EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 13, Issue 2 (2025) 4 – 12

APPLICATION OF MACHINE LEARNING METHODS TO CHANNEL FLOW MODELLING

Bernard A., Yakovenko S.N. [1]

Abstract Machine-learning methods to enhance an approximation for the Reynolds-stress anisotropy tensor are presented. The approach of tensor basis random forest is applied for this. The set of input features and tensors in the basis are discussed. Different ways to propagate the Reynolds-stress anisotropy tensor into the Reynolds-averaged Navier–Stokes equation solver are explored. It is demonstrated that the conventional expression for Reynolds-stress anisotropy based on the linear eddy-viscosity model is not able to reproduce a secondary flow in the square duct cross-section, whereas the machine-learning modifications can fix such a disadvantage and have potentials for further improvements of turbulence models.

Key words: tensor basis random forest, Reynolds stress, RANS, LES, DNS, channel flow.

AMS Mathematics Subject Classification: 76-10, 76D05, 76F99.

DOI: 10.32523/2306-6172-2025-13-2-4-12

1 Introduction

The simulation of a turbulent flow has numerous applications in varied domains, such as weather forecasting, dam flow control management, aerospace, energy, which require accurate predictions. Such a flow has high values of the Reynolds number (Re) which is based on typical length and velocity scales. The eddy-resolving methods, large eddy simulation (LES) and direct numerical simulation (DNS), provide the high-fidelity predictions, but have computational costs that grow exponentially as Re increases. Another conventional approach is the Reynolds-averaged Navier–Stokes (RANS) equation model, which has the advantage of much lower computational costs, at the expense of its accuracy. In order to overcome the low-fidelity limitations, the use of machine learning (ML) in conjunction with traditional models has been proposed [1, 2].

Multiple ML approaches have been developed and applied to improve the predictions of RANS models, based on the tensor basis expansion [3], in particular: tensor basis neural network (TBNN) [4, 5], tensor basis random forest (TBRF) [6–8], multi-dimensional gene expression programming (MGEP) [9, 10], physics-informed machine learning (PIML) [11]. Moreover, other modern ML methods, e.g. physics-informed neural networks (PINNs) [12,13], can be mentioned here to give more accurate solutions to the RANS equations.

The present paper focuses on the TBRF method which is similar to the TBNN where the Reynolds-stress anisotropy (RSA) tensor is expressed as the linear combination of tensors from a tensor basis [3]. Two canonical turbulent flow cases in channels without and with bumps are considered here due to the available high-fidelity data sets collected together in the extensive studies by RANS models [14,15]. The novelty is to examine the possibility of TBRF to obtain the RSA distributions in a flow with a circular-shaped bump [16] which has not been done elsewhere before [6,7], as well as to perform such an examination for a flow in a squire duct [17] which is a simple prototype of more complex-geometry cases of engineering interest, e.g. a

[1]Corresponding Author.

flow in a rod bundle [18]. Such a study of enhancing the RANS model approximations is quite significant, since the conventional turbulence closures often fail when predicting the flow cases with recirculation, streamline curvature, rotation, secondary currents, and other complicated physical effects, so the new data-driven methods could substantially help to overcome the model inaccuracy.

2 Reynolds stress approximation

The RANS equations are formulated for incompressible fluid as

$$\frac{\partial U_i}{\partial x_i} = 0, \quad \frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left[-\frac{\overline{p}}{\rho}\delta_{ij} + \nu\frac{\partial U_i}{\partial x_j} - \tau_{ij}\right] = \frac{\partial}{\partial x_j}\left[-\frac{P}{\rho}\delta_{ij} + \nu\frac{\partial U_i}{\partial x_j} - a_{ij}\right],$$

where $U_i = \overline{u}_i$ is the mean velocity vector, ρ is the constant density, $\tau_{ij} = \overline{u_i' u_j'}$ is the Reynolds stress, $k = \frac{1}{2}\tau_{ii}$ is the turbulent kinetic energy, $\overline{p}$ is the true mean pressure, which is replaced by the modified mean pressure $P = \overline{p} + \frac{2}{3}\rho k$ [10] in many codes including OpenFOAM [19]. Boundary conditions differ for different flows and are set below when introducing the test cases. The Reynolds-stress anisotropy tensor is defined as:

$$a_{ij} = \tau_{ij} - \frac{2}{3}\delta_{ij}k, \tag{1}$$

where a_{ij} are the components of the Reynolds-stress anisotropy tensor, which can be normalized:

$$b_{ij} = \frac{a_{ij}}{2k} = \frac{\tau_{ij}}{2k} - \frac{1}{3}\delta_{ij}. \tag{2}$$

The unknown τ_{ij} or a_{ij} quantity should be modeled. In conventional RANS closures, the linear eddy viscosity model (LEVM) is applied to close the RANS equations:

$$a_{ij} = -2\nu_t S_{ij}, \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \tag{3}$$

where a_{ij} depends linearly on the mean strain rate tensor S_{ij} using the turbulent viscosity $\nu_t = C_\mu k^2/\varepsilon = k/\omega$ where $C_\mu = 0.09$. Extra equations for k and ε or ω are added in two-parameter ($k-\varepsilon$ or $k-\omega$) models to close the set of governing equations.

For more accurate prediction, the LEVM approximation can be generalized [3] with the b_{ij} expression as the sum of tensors from a tensor basis (hereafter, tensors are written in their compact form such as $\boldsymbol{\tau} = \tau_{ij}$, $\mathbf{b} = b_{ij}$, $\hat{\mathbf{S}}\hat{\mathbf{\Omega}} = \hat{S}_{ik}\hat{\Omega}_{kj}$):

$$\mathbf{b} = \sum_{m=1}^{10} g^{(m)}(\lambda_1, ..., \lambda_P)\mathbf{T}^{(m)},$$

where $g^{(m)}$, $\forall m \in \{1, \dots, 10\}$ are the scalar coefficients predicted from the features λ_i, $\forall i \in \{1, \dots, P\}$. The tensor basis $\mathbf{T}^{(m)}$ is defined as:

$$\begin{aligned}
&\mathbf{T}^{(1)} = \hat{\mathbf{S}} && \mathbf{T}^{(6)} = \hat{\mathbf{\Omega}}^2\hat{\mathbf{S}} + \hat{\mathbf{S}}\hat{\mathbf{\Omega}}^2 + \frac{2}{3}\mathbf{I}\cdot\mathrm{Tr}(\hat{\mathbf{S}}\hat{\mathbf{\Omega}}^2)\\
&\mathbf{T}^{(2)} = \hat{\mathbf{S}}\hat{\mathbf{\Omega}} - \hat{\mathbf{\Omega}}\hat{\mathbf{S}} && \mathbf{T}^{(7)} = \hat{\mathbf{\Omega}}\hat{\mathbf{S}}\hat{\mathbf{\Omega}}^2 - \hat{\mathbf{\Omega}}^2\hat{\mathbf{S}}\hat{\mathbf{\Omega}}\\
&\mathbf{T}^{(3)} = \hat{\mathbf{S}}^2 - \frac{1}{3}\mathbf{I}\cdot\mathrm{Tr}(\hat{\mathbf{S}}^2) && \mathbf{T}^{(8)} = \hat{\mathbf{S}}\hat{\mathbf{\Omega}}\hat{\mathbf{S}}^2 - \hat{\mathbf{S}}^2\hat{\mathbf{\Omega}}\hat{\mathbf{S}}\\
&\mathbf{T}^{(4)} = \hat{\mathbf{\Omega}}^2 - \frac{1}{3}\mathbf{I}\cdot\mathrm{Tr}(\hat{\mathbf{\Omega}}^2) && \mathbf{T}^{(9)} = \hat{\mathbf{\Omega}}^2\hat{\mathbf{S}}^2 + \hat{\mathbf{S}}^2\hat{\mathbf{\Omega}}^2 - \frac{2}{3}\mathbf{I}\cdot\mathrm{Tr}(\hat{\mathbf{S}}^2\hat{\mathbf{\Omega}}^2)\\
&\mathbf{T}^{(5)} = \hat{\mathbf{\Omega}}\hat{\mathbf{S}}^2 - \hat{\mathbf{S}}^2\hat{\mathbf{\Omega}} && \mathbf{T}^{(10)} = \hat{\mathbf{\Omega}}\hat{\mathbf{S}}^2\hat{\mathbf{\Omega}}^2 - \hat{\mathbf{\Omega}}^2\hat{\mathbf{S}}^2\hat{\mathbf{\Omega}},
\end{aligned} \tag{4}$$

where $\hat{\mathbf{S}} = \frac{k}{\epsilon}\left[\frac{1}{2}\left(\nabla\overline{\boldsymbol{u}} + \nabla\overline{\boldsymbol{u}}^T\right)\right]$ is the normalized mean strain rate tensor, $\hat{\mathbf{\Omega}} = \frac{k}{\epsilon}\left[\frac{1}{2}\left(\nabla\overline{\boldsymbol{u}} - \nabla\overline{\boldsymbol{u}}^T\right)\right]$ is the normalized mean rotation rate tensor, Tr denotes the trace.

3 TBRF Model

The TBRF is an agglomeration of tensor basis decision tree (TBDT). Each TBDT is a decision tree that is constructed using a modified version of the classification and regression tree (CART) algorithm. The CART algorithm performs the following process when creating any node from the root: it repeatedly selects a feature j (at random $p_{\max}$ times) and partitions the feature space into two bins, denoted as $\mathcal{B}_{\mathrm{L}}$ (left bin), $\mathcal{B}_{\mathrm{R}}$ (right bin) and written as:

$$\mathcal{B}_{\mathrm{L}}(j,s) = \{\boldsymbol{X} | \forall x \in \boldsymbol{X}, x_j \leq s\}, \quad \mathcal{B}_{\mathrm{R}}(j,s) = \{\boldsymbol{X} | \forall x \in \boldsymbol{X}, x_j > s\}, \tag{5}$$

where $\boldsymbol{X}$ denotes all the observations, x is one observation with P features such that $1 \leq j \leq P$, and s is the value of the split on the feature j.

The goal of the CART algorithm is to find the values that minimize the cost function defined as follows:

$$J = \sum_{x \in \mathcal{B}_{\mathrm{L}}} (c_1 - y)^2 + \sum_{x \in \mathcal{B}_{\mathrm{R}}} (c_2 - y)^2, \tag{6}$$

where y is the observed target of the observation x, and (c_1, c_2) are the predictions of $(\mathcal{B}_{\mathrm{L}}, \mathcal{B}_{\mathrm{R}})$. Usually, the predictions of the bins $(\mathcal{B}_{\mathrm{L}}, \mathcal{B}_{\mathrm{R}})$ are their respective average. To find the optimal value s for the feature j, the minimization problem can be written as:

$$\min_{j,s} \left[\min_{c_1} \sum_{x \in \mathcal{B}_{\mathrm{L}}} (c_1 - y)^2 + \min_{c_2} \sum_{x \in \mathcal{B}_{\mathrm{R}}} (c_2 - y)^2 \right] \tag{7}$$

The optimum value of s can be found by comparing the cost value of J for all s between two successive observed values of x_j for the feature considered.

Once the optimum value for s together with its cost value has been found for $p_{\max}$ random features, the s value of the feature j that best minimizes the cost function is kept in the node as the splitting value for this feature. This process is then repeated for all subsequent nodes.

The adaptation for the tensor basis approach redefines Equations (6) and (7) as:

$$J = \sum_{x \in \mathcal{B}_{\mathrm{L}}} \left\| \sum_{m=1}^{10} g_{\mathrm{L}}^{(m)} \mathsf{T}_n^{(m)} - y \right\|^2 + \sum_{x \in \mathcal{B}_{\mathrm{R}}} \left\| \sum_{m=1}^{10} g_{\mathrm{R}}^{(m)} \mathsf{T}_n^{(m)} - y \right\|^2 \tag{8}$$

$$\min_{j,s} \left[\min_{\boldsymbol{g}_{\mathrm{L}}} \sum_{x \in \mathcal{B}_{\mathrm{L}}} \left\| \sum_{m=1}^{10} g_{\mathrm{L}}^{(m)} \mathsf{T}_n^{(m)} - y \right\|^2 + \min_{\boldsymbol{g}_{\mathrm{R}}} \sum_{x \in \mathcal{B}_{\mathrm{R}}} \left\| \sum_{m=1}^{10} g_{\mathrm{R}}^{(m)} \mathsf{T}_n^{(m)} - y \right\|^2 \right], \tag{9}$$

where $\|z\| = \sqrt{\sum_{ij} z_{ij}^2}$ denotes the Frobenius norm. Noticing that this minimization is the sum of two least-square problems, and using the following matrix notation:

$$\mathbf{T} = \begin{pmatrix} T_{11}^{(1)} & T_{11}^{(2)} & \cdots & T_{11}^{(10)} \\ T_{12}^{(1)} & T_{12}^{(2)} & \cdots & T_{12}^{(10)} \\ \vdots & \vdots & \ddots & \vdots \\ T_{33}^{(1)} & T_{33}^{(2)} & \cdots & T_{33}^{(10)} \end{pmatrix}, \quad \boldsymbol{g} = \begin{pmatrix} g^{(1)} \\ g^{(2)} \\ \vdots \\ g^{(10)} \end{pmatrix}, \quad \mathsf{b} = \begin{pmatrix} b_{11} \\ b_{12} \\ \vdots \\ b_{33} \end{pmatrix}, \tag{10}$$

where $\mathbf{T}$ is the matrix of flatten tensors from T, $\boldsymbol{g}$ is the vector of coefficients of T tensors, and b is the flatten Reynolds-stress anisotropy tensor, the cost function can be rewritten as:

$$J = \sum_{n=1}^{N} \left[\|\mathbf{T}_n \boldsymbol{g} - \mathsf{b}_n\|^2 + \|\alpha \mathbf{I} \boldsymbol{g}\|^2 \right], \tag{11}$$

where the second term $\|\alpha \mathbf{I} \boldsymbol{g}\|^2$ is an added regularization that is controlled with the hyper-parameter $\gamma = \alpha^2$ from the TBDT. This regularization term is an incentive to have lower values for $\boldsymbol{g}$ and to provide a smoother cost function [6].

Using such a least-square problem, it is possible to obtain the tensor basis coefficients $\boldsymbol{g}$. The derivative of J, with respect to $\boldsymbol{g}$, can be set as equal to zero (implying an optimum) and the coefficients are then defined as:

$$\boldsymbol{g} = \left(\sum_{n=1}^{N} [\mathbf{T}_n^T \mathbf{T}_n + \gamma \mathbf{I}] \right)^{-1} \left(\sum_{n=1}^{N} \mathbf{T}_n^T \mathbf{b}_n \right).$$

This whole process is used to construct multiple TBDT which form the TBRF. Some extra hyper-parameters also define the TBRF model and their values can be tuned. Tab. 1 shows the values used for the subsequent results. The numerical experiments have shown that the small value for γ given in Tab. 1 yields better results, and the values for other parameters provide good enough results while keeping the computational time low. The ceiling function is denoted as $\lceil \cdot \rceil$.

The ML model is given by a set of input features summarized in Tab. 2.

4 Test cases and data sets

Two canonical flows in the channel with a bump (CB) and square duct (SD) are considered. Both of these cases are taken from a dataset compilation for ML applications [14], which provides the high-fidelity data extracted from the previous LES [16] and DNS [17] studies. Different parameters are selected for each flow, adding up to a total of eight cases. The reference data sets are also complemented by the results of five popular turbulence models obtained at the optimal meshes which have been chosen after the careful mesh-convergence and grid-independence studies [14]. These optimal mesh arrangements (Figs. 1 and 2) are taken in the present computations too.

The Reynolds number for the CB case is defined as $\mathrm{Re}_h = U_\infty h/\nu$ where h is the bump height, and U_∞ is the free stream inlet velocity. The Re_h values vary from 13260 to 27850 depending on the bump height range $h \in \{26, 31, 38, 42\}$ mm while the cord length is set to $C = 305$ mm [16] (Fig. 1). At the bottom wall, no-slip boundary condition is applied, whereas at the top (upper boundary) free-stream condition is specified, which is identical to the symmetry one. At the inlet boundary the distributions for the velocity, k and ε or ω values are assigned, corresponding to the turbulent boundary layer on the channel bottom.

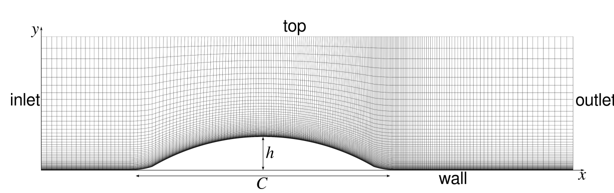

Figure 1: Domain and mesh scheme for CB case; each fourth grid line is shown.

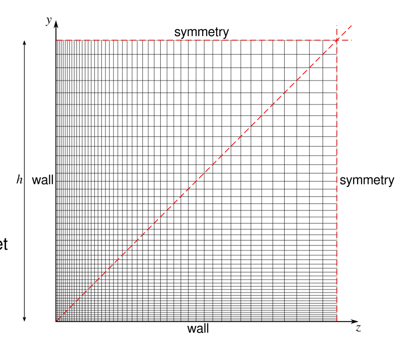

Figure 2: Domain and mesh scheme for SD case; symmetry planes are depicted in red.

At the outlet boundary, zero gradients in the normal direction are set for these quantities. For pressure, zero gradients in the normal direction are fixed for all boundaries.

For a SD flow, only the bottom-left quadrant of its cross-section is taken into account due to symmetry properties (Fig. 2), and the Reynolds number is defined as $\mathrm{Re}_b = U_b h/\nu \in \{2600, 2900, 3200, 3500\}$ where h is the duct half-width, U_b is the bulk inlet velocity [17]. At the bottom and left walls, no-slip boundary condition is applied, whereas the top and right boundaries have symmetry conditions. Along the third direction (x), there is an only cell with periodic conditions at its downstream and upstream faces. For pressure, zero gradients in the normal direction are fixed for all boundaries. The mean-pressure-gradient source term is added to the mean momentum equation (via meanVelocityForce inside fvOptions in OpenFOAM) to maintain a specified bulk velocity along the square duct as in [18].

For both flow cases, due to homogeneity of the mean quantities in time, the time derivatives in the momentum equation can be omitted (or used in the solver to obtain relaxation to a

Table 1: Hyper-parameters of the TBRF, and their values.

Hyper-parameter	Value	Description
γ	1.0×10^{-8}	Regularization parameter
$p_{\max}$	$\left\lceil \sqrt{P} \right\rceil = 5$	Number of features to choose from when performing a split
$n_{\text{estimators}}$	50	Number of TBDTs in the TBRF
r_{samples}	0.3	The fraction of samples to use from each dataset when bootstrapping

Table 2: List of features $\{\lambda_1, ..., \lambda_{17}\} = \mathrm{FS1} \cup \mathrm{FS2} \cup \mathrm{FS3}$ used as input in the ML algorithms, taken from [6, 11]; amounting to a total of $P = 17$ features. For each of the tensor quantities in FS1 and FS2, their trace is taken instead of the tensor quantities.

Set	Raw feature	Normalization	Details
FS1	$\hat{\mathbf{S}}^2, \hat{\mathbf{\Omega}}^2, \hat{\mathbf{\Omega}}^2\hat{\mathbf{S}}^2$	-	Invariants set based on $\hat{\mathbf{S}}$, $\hat{\mathbf{\Omega}}$
FS2	$\hat{\mathbf{A}}_{\mathbf{k}}^2, \hat{\mathbf{A}}_{\mathbf{k}}^2\hat{\mathbf{S}}, \hat{\mathbf{A}}_{\mathbf{k}}^2\hat{\mathbf{S}}^2, \hat{\mathbf{A}}_{\mathbf{k}}^2\hat{\mathbf{\Omega}}\hat{\mathbf{S}}, \hat{\mathbf{A}}_{\mathbf{k}}^2\hat{\mathbf{S}}\hat{\mathbf{\Omega}}\hat{\mathbf{S}}^2$	-	Invariants set based on the TKE gradient $\boldsymbol{\nabla}\mathbf{k}$, where $\hat{\mathbf{A}}_{\mathbf{k}} = -\mathbf{I} \times \frac{\sqrt{k}}{\epsilon}\nabla k$
FS3	$\frac{1}{2}\left(\lVert\hat{\mathbf{\Omega}}\rVert^2 - \lVert\hat{\mathbf{S}}\rVert^2\right)$	$\lVert\hat{\mathbf{S}}\rVert^2$	Ratio of excess rotation rate to strain rate
	k	$\frac{1}{2}\bar{u}_i\bar{u}_i$	Turbulent kinetic energy (TKE)
	$\min\left(\frac{\sqrt{k}d}{50\nu}, 2\right)$	-	Wall-distance based Reynolds number
	$\bar{u}_k\frac{\partial p}{\partial x_k}$	$\sqrt{\frac{\partial p}{\partial x_j}\frac{\partial p}{\partial x_j}\bar{u}_i\bar{u}_i}$	Pressure gradient along the streamline
	$\frac{k}{\epsilon}$	$\frac{1}{\lVert\mathbf{S}\rVert}$	Ratio of turbulent time scale to mean strain time scale
	$\sqrt{\frac{\partial p}{\partial x_j}\frac{\partial p}{\partial x_j}}$	$\frac{1}{2}\rho\frac{\partial}{\partial x_k}\bar{u}_k^2$	Ratio of pressure normal stresses to shear stresses
	$\bar{u}_i\frac{\partial k}{\partial x_i}$	$\lvert\tau_{jk}S_{jk}\rvert$	Ratio of convection to production of TKE
	$\lVert\boldsymbol{\tau}\rVert$	k	Ratio of total to normal Reynolds-stress
	$\left\lvert\bar{u}_i\bar{u}_j\frac{\partial\bar{u}_i}{\partial x_j}\right\rvert$	$\sqrt{\bar{u}_l\bar{u}_l\bar{u}_i\frac{\partial\bar{u}_i}{\partial x_j}\bar{u}_k\frac{\partial\bar{u}_k}{\partial x_j}}$	Non-orthogonality between velocity and its gradient

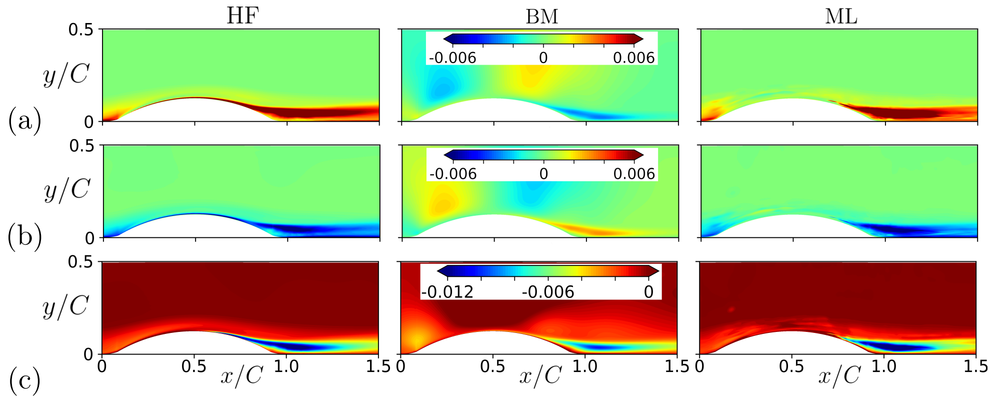

Figure 3: Components of a_{ij} for the CB testing case at $h = 38$ mm obtained from the data of LES (denoted as HF) [16], LEVM model (denoted as BM), ML prediction: (a) a_{11}/U_∞^2, (b) a_{22}/U_∞^2, (c) a_{12}/U_∞^2; colour legend is shown once for each row.

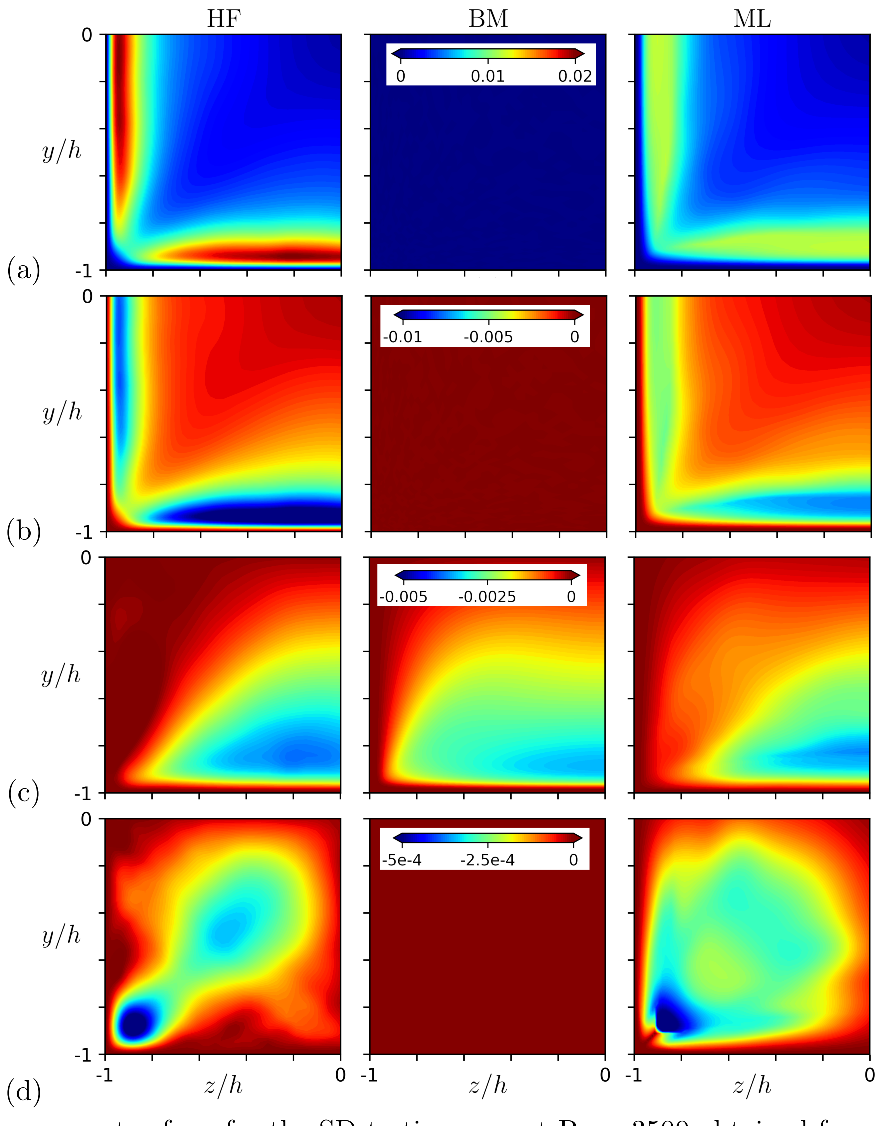

Figure 4: Components of a_{ij} for the SD testing case at Re $= 3500$ obtained from the data of DNS (denoted as HF) [17], LEVM model (denoted as BM), ML prediction: (a) a_{11}/U_b^2, (b) a_{22}/U_b^2, (c) a_{12}/U_b^2, (d) a_{23}/U_b^2; colour legend is shown once for each row.

time-independent solution with pseudo-time steps serving as iterations). Therefore, the initial conditions can be set arbitrarily, for instance, to be as in a fluid at rest, and solution proceeds up to its convergence to a steady-state solution with pseudo-time step iterations. Moreover, the steady-state two-dimensional solutions depend only on the two spatial coordinates x and y (for CB case), y and z (for SD case).

5 Results

The numerical simulations (which results for the cases introduced above are presented below) use the hyper-parameters and features shown in Tabs. 1 and 2.

For the CB cases, the ML model has been trained on the data sets with $h \in \{26, 31, 42\}$ mm and tested on the case with $h = 38$ mm. In Fig. 3, it is observed that the LEVM model yields the incorrect RSA distributions over the bump, while the ML plots better reproduce the HF data.

For the SD cases, after training the ML model on the high-fidelity data sets with $\mathrm{Re} \in \{2600, 2900, 3200\}$, the resulting model has been used to predict the Reynolds-stress anisotropy for the case with Re = 3500. The results are given in Fig. 4 where the a_{33} and a_{13} components are omitted as they are equivalent to a_{22} and a_{12} respectively after mirroring along the bisector plane $y = z$. It can be observed that the conventional LEVM approximation predicts zero values for a_{11}, a_{22} (and a_{33}), a_{23}, while the ML model provides more realistic and accurate results.

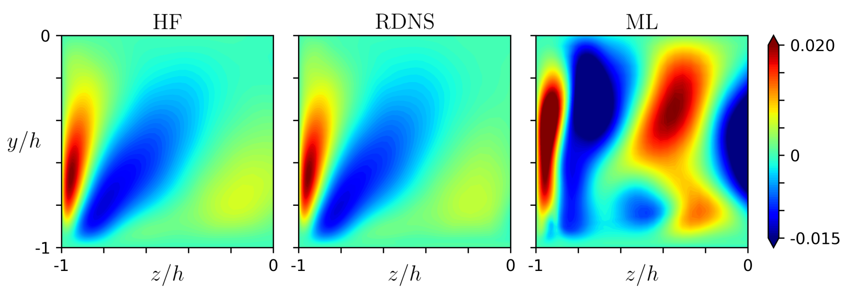

Figure 5: The normalized mean velocity component U_y/U_b after propagation of RSA into the solver, for the SD testing case at Re = 3500: DNS data (denoted as HF) [17]; RDNS prediction after propagation of RSA taken from DNS; ML prediction.

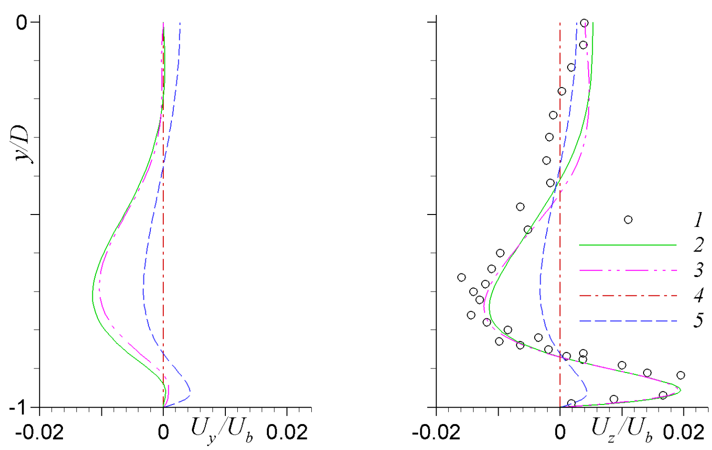

Figure 6: The normalized mean velocity vector components, for a SD flow at Re = 3500: *1* – DNS [17], *2* – measurements [17], *3* – RDNS predictions (present study), *4* – LEVM model [14], *5* – MGEP predictions [21].

Next, the RANS equations closed with the standard $k-\omega$ model [20] are solved numerically for the SD case at Re=3500 in OpenFOAM, with the Reynolds stress taken directly from the DNS data [17], or from the linear LEVM approximation, or from the non-linear modification using the TBRF model. In Fig. 5, the mean velocity of a secondary flow obtained after the Reynolds-stress propagation into the RANS equations is shown. Again, the component U_z has been omitted as it is equivalent to U_y after its mirroring along the bisector plane. The LEVM model does not predict secondary flows with non-zero U_y and U_z at all, while the ML model provides more accurate results.

The Reynolds-stress propagation from the conventional LEVM model yields the invalid (zero) reproduction for a secondary flow in the SD cross section, whereas the data-driven quadratic MGEP approximation [21] improves the predictions (Fig. 6). On the other hand, RANS-DNS runs (RDNS), where the frozen a_{ij} values from the DNS data [17] are inserted into the RANS equations, reveal as in [6, 11] the better performance versus that with LEVM or MGEP. It yields possibility of further model enhancement by ML algorithms (like TBNN, TBRF or their extensions) where high-fidelity data serve as target solutions, if their ML mimics correspond well to targets.

6 Conclusion

The TBRF algorithm has been implemented to enhance the performance of the baseline k-ω model, using the high-fidelity data of DNS or LES for canonical flows in channels with and without bumps. It has been shown that the predictions for the Reynolds-stress anisotropy tensor components are improved in comparison with those for the baseline LEVM approximation as shown in a priori test. The propagation of the Reynolds-stress anisotropy from DNS/LES or ML results can improve the mean velocity predictions, compared to those for LEVM, in particular, for a secondary flow in the SD case.

In future studies, the further improvement in accuracy of results is possible when carefully selecting optimal sets of appropriate training cases, tensor basis terms, input features, hyperparameters of the ML algorithm, and making global iterations to refine the Reynolds-stress model. Moreover, the comparative analyses of the TBRF algorithm results versus those for other tensor basis methods, like TBNN, MGEP, and against other modern ML methods, like PINNs, would be of considerable interest too.

Acknowledgement

The study has been supported by a grant No. 22-19-00587 of Russian Science Foundation, `https://rscf.ru/en/project/22-19-00587/`.

References

[1] Ling J., Templeton J. (2015). *Evaluation of Machine Learning Algorithms for Prediction of Regions of High Reynolds Averaged Navier Stokes Uncertainty*, Physics of Fluids, 27 (8), 085103. https://doi.org/10.1063/1.4927765

[2] Duraisamy K, Iaccarino G., Xiao H. (2019). *Turbulence modeling in the age of data*, Annual Review of Fluid Mechanics, 51, 1-23. https://doi.org/10.1146/annurev-fluid-010518-040547

[3] Pope S.B. (1975). *A more general effective-viscosity hypothesis*, Journal of Fluid Mechanics, 72, 331-340. https://doi.org/10.1017/S0022112075003382

[4] Ling J., Kurzawski A., Templeton J. (2016).*Reynolds averaged turbulence modelling using deep neural networks with embedded invariance*, Journal of Fluid Mechanics, 807, 155-166. https://doi.org/10.1017/jfm.2016.615

[5] Sandia National Laboratories, TBNN: Tensor Basis Neural Network. [Online]. Available: https://github.com/sandialabs/tbnn.

[6] Kaandorp M.L.A., Dwight R.P. (2020). *Data-driven modelling of the Reynolds stress tensor using random forests with invariance*, Computers and Fluids, 202, 104497. https://doi.org/10.1016/j.compfluid.2020.104497

[7] Bernard A., Yakovenko S.N. (2023). *Enhancement of RANS models by means of the tensor basis random forest for turbulent flows in two-dimensional channels with bumps*, Journal of Applied Mechanics and Technical Physics, 64(3), 437-441. https://doi.org/10.1134/S0021894423030094

[8] A. Bernard, FluidML: Implementation of Machine Learning Algorithms for Fluid Dynamics. [Online]. Available: https://github.com/AlixBernard/FluidML.

[9] Weatheritt J., Sandberg R. (2016). *A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship*, Journal of Computational Physics, 325, 22-37. https://doi.org/10.1016/j.jcp.2016.08.015

[10] Weatheritt J., Sandberg R. (2017). *The development of algebraic stress models using a novel evolutionary algorithm*, International Journal of Heat and Fluid Flow, 68, 298-318. https://doi.org/10.1016/j.ijheatfluidflow.2017.09.017

[11] Wu J.-L., Xiao H., Paterson P. (2018). *Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework*, Physical Review Fluids, 3(7), 074602. https://doi.org/10.1103/PhysRevFluids.3.074602

[12] Eivazi H., Tahani M., Schlatter P., Vinuesa R. (2022). *Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations*, Physics of Fluids, 34(7), 075117. https://doi.org/10.1063/5.0095270

[13] Harmening J.H., Pioch F., Fuhrig L., Peitzmann F.-J., Schramm D., Moctar O. (2024). *Data-assisted training of a physics-informed neural network to predict the separated Reynolds-averaged turbulent flow field around an airfoil under variable angles of attack*, Neural Computing and Applications, 36, 15353-15371. https://doi.org/10.1007/s00521-024-09883-9

[14] McConkey R., Yee E., Lien, F.-S. (2021). *A curated dataset for data-driven turbulence modelling*, Scientific Data, 8, Article 255. https://doi.org/10.1038/s41597-021-01034-2

[15] R. McConkey, "ML turbulence dataset" [Online]. Available: https://www.kaggle.com/datasets/ryleymcconkey/ml-turbulence-dataset.

[16] Matai R., Durbin P. (2019). *Large-eddy simulation of turbulent flow over a parametric set of bumps*, Journal of Fluid Mechanics, 866, 503-525. https://doi.org/10.1017/jfm.2019.98

[17] Pinelli A., Uhlmann M., Sekimoto A., Kawahara G. (2010). *Reynolds number dependence of mean flow structure in square duct turbulence*, Journal of Fluid Mechanics, 644, 107-122. https://doi.org/10.1017/S0022112009992242

[18] Li H., Yakovenko S.N., Ivashchenko V., Lukyanov A., Mullyadzhanov R., Tokarev M. (2024). *Data-driven turbulence modeling for fluid flow and heat transfer in peripheral subchannels of a rod bundle*, Physics of Fluids, 36(2), 025141. https://doi.org/10.1063/5.0184157

[19] OpenFOAM, "The OpenFOAM Foundation" [Online]. Available: https://openfoam.org/.

[20] Wilcox D.C. (1988). *Reassessment of the scale-determining equation for advanced turbulence models*, AIAA Journal, 26(11), 1299-1310. https://doi.org/10.2514/3.10041

[21] Li H., Yakovenko S.N. (2023). *Application of machine learning methods to develop algebraic Reynolds-stress models for flows in channels*. Turbulence, Heat and Mass Transfer (Vol. 10, pp. 345-348). Begell House. https://doi.org/10.1615/ICHMT.THMT-23.710

Alix Bernard,
Novosibirsk State University,
Pirogova st. 1, 630090 Novosibirsk, Russia,
Email: `alix.bernard9@gmail.com`

Sergey N. Yakovenko,
Kutateladze Institute of Thermophysics SB RAS,
pr. Lavrentyev 1, 630090 Novosibirsk, Russia,
Email: `s.yakovenko@mail.ru`

Received 01.08.2024 , Revised 20.12.2024, Accepted 15.01.2025, Available online 30.06.2025.