Using ML to improve RANS Turbulence models

OMID RAZIZADEH

Department of Mechanics and Mathematics
Novosibirsk State University

Contents :

Turbulence :

Almost all of the turbulence theory is based on Navier Stokes equations.

Till today it is still one of the topics in classical physics problems.

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\rho g_x$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right)+\rho g_y$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)+\rho g_z$$

We have three common types of turbulence models :

DNS :

A direct numerical simulation is a simulation in computational fluid dynamics in which the Navier–Stokes equations are numerically solved without any turbulence model.

LES :

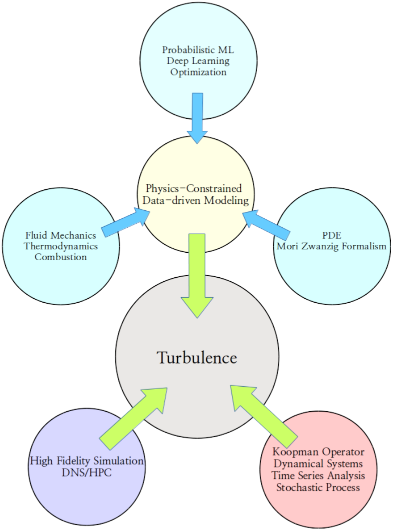
Moderately accurate and computationally costly.

RANS :

We take average usually in time.Computing this model is not expensive but it's not also that exact.

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \nu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \overline{u_i' u_j'}\right]$$

How Turbulence and Data science can be relevant :



Data collection :

Low fidelity data by simulations(OpenFoam)

High fidelity data by experiments that is available like Johns Hopkins Dataset

Low fidelity data(Simulation Data) :

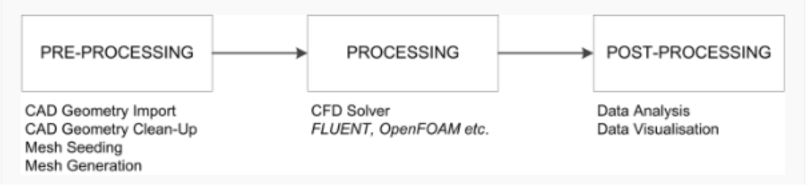
We obtain it by OpenFoam :

It is a C++ toolbox

It is used for continuum mechanics particularly fluid dynamic problems

It is completely free !

Structure of the research :



Steps :

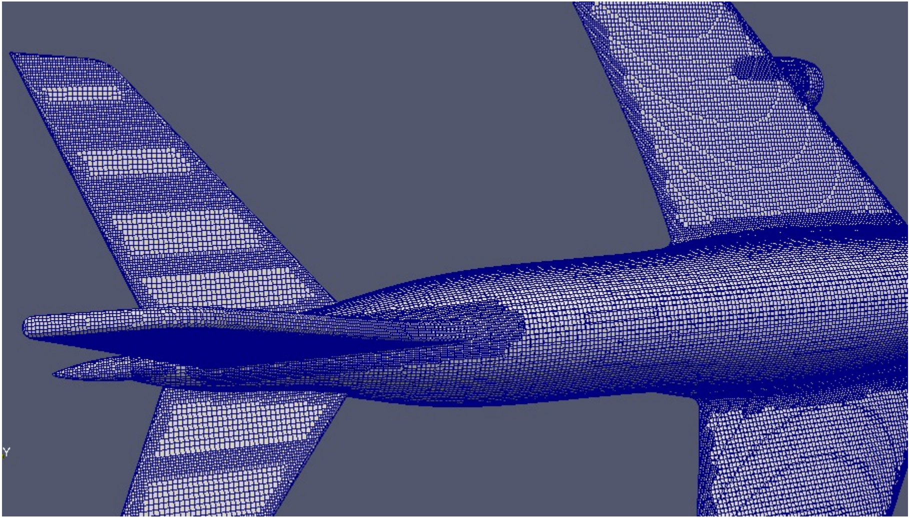
1. Get low fidelity data by simulation.

2. Build ML model and train model with step 1 data to decrease the discrepancy between model.

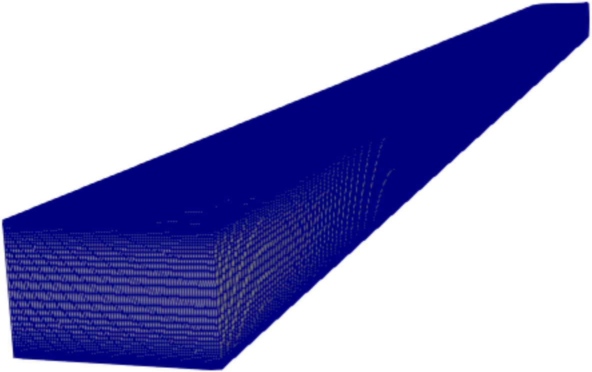
3.Compare the result with DNS data

4.Backpropogate ML+RANS model into our simulation solver.

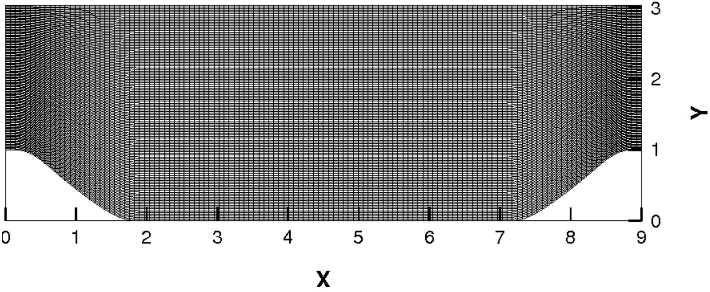
Mesh :



3D-channel flow :



Periodic hills :



Check convergence :

It's easy to do simulation but what about checking it ?It is difficult !

1.Residuals (We can see numbers during run)

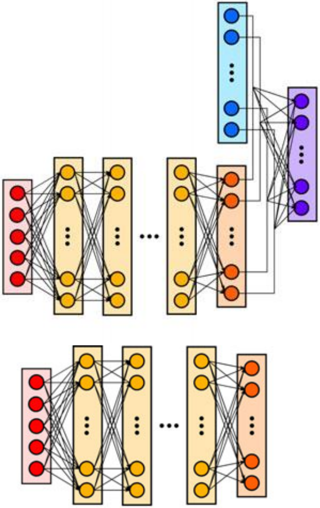
2.Check Mesh (Smaller meshes are better but more expensive)

3.experiment (simulation should be close to experiment)

4.Mesh convergence (mesh refinement should give better result)

Reynolds Averaged Turbulence Modeling using Deep Neural Networks with Embedded Invariance

TBNN and MLP :



Input features :

$T^{(1)} = S \quad T^{(6)} = R^2 S + S R^2 - \frac{2}{3} I \cdot Tr\left(S R^2\right)$

$T^{(2)} = SR - RS \; \mathrm{T}^{(7)} = RSR^2$

$\boldsymbol{T}^{(3)} = \boldsymbol{S}^2 - \frac{1}{3} \boldsymbol{I} \cdot \mathrm{Tr}\left(\boldsymbol{S}^2\right) \quad \boldsymbol{T}^{(8)} = \boldsymbol{S R S}^2 - \boldsymbol{S}^2 \boldsymbol{R S}$

$T^{(4)} = R^2 - \frac{1}{3} I \cdot \mathrm{Tr}\left(R^2\right) \quad T^{(9)} = R^2 S^2 + S^2 R^2 - \frac{2}{3} I \cdot \mathrm{Tr}\left(S^2 R^2\right)$

$T^{(5)} = RS^2 - S^2 R \quad T^{(10)} = RS^2 R^2 - R^2 S^2 R$

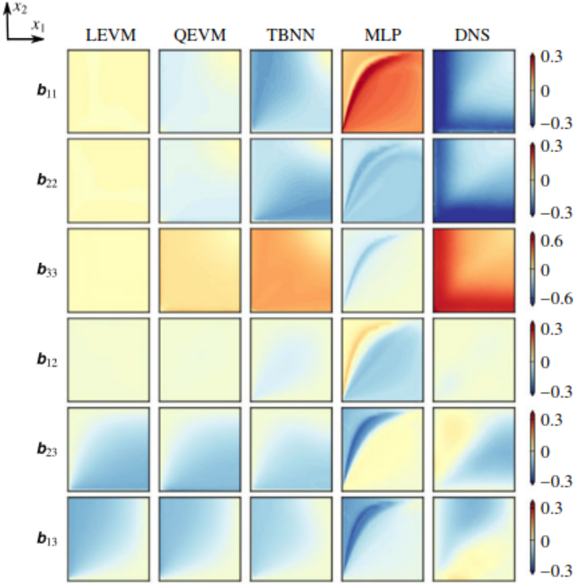
Dataset :

They used Pinelli dataset at different reynolds number for square duct.

QEVM :

$$b_{ij} = -\frac{v_t S_{ij}}{k} + C_1 \frac{v_t}{\tilde{\epsilon}} \left(2S_{ik}S_{kj} - \frac{2}{3} S_{kl}S_{kl}\delta_{ij} \right)$$
$$+ C_2 \frac{v_t}{\tilde{\epsilon}} \left(2R_{ik}S_{kj} + 2R_{jk}S_{ki} \right) + C_3 \frac{v_t}{\tilde{\epsilon}} \left(2R_{ik}R_{jk} - \frac{2}{3} R_{kl}R_{kl}\delta_{ij} \right)$$

Results :



x_2
x_1
LEVM
QEVM
TBNN
MLP
DNS
b_{11}
b_{22}
b_{33}
b_{12}
b_{23}
b_{13}
0.3
0
−0.3
0.6
0
−0.6

Results :

Model	Duct flow	Flow over wavy wall
LEVM	0.25	0.18
QEVM	0.20	0.11
TBNN	0.14	0.08
MLP	0.31	0.09

TABLE 1. Root mean squared error on test cases.

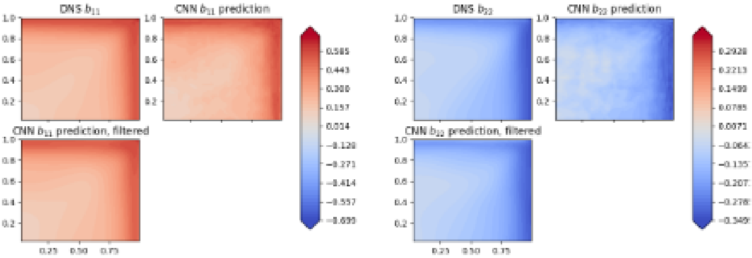
Master thesis :

Motivation :

Till now only simple CNN or the specific TBNN architecture is used in most research papers. I would like to see how it will work on more complicate architectures.

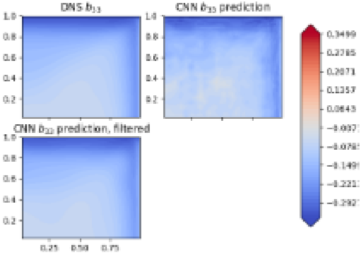
CNN-Decision tree or CNN-SVM. Here SVM and Decision tree will be used as feature extractors.

Results of Kaandorp-Delft University of Technology :

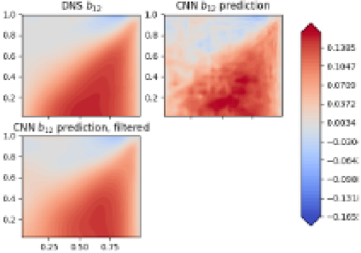


(a) b_{11} **(b)** b_{22}

Results of Kaandorp-Delft University of Technology :

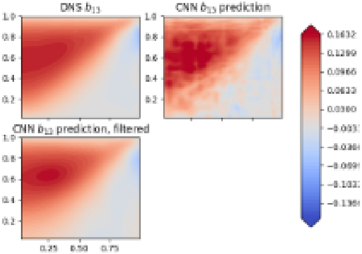


(c) b_{33}

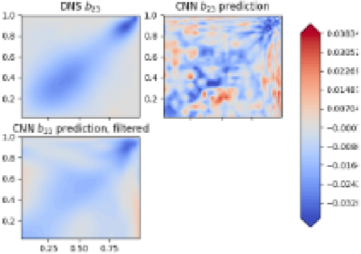


(d) b_{12}

Results of Kaandorp-Delft University of Technology :

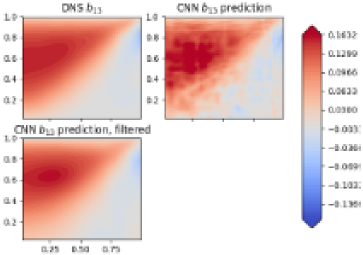


(e) b_{13}

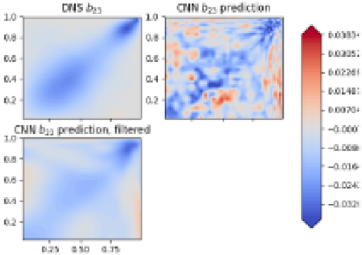


(f) b_{23}

Results of Kaandorp-Delft University of Technology :



(e) b_{13}



(f) b_{23}

Features :

Features were proposed by Wang et al. (Wang, Wu, Xiao, 2017)

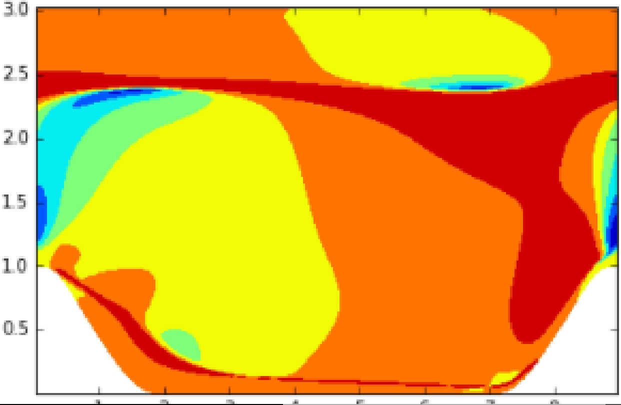
The features are normalized according to :

$q_\beta = \frac{\hat{q}_\beta}{|\hat{q}_\beta|+|q^*_\beta|}$

1.Ratio of excess rotation rate to strain rate

$\hat{q}_\beta = \frac{1}{2}\left(\|\Omega\|^2 + \|S\|^2\right)$

$q^*_\beta = \|S\|^2$

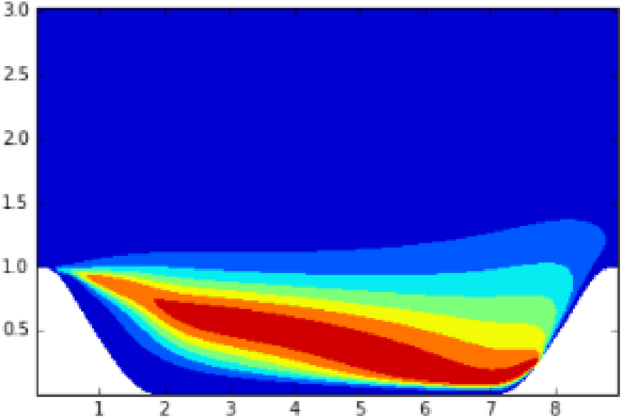


Features :

2.Turbulence intensity

$\hat{q}_\beta = k$

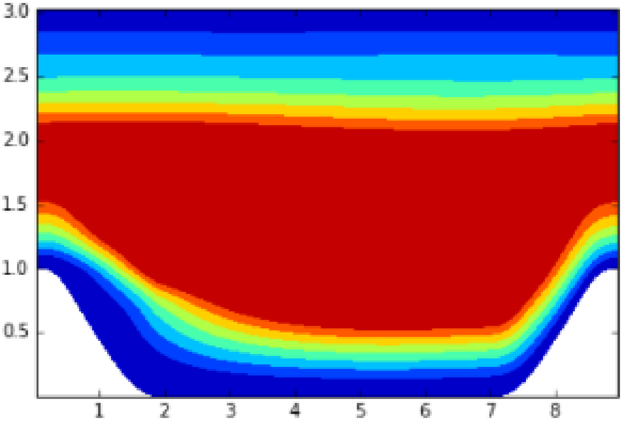
$q^*_\beta = \frac{1}{2} U_i U_i$



Features :

3.Wall-distance based on Reynolds number

$$q_\beta = \min\left(\frac{\sqrt{k}d}{50\nu}, 2\right)$$

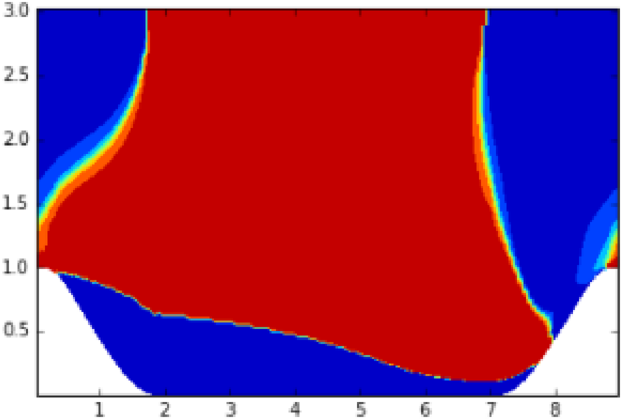


Features :

4.Pressure gradient along streamline

$\hat{q}_\beta = U_i \frac{\partial P}{\partial x_i}$

$q^*_\beta = \sqrt{\frac{\partial P}{\partial x_j} \frac{\partial P}{\partial x_j} U_i U_i}$

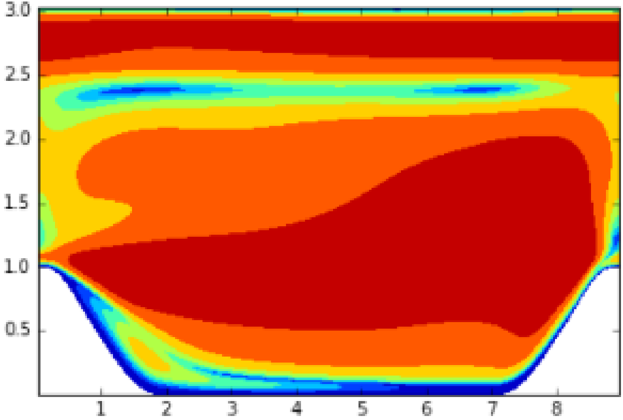


Features :

5.Ratio of turbulent time scale to mean strain time scale

$\hat{q}_\beta = \frac{k}{\epsilon}$

$q^*_\beta = \frac{1}{\|S\|}$

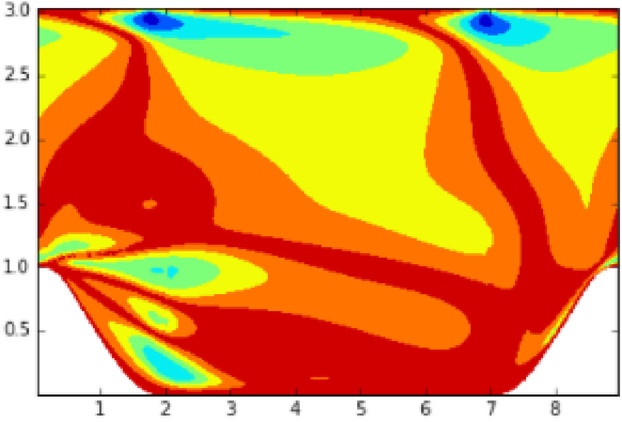


Features :

6.Cratio of pressure normal stresses to shear stresses

$$\hat{q}_\beta = \sqrt{\frac{\partial p}{x_i}\frac{\partial p}{\partial x_i}}$$

$$q^*_\beta = \frac{1}{2}\rho\left(\frac{\partial U_k}{\partial x_k}\right)^2$$

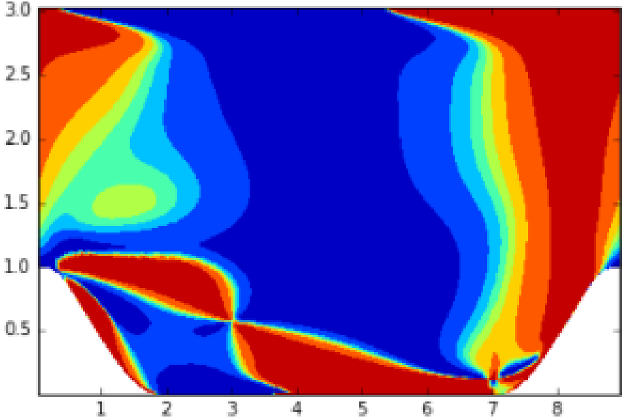


Features :

7.Non-orthogonality between velocity and its gradient

$\hat{q}_\beta = \left| U_i U_j \frac{\partial U_i}{\partial x_j} \right|$

$q^*_\beta = \sqrt{U_l U_l U_i \frac{\partial U_i}{\partial x_j} U_k \frac{\partial U_k}{\partial x_j}}$

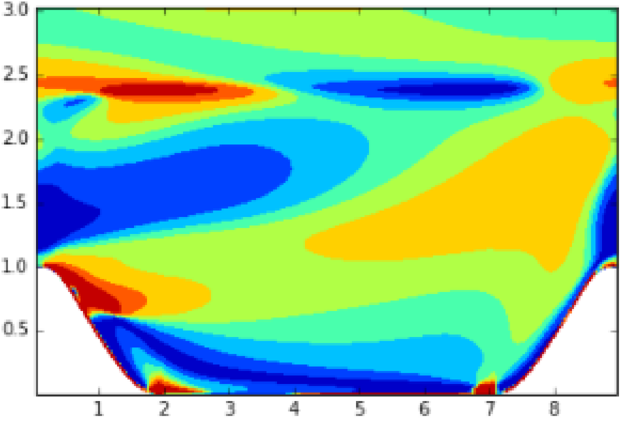


Features :

8.Ratio of convection to production of TKE

$$\hat{q}_\beta = U_i \frac{dk}{dx_i}$$

$$q^*_\beta = \overline{\left| u'_i u'_j S_{jk} \right|}$$

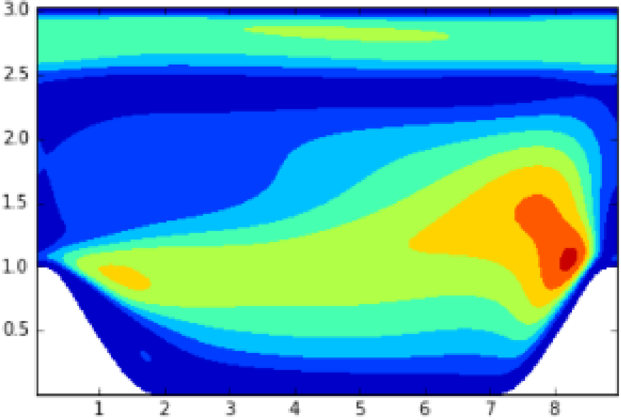


Features :

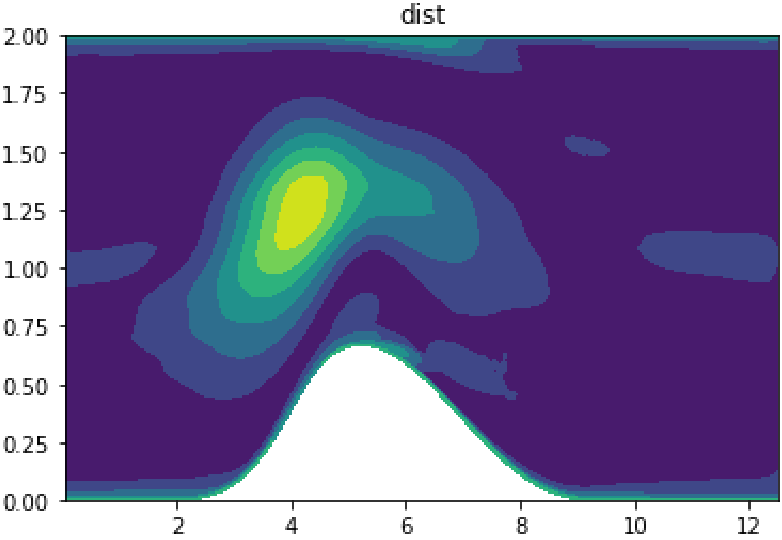
9.Ratio of total to normal Reynolds stresses

$\hat{q}_\beta = \|\overline{u_i' u_j'}\|$

$q^*_\beta = k$



Random forests+ML(Improved) :



Other researcher approaches :

Field inversion and machine learning-University of Michigan-Karthik duraisamy

Gene expression programming-University of Melbourne- Richard Sandberg

Plan :

1.CNN+SVM for three cases(periodic hill,square duct,channel flow) will be implemented till 15th of April. Abstract for this approach already submitted to Turbulence,Heat and mass transfer conference in Saint Petersburg and got accepted.

2.CNN+Decision tree result for one case(periodic hill) will be submitted for Marchuk Scientific Readings 2020 conference- Abstract already submitted.

3.CNN+Decision tree result for one case(square duct) will be submitted for High Energy Processes in Condensed Matter 2020- Abstract already submitted.

4.CNN+Decision tree result for one case(periodic hill) will be submitted for Thermophysics and Aeromechanics (TA) journal.

Preliminary Next Stage plan :

Heat transfer(Active and Passive scalar) will be added to these three cases and also gene expression programming will be implemented. Then the results could be submitted to a top journal such as Journal of Fluid Mechanics(IF=3.137)
Based on Richard Sandberg paper :
Data-driven scalar-flux model development with application to jet in cross flow.

References :

-Master thesis "Machine Learning for Data-Driven RANS Turbulence Modelling" by Mikael Kaandorp.
-Reynolds Averaged Turbulence Modeling using Deep Neural Networks with Embedded Invariance by Julia Ling

Thank you for your attention !