

HOW TO IMPROVE A $k - \omega$ TURBULENCE MODEL USING PHYSICS INFORMED NEURAL NETWORK (PINN) AND NEURAL NETWORK (NN) [2, 4]

Lars Davidson, Div. of Fluid Dynamics
Dept. of Mechanical Engineering (ME)
Chalmers University of Technology
Gothenburg, Sweden

14 Int, National Conf. on Fluid Mechanics (NCFM14)
Qingdao 7-11 August 2026
Invited speaker

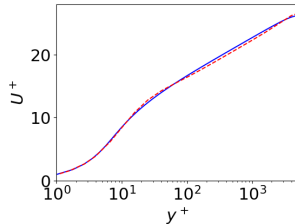
THE $k - \omega$ TURBULENCE MODEL

The Wilcox $k - \omega$ turbulence model reads [9]

$$\begin{aligned}\frac{\partial \bar{v}_i}{\partial x_i} &= 0 \\ \frac{\partial \bar{v}_i}{\partial t} + \frac{\partial \bar{v}_i \bar{v}_j}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \frac{\partial \bar{v}_i}{\partial x_j} \right] \\ \frac{\partial \bar{v}_j k}{\partial x_j} &= P^k + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_{k,NN}} \right) \frac{\partial k}{\partial x_j} \right] - C_{k,NN} C_\mu k \omega \\ \frac{\partial \bar{v}_j \omega}{\partial x_j} &= C_{\omega 1} \frac{\omega}{k} P^k + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right] - C_{\omega 2,NN} \omega^2 \\ P^k &= \nu_t \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) \frac{\partial \bar{v}_i}{\partial x_j}, \quad \nu_t = \frac{k}{\omega}\end{aligned}\tag{1}$$

The standard coefficients are used, i.e. $C_{\omega 1} = 5/9$, $C_{\omega 2} = 3/40$, $\sigma_k = \sigma_\omega = 2$ and $C_\mu = 0.09$.

FULLY-DEVELOPED CHANNEL FLOW, $k - \omega$ MODEL, $Re_\tau = 5\,200$

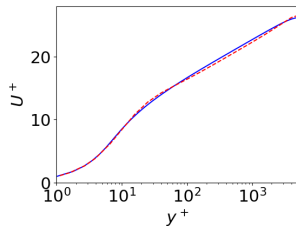


(A) Mean velocity.

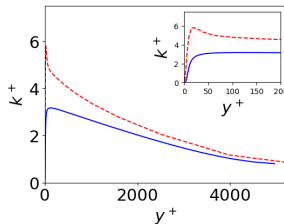
FIGURE: Fully-developed channel flow. Solid lines: $k - \omega$; dashed lines: DNS [6].

- The mean flow, shear stress (and hence the turbulent viscosity, ν_t) agree well

FULLY-DEVELOPED CHANNEL FLOW, $k - \omega$ MODEL, $Re_\tau = 5200$



(A) Mean velocity.



(B) Turb. kinetic energy.

FIGURE: Fully-developed channel flow. Solid lines: $k - \omega$; dashed lines: DNS [6].

- The mean flow, shear stress (and hence the turbulent viscosity, ν_t) agree well
- But not the turbulent, kinetic energy

FULLY-DEVELOPED CHANNEL FLOW, $k - \omega$ MODEL, $Re_\tau = 5200$

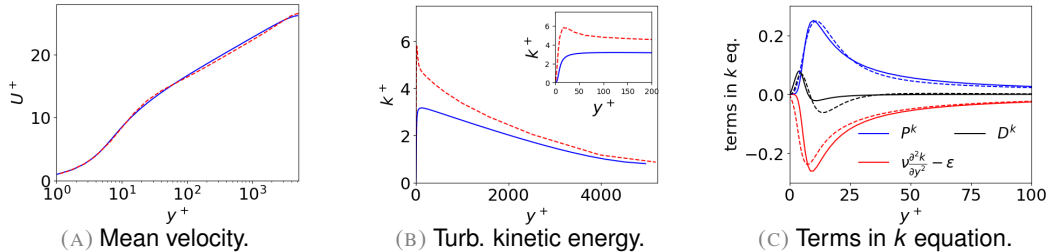


FIGURE: Fully-developed channel flow. Solid lines: $k - \omega$; dashed lines: DNS [6].

- The mean flow, shear stress (and hence the turbulent viscosity, ν_t) agree well
- But not the turbulent, kinetic energy
- It seems to be because the diffusion of k is poorly predicted

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

$$(\nu + \nu_{t,PINN}) \frac{d^2 k_{DNS}}{dy^2} + \frac{dk_{DNS}}{dy} \frac{d\nu_{t,PINN}}{dy} + P_{DNS}^k - \varepsilon_{DNS} = Q \quad (2)$$

where $\nu_{t,PINN}$ is the turbulent viscosity in the k_{DNS} equation and $Q = 0$.

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

$$(\nu + \nu_{t,PINN}) \frac{d^2 k_{DNS}}{dy^2} + \frac{dk_{DNS}}{dy} \frac{d\nu_{t,PINN}}{dy} + P_{DNS}^k - \varepsilon_{DNS} = Q \quad (2)$$

where $\nu_{t,PINN}$ is the turbulent viscosity in the k_{DNS} equation and $Q = 0$.

- $\nu_{t,PINN}$ is the unknown

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

$$(\nu + \nu_{t,PINN}) \frac{d^2 k_{DNS}}{dy^2} + \frac{dk_{DNS}}{dy} \frac{d\nu_{t,PINN}}{dy} + P_{DNS}^k - \varepsilon_{DNS} = Q \quad (2)$$

where $\nu_{t,PINN}$ is the turbulent viscosity in the k_{DNS} equation and $Q = 0$.

- $\nu_{t,PINN}$ is the unknown
- k_{DNS} , P_{DNS}^k and ε_{DNS} are known (taken from DNS),

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

$$(\nu + \nu_{t,PINN}) \frac{d^2 k_{DNS}}{dy^2} + \frac{dk_{DNS}}{dy} \frac{d\nu_{t,PINN}}{dy} + P_{DNS}^k - \varepsilon_{DNS} = Q \quad (2)$$

where $\nu_{t,PINN}$ is the turbulent viscosity in the k_{DNS} equation and $Q = 0$.

- $\nu_{t,PINN}$ is the unknown
- k_{DNS} , P_{DNS}^k and ε_{DNS} are known (taken from DNS),
- First I tried to use the finite volume method, but failed

FIND A NEW $\nu_{t,PINN}$

- Our ordinary differential equation reads in fully-developed channel flow

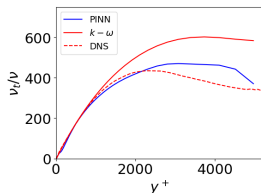
$$\frac{d}{dy} \left((\nu + \nu_{t,PINN}) \frac{dk}{dy} \right) + P^k - \varepsilon = Q$$

$$(\nu + \nu_{t,PINN}) \frac{d^2 k_{DNS}}{dy^2} + \frac{dk_{DNS}}{dy} \frac{d\nu_{t,PINN}}{dy} + P_{DNS}^k - \varepsilon_{DNS} = Q \quad (2)$$

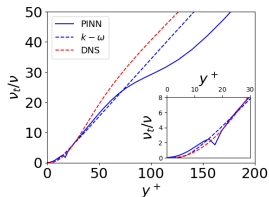
where $\nu_{t,PINN}$ is the turbulent viscosity in the k_{DNS} equation and $Q = 0$.

- $\nu_{t,PINN}$ is the unknown
- k_{DNS} , P_{DNS}^k and ε_{DNS} are known (taken from DNS),
- First I tried to use the finite volume method, but failed
- $\nu_{t,PINN}$ in Eq. 2, will be predicted by PINN while minimizing the error Q^2 .

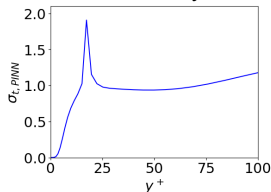
SOLVING EQ. 2 WITH PINN.



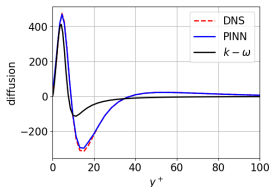
(A) Turbulent viscosity. Full view



(B) Turbulent viscosity. Zoomed view



(C) Prandtl number.



(D) Turbulent diffusion.

FIGURE: Prediction by PINN compared to DNS and the Wilcox $k-\omega$ model. $Re_\tau = 5\,200$.

CFD, $Re_\tau = 5\,200$

The equation below is solved is using **pyCALC-RANS**

$$\frac{d}{dy} \left(\nu + \nu_{t,PINN} \frac{dk}{dy} \right) + P_{DNS}^k - \varepsilon_{DNS} = 0 \quad (3)$$

where $\nu_{t,PINN}$ is known (given by PINN) and P_{DNS}^k and ε_{DNS} are taken from DNS.

CFD, $Re_\tau = 5\,200$

The equation below is solved is using **pyCALC-RANS**

$$\frac{d}{dy} \left(\nu + \nu_{t,PINN} \frac{dk}{dy} \right) + P_{DNS}^k - \varepsilon_{DNS} = 0 \quad (3)$$

where $\nu_{t,PINN}$ is known (given by PINN) and P_{DNS}^k and ε_{DNS} are taken from DNS.

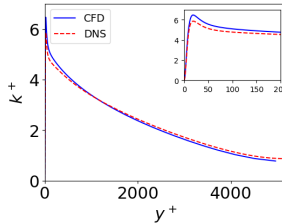


FIGURE: CFD prediction of Eq. 3 using $\sigma_{k,PINN}$.

FIND $C_{k,PINN}$ AND $C_{\omega2,PINN}$

- Recall that the standard $k - \omega$ gives correct $\nu_t = k/\omega = k_{DNS}/\omega_{DNS}$

FIND $C_{k,PINN}$ AND $C_{\omega 2,PINN}$

- Recall that the standard $k - \omega$ gives correct $\nu_t = k/\omega = k_{DNS}/\omega_{DNS}$
- I must predict a correct $\varepsilon = \varepsilon_{DNS}$, i.e.

FIND $C_{k,PINN}$ AND $C_{\omega2,PINN}$

- Recall that the standard $k - \omega$ gives correct $\nu_t = k/\omega = k_{DNS}/\omega_{DNS}$
- I must predict a correct $\varepsilon = \varepsilon_{DNS}$, i.e.

$$P_{DNS}^k + D_{DNS}^k - \underbrace{C_{k,PINN} C_\mu k_{DNS} \omega_{DNS}}_{\varepsilon_{DNS}} = 0 \quad \Rightarrow \quad C_{k,PINN} = \frac{P_{DNS}^k + D_{DNS}^k}{C_\mu k_{DNS} \omega_{DNS}} \quad (4)$$

FIND $C_{k,PINN}$ AND $C_{\omega 2,PINN}$

- Recall that the standard $k - \omega$ gives correct $\nu_t = k/\omega = k_{DNS}/\omega_{DNS}$
- I must predict a correct $\varepsilon = \varepsilon_{DNS}$, i.e.

$$P_{DNS}^k + D_{DNS}^k - \underbrace{C_{k,PINN} C_\mu k_{DNS} \omega_{DNS}}_{\varepsilon_{DNS}} = 0 \quad \Rightarrow \quad C_{k,PINN} = \frac{P_{DNS}^k + D_{DNS}^k}{C_\mu k_{DNS} \omega_{DNS}} \quad (4)$$

- Finally, the ω equation in the new $k - \omega$ model must predict $\omega = \omega_{DNS}$

$$\underbrace{\frac{d}{dy} \left(\frac{\nu_t}{\sigma_\omega} \frac{d\omega_{DNS}}{dy} \right)}_{D^\omega} + C_{\omega 1} \frac{\omega_{DNS}}{k_{DNS}} P_{DNS}^k - C_{\omega 2,PINN} \omega_{DNS}^2 = 0 \Rightarrow \quad (5)$$

$$C_{\omega 2,PINN} = \frac{\frac{d}{dy} \left(\frac{\nu_t}{\sigma_\omega} \frac{d\omega_{DNS}}{dy} \right) + C_{\omega 1} \frac{P_{DNS}^k}{\nu_{t,DNS}}}{\omega_{DNS}^2}$$

PLOT C_k AND $C_{\omega 2}$

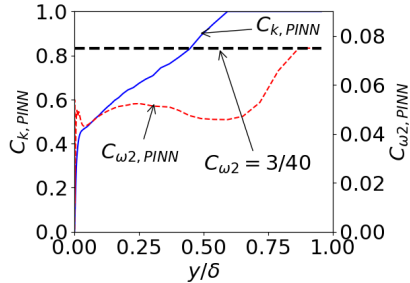
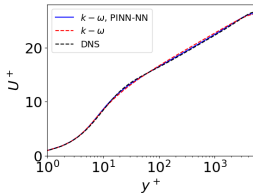


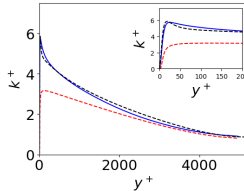
FIGURE: C_k and $C_{\omega 2}$ vs. y/δ .

IMPORT PINN TO PYCALC-RANS

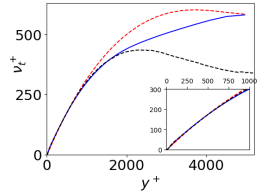
- Solve U , k , ω using $\sigma_{k,PINN}$, $C_{k,PINN}$, $C_{\omega 2,PINN}$



(A) Velocity.



(B) Turbulent kinetic energy.



(C) Turbulent viscosity.

FIGURE: Fully-developed channel flow comparing $k - \omega$ PINN and standard $k - \omega$ model. $\sigma_{k,PINN}$ (Eq.3), $C_{k,PINN}$ (Eq. 4) and $C_{\omega 2,PINN}$ (Eq. 5) are used. $Re_\tau = 5\,200$.

DRAWBACK/PROBLEM

- I have made $\sigma_{k,PINN}$, $C_{k,PINN}$ and $C_{\omega 2,PINN}$ functions of y/δ

DRAWBACK/PROBLEM

- I have made $\sigma_{k,PINN}$, $C_{k,PINN}$ and $C_{\omega 2,PINN}$ functions of y/δ
- Hence, the current formulation of the model is not applicable to re-circulating flow

DRAWBACK/PROBLEM

- I have made $\sigma_{k,PINN}$, $C_{k,PINN}$ and $C_{\omega 2,PINN}$ functions of y/δ
- Hence, the current formulation of the model is not applicable to re-circulating flow
- Using Neural Network (NN), I've tried to make them functions of different input parameters such as P_k/ε , P_k^+ , $\nu_t/(y u_\tau)$, ...

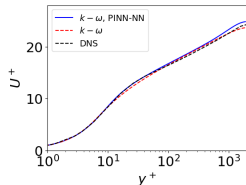
DRAWBACK/PROBLEM

- I have made $\sigma_{k,PINN}$, $C_{k,PINN}$ and $C_{\omega 2,PINN}$ functions of y/δ
- Hence, the current formulation of the model is not applicable to re-circulating flow
- Using Neural Network (NN), I've tried to make them functions of different input parameters such as P_k/ε , P_k^+ , $\nu_t/(y u_\tau)$, ...
- Finally, I found a good combination input parameters: τ_{tot}/u_τ^2 and $\nu_t/(y u_\tau)$

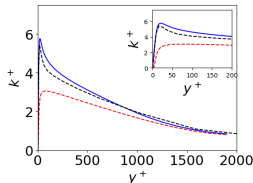
$$\tau_{tot} = 2 (\nu + \nu_t) (\bar{s}_{ij} \bar{s}_{ij})^{1/2}, \quad \bar{s}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right)$$

- During training, I find min/max of τ_{tot} and $\nu_t/(y u_\tau)$. When the NN models are used in the CFD code, I use min/max as limiters on τ_{tot} and $\nu_t/(y u_\tau)$

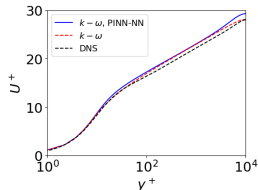
RESULTS. CHANNEL FLOW & FLAT-PLATE BOUNDARY LAYER



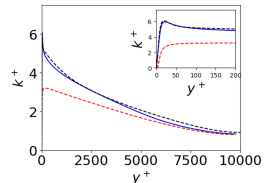
(A) Velocity. $Re_\tau = 2000$



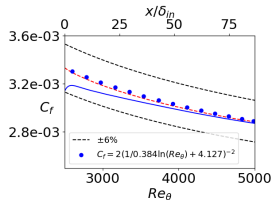
(B) Turb. kinetic energy.
 $Re_\tau = 2000$



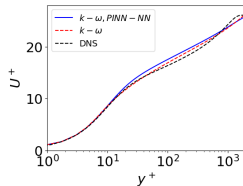
(C) Velocity.
 $Re_\tau = 10000$



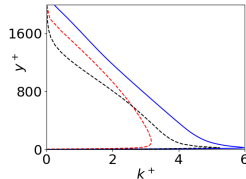
(D) Turb. kinetic energy.
 $Re_\tau = 10000$



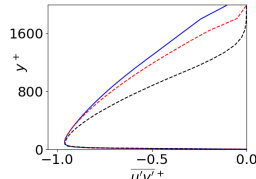
(A) Skin friction.



(B) Velocity.



(C) Turbulent kinetic energy.



(D) Turbulent shear stress.

HILL FLOW.

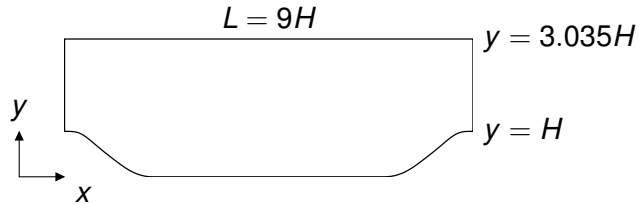


FIGURE: The geometry of the hill.

HILL FLOW. EXCELLENT RESULTS. PURE LUCK!

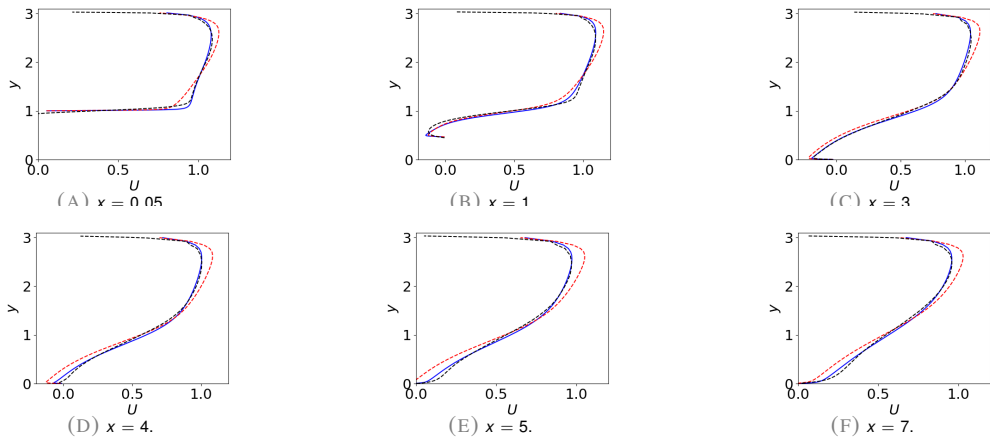
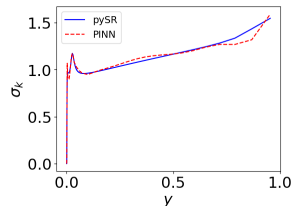


FIGURE: Hill flow. Velocity. — : $k - \omega$ -PINN-NN; - - : standard $k - \omega$; - - : DNS by [5]

NN models predict $\sigma_{k,NN} = 1.42$, $C_{k,NN} = 0.42$ and $C_{\omega 2,NN} = 0.043$ at $y - y_{wall} \gtrsim 0.03$.

SYMBOLIC REGRESSION

- Above I have used three NN models for $\sigma_{k,NN}$, $C_{k,NN}$ and $C_{\omega 2,NN}$
- They can be replaced with symbolic regression (pySR).
- The formula for σ_k is given below
- Advantage of PySR: these formulas can conveniently be used in commercial CFD codes



(A) σ_k predicted with pySR.

$$0.0133x_0 \left(-27.4x_1 + \frac{1}{-1.55 + \frac{1}{x_1(x_0+x_0)(x_1-0.0595)}} + \frac{1}{x_1^2} \right) \\ + x_0 (x_0 (x_0 + 0.385) + (x_1 + 0.385) 0.0575)^{-1}, \quad x_0 = \nu_t / (y u_\tau), x_1 = \tau_{tot} / u_\tau^2$$

ANALYTICAL SOLUTION

- The k equation reads

$$\frac{d}{dy} \left[(\nu_{t,PINN} + \nu) \frac{dk}{dy} \right] + \frac{P^k - \varepsilon}{S^k} = 0 \quad (6)$$

- The heat conduction equation reads

$$\frac{\partial}{\partial x} \left(\alpha \frac{\partial T}{\partial x} \right) = S$$

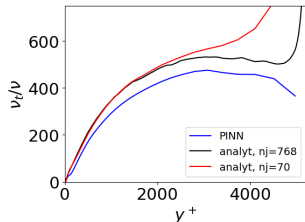
This equation can be solved by simply integrating from 0 to x , i.e.

$$T'(x)\alpha(x) - T'(0)\alpha(0) = \int_0^x S(\xi) d\xi \Rightarrow \alpha(x) = \frac{T'(0)}{T'(x)}\alpha(0) + \frac{1}{T'(x)} \int_0^x S(\xi) d\xi, T'(x) \neq 0$$

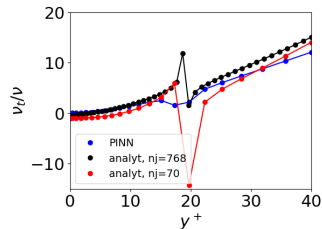
- The analytical solution of Eq 6 reads

$$\nu_{t,PINN}(x) = \frac{1}{dk/dy} \int_0^x S^k(\xi) d\xi \quad (7)$$

ANALYTICAL SOLUTION



(A) Turbulent viscosity. Full view



(B) Turbulent viscosity. Zoomed view

FIGURE: Prediction by PINN (Eq. 2) compared to the analytical solution (Eq. 7). $Re_\tau = 5\,200$.

- For in-depth understanding of autograd and NN, the reader is referred to [this report \[3\]](#) and the references therein.
- **Neural Network** and **PINN** in Python.
 - Good YouTube lectures: "3Blue1Brown: But what is a neural network"; "3Blue1Brown: gradient descent, how neural networks learn"; "3Blue1Brown: backpropagation, intuitively"; "3Blue1Brown: backpropagation, calculus"; "Sebastian Lagre: how to create a neural network".

CONCLUSIONS

- The k equation is turned into an ODE for the turbulent viscosity, $\nu_{t,PINN}$
- Then the turbulent Prandtl number in the k equation is given by $\sigma_{k,PINN} = \nu_t / \nu_{t,PINN}$.
- In order to keep the turbulent viscosity, ν_t , the same as in the standard $k - \omega$ model, two new functions are introduced:
 - one, $C_{k,PINN}$, in front of the dissipation term in the k equation
 - and another, $C_{\omega 2,PINN}$, in front of the destruction term in the ω equation.
- Three NN models, are then created for $\sigma_{k,PINN}$, $C_{k,PINN}$ and $C_{\omega 2,PINN}$ to generalize these results and thus construct a turbulence model.
- **Download paper** Python scripts and CFD codes at:
<https://www.tfd.chalmers.se/lada/Using-Physical-Informed-Neural-Network-PINN-and-NN-improve-a-k-omega-turbulence-model.html>



USING PHYSICAL INFORMED NEURAL NETWORK (PINN) AND NEURAL NETWORK (NN) TO IMPROVE A REYNOLDS STRESS TRANSPORT MODEL

THE REYNOLDS STRESS TRANSPORT EQ. OF $\overline{v'_i v'_j}$

The PDE for the Reynolds Stresses reads (Eq. 9.12 in [1])

$$\begin{aligned}
 \underbrace{\frac{\partial}{\partial x_k} (\bar{v}_k \overline{v'_i v'_j})}_{C_{ij}} &= \underbrace{-\overline{v'_j v'_k} \frac{\partial \bar{v}_i}{\partial x_k} - \overline{v'_i v'_k} \frac{\partial \bar{v}_j}{\partial x_k}}_{P_{ij}} \\
 &+ \underbrace{\nu \frac{\partial^2 \overline{v'_i v'_j}}{\partial x_k \partial x_k} - \frac{\partial}{\partial x_k} \left(\overline{v'_i v'_j v'_k} + \frac{1}{\rho} \delta_{jk} \overline{v'_i p'} + \frac{1}{\rho} \delta_{ik} \overline{v'_j p'} \right)}_{D_{ij}^t} + \underbrace{\frac{1}{\rho} \overline{p' \left(\frac{\partial v'_i}{\partial x_j} + \frac{\partial v'_j}{\partial x_i} \right)}}_{\Phi_{ij}} - \underbrace{2\nu \frac{\partial v'_i}{\partial x_k} \frac{\partial v'_j}{\partial x_k}}_{\varepsilon_{ij}}
 \end{aligned} \tag{8}$$

where

- C_{ij} , P_{ij} , D_{ij}^ν are convection, production and viscous diffusion, respectively (exact)
- D_{ij}^t , Φ_{ij} , ε_{ij} are turb diffusion, pressure strain and dissipation (need to be modeled)

FULLY DEVELOPED CHANNEL FLOW

- In fully-developed channel flow, the PDE for $\overline{v_1'^2}$ reads (see Eq. 8)

$$0 = \underbrace{-2\overline{v_1'v_2'}\frac{d\bar{v}_1}{dx_2}}_{P_{11}} + \underbrace{\nu\frac{d^2\overline{v_1'^2}}{dx_2^2}}_{D_{11}^\nu} - \underbrace{\frac{d}{dx_2}\left(\overline{v_1'v_1'v_2'} + 2\overline{v_2'p'}\right)}_{D_{11}^t} + \underbrace{\frac{2}{\rho}\overline{p'}\frac{\partial\overline{v_1'}}{\partial x_1}}_{\Phi_{11}} - \underbrace{2\nu\frac{\partial\overline{v_1'}}{\partial x_k}\frac{\partial\overline{v_1'}}{\partial x_k}}_{\varepsilon_{11}} \quad (9)$$

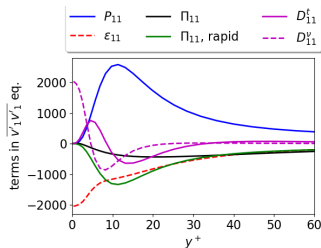


FIGURE: Terms in Eq. 9.

THE REYNOLDS STRESS TRANSPORT MODEL

The Reynolds Stress Transport Model (RSTM) reads (Eq.11.101 in [1]))

$$\begin{aligned}
 \underbrace{\frac{\partial \bar{v}_k \overline{v'_i v'_j}}{\partial x_k}}_{\text{convection}} = & \underbrace{-\overline{v'_i v'_k} \frac{\partial \bar{v}_j}{\partial x_k} - \overline{v'_j v'_k} \frac{\partial \bar{v}_i}{\partial x_k}}_{\text{production}} - \underbrace{c_1 \frac{\varepsilon}{k} \left(\overline{v'_i v'_j} - \frac{2}{3} \delta_{ij} k \right)}_{\text{pressure strain, slow part}} - \underbrace{c_2 \left(P_{ij} - \frac{2}{3} \delta_{ij} P^k \right)}_{\text{pressure strain, rapid part, IP model}} \\
 & + \underbrace{c_{1w} \rho \frac{\varepsilon}{k} \left[\overline{v'_k v'_m} n_k n_m \delta_{ij} - \frac{3}{2} \overline{v'_i v'_k} n_k n_j - \frac{3}{2} \overline{v'_k v'_j} n_k n_i \right]}_{\text{pressure strain, wall, slow part}} \\
 & + \underbrace{c_{2w} \left[\phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2} \phi_{ik,2} n_k n_j - \frac{3}{2} \phi_{jk,2} n_k n_i \right]}_{\text{pressure strain, wall, rapid part, IP model}} f + \underbrace{\nu \frac{\partial^2 \overline{v'_i v'_j}}{\partial x_k \partial x_k}}_{\text{viscous diffusion}} + \underbrace{\frac{\partial}{\partial x_m} \left[\frac{\nu_t}{\sigma_k} \frac{\partial \overline{v'_i v'_j}}{\partial x_m} \right]}_{\text{turbulent diffusion}} - \underbrace{\frac{2}{3} \varepsilon \delta_{ij}}_{\text{dissipation}}
 \end{aligned}
 \tag{10}$$

FULLY-DEVELOPED CHANNEL FLOW

- In fully-developed channel flow, the modelled equation for $\overline{v_1'^2}$ reads (see Eq. 10)

$$\frac{d}{dx_2} \left[(\nu + \nu_t) \frac{d\overline{v_1'^2}}{dx_2} \right] - \underbrace{\overline{v_1' v_2'} \frac{\partial \bar{v}_1}{\partial x_2}}_{P_{11}} - \underbrace{C_1 \frac{\varepsilon}{k} \left(\overline{v_1'^2} - 2k/3 \right)}_{\Phi_{11,1}} - \underbrace{\frac{C_2 + 8}{11} \left(P_{11} - 2P^k/3 \right)}_{\Phi_{11,2}} \quad (11)$$

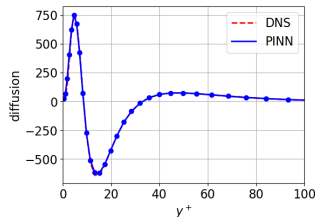
$$+ \underbrace{c_{1w} \frac{\varepsilon}{k} \overline{v_2'^2} f}_{\phi_{11,1w}} + \underbrace{c_{2w} \phi_{22,2} f}_{\phi_{11,2w}} - \frac{2}{3} \varepsilon = 0, \quad f = \frac{k^{3/2}}{2.55 y \varepsilon}$$

where $\phi_{22,2} = c_2 \frac{2}{3} P^k = -c_2 \frac{2}{3} \overline{v_1' v_2'} \frac{\partial \bar{v}_1}{\partial x_2}$.

- In the same way as in [2], we can use PINN to improve the modeling of the turbulent diffusion in the $\overline{v'_i v'_j}$ equations. The equation for $\overline{v_1'^2}$, for example, reads

$$\frac{d}{dx_2} \left[(\nu + \nu_{t,PINN}) \frac{d\overline{v_1'^2}_{DNS}}{dx_2} \right] + P_{11,DNS} + \Phi_{11,DNS} - \varepsilon_{11,DNS} = 0 \quad (12)$$

- Use PINN to predict $\nu_{t,PINN}$ from Eq. 12
- Then, $\nu_{t,PINN}$ is inserted into Eq. 12 and $\overline{v_1'^2}$ is solved.
- The predicted turbulent diffusion is shown to the right

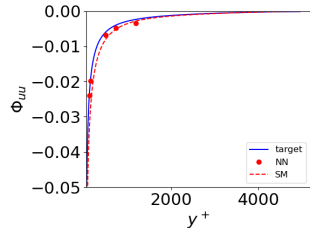
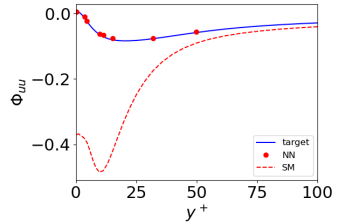


NN MODEL FOR PRESSURE STRAIN IN THE $\overline{v_1'^2}$ EQUATION

- In Eqs. 10 and 11 you find the models for pressure strain in 3D (tensor notation) and 1D, respectively. They are rather complex and lengthy.
- Let's try to make a (simpler) NN model of the pressure strain for $\overline{v_1'^2}$.
- Which input parameters should we choose? Looking at the models for the pressure strain in Eq. 11, we find that they include k , $\overline{v_1'^2}$, P_{11} , P^k and ε .

NN MODEL FOR PRESSURE STRAIN IN THE $\overline{v_1'^2}$ EQUATION

- I made a NN model for the entire pressures strain term, Φ_{11} , in the $\overline{v_1'^2}$ equation
- Input parameters: k , $\overline{v_1'^2}$, P_{11} , P^k
- I compare NN model with standard model (SM) to the right



CONCLUSIONS

- I've given some suggestions on how to improve a Reynolds Stress Transport Model (RSTM) using PINN and NN.

REFERENCES

- [1] L. Davidson. Fluid mechanics, turbulent flow and turbulence modeling. eBook, Division of Fluid Dynamics, Dept. of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, 2021. URL https://www.cfd-sweden.se/lada/postscript_files/solids-and-fluids_turbulent-flow_turbulence-modelling.pdf.
- [2] L. Davidson. Using Physical Informed Neural Network (PINN) and Neural Network (NN) to improve a $k - \omega$ turbulence model: Python CFD code and PINN script. Division of Fluid Dynamics, Dept. of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, 2025. URL <https://www.cfd-sweden.se/lada/Using-Physical-Informed-Neural-Network-PINN-and-NN-improve-a-k-omega-turbulence-model-Python-CFD-code-and-PINN-script.html>.

REFERENCES

- [3] L. Davidson. Understanding autograd and neural network in pytorch. Division of Fluid Dynamics, Dept. of Mechanic Engineering, Chalmers University of Technology, Gothenburg, 2026. URL https://www.cfd-sweden.se/lada/postscript_files/understanding-autograd-and-neural-network-in-pytorch.pdf.
- [4] Lars Davidson. Using physics informed neural network (PINN) and neural network (NN) to improve a k-omega turbulence model. *Journal of Turbulence*, 27(7):187–208, 2026. doi: <https://doi.org/10.1080/14685248.2026.2665148>. URL <https://doi.org/10.1080/14685248.2026.2665148>.
- [5] J. Froehlich, C. Mellen, W. Rodi, L. Temmerman, M. A., and Leschziner. Highly-resolved large eddy simulations of separated flow in a channel with streamwise periodic constrictions. *Journal of Fluid Mechanics*, 526:19–66, 2005.
- [6] M. Lee and R. D. Moser. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *Journal of Fluid Mechanics*, 774:395–415, 2015. doi: [10.1017/jfm.2015.268](https://doi.org/10.1017/jfm.2015.268). URL <https://doi.org/10.1017/jfm.2015.268>.

REFERENCES

- [7] N. N. Mansour, J. Kim, and P. Moin. Reynolds-stress and dissipation-rate budgets in a turbulent channel flow. *Journal of Fluid Mechanics*, 194:15–44, 1988. doi: doi:10.1017/S0022112088002885. URL <https://doi.org/10.1017/S0022112088002885>.
- [8] J.A. Sillero, J. Jimenez, and R.D. Moser. One-point statistics for turbulent wall-bounded flows at Reynolds numbers up to $\delta^+ \simeq 2000$. *Physics of Fluids*, 25(105102), 2014. doi: <https://doi.org/10.1063/1.4823831>. URL <https://doi.org/10.1063/1.4823831>.
- [9] D. C. Wilcox. Reassessment of the scale-determining equation. *AIAA Journal*, 26(11): 1299–1310, 1988.