SAUDI INTERNATIONAL CONFERENCE
ON NUCLEAR POWER ENGINEERING

Proceedings of SCOPE
13-15 Nov. 2023 – KFUPM
Paper 23217

Machine Learning Implementation on Reynold Average Navier-Stoke Equations

Adeel Ahmad[1], Faizan Ahmed[2], Junaid Anjum[1], Afaque Shams[3, 4]

[1] Department of Mathematics, COMSATS University Islamabad (CUI), Pakistan
[2] *Formal Methods and Tools Groups, University of Twente,The Netherlands*
[3] Mechanical Engineering Department, King Fahd University of Petroleum & Minerals (KFUPM), Saudi Arabia
[4] Interdisciplinary Research Center for Renewable Energy and Power Systems (IRC-REPS), KFUPM, Saudi Arabia
Adeel Ahmad: adeelahmed@comsats.edu.pk, Faizan Ahmed: faizan-ahmed@utwente.nl,
Junaid Anjum: junaid.anjum@comsats.edu.pk, Afaque Shams: afaque,shams@kfupm.edu.sa,

Abstract – *The Reynolds-Averaged Navier-Stokes (RANS) equations are widely used in computational fluid dynamics (CFD) simulations for predicting turbulent flows. The turbulent stresses in RANS equations are computed using additional turbulence models such as the $k-\varepsilon$ model. From implementation point of view, $k-\varepsilon$ model requires values of involved parameters that are usually approximated using calibrated relations. In recent years, machine learning has shown great potential in improving the accuracy and efficiency of turbulent models by enabling the development of data-driven closures for turbulence models. In this article, we use machine learning to compute $k-\varepsilon$ simulation parameters instead of using empirical relations. Our results show that machine learning accurately computes the solution which is very close to actual DNS data. In comparison to conventional turbulence models, machine learning exhibits low numerical error.*

Keywords: *Reynolds-Averaged Navier-Stokes equations*, $k-\varepsilon$ model, machine learning

I. Introduction

The Navier-Stokes equations are widely acknowledged as one of the most intricate sets of equations in mathematical physics, given their nonlinear and multi-dimensional nature. The mathematical and computational complexities associated with Navier-Stokes equations inspired many theoretical modifications, one of which is the Reynolds-averaged Navier-Stokes (RANS) equations. By separating the steady-state solution from the time-varying fluctuations in the system through Reynolds decomposition, the RANS equations incorporate the effects of turbulence in various flow regimes. This approach enables the RANS equations to account for the average behavior of the fluid flow and its turbulent fluctuations, providing a powerful tool for predicting the behavior of turbulent flows in practical applications. The RANS equations use a solution that is split into a time-independent mean flow velocity and time-varying fluctuations about the mean:

$$\mathbf{u}(x,t)=\mathbf{u}(x)+\mathbf{u}'(x,t)$$

Utilizing the time averaging operation and above decomposition, one gets the following nonlinear equations:

$$u_i\frac{\partial u_j}{\partial x_i}=-\frac{1}{\rho}\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\upsilon\frac{\partial u_i}{\partial x_j}-\left\langle u_i'u_j'\right\rangle\right) \qquad (1)$$

known as Reynolds-averaged Navier-Stokes (RANS) equations for incompressible flows. The last term on right hand side of Eq. (1) is known as Reynolds stress. In RANS, the turbulent fluctuations are averaged over time, leading to the appearance of additional terms in the equations that cannot be solved directly. These terms require the specification of a turbulence model or closure model to describe the turbulent stresses and fluxes. These models require calibration against experimental or numerical data, and their accuracy may vary depending on the flow conditions and geometry. As a result, the choice of turbulence model and its parameters can have a significant impact on the accuracy of the RANS simulation results. The closure problem remains an active area of research in computational fluid dynamics, with ongoing efforts to develop more accurate and reliable turbulence models. Boussinesq [1] proposed a relation among the turbulence stresses and the mean flow to close the system of equations by introducing a new proportionality constant υ_t , the turbulence eddy viscosity. This model is known as eddy viscosity models given as:

$$\left\langle u_i' u_j' \right\rangle = \frac{2}{3} k \delta_{ij} - \upsilon_t S_{ij} \qquad (2)$$

where the mean strain rate tensor is given by:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \qquad (3)$$

where

$$\upsilon_t = C_\mu f_\mu \frac{k^2}{\varepsilon} \qquad (4)$$

k is the turbulent kinetic energy, ε is the dissipation and δ_{ij} is the Kronecker delta.

The k -epsilon ($k-\varepsilon$) model is one of the most common turbulence models with two equations. In collaboration with the RANS equations, the $k-\varepsilon$ model determines the turbulence kinetic energy k and its dissipation rate ε . In the $k-\varepsilon$ model, the turbulence is assumed to be isotropic, homogeneous, and stationary, and the transport equations for k and ε are closed by assuming a linear relationship between the turbulent viscosity and the turbulent kinetic energy. The model includes additional empirical constants and coefficients that are determined by means of experimental or numerical data.

In the $k-\varepsilon$ model, the reference velocity of turbulence is represented by $k^{1/2}$ determined from the turbulent kinetic energy k , and the characteristic length scale is given by $L_e = k^{3/2} / \varepsilon$ which is the typical length scale of energy-containing eddies. Using the concept of eddy viscosity, the governing equations of the $k-\varepsilon$ model for wall turbulence may be written as:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right) - \left\langle u_i' u_j' \right\rangle \frac{\partial u_i}{\partial x_j} - \varepsilon \qquad (5)$$

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right) - C_{\varepsilon 1} f_1 \frac{\varepsilon}{k} \left\langle u_i' u_j' \right\rangle \frac{\partial u_i}{\partial x_j} - C_{\varepsilon 2} f_2 \frac{\varepsilon^2}{k} \qquad (6)$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + u_j \frac{\partial}{\partial x_j}$. The values of constants ($C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_ε, σ_k), function (f_1, f_2) and turbulent viscosity (υ_t) in the above model are well debated in the literature and are determined through calibration against experimental or numerical data.

Jones and Launder [2, 3] extend the original $k-\varepsilon$ model to the low Reynolds number form which allows calculations right to the wall but included extra terms in the transport equations to improve predictions in the wall region, or for computational expediency. The formations proposed by of Launder and Sharma [4], also known as the standard $k-\varepsilon$ model, is one of the commonly used turbulent model. This formulation is a modification of the k-epsilon model that aims to improve its performance in simulating wall-bounded turbulent flows. It includes additional transport equations for the dissipation rate of the turbulent kinetic energy and the dissipation rate of the specific dissipation rate, which is a measure of the dissipation rate normalized by the turbulent kinetic energy. The model also includes additional empirical constants and coefficients that are determined through calibration against experimental or numerical data. The Launder and Sharma formulation is particularly effective in simulating wall-bounded turbulent flows, such as turbulent boundary layers and fully developed pipe flows. It is known to provide more accurate predictions of the flow characteristics near the wall, where the turbulence is strongly influenced by the presence of the wall. However, it is also computationally more expensive than the standard $k-\varepsilon$ model, as it requires the solution of additional transport equations. Chein [5] studied presented a general finite difference formulation of the incompressible Navier-Stokes equations in terms of the vorticity and the stream function for turbulent internal flows. Turbulent models such as algebraic eddy viscosity models, and the low Reynolds number two-equation k-epsilon models were systematically studied. Lam and Bremhorst [6] described the development of a new form of the high Reynolds number $k-\varepsilon$ model which in contrast to previous forms of the high Reynolds number $k-\varepsilon$ model, does not required the use of wall function formulas and the introduction of additional terms into the transport equations. In 1987, Nagano and Hishida [7] proposed an enhanced version of the k-epsilon model, which demonstrated exceptional accuracy in predicting various types of turbulent shear flows near walls, including pipe flows, flat-plate boundary layers, diffuser flows, and relaminarizing flows. Although the k-epsilon model has shown to be effective in predicting ordinary turbulent flows, including those with heat transfer, as demonstrated by Nagano and Kim [8], it does not fully account for the wall-limiting behavior of velocity fluctuations as identified by Chapman and Kuhn [9]. As a result, when near-wall turbulence needs to be accurately predicted in numerical analyses, such as in the heat transfer analysis of high Prandtl number fluids, errors may arise in the predictions, as pointed out by Myong et al. [10]. In 1990, Nagano and Tagawa [11] presents an enhanced version of [8] that considers the limiting behavior of turbulence as well as the impact of an adverse pressure gradient on shear layers. To see the literature related to variations and improvements in $k-\varepsilon$ model read the articles [12]-[17] and references given therein. Calibrated functions and constants of some of the important models are given in Table 1.

In summary, the RANS equations possessing reduced computational cost [18] has a closure problem originating from the introduction of new Reynolds stress terms when averaging the Navier-Stokes equations. In response, many turbulence models have been developed using a physics-driven modeling process over the years (in addition to previously discussed literature see also [19]-[20]). This process involves forming hypotheses based on physical intuition, constructing differential or algebraic mathematical models, parameterizing the models, and calibrating the coefficients. These physics-driven models have the advantage of capturing the primary turbulence transport characteristics and are useful for a broad range of engineering applications. However, the turbulence model remains a significant source of uncertainty in RANS simulations, as noted by many researchers ([21]-[22]), due to the divergence of real engineering applications from the ideal conditions used to construct these models.

II. RANS and Machine Learning

In the field of engineering, simulating fluid flow is an essential task for understanding and optimizing many systems, ranging from aircraft design to energy production. While high-fidelity simulations such as DNS (Direct Numerical Simulation) and LES (Large Eddy Simulation) can accurately capture the complex and chaotic nature of turbulent flows, their computational cost can be prohibitive for many practical applications. As a result, RANS (Reynolds-Averaged Navier-Stokes) simulations, which are computationally less expensive, are often preferred. However, it is important to note that the latest physics knowledge and data obtained through high-fidelity simulations provide a solid foundation for advanced turbulent model development. With an increasing amount of high-fidelity simulation data from DNS and LES, we can obtain detailed information about both averages and turbulent flow fields close to real physical conditions. This rich and precise database of flow phenomena can be leveraged to develop more accurate and reliable RANS turbulent models. By incorporating this data, RANS simulations can achieve a level of accuracy and realism that was previously unattainable. This can lead to significant improvements in engineering design and optimization, as well as a deeper understanding of the underlying physics of turbulent flows. Moreover, as computational resources continue to advance, it is likely that high-fidelity simulations will become more accessible, further improving the accuracy and applicability of RANS models.

Traditional numerical techniques, such as finite difference and finite element methods, have long been employed to solve differential equations. However, in recent years, the application of machine learning (ML) for solution approximation has gained significant attraction [23-24]. Neural networks, particularly deep architectures, possess the capability to approximate any continuous function, given adequate depth and breadth. This universal approximation property makes them a compelling choice for representing intricate solutions of differential equations without the constraints of predefined meshes or grids. By recasting the PDE as an optimization challenge, where the loss function embodies the deviation from the differential equation, neural networks can be trained to identify an approximate solution that closely adheres to the PDE and its boundary conditions [25]. Such a data-driven methodology facilitates adaptive resolution, concentrating computational resources where the solution displays rapid variations or complex features [26]. As the intersection of deep learning and PDEs continues to evolve, neural network-based approaches are set to redefine our capabilities in understanding and solving intricate PDE systems.

Machine learning (ML) techniques have been proposed as a potential solution to the RANS closure problem, as they could learn complex relationships between input and output data. ML can play a significant role in the solution of RANS equations in several ways, such as:

1. Training data generation: One of the primary challenges in solving RANS equations is the generation of high-quality training data. ML and AI techniques can be used to generate training data for RANS simulations, allowing for more efficient and accurate solutions. According to the study conducted by Zhang et al. [27], the use of generative adversarial networks (GANs) can aid in the efficient and accurate generation of RANS training data.

2. Surrogate modelling: ML and AI can be used to develop surrogate models that approximate the solution of RANS equations. These models can be used to simulate complex fluid flows with high accuracy and reduced computational cost. The study by Xiao et al. [28] demonstrated the efficacy of using deep neural networks to develop surrogate models for RANS simulations.

3. Optimization: ML and AI can be used to optimize the parameters of the RANS simulation. This can help to improve the accuracy and efficiency of the simulation and reduce computational cost. The study conducted by Bhushan et al. [29] showed that the use of genetic algorithms can help optimize the parameters of the RANS simulation, leading to improved accuracy and reduced computational cost.

4. Uncertainty quantification: ML and AI techniques can be used to quantify the uncertainty in the RANS solution. This can help to identify the sources of error and improve the accuracy of the simulation as demonstrated in the study by Ray et al. [30].

Considering the preceding discussion, it becomes evident that an extensive body of literature exists concerning the calibration of functions and constants employed in various RANS models. Furthermore, considering the remarkable capacity of machine learning to approximate solutions to differential equations, in this article, we propose to enhance the efficacy of RANS models by using neural network in place of calibrated functions. Keeping to its promise, machine learning indeed provides results that are very close to DNS data and in many situations, perform better than conventional turbulence models.

III. The Architecture of Neural Network

To estimate the RANS simulation parameters, we train an artificial neural network for a fixed value of Reynolds using DNS data to train the network. This deep feedforward network comprises a series of fully connected layers, punctuated by activation functions and dropout layers to enhance its generalization capabilities. The considered neural network employs 50 neurons within the hidden layers. Notably, the hyperbolic tangent (Tanh) activation function is employed within the hidden layers. The Tanh function, which maps its input to a range between -1 and 1, introduces a smooth gradient, making it a suitable choice for the solution approximation tasks of differential equation [23]. Such a characteristic can be instrumental in mitigating challenges related to vanishing or exploding gradients, which are often encountered in deep network training. The architecture concludes with a LeakyReLU activation function, providing a safeguard against inactive neurons and ensuring gradient flow during the training phase. Strategically positioned dropout layers further augment the network by introducing a regularization mechanism, which combats overfitting by randomly nullifying a fraction of input units during the training process. We have used a dropout rate of 20%. The architecture and internal working are shown in Figure 1.

III. Results and Discussion

The problem of fully developed plane turbulent channel flow of a viscous fluid is considered in this article. The data from direct numerical simulations of the problem (URL:http://torroja.dmt.upm.es/channels/data/) is used to train the artificial neural network, as described above, for approximating the simulation parameter, f_μ, which is usually computed using some pre-defined analytical relation. The numerical results of the $k-\varepsilon$ model for the problem is calculated using the calibrated functions and constants given in Nagano Tagawa model [11] and Chien model [5].

To approximate f_μ , it is important to note that the term most directly influenced by f_μ is ν_t. Consequently, we opted to employ the mean squared error between the

RK approximation and the DNS solution as our error function. Specifically, we utilized the following error function:

$$MSE_{\nu_t} = \frac{1}{n}\sum\left(\nu_{t_i} - \nu_{DNS_i}\right)^2$$

It's noteworthy that the chosen neural network architecture attained the desired accuracy right in its initial iteration. We have therefore not presented any graphical results showing error decay and convergence of the network training. However, a detailed comparison of the results obtained through machine learning with those obtained using Nagano Tagawa model [11] and the Chien model [5] is presented in Figures 2-5.

In Figure 2(a), we show results in terms of computed velocities using DNS (solid black curve), machine learning results (red dashed lines with filled triangles) and simulation results corresponding to Chien model [5] (dashed blue line with filled squares). It is noted that the results at the end points i.e., around y=0 and y=1 are similar for DNS, machine learning and Chien model, however, in the interior region, where velocities increase sharply, machine learning results are in better agreement with the DNS whereas error in Chien model is relatively large. Similar results are shown in Figure 2(b) where we compare DNS results with machine learning results and Nagano Tagawa model [11] (blue line with squares). It is seen that Tagawa model tend to overestimate velocities, whereas machine learning results lie very close to the DNS results, exhibiting a negligible error. Hence, it is shown that the velocities computed using machine learning are more accurate compared to the results obtained using Chien model [5] and Nagano Tagawa model [11].

In Figure 3, we discuss results in terms of turbulence kinetic energy k. In the presented results, solid black curve represents the DNS data whereas results in dashed red (with triangles) are due to machine learning. Also plotted in Figure 3(a) are the results computed using the Chien model, shown in dashed blue line with squares. It is seen that the values of k computed using machine learning are much closer to the DNS data whereas the results of Chien model show considerably large error in the computed values. Figure 3(b) shows simulation results, given in dashed blue line with squares, corresponding to Nagano Tagawa model. Even in this case, the machine learning results are better as these lie close to DNS data. Even though the results obtained using the Chien model and Nagano Tagawa model, both tend to show the same qualitative patterns as DNS, however, the quantitative difference between the computed values of k is very large. In contrast, machine learning results are in excellent agreement with the DNS data both in qualitative and quantitative sense.

We compare the simulation results, in terms of dissipation rate of turbulence kinetic energy ε in Figure 3. Following the same color notation, solid black curve represents DNS data whereas dashed red (with filled triangles) and dashed blue (with filled squares) shows results due to machine learning and Chien model, respectively. In Figure 3(b), the results in blue dashed with solid squares represent Nagano Tagawa model. In this case, the results from machine learning, Chien model and Nagano Tagawa model, all exhibit similar performance. The numerical error between computed results and DNS is small except in the early phase (around y=0) where a sharp increase is seen which all the models fail to capture accurately.

In Figure 5, we compare results in terms of turbulent viscosity, υ_t, computed using DNS data (solid black curve), machine learning results (dashed red with triangles) and Chien model (dashed blue with squares). The results from Nagano Tagawa model are shown in Figure 5(b) using dashed blue line with squares. It is seen that none of the method seems to accurately capture the variation of υ_t with y. Nagano Tagawa model seem to be working relatively better but still exhibits large errors compared to actual DNS data. For our machine learning results, the underperformance of the network is due to the fact the numeric values of turbulent viscosity υ_t is very small i.e., 10^{-3}. We are currently looking into this scaling issue of the data and hope that we will find a work around to encounter this. Nonetheless, the results in terms of u, k and ε have shown great performance of machine learning compared to conventional Chien model and Nagano Tagawa model.

A tabulated comparison of the DNS data, machine learning results, Chien model and Nagano Tagawa mode is presented in Table 2. In the table, we show numerical error between the DNS data and the simulation results of machine learning and considered models. The L_2-norm error, for computed velocities u, k, ε and turbulent viscosity υ_t is given. The numerical error for velocity u, k and ε is smallest for machine learning results. However, machine learning results in terms of turbulent viscosity υ_t, shows large error for machine learning results which is due to the data scaling issue as discussed in the preceding paragraph.

Table 1: Calibrated functions and constants of some of the important k-ε models

Model	Functions	Constants
Launder Sharma Model [4]	$C_\mu = 0.09\exp\left(-\frac{3.4}{(1+R_t/50)^2}\right)$ $C_2 = 1.92\left(1-0.3\exp(-R_t^2)\right)$	$\sigma_k = 1.0,$ $\sigma_\varepsilon = 1.3,$ $C_1 = 1.44$
Lam Bremhorst model [6]	$f_\mu = \left(1-\mathrm{e}^{-A}\mu^{R_k}\right)^2\left(1+\frac{A_t}{R_t}\right);$ $f_2 = 1-e^{-R_t{}^2}$ $f_1 = 1+\left(\frac{A_{c1}}{f_\mu}\right)^n$ (At and Aμ are determined by trial and error)	$C_\mu = 0.09,$ $C_1 = 1.44,$ $C_2 = 1.92,$ $\sigma_k = 1.0,$ $\sigma_\varepsilon = 1.3$
Nagano Tagawa model [11]	$f_\mu = \left(1-\exp\left(-\frac{y^+}{A_\mu}\right)\right)^2$ $\left(1+\frac{B_\mu}{R_t^{3/4}}\right);$ $f_2 = \left(1-\exp\left(-\frac{y^+}{B_u}\right)\right)^2$ $\left(1-0.3\exp\left(\frac{R_t}{A_u}\right)^2\right);$	$C_\mu = 0.09,$ $C_{\varepsilon 1} = 1.45,$ $C_{\varepsilon 2} = 1.9,$ $f_1 = 1.0,$ $\sigma_k = 1.4,$ $\sigma_\varepsilon = 1.3$ $A_u = 6.5,$ $B_u = 6$ $A_\mu = 26,$ $B_\mu = 4.1$
Chien model [5]	$f_\mu = 1-\exp\left(-0.0115y^+\right);$ $f_2 = 1-0.22\exp\left(-\frac{R_t}{6}\right)^2;$	$C_\mu = 0.09,$ $C_{\varepsilon 1} = 1.35,$ $C_{\varepsilon 2} = 1.8,$ $f_1 = 1.0,$ $\sigma_k = 1.0,$ $\sigma_\varepsilon = 1.3$

Table 2: *Error of the results of velocity, turbulent kinetic energy (k), dissipation rate obtained from neural network, Chien mode and Nagano Tagawa model with the DNS (solid black curve).*

Models	$\lVert u-u_{DNS}\rVert_2$	$\lVert k-k_{DNS}\rVert_2$	$\lVert \varepsilon-\varepsilon_{DNS}\rVert_2$	$\lVert \upsilon_T-\upsilon_{T\ DNS}\rVert_2$
ML	0.0189	0.0287	0.4211	1.7554
Chien model [5]	0.1823	0.1347	1.4914	0.0035
Nagano Tagawa model [11]	0.0642	0.0571	0.9718	0.0005

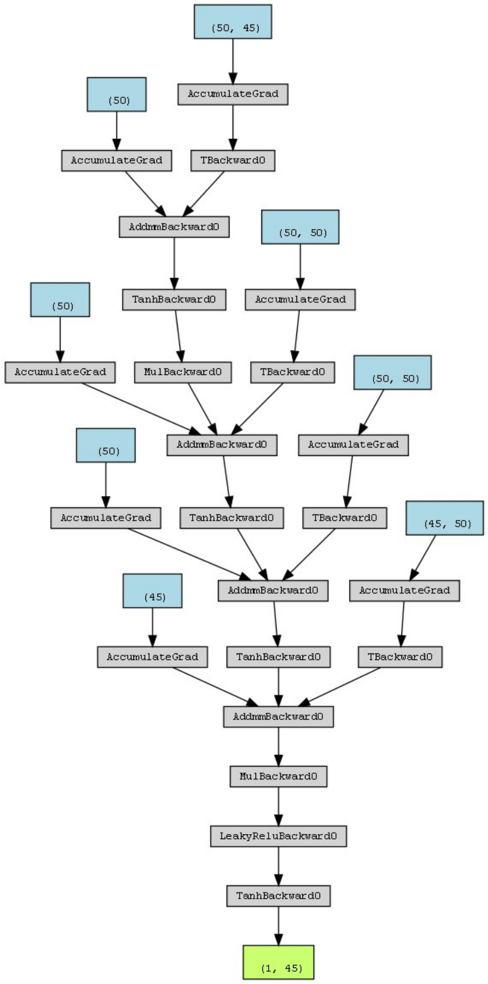

Fig. 1. Structure of the neural network used to replace the calibrated value of f_μ.

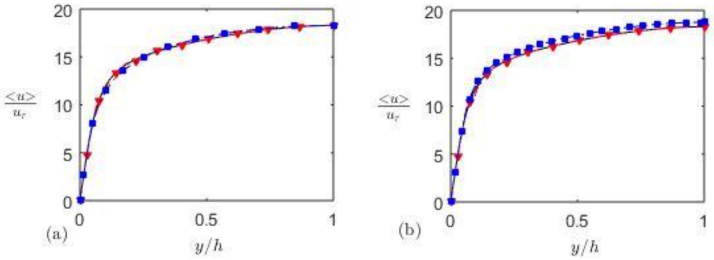

Fig. 2. Comparison of the results of velocity profile obtained from neural network (red dashed curve with filled triangles), Chien model and Nagano Tagawa model (dashed blue line with filled squares in (a) and (b) respectively) with the DNS (solid black curve).

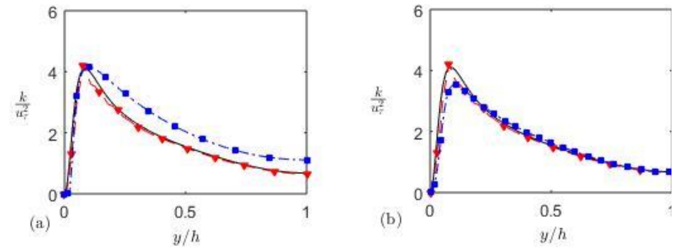

Fig. 3. Comparison of the results of turbulent kinetic energy (k) obtained from neural network (red dashed curve with filled triangles), Chien model and Nagano Tagawa model (dashed blue line with filled squares in (a) and (b) respectively) with the DNS (solid black curve).

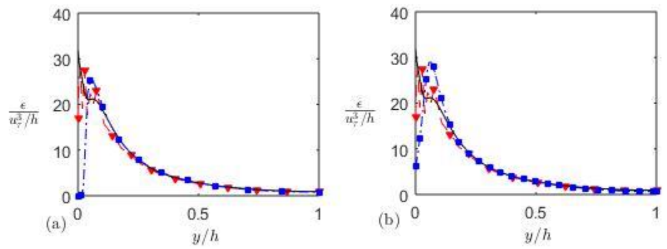

Fig. 4. Comparison of the results of dissipation rate of kinetic energy (ε) obtained from neural network (red dashed curve with filled triangles), Chien model and Nagano Tagawa model (dashed blue line with filled squares in (a) and (b) respectively) with the DNS (solid black curve).

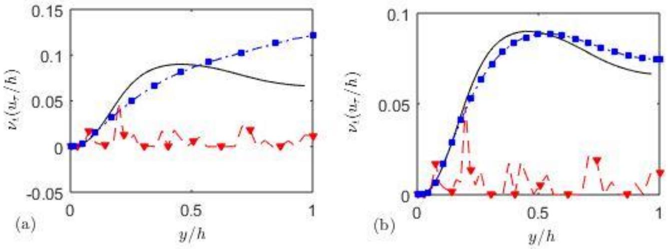

Fig. 5. Comparison of the results of turbulent viscosity (υ_t) obtained from neural network (red dashed curve with filled triangles), Chien model and Nagano Tagawa model (dashed blue line with filled squares in (a) and (b) respectively) with the DNS (solid black curve).

IV. Conclusions

The key objective of this paper was to introduce an alternative approach to $k-\varepsilon$ simulations incorporating machine learning to approximate simulation parameters such as f_μ (and others) in comparison to conventional approaches where these are computed as calibrated functions.

For this purpose, we develop a neural network to approximate the values of f_μ and iteratively improve it by minimizing the loss function which compares the DNS results and $k-\varepsilon$ results. The results show that the machine learning approach tends to show much better performance compared to conventional turbulence models. We compare results in terms of flow velocities *u*, *k*, ε and turbulent viscosity υ_t. The machine learning results show excellent agreement with the DNS data. For the case of *u*, *k*, and ε, machine learning results tend to be better than the conventional turbulence models. However, for the turbulent viscosity υ_t, the errors in the machine learning results and DNS data is relatively large which we believe is because the numeric values of are very small. It should be noted that for υ_t, the performance of turbulence models is also poor.

It is also worth mentioning that the great performance of the neural network can be attributed to the availability of high precision DNS data. Also, the training is limited to a fixed value of the Reynold number. The future work will be aimed at training neural networks that can work with varying values of Reynolds number.

References

1. J. Boussinesq, "Thōrie analytique de la chaleur mise en harmonie avec la thermodynamique et avec la thōrie mc̄anique de la lumi_re: Refroidissement et c̄hauffement par rayonnement, conductibilit¯des tiges, lames et masses cristallines, courants de convection, thōrie mc̄anique de la lumi_re," Gauthier-Villars, 1903.
2. W. P. Jones and B. E. Launder, "The prediction of laminarization with a 2-equation model of turbulence," International Journal of Heat and Mass Transfer, vol. 15, pp. 301-314, 1972.
3. W. P. Jones and B. E. Launder, "The calculation of low-Reynolds number phenomena with a two-equation model of turbulence," International Journal of Heat and Mass Transfer, vol. 16, pp. 1119-1130, 1973.
4. B. E. Launder and B. I. Sharma, "Application of the energy dissipation model of turbulence to the calculation of flow near a spinning disc," Letters in Heat and Mass Transfer, vol. 1, no. 2, pp. 131-138, 1974.
5. J. C. Chien, "Numerical Analysis of Turbulent Separated Subsonic Diffuser Flow," Symposium on Turbulent Shear Flows, University Park, Pennsylvania, Vol. 1, pp. 18.19-18.25, 1977.
6. C. K. G. Lam and K. Bremhorst, "A modified form of the k-ε model for predicting wall turbulence," Journal of Fluids Engineering, vol. 103, no. 3, pp. 456-460, 1981.

7. Y. Nagano and M. Hishida, "Improved form of the k-e model for wall turbulent shear flows," ASME Journal of Fluids Engineering, vol. 109, pp. 156-160, 1987.
8. Y. Nagano and C. Kim, "A two-equation model for heat transport in wall turbulent shear flows," ASME Journal of Heat Transfer, vol. 110, pp. 583-589, 1988.
9. D. R. Chapman and G. D. Kuhn, "The limiting behaviour of turbulence near the wall," Journal of Fluid Mechanics, vol. 170, pp. 265-292, 1986.
10. H. K. Myong, N. Kasagi, and M. Hirata, "Prediction of turbulent heat transfer by the modified k-e model taking into account the near-wall behavior of turbulence," Proceedings of the 24th National Heat Transfer Symposium of Japan, pp. 10-12, 1987.
11. Y. Nagano and M. Tagawa, "An improved k-e model for boundary layer flows," ASME Journal of Fluids Engineering, vol. 112, pp. 33-39, 1990.
12. V. C. Patel, W. Rodi, and G. G. Scheuerer, "Turbulence models for near-wall and low Reynolds number flow: a review," AIAA Journal, vol. 23, pp. 1308-1319, 1985.
13. Y. Nagano, M. Kondoh, and M. Shimada, "Multiple time scale turbulence model for wall and homogeneous flows based on direct numerical simulations," International Journal of Heat and Fluid Flow, vol. 18, pp. 346-359, 1997.
14. M. M. Rahman, P. Rautaheimo, and T. Siikonen, "Modifications for an explicit algebraic stress model," International Journal for Numerical Methods in Fluids, vol. 35, pp. 221-245, 2001.
15. T. Nagano and M. Tagawa, "An improved k-ε model for boundary layer flows," Journal of Fluids Engineering, vol. 112, pp. 33-39, 1990.
16. K-Y. Chien, "Predictions of channel and boundary layer flows with a low-Reynolds number turbulence model," AIAA Journal, vol. 20, no. 1, pp. 33-38, 1982.
17. M. M. Rahman and T. Siikonen, "Improved low-Reynolds-number k-ε model," AIAA Journal, vol. 38, no. 7, pp. 1298-1300, 2000.
18. J. Slotnick et al., "CFD vision 2030 study: a path to revolutionary computational aerosciences," Technical Report, 2014.
19. F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," AIAA Journal, vol. 32, pp. 1598-1605, 1994.
20. S. Wallin and A. Johansson, "An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows," Journal of Fluid Mechanics, vol. 403, pp. 89-132, 2000.
21. C. J. Roy and F. G. Blottner, "Review and assessment of turbulence models for hypersonic flows," Progress in Aerospace Sciences, vol. 42, pp. 469-530, 2006.
22. Y. Liu, X. Yu, and B. Liu, "Turbulence models assessment for large-scale tip vortices in an axial compressor rotor," Journal of Propulsion and Power, vol. 24, pp. 15–25, 2008.
23. J. Blechschmidt and O.G. Ernst, "Three ways to solve partial differential equations with neural networks—A review," GAMM-Mitteilungen, vol. 44, no. 2, p. e202100006, 2021.
24. S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," Journal of Scientific Computing, vol. 92, no. 3, 2022, p. 88.
25. K. Srivastava, M. Ahlawat, J. Singh, and V. Kumar, "Learning partial differential equations from noisy data using neural networks," Journal of Physics: Conference Series, vol. 1655, no. 1, p. 012075, 2020.
26. H. Son, J.W. Jang, W.J. Han, and H.J. Hwang, "Sobolev training for the neural network solutions of PDEs," 2020.
27. Z. Zhang et al., "Application of deep learning method to Reynolds stress models of channel flow based on reduced-order modeling of DNS data," International Journal of Heat and Fluid Flow, vol. 79, pp. 108441, 2019.
28. H. Xiao, A. Rasheed, and K. Duraisamy, "Surrogate modeling of turbulence models using deep neural networks," Journal of Computational Physics, vol. 371, pp. 443-467, 2018. https://doi.org/10.1016/j.jcp.2018.06.037
29. S. Bhushan, K. Duraisamy, and J.J. Alonso, "Data-driven RANS surrogate models for the simulation-based optimization of airfoil shapes," AIAA Journal, vol. 57, no. 5, pp. 1819-1831, 2019. https://doi.org/10.2514/1.J057059
30. P. Ray, S. Bhattacharya, and K. Duraisamy, "A neural network approach for uncertainty quantification of RANS simulations," Journal of Computational Physics, vol. 406, p. 109178, 2020. https://doi.org/10.1016/j.jcp.2019.109178