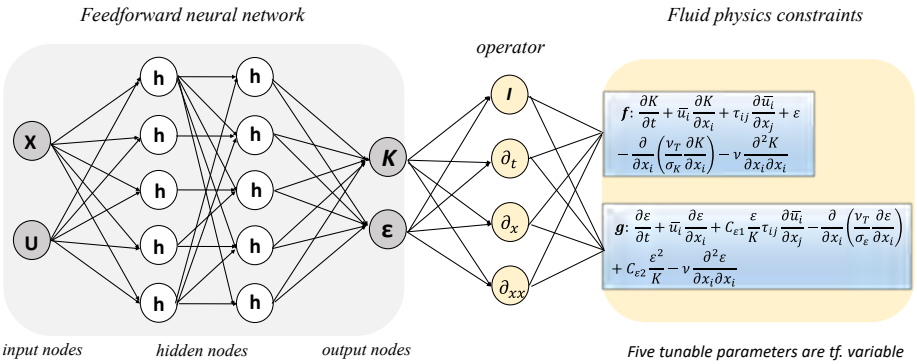

Figure 1. The architecture of physics informed neural network for turbulence modeling application.

Automatic differentiation in general, and the back-propagation algorithm is currently the dominant approach for training deep models by taking their derivatives with respect to the parameters (e.g., weights and biases) of the models. Here, we use the exact same automatic differentiation techniques, employed by the deep learning community, to physics-inform neural networks by taking their derivatives with respect to their input coordinates (i.e., space and time) where the physics is described by partial differential equations. It has been empirically observed that this structured approach introduces a regularization mechanism that allows us to use relatively simple feed-forward neural network architectures and train them with small amounts of data.

4. Case Study: Channel flow with a lower curved wall

The high-fidelity DNS dataset is composed of DNS of converging-diverging turbulent channel flows at two Reynolds numbers (Re_τ=395 and Re_τ=617).[23,24] The dataset is about turbulent boundary layers (TBL) with strong adverse pressure gradient. It is critical to understand flow which undergoes separation and subsequent turbulent reattachment in the TBL to correctly predict the efficiency of many aerodynamic devices. Such turbulent flows have been regarded as being among the most challenging flow dynamics to predict using turbulence models.[25] It is thus of great interests to study whether turbulence models leveraging high-fidelity data can have a more satisfying performance. The dataset includes 438 and 930 3D velocity and pressure fields for the two Reynolds respectively.[26] All the terms involved in the balance of each Reynolds stress component are provided. These DNS have been designed to test and validate turbulence model as flat channel flow data are used for inflow condition. The RANS calculations are performed using the Ansys Fluent v.14.0 commercial CFD package. The steady-state, two-dimensional, incompressible pressure-based solver, SIMPLE method, is used with the default settings of the Fluent package. [27]

Figure 2 shows a comparison of the contour plots of the turbulent kinetic energy. The RANS models are found to predict an incorrect evolution in regions of adverse pressure gradient. It seems to be related to the fact that these models do not correctly describe the evolution of the turbulent kinetic energy close to the walls in adverse pressure gradient regions. It is then urgent to optimize the RANS parameters in the hope that the discrepancy will, at least to some extent, be attenuated.

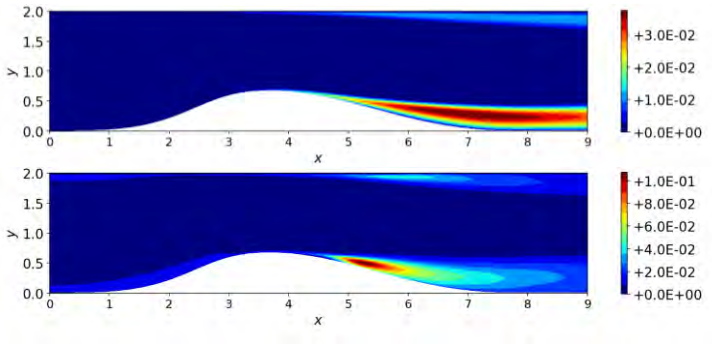

Figure 2. A comparison of the contour plots of the time averaged turbulent kinetic energy (top) RANS (bottom) DNS when Re_τ=395.

To improve the RANS performance, we calibrate the five tunable parameters by the PINNs with the DNS data. The velocities, velocity gradients, pressure, along with other terms involved in the turbulent kinetic energy budgets are time and spanwise-averaged. The pre-processed DNS data are feed to the PINNs to optimize the five parameters by minimizing the customized loss function. The optimized parameters from the PINNs are:

$$C_{\varepsilon 1} = 1.302,\ C_{\varepsilon 2} = 1.862,\ C_{\mu} = 0.09,\ \sigma_{\kappa} = 0.75,\ \sigma_{\varepsilon} = 0.273$$

Figure 3&4 shows the comparison of the time averaged turbulent kinetic energy (κ) and dissipation rate (ε) from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS. It is clear there is a big discrepancy between DNS with RANS, especially near the downstream wall region, where the adverse pressure gradient is most severe. While it is not very intuitive to tell from the contour plots, nevertheless, it is not hard to see that RANS with PINNs inferred parameters (bottom ones) are more closely agree to the DNS data than the RANS with default parameters (middle ones).

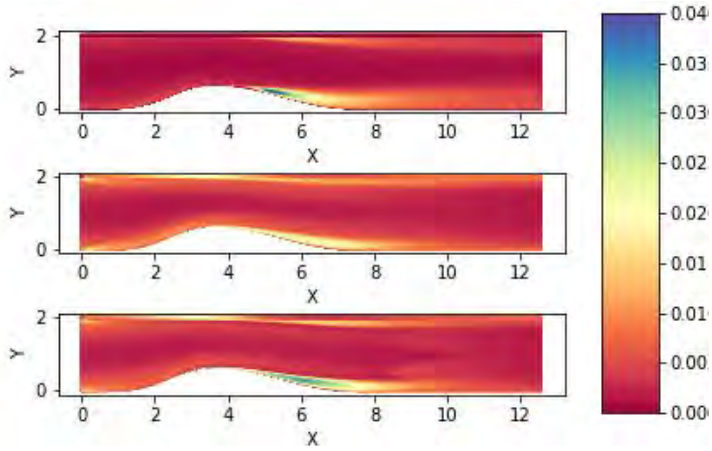

Figure 3. A prior: Comparison of the time averaged turbulent kinetic energy from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

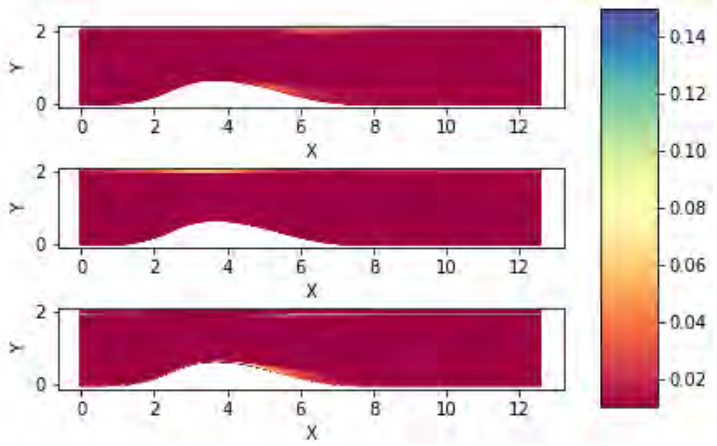

Figure 4. A prior: Comparison of the time averaged dissipation rate from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

To have a more straightforward understanding of the performance comparison, we plot the mean profiles of velocity and turbulence kinetic energy at three locations. These plots can allow us to more closely examine the difference of three simulations, especially near the wall region. Figure 5-7 plot the mean profile of TKE, x- and y- velocity along the y axis at three location when x = 5.7306, x = 6.1399, and x = 6.5493. It shows that RANS with PINN inferred parameters gives better results near the wall region.

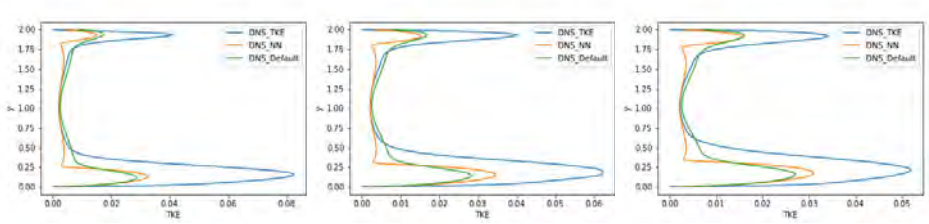

Figure 5. Plots of the turbulent kinetic energy along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).

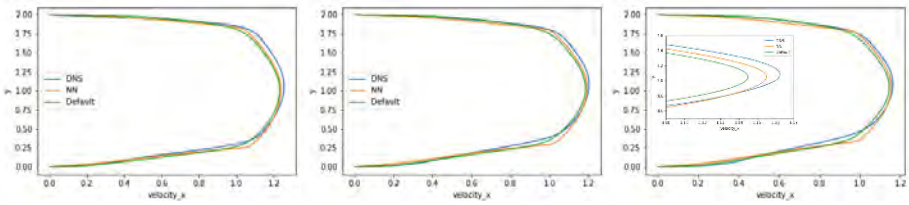

Figure 6. Plots of the x-velocity along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).

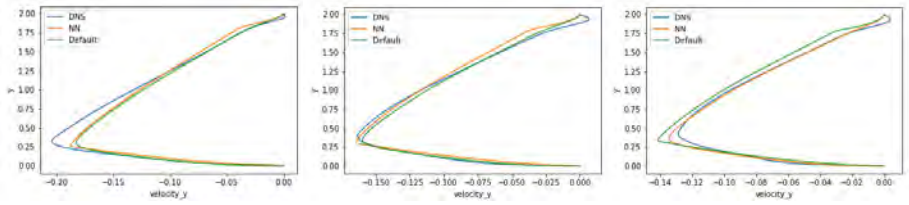

Figure 7. Plots of the y-velocity along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).

We also have the error contour plots showing the difference of two RANS models with DNS results, as showing in Figure 8. Apparently, the error between DNS-NN is relatively smaller than the error between DNS-Default. To quantify the performance improvement, we measure the mean absolute error (MAE) of the velocity magnitude between DNS and RANS. The MAE $= \frac{1}{N}\sum|V^{DNS} - V^{RANS}|$ shows that the error of velocity magnitude for all the collocation points can be reduced by 22% (from 0.069 to 0.054).

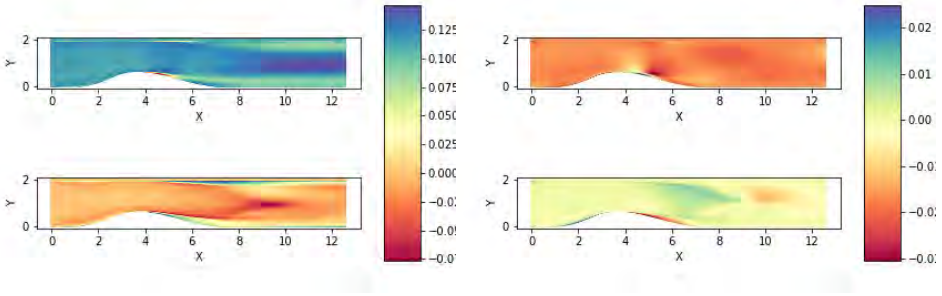

Figure 8. A posteriori: Comparison of the error contour plots between RANS and DNS (left is the x-velocity, right is the y-velocity). (top) error between DNS-Default; (bottom) error between DNS-PINNs, apparently the error in the bottom is relatively much smaller than the error in the top.

It is recognizable that while there is some improvement for the RANS with inferred parameters from PINNs, the error between RANS and DNS is still noteworthy. This inaccuracy is due to RANS's inherent simplifications rather than the inappropriate use of RANS constants which usually estimated by fitting to experimental results of simple flows. If a more accurate result is required, switch to a more sophisticated model like LES is a more feasible option.

5. Conclusion

While high fidelity fluid simulation on high performance computing cluster receive great achievements recently[28,29], Reynolds-averaged Navier–Stokes (RANS) simulation remains the workhorse computational fluid dynamics (CFD) method for industrial enterprises. In this paper we demonstrate an alternative method to calibrate the parameters in the RANS turbulence model with high-fidelity DNS data. We leverage high-resolution DNS data to train a deep neural network to learn the mapping between the low-resolution flow and its high-resolution counterpart. We used a physics informed neural network (PINN) that is embedded with the turbulent transport equations, physical loss functions are proposed to explicitly impose information of the transport equations to deep learning networks. This approach is an inverse problem by treating the five parameters in turbulence model as random variables, with the turbulent kinetic energy and dissipation rate as known quantities from DNS simulation. The objective is to optimize the five parameters in turbulence closures using the PINN leveraging limited data available from costly high-fidelity DNS data. We validated this method on two test cases of flow over bump. The recommended values were found to be $C_{\varepsilon 1} = 1.302$, $C_{\varepsilon 2} = 1.862$, $C_{\mu} = 0.09$, $\sigma_{\kappa} = 0.75$, $\sigma_{\varepsilon} = 0.273$, the mean absolute error of the velocity profile between RANS and DNS decreased by 22% when used the neural network inferred parameters.

The PINNs for turbulence modeling is an example of approaches that "bake in" the physics to address the technical challenges in application of artificial intelligence in scientific discovery. This study gives an example of optimize parameters for κ-ε turbulence model, it can also be similarly applied to other models, new classes of numerical solvers for partial differential equations, as well as new data-driven approaches for model inversion and systems identification.

ACKNOWLEDGMENTS

This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign.

Appendix

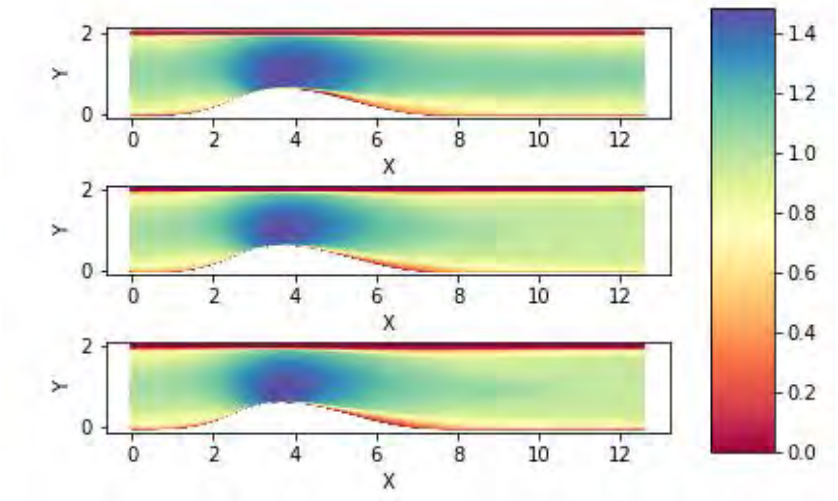

Figure 9. Comparison of the time averaged x-velocity from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

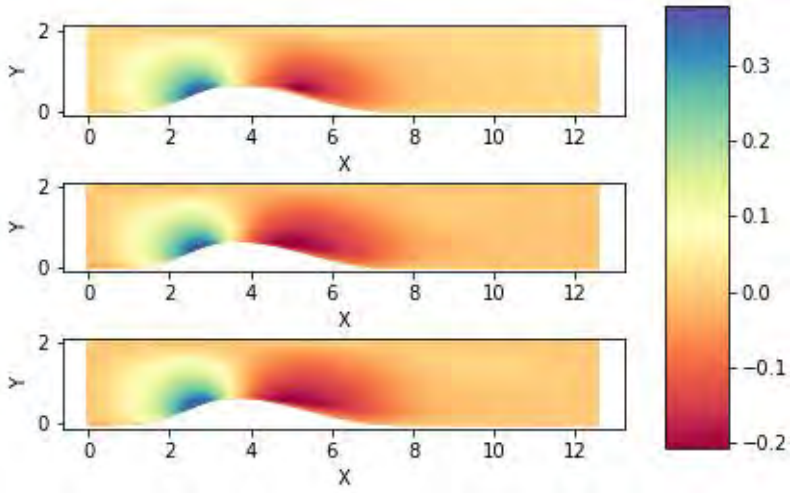

Figure 10 Comparison of the time averaged y-velocity from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

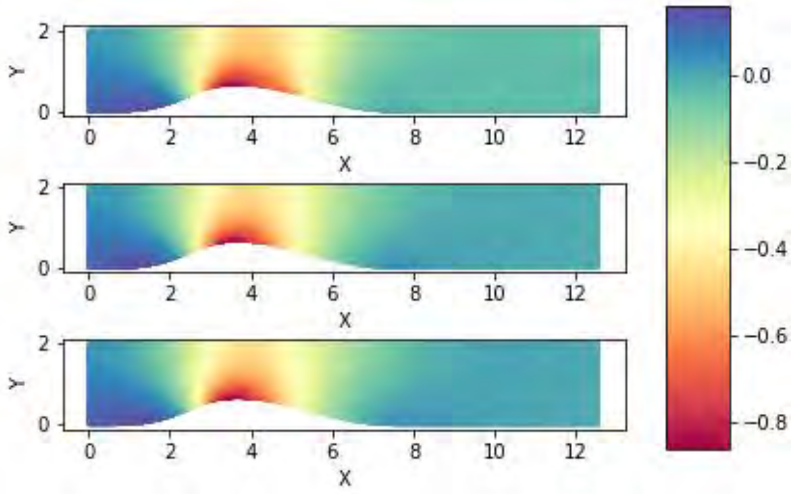

Figure 11 Comparison of the time averaged pressure from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

Reference

1. Ray J, Dechant L, Lefantzi S, Ling J, Arunajatesan S. Robust bayesian calibration of a κ-ε model for compressible jet-in-crossflow simulations. *AIAA J*. 2018;56(12):4893-4909.

2. Thies AT, Tam CK. Computation of turbulent axisymmetric and nonaxisymmetric jet flows using the k-epsilon model. *AIAA J*. 1996;34(2):309-316.

3. Yakhot V, Orszag SA. Renormalization group analysis of turbulence. I. basic theory. *J Sci Comput*. 1986;1(1):3-51.

4. Durbin PA. Separated flow computations with the k-epsilon-v-squared model. *AIAA J*. 1995;33(4):659-664.

5. Shih T, Liou WW, Shabbir A, Yang Z, Zhu J. A new κ-ε eddy viscosity model for high reynolds number turbulent flows. *Comput Fluids*. 1995;24(3):227-238.

6. Shirzadi M, Mirzaei PA, Naghashzadegan M. Improvement of k-epsilon turbulence model for CFD simulation of atmospheric boundary layer around a

high-rise building using stochastic optimization and monte carlo sampling technique. *J Wind Eng Ind Aerodyn*. 2017;171:366-379.

7. Dow E, Wang Q. Quantification of structural uncertainties in the k-w turbulence model. . 2011:1762.

8. Kato H, Obayashi S. Statistical approach for determining parameters of a turbulence model. . 2012:2452-2457.

9. Launder BE, Sharma B. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in heat and mass transfer*. 1974;1(2):131-137.

10. Hanjalić K, Launder BE. A reynolds stress model of turbulence and its application to thin shear flows. *J Fluid Mech*. 1972;52(4):609-638.

11. Platteeuw P, Loeven G, Bijl H. Uncertainty quantification applied to the k-epsilon model of turbulence using the probabilistic collocation method. . 2008:2150.

12. Jones W, Launder BE. The prediction of laminarization with a two-equation model of turbulence. *Int J Heat Mass Transfer*. 1972;15(2):301-314.

13. Chien K. Predictions of channel and boundary-layer flows with a low-reynolds-number turbulence model. *AIAA J*. 1982;20(1):33-38.

14. Zhang J, Fu S. An efficient approach for quantifying parameter uncertainty in the SST turbulence model. *Comput Fluids*. 2019;181:173-187.

15. Schaefer J, Hosder S, West T, Rumsey C, Carlson J, Kleb W. Uncertainty quantification of turbulence model closure coefficients for transonic wall-bounded flows. *AIAA J*. 2017;55(1):195-213.

16. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*. 2019;378:686-707.

17. Raissi M, Karniadakis GE. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*. 2018;357:125-141.

18. Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*. 2019.

19. Mao Z, Jagtap AD, Karniadakis GE. Physics-informed neural networks for high-speed flows. *Comput Methods Appl Mech Eng*. 2020;360:112789.

20. Nabian MA, Meidani H. A deep neural network surrogate for high-dimensional random partial differential equations. *arXiv preprint arXiv:1806.02957*. 2018.

21. Nabian MA, Meidani H. Physics-driven regularization of deep neural networks for enhanced engineering design and analysis. *Journal of Computing and Information Science in Engineering*. 2020;20(1).

22. Sirignano J, Spiliopoulos K. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*. 2018;375:1339-1364.

23. Marquillie M, Laval J, Dolganov R. Direct numerical simulation of a separated channel flow with a smooth profile. *Journal of Turbulence*. 2008(9):N1.

24. Marquillie M, Ehrenstein U, Laval J. Instability of streaks in wall turbulence with adverse pressure gradient. *J Fluid Mech*. 2011;681:205-240.

25. Wilcox DC. *Turbulence modeling for CFD.* Vol 2. DCW industries La Canada, CA; 1998.

26. Benzi R, Biferale L, Bonaccorso F, et al. TurBase: A software platform for research in experimental and numerical fluid dynamics. . 2017:51-57.

27. Jesus A, Azevedo JL, Laval J. Large eddy simulations and RANS computations of adverse pressure gradient flows. . 2013:267.

28. R.Borrell, J.C.Cajas D.Mira, A.Taha, S.Koric, M.Vázquez, G.Houzeaux. Parallel mesh partitioning based on space filling curves. *Computers & Fluids*. 2018:264-272.

29. Vazquez, M., Houzeaux, G., Koric, S., Artigues, A., Aguado-Sierra, J., Ars, R.,Mira, D., Calmet, H., Cucchietti, F., Owen, H. et al. (2016). Alya: Multiphysics engineering simulation toward exascale. *Journal of computational science*, 14, 15-27.