An invariances-preserving vector basis neural network for the closure of Reynolds-averaged Navier-Stokes equations by the divergence of the Reynolds stress tensor

Accepted to Phys. Fluids 10.1063/5.0104605

An invariances-preserving Vector Basis Neural Network for the closure of Reynolds-averaged Navier-Stokes equations by the divergence of the Reynolds Stress Tensor

S. Berrone[1] and D. Oberto[1]

Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.

(*Electronic mail: davide.oberto@polito.it)

(Dated: 13 August 2022)

In the present paper a new data-driven model is proposed to close and increase accuracy of Reynolds-averaged Navier-Stokes (RANS) equations. Among the variety of turbulent quantities, it has been decided to predict the divergence of the Reynolds Stress Tensor (RST). Recent literature works highlighted the potential of this choice. The key novelty of this work is to obtain the divergence of the Reynolds Stress Tensor through a Neural Network (NN) whose architecture and input choice guarantee both Galilean and coordinates-frame rotation. The former derives from the input choice of the NN while the latter from the expansion of the divergence of the RST into a vector basis. This approach has been widely used for data-driven models for the RST anisotropy or the RST discrepancies but surprisingly not for the divergence of the RST. The present paper tries to fill this literature gap. Hence, a constitutive relation of the divergence of the RST from mean quantities is proposed to obtain such expansion. Moreover, once the proposed data-driven approach is trained, there is no need to run any classic turbulence model to close the equations.

The well-known tests of flow in a square duct and over periodic hills are used to show advantages of the present method compared to standard turbulence models.

I. INTRODUCTION

Reynolds-Averaged Navier-Stokes (RANS) equations are widely used in engineering for turbulent flow simulations. Their popularity comes from the low computational cost compared to Large-Eddy Simulations (LES) and Direct Numerical Simulation (DNS) approaches. However, RANS predictions may be highly inaccurate for some classes of flows[1] due to the bad description of the effects of the Reynolds stresses on the mean flow[2]. On the other hand, thanks to the remarkable growth of High Performance Computing facilities, more and more DNS data coming from simple geometries and moderate Reynolds numbers are becoming available. Some classes of flows with DNS datasets are: channel flows[3,4], pipe and duct flows[5–8], flows over periodic hills[9,10], flows around cylinders[11–13]. As a consequence, in the past years an increasing number of studies took advantage of Machine Learning techniques to exploit DNS data. The remarkable variety of review articles in the recent years on the subject highlights this trend[14–17]. In particular, one active research area is focused on data-driven RANS turbulence models that increase accuracy through DNS (or highly accurate LES) data.

In this framework, data-driven models must satisfy the same invariance properties of the physical system they are modelling. Ling et al.[18] proved that invariance to coordinates-frame rotation can be guaranteed by taking for each physical dimension 10 rotations of the initial dataset and by including them in an augmented dataset. Even if this approach is conceptually valid, it has the huge drawback that for 3D problems the dataset would become 1000 bigger than the initial one making this method impracticable both in terms of learning time and of memory storage.

Another approach to guarantee invariance properties has been proposed in the pioneering paper of Ling et al.[19]. In this work, invariance is automatically satisfied by the architecture of the trained Neural Network (NN). In particular, the NN outputs were the coefficients of the decomposition of the Reynolds Stress Tensor anisotropy into a tensor basis defined in Pope[20], while the inputs were invariant quantities. The obtained neural network was named Tensor Basis Neural Network (TBNN). Since then, various studies have been performed to analyse data-driven approaches for the Reynolds Stress Tensor anisotropy. A change to the TBNN was proposed to increase accuracy[21]. A Convolutional Neural Network was used for 1D turbulent flows and its interpretability was discussed[22]. Both the coefficients and the basis of the Reynolds Stress Tensor anisotropy were inferred[23]. This approach was also applied for wind turbine wakes under neutral conditions[24]. The same rotation-invariance idea was used to train random forests able to predict the discrepancy between the Reynolds Stress Tensor (RST) obtained by a baseline RANS turbulence model and the DNS one[25,26].

A recent work[27] showed that predicting the divergence of the Reynolds Stress Tensor, denoted as Reynolds Force Vector (RFV), instead of the RST itself can effectively increase the accuracy of data-driven RANS turbulence models. From one hand, the RFV can be obtained from first order statistics reducing intrinsic statistical errors of DNS data. On the other hand, the RFV directly compares into the RANS equations and it seems natural to directly predict it. This work is grounded on these observations and aims to enforce into the RANS system physical invariance properties by construction without data augmentation. Analogously to Ling et al. approach[19], in this work a constitutive assumption of the RFV from mean fields is derived. This hypothesis is fundamental to derive the inputs of the data-driven model and the vector basis used to expand the RFV. The obtained neural network is called Vector Basis Neural Network (VBNN). The proposed approach closes the RANS system without the requirements of additional PDEs for turbulent scalar quantities or for the RST discrepancies as in previous works. As a consequence, once the model is satisfactorily trained, it does not require any coupling with a classic turbulence model.

Besides this introduction, the paper is organized into four more sections. In Section 2 a brief overview on RANS models is given to successively describe the TBNN[19]. In Section 3 the constitutive dependencies of the divergence of the RST are derived. The properties of the VBNN are consequently discussed. The numerical results are presented and discussed in Section 4. Classic benchmark flows in a square duct and over periodic hills are chosen as numerical experiments to analyze the data-driven model for two main reasons: i) availability of highly reliable DNS data in literature; ii) despite their geometrical simplicity, they represent a powerful proof-of-concept flows because classic RANS turbulence models fail in the prediction of their velocity fields. Finally, in the last section conclusions are drawn.

II. TENSOR BASIS NEURAL NETWORK

A. Reynolds-Averaged Navier-Stokes equation and turbulence models

The RANS equations for incompressible flows read

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} = -\nabla p - \nabla \cdot \boldsymbol{\tau}, \end{cases} \tag{1}$$

where $\mathbf{u}$ is the averaged velocity field, ν is the kinematic viscosity of the fluid, p is the averaged pressure field divided by the constant density of the fluid and $\boldsymbol{\tau}$ is the Reynolds Stress Tensor. The latter is a symmetric tensor that needs to be modelled to close the RANS equations and whose components are associated to the correlations of the turbulent fluctuating components of the velocity field. Hence, the divergence of the RST describes the effects of the turbulence on the averaged fields.

One class of turbulence models, called *linear isotropic models*, is based on the well known Boussinesq hypothesis. The RST is modelled as

$$\boldsymbol{\tau} = \frac{2}{3}k\mathbf{I} - 2\nu_t\mathbf{S} \tag{2}$$

where $k = \frac{1}{2}\mathrm{tr}(\boldsymbol{\tau})$ is the *turbulent kinetic energy* (tr denotes the trace operator), $\mathbf{I}$ is the identity tensor, ν_t is the *turbulent viscosity* and $\mathbf{S} = \frac{1}{2}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]$ is the mean strain rate tensor. The quantity ν_t must be modelled and the system is usually closed by two PDEs (for example one for the turbulent kinetic energy k and one for its dissipation rate ε using the relation $\nu_t = C_\mu k^2/\varepsilon$ being C_μ a model's constant). In literature many different linear turbulence models are defined depending, for example, on the choice of the variables solved to model ν_t.

The linear isotropic models fail in the description of some physical behaviours. As a consequence, more advanced non-linear models have been proposed in literature[1,20]. These models assume an algebraic representation of $\boldsymbol{\tau}$ more complex than in (2). Indeed, they include dependences on high-order powers of the mean strain rate tensor and the mean rotation rate tensor $\mathbf{W} = \frac{1}{2}[\nabla\mathbf{u} - (\nabla\mathbf{u})^T]$.

Another classical approach to close the RANS system (1) is to solve a PDE for each component of the RST tensor[28–30]. This class of models are called Reynolds Stress Transport Models (RSTM). This procedure does not require any modeling of the RST but, on the other hand, require the modeling of some terms inside the PDEs.

Unfortunately, both non-linear and RSTM turbulence models are more likely to diverge compared to classic linear models and, consequently, the latter are still the main option for many flow cases[31,32].

B. Tensor Basis Neural Network

Let $a_{ij} = \tau_{ij}/(2k) - 1/3\ \delta_{ij}$ be the anisotropic Reynolds stresses where τ_{ij} are the Reynolds stresses and δ_{ij} is the Kronecker delta. In the following, the tensor $\mathbf{a} = a_{ij}$ will be referred as the

Reynolds stress tensor anisotropy. The tensor $\mathbf{a}$ is dimensionless with vanishing trace by definition.

Let $\mathbf{s} = \frac{1}{2}\frac{k}{\varepsilon}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]$ and $\mathbf{w} = \frac{1}{2}\frac{k}{\varepsilon}[\nabla\mathbf{u} - (\nabla\mathbf{u})^T]$ be the dimensionless counterparts of the mean strain rate tensor $\mathbf{S}$ and mean rotation rate tensor $\mathbf{W}$ respectively, where ε is the turbulent kinetic energy dissipation rate.

Some classic algebraic turbulence models for the RST can be rephrased as algebraic models for its anisotropic part in terms of $\mathbf{s}$ and $\mathbf{w}$. For example the classic linear closure (2) is equivalent to $\mathbf{a} = -\frac{\varepsilon}{k^2}\nu_t\mathbf{s}$.

Pope[20] supposed a more general constitutive relation $\mathbf{a} = \mathbf{a}(\mathbf{s},\mathbf{w})$. This assumption and representation theorems[33] lead to

$$\mathbf{a} = \sum_{j=1}^{10} c_j(\lambda_1,\dots,\lambda_6)\mathbf{T}_j, \tag{3}$$

where λ_i, $i = 1,\dots,6$, are invariant scalar quantities that depend on $\mathbf{s}$ and $\mathbf{w}$. They are[20,33]

$$\begin{aligned} &\lambda_1 = \mathrm{tr}(\mathbf{s}^2), \quad \lambda_2 = \mathrm{tr}(\mathbf{s}^3), \quad \lambda_3 = \mathrm{tr}(\mathbf{w}^2), \\ &\lambda_4 = \mathrm{tr}(\mathbf{s}\mathbf{w}^2), \quad \lambda_5 = \mathrm{tr}(\mathbf{s}^2\mathbf{w}^2), \quad \lambda_6 = \mathrm{tr}(\mathbf{s}^2\mathbf{w}^2\mathbf{s}\mathbf{w}). \end{aligned} \tag{4}$$

Furthermore, $\mathbf{T}_j$, $j = 1,\dots,10$, are

$$\begin{aligned} &\mathbf{T}_1 = \mathbf{s}, \quad \mathbf{T}_2 = \mathbf{s}\mathbf{w} + \mathbf{w}\mathbf{s}, \quad \mathbf{T}_3 = \mathbf{s}^2 - \frac{1}{3}\mathrm{tr}(\mathbf{s}^2)\mathbf{I}, \quad \mathbf{T}_4 = \mathbf{w}^2 - \frac{1}{3}\mathrm{tr}(\mathbf{w}^2)\mathbf{I}, \\ &\mathbf{T}_5 = \mathbf{w}\mathbf{s}^2 - \mathbf{s}^2\mathbf{w}, \quad \mathbf{T}_6 = \mathbf{w}^2\mathbf{s} + \mathbf{s}\mathbf{w}^2 - \frac{2}{3}\mathrm{tr}(\mathbf{s}\mathbf{w}^2)\mathbf{I}, \quad \mathbf{T}_7 = \mathbf{w}\mathbf{s}\mathbf{w}^2 - \mathbf{w}^2\mathbf{s}\mathbf{w}, \\ &\mathbf{T}_8 = \mathbf{s}\mathbf{w}\mathbf{s}^2 - \mathbf{s}^2\mathbf{w}\mathbf{s}, \quad \mathbf{T}_9 = \mathbf{w}^2\mathbf{s}^2 + \mathbf{s}^2\mathbf{w}^2 - \frac{2}{3}\mathrm{tr}(\mathbf{s}^2\mathbf{w}^2)\mathbf{I}, \\ &\mathbf{T}_{10} = \mathbf{w}\mathbf{s}^2\mathbf{w}^2 - \mathbf{w}^2\mathbf{s}^2\mathbf{w}. \end{aligned} \tag{5}$$

Equation (3) states that the Reynolds Stress Tensor anisotropy can be expressed as a finite linear combination of the 10 tensor basis elements $\{\mathbf{T}_j\}$ listed above. Moreover, the coefficients involved are functions of the 6 invariants $\{\lambda_i\}$. Both the tensor basis and the invariants are known *a priori*. The only unknowns are the expressions of the 10 coefficients $\{c_j\}$.

The formula (3) has been the starting point for Ling et al.[19] to define a Tensor Basis Neural Network able to predict the coefficients using the invariants as inputs. Once the coefficients are obtained, the linear combination is computed to obtain $\mathbf{a}$. This approach has the huge advantage that the coefficients are automatically invariant to coordinates-frame rotations and Galilean transformations. This property arises from the Galilean and coordinates-frame rotation invariance of the inputs $\{\lambda_i\}$ that are fed into the TBNN.

III. VECTOR BASIS NEURAL NETWORK

This section focuses on the Vector Basis Neural Network used in this work to close the RANS system by obtaining the $\nabla\cdot\boldsymbol{\tau}$ term.

A. Constitutive dependencies

Let us define the dimensionless quantity $\widetilde{\nabla\cdot\boldsymbol{\tau}} = \frac{k^{1/2}}{\varepsilon}\nabla\cdot\boldsymbol{\tau}$. In the present work we assume the constitutive hypothesis

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \mathbf{f}(\mathbf{s},\mathbf{w},\widetilde{\nabla\cdot\mathbf{S}},\widetilde{\nabla k},Re_d), \tag{6}$$

where $\mathbf{s} = \frac{k}{\varepsilon}\mathbf{S}$, $\mathbf{w} = \frac{k}{\varepsilon}\mathbf{W}$, $\widetilde{\nabla\cdot\mathbf{S}} = \frac{k^{5/2}}{\varepsilon^2}\nabla\cdot\mathbf{S}$ and $\widetilde{\nabla k} = \frac{k^{1/2}}{\varepsilon}\nabla k$ are the dimensionless counterparts of the symmetric part of the velocity gradient $\mathbf{S}$, the antisymmetric part of the velocity gradient $\mathbf{W}$, the divergence of $\mathbf{S}$ and the gradient of k, respectively. Finally, $Re_d = \min(\frac{\sqrt{k}d}{50\nu},2)$ is the wall-distance based Reynolds number, where d is the wall distance. This quantity is relevant during the training process as reported in literature[25,34]. The motivations behind this constitutive choice are discussed in Appendix.

With the above hypothesis, following the idea in Ling et al.[19], $\widetilde{\nabla\cdot\boldsymbol{\tau}}$ can be written in a basis made by $N_c = 12$ vectors $\{\mathbf{t}_k\}_{k=1}^{N_c}$ with corresponding coefficients that depend on $N_i = 26$ invariant scalar quantities $\{\lambda_k\}_{k=1}^{N_i}$. In particular, it reads

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \sum_{k=1}^{N_c} c_k(\lambda_1,\ldots,\lambda_{N_i})\,\mathbf{t}_k. \tag{7}$$

The appropriate vector basis and the invariants can be obtained from Zheng[33] (Tables 1,2). The vector basis reads

$$\begin{aligned}
&\mathbf{t}_1 = \widetilde{\nabla\cdot\mathbf{S}}, \quad \mathbf{t}_2 = \mathbf{s}\,\widetilde{\nabla\cdot\mathbf{S}}, \quad \mathbf{t}_3 = \mathbf{s}^2\,\widetilde{\nabla\cdot\mathbf{S}},\\
&\mathbf{t}_4 = \mathbf{w}\,\widetilde{\nabla\cdot\mathbf{S}}, \quad \mathbf{t}_5 = \mathbf{w}^2\,\widetilde{\nabla\cdot\mathbf{S}}, \quad \mathbf{t}_6 = (\mathbf{sw}+\mathbf{ws})\,\widetilde{\nabla\cdot\mathbf{S}},\\
&\mathbf{t}_7 = \widetilde{\nabla k}, \quad \mathbf{t}_8 = \mathbf{s}\,\widetilde{\nabla k}, \quad \mathbf{t}_9 = \mathbf{s}^2\,\widetilde{\nabla k},\\
&\mathbf{t}_{10} = \mathbf{w}\,\widetilde{\nabla k}, \quad \mathbf{t}_{11} = \mathbf{w}^2\,\widetilde{\nabla k}, \quad \mathbf{t}_{12} = (\mathbf{sw}+\mathbf{ws})\,\widetilde{\nabla k}.
\end{aligned} \tag{8}$$

The invariants are

$$
\begin{aligned}
&\lambda_1 = (\widetilde{\nabla\cdot\mathbf{S}})^T(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_2 = \mathrm{tr}(\mathbf{s}^2), \quad \lambda_3 = \mathrm{tr}(\mathbf{s}^3), \quad \lambda_4 = \mathrm{tr}(\mathbf{w}^2),\\
&\lambda_5 = \mathrm{tr}(\mathbf{s}\mathbf{w}^2), \quad \lambda_6 = \mathrm{tr}(\mathbf{s}^2\mathbf{w}^2), \quad \lambda_7 = \mathrm{tr}(\mathbf{s}^2\mathbf{w}^2\mathbf{s}\mathbf{w}), \quad \lambda_8 = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{s}(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_9 = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{s}^2(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_{10} = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{w}^2(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_{11} = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{s}\mathbf{w}(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_{12} = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{s}^2\mathbf{w}(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_{13} = (\widetilde{\nabla\cdot\mathbf{S}})^T\mathbf{w}\mathbf{s}\mathbf{w}^2(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_{14} = (\widetilde{\nabla k})^T(\widetilde{\nabla k}), \quad \lambda_{15} = (\widetilde{\nabla k})^T\mathbf{s}(\widetilde{\nabla k}), \quad \lambda_{16} = (\widetilde{\nabla k})^T\mathbf{s}^2(\widetilde{\nabla k}),\\
&\lambda_{17} = (\widetilde{\nabla k})^T\mathbf{w}^2(\widetilde{\nabla k}), \quad \lambda_{18} = (\widetilde{\nabla k})^T\ \widetilde{\nabla\cdot\mathbf{S}}, \quad \lambda_{19} = (\widetilde{\nabla k})^T\mathbf{s}\mathbf{w}(\widetilde{\nabla k}),\\
&\lambda_{20} = (\widetilde{\nabla k})^T\mathbf{s}^2\mathbf{w}(\widetilde{\nabla k}), \quad \lambda_{21} = (\widetilde{\nabla k})^T\mathbf{w}\mathbf{s}\mathbf{w}^2(\widetilde{\nabla k}), \quad \lambda_{22} = (\widetilde{\nabla k})^T\mathbf{s}\mathbf{w}(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_{23} = (\widetilde{\nabla k})^T\mathbf{s}^2\mathbf{w}(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_{24} = (\widetilde{\nabla k})^T\mathbf{w}(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_{25} = (\widetilde{\nabla k})^T\mathbf{w}\mathbf{s}\mathbf{w}^2(\widetilde{\nabla\cdot\mathbf{S}}), \quad \lambda_{26} = (\widetilde{\nabla k})^T(\mathbf{s}\mathbf{w}+\mathbf{w}\mathbf{s})(\widetilde{\nabla\cdot\mathbf{S}}),\\
&\lambda_{27} = Re_d,
\end{aligned}
\tag{9}
$$

where the first 26 invariants derive from the dependencies on $\mathbf{s}, \mathbf{w}, \widetilde{\nabla\cdot\mathbf{S}}, \widetilde{\nabla k}$ while the last one is the scalar quantity added in the dependencies assumption (6). The latter is invariant to the choice of the coordinates-frame. The invariant $\mathrm{tr}(\mathbf{s})$ is neglected because identically zero due to the incompressibility constraint.

In Section IV, we will consider also the simplified assumption

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \mathbf{f}(\mathbf{s}, \mathbf{w}, \widetilde{\nabla\cdot\mathbf{S}}, Re_d).$$

In this case, the vector basis is formed by the first 6 vectors in (8) while the invariants are the first 13 and the last one in (9) (they are the expressions that do not involve $\widetilde{\nabla k}$).

B. Vector Basis Neural Network

1. *Inputs and outputs*

The Vector Basis Neural Network outputs are the coefficients c_j, $j = 1, \ldots, N_c$, in (7). The VBNN should be able to reproduce the divergence of the RST using only informations coming from RANS simulations. In particular, during the training stage, the quantity $\nabla\cdot\boldsymbol{\tau}$ comes from the DNS while the invariants, the vectors and the adimensionalization factors come from the RANS.

Therefore, during the training the optimization process aims to reduce

$$|| \, (\widetilde{\nabla\cdot\boldsymbol{\tau}})^{\mathrm{DNS}} - \sum_{k=1}^{N_c} c_k^{\mathrm{NN}}(\lambda_1^{\mathrm{RANS}},\dots,\lambda_{N_i}^{\mathrm{RANS}}) \, \mathbf{t}_k^{\mathrm{RANS}} \, ||_2. \tag{10}$$

Here, the quantities $\lambda_i^{\mathrm{RANS}}$, $i = 1,\dots,N_i$, and $\mathbf{t}_k^{\mathrm{RANS}}$, $k = 1,\dots,N_c$, come from RANS simulations whereas the quantities c_k^{NN}, $k = 1,\dots,N_c$, are the Neural Network outputs. In equation (10), with an abuse of notation, we define $(\widetilde{\nabla\cdot\boldsymbol{\tau}})^{\mathrm{DNS}} = \frac{(k^{1/2})^{\mathrm{RANS}}}{\varepsilon^{\mathrm{RANS}}}\nabla\cdot\boldsymbol{\tau}^{\mathrm{DNS}}$. In some cases, the DNS ε field is not available. For this reason, the dimensionless VBNN output must be successively dimensionalized using RANS fields.

Recent works[27,35,36] proposed indirect approaches to obtain $(\nabla\cdot\boldsymbol{\tau})^{\mathrm{DNS}}$ aiming to reduce errors associated to the lack of statistical convergence in DNS databases. In this work, $(\nabla\cdot\boldsymbol{\tau})^{\mathrm{DNS}}$ is obtained by interpolation of the available $\boldsymbol{\tau}^{\mathrm{DNS}}$ into the RANS mesh followed by computation of its divergence on the RANS mesh.

2. *Architecture and hyperparameters*

The input and output layers of the VBNN have a number of nodes that is constrained by the assumption (6). In particular, the input layer has N_i nodes (as many as the invariants) while the output layer has N_c nodes (as many as the coefficients to be predicted), see Figure 1. In this work, N_i is smaller compared to other models in literature such as Cruz et al.[27] where $N_i = 72$.

After some tests, it has been noticed that increasing the depth to more than 6 hidden layers or the width to more than 30 nodes per hidden layer did not increase the prediction ability of the network. This behavior was observed also in literature[34]. At the end, 8 hidden layers have been defined with 30 nodes each as made by Ling et al.[19].

It has been observed that the network is not affected by overfitting issues. Thus, the regularization term associated to the weights norm has been shut down. The Adam optimizer[37] is used with learning rate that decreases during the training stage from 10^{-3} to 10^{-5} and batch size equal to 50. The Exponential Linear Unit[38] (ELU) function has been chosen as activation function because of the better observed performances.

Due to the intrinsic stochasticity of the optimization process, several training runs with the same hyperparameters have been performed. Among them, the run that minimized the validation error has been identified as the reference one for the specific hyperparameters choice.

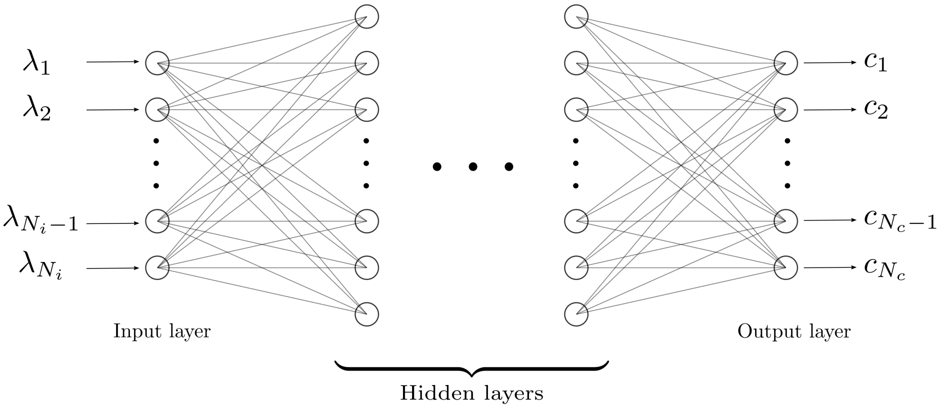


FIG. 1. Architecture of the Vector Basis Neural Network.

C. Invariance properties

1. Galilean invariance

Galilean invariance states that the laws of motion are the same in all frames with constant velocities[26]. All the inputs of the VBNN are Galilean invariant because k, ε, $\mathbf{S}$, $\mathbf{W}$, $\nabla \cdot \mathbf{S}$ and ∇k are all Galilean invariant. Consequently, the outputs of the VBNN, that depends on the inputs, do not change through a Galilean transformation.

2. Coordinates-frame rotation invariance

Given a rotation matrix $\mathbf{Q}$, the invariants (9) and the vectors (8) follow the transformation law[18]

$$\begin{aligned} \lambda_i^{\mathbf{Q}} &= \lambda_i, \qquad i = 1,\dots,N_i, \\ \mathbf{t}_k^{\mathbf{Q}} &= \mathbf{Q}\mathbf{t}_k, \qquad k = 1,\dots,N_c, \end{aligned} \tag{11}$$

where the superscript $\mathbf{Q}$ denotes the representation of the quantity in the rotated coordinates system.

The VBNN is coordinates-frame rotation invariant in the sense that all the scalar outputs are coordinates-frame rotation invariant. This property directly derives from the invariance of the scalar inputs, i.e. $\lambda_i^{\mathbf{Q}} = \lambda_i$. It implies that $\widetilde{\nabla\cdot\boldsymbol{\tau}}$ transforms correctly under rotations. Indeed

$$\begin{aligned}\widetilde{\nabla\cdot\boldsymbol{\tau}}^{\mathbf{Q}} &= \sum_{k=1}^{N_c} c_k(\lambda_1^{\mathbf{Q}},\dots,\lambda_{N_i}^{\mathbf{Q}})\,\mathbf{t}_k^{\mathbf{Q}} = \sum_{k=1}^{N_c} c_k(\lambda_1,\dots,\lambda_{N_i})\,\mathbf{Q}\mathbf{t}_k = \\ &= \mathbf{Q}\Big[\sum_{k=1}^{N_c} c_k(\lambda_1,\dots,\lambda_{N_i})\,\mathbf{t}_k\Big] = \mathbf{Q}\widetilde{\nabla\cdot\boldsymbol{\tau}}.\end{aligned} \tag{12}$$

D. Implicit-Explicit treatment of $\nabla\cdot\boldsymbol{\tau}$

Once the term $\nabla\cdot\boldsymbol{\tau}$ is obtained, the RANS system (1) has to be solved. The easiest approach is to treat explicitly this term like a source term. However, the ill-conditioning of this approach is known[39] when dealing with data-driven Reynolds Stress Tensor. Consequently, Wu et al. proposed to treat implicitly the Reynolds Stress Tensor component aligned to the mean strain rate tensor $\mathbf{S}$ into the diffusive term. A better conditioning of the system was observed with this approach. The present work takes inspiration on this remark with the slight change imposed by dealing with the divergence of the RST instead of the RST itself. Hence, the attention will be devoted to the term aligned with $\nabla\cdot\mathbf{S}$.

Let us take the expression (7) with the first term explicitly written

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = c_1\widetilde{\nabla\cdot\mathbf{S}} + \sum_{k=2}^{N_c} c_k\,\mathbf{t}_k. \tag{13}$$

Recalling $\widetilde{\nabla\cdot\boldsymbol{\tau}} = \frac{k^{1/2}}{\varepsilon}\nabla\cdot\boldsymbol{\tau}$ and $\widetilde{\nabla\cdot\mathbf{S}} = \frac{k^{5/2}}{\varepsilon^2}\nabla\cdot\mathbf{S}$, the above expression becomes

$$\nabla\cdot\boldsymbol{\tau} = \frac{k^2}{\varepsilon}\,c_1\nabla\cdot\mathbf{S} + \frac{\varepsilon}{k^{1/2}}\sum_{k=2}^{N_c} c_k\,\mathbf{t}_k. \tag{14}$$

The scalar term $\frac{k^2}{\varepsilon}$ is dimensionally a viscosity. This remark drives to the definition of the *turbulent-like viscosity*

$$\nu_{tl} := -\frac{k^2}{2\varepsilon}\,c_1. \tag{15}$$

Thus, the momentum equation of the RANS system reads

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - (\nu+\nu_{tl})\Delta\mathbf{u} = -\nabla p - \frac{\varepsilon}{k^{1/2}}\sum_{k=2}^{N_c} c_k\,\mathbf{t}_k. \tag{16}$$

Looking to the obtained system, the difference between the turbulent-like and the turbulent viscosity consists in their positioning with respect to the divergence operator. Indeed the former is located outside the divergence, i.e. $\nu_{tl}\nabla\cdot(\nabla\mathbf{u})$, while the former inside it, i.e. $\nabla\cdot(\nu_t\nabla\mathbf{u})$, see (1) and (2).

In general, it is not guaranteed that $\nu_{tl} > 0$ (corresponding to $c_1 < 0$) holds everywhere. Let us write $\nu_{tl} = \nu_{tl}^{+} + \nu_{tl}^{-}$ where $\nu_{tl}^{+}(x) = \max(\nu_{tl}, 0)$ is the positive part of the turbulent-like viscosity. Finally, let define

$$(\nabla\cdot\boldsymbol{\tau})^{\dagger} := -\nu_{tl}^{-}\nabla\cdot\mathbf{S} + \frac{\varepsilon}{k^{1/2}}\sum_{k=2}^{N_c} c_k\,\mathbf{t}_k. \tag{17}$$

The final RANS system with Implicit-Explicit treatment reads

$$\begin{cases} \nabla\cdot\mathbf{u} = 0, \\ \dfrac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - (\nu + \nu_{tl}^{+})\Delta\mathbf{u} = -\nabla p - (\nabla\cdot\boldsymbol{\tau})^{\dagger}, \end{cases} \tag{18}$$

where the term associated to ν_{tl}^{+} is treated implicitly into the diffusion term while the term $(\nabla\cdot\boldsymbol{\tau})^{\dagger}$ is treated explicitly.

More details about the implementation of the Implicit-Explicit treatment in OpenFOAM are given in Appendix.

IV. NUMERICAL RESULTS

This section discusses about the application of the VBNN into two classical benchmark flows: the flow in a square duct and the flow over periodic hills. As it will be discussed, standard RANS models fails in the description of the velocity field in these configurations.

The VBNN is implemented and trained in Python using the Tensorflow package[40] while all the RANS computations are performed with the Finite Volume Method based OpenFOAM opensource code[41].

A. Flow in a square duct

1. Dataset

DNS data[6] at several bulk Reynolds numbers Re_b have been used. The simulations with $Re_b = 2200, 2600, 2900$ are employed for training purposes. In particular, 80% of the data are used for training while the remaining 20% for validation. The flow at $Re_b = 3500$ is employed to test the network prediction ability. The test flow Reynolds number is higher than the training ones to

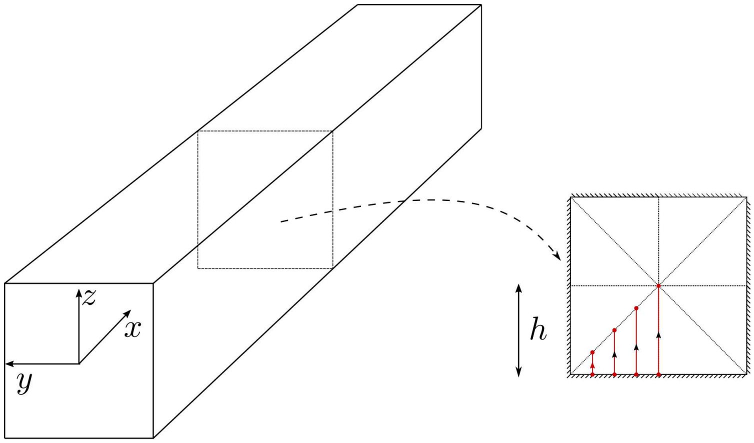
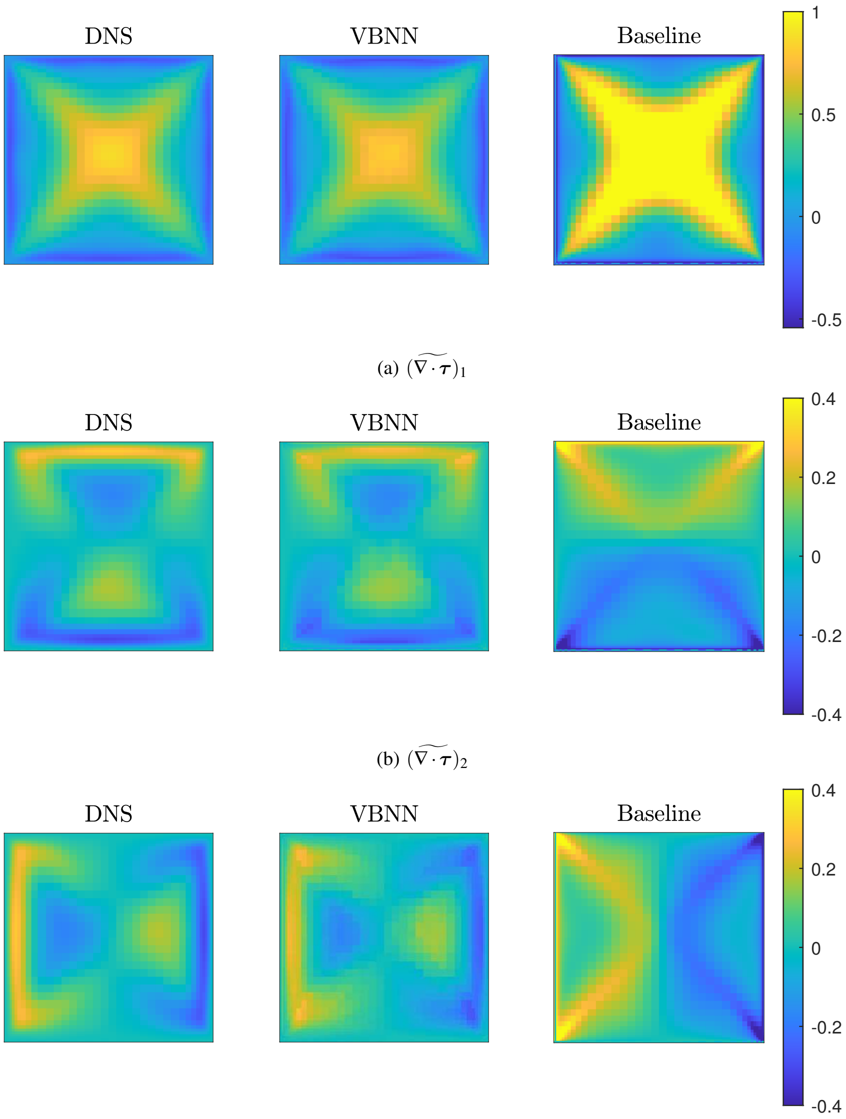


FIG. 2. Square duct domain.

analyse the extrapolation property of the VBNN. This particular flow is (in average) stationary and uniform across the main streamwise direction. Only the data coming from three square sections in the central region of the duct are used to reduce considerably the training effort. Figure 2 shows the domain and one square section. The obtained dataset counts roughly $2 \cdot 10^4$ simulation cells. The Launder, Reece and Rodi (LRR) RSTM[29] is used as RANS model with a low-Reynolds number approach at wall (with the first mesh layer thickness such that the maximum value of y^+ is around 1). In the following, this turbulence model will be referred as Baseline.

2. *Results analysis*

Figure 3 compares the components of the vector $\widetilde{\nabla \cdot \boldsymbol{\tau}}$ obtained from DNS, VBNN and Baseline model respectively. Regarding the first component, the VBNN is in agreement with the DNS both qualitatively and quantitatively while the Baseline overpredicts it in the center and along the diagonals of the square section. Regarding the second and third components, the Baseline have positive and negative values located in two separated square section's halves. In addition maxima and minima are overestimated in absolute value. On the other hand, the VBNN describes correctly the values of these components and where these are positive and negative. However, VBNN predicts in few cells near the corners maxima or minima that are not in the DNS.



(a) $(\widetilde{\nabla\cdot\boldsymbol{\tau}})_1$

(b) $(\widetilde{\nabla\cdot\boldsymbol{\tau}})_2$

(c) $(\widetilde{\nabla\cdot\boldsymbol{\tau}})_3$

FIG. 3. Comparison between first (a), second (b) and third (c) components of $\widetilde{\nabla\cdot\boldsymbol{\tau}}$ from DNS (on the left), VBNN (in the middle) and RSTM Baseline one (on the right).

TABLE I. Root Mean Square Error (RMSE) of VBNN and Baseline models using $\widetilde{\nabla\cdot\boldsymbol{\tau}}^{\text{DNS}}$ as reference.

model	RMSE
VBNN	0.32 e-1
Baseline	2.43 e-1

Table I shows the Root Mean Square Error (RMSE) defined as

$$\text{RMSE} = \sqrt{\frac{1}{3N_{cells}}\sum_{i=1}^{N_{cells}} ||\widetilde{\nabla\cdot\boldsymbol{\tau}_i}^{\text{DNS}} - \widetilde{\nabla\cdot\boldsymbol{\tau}_i}^{\text{model}}||^2}, \tag{19}$$

where N_{cells} is the number of cells in the RANS square section grid. This metric quantitatively measures the distance between the DNS dimensionless target and the turbulence model ones. The Baseline RMSE is one order of magnitude higher than the VBNN.

Among all the predicted coefficients in the expansion (7), the first one plays a key role in the conditioning of the RANS system. In particular, the more extended are the regions with a negative predicted first coefficient (and consequently positive ν_{tl}), the better conditioned is the system. Figure 4 shows the ratio between the turbulent-like viscosity ν_{tl}, defined in (15), and the kinematic viscosity ν. The ratio is positive in the majority of the square section with values bigger than 8 frequently occurring. The negative regions are very limited and located on the square diagonals near the corners. The minimum value of the ratio is lower that -1, in particular $\min(\nu_{tl}/\nu) = -1.70$. If the turbulent-like viscosity was treated completely implicitly, the total viscosity associated to the laplacian operator in (16) would be negative in some regions. This observation justifies the splitting of ν_{tl} into its positive and negative part being the former only treated implicitly.

The obtained data-driven $\nabla\cdot\boldsymbol{\tau}$ is successively inserted into the RANS solver to obtain new steady fields. Figure 5 shows the magnitude of the secondary motion $||(u_y,u_z)^T||_2/u_b$ (assuming the streamwise velocity aligned to the x axis), where u_b is the bulk velocity. Lighter colors correspond to higher values of the norm. The different resolution between the models is due to the grid density, being the DNS one much finer than the VBNN and Baseline one (the same grid is employed for both models). Even if the Baseline model describes correctly the regions where the secondary motion is more prominent, it drastically overpredicts it. On the other hand, the VBNN secondary motion is still overpredicted, but its magnitude is in between the DNS case and the Base-

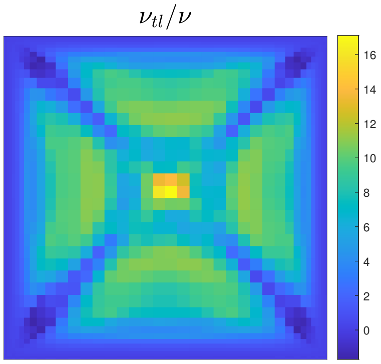


FIG. 4. Ratio between turbulent-like viscosity ν_{tl} and kinematic viscosity ν.

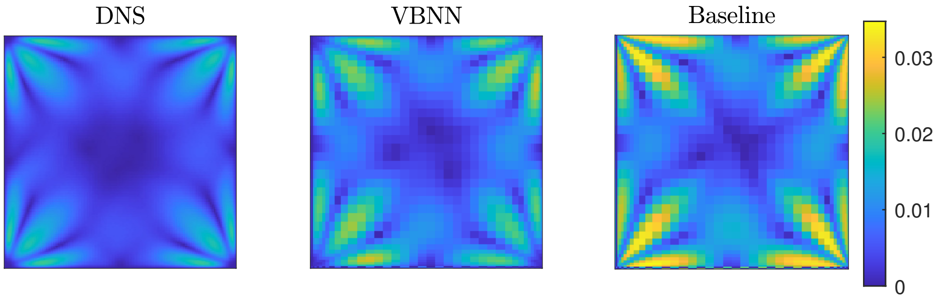


FIG. 5. Magnitude of the secondary motion in the DNS (left), the VBNN (middle) and the Baseline model (right). Lighter colors correspond to higher magnitudes of the secondary flow.

line one. To make a quantitative comparison, Table II reports the values of $\max(||(u_y,u_z)^T||_2)/u_b$, and the relative amplification using the DNS value as reference. The VBNN approach reduces the overestimation from 70% of the Baseline model to 30%. Finally, the VBNN secondary motion is characterized by symmetry (up to numerical discrepancies in the central and on the peaks regions) with respect to square section diagonals, while the Baseline case does not correctly respect the symmetry.

Figure 6 shows the velocity components profiles along the red lines defined in Figure 2 in the lower-left square section quadrant. The u_y has been flipped of sign to make comparison with literature curves[26,39] easier. The Baseline model correctly describes the main motion u_x. However, it overpredicts the magnitudes of both secondary motion's velocity components. On the other

TABLE II. Maxima of the secondary motion norm and corresponding amplification factor

model	$\max(\|\|(u_y, u_z)^T\|\|_2)/u_b$	$\frac{\max(\|\|(u_y,u_z)^T\|\|_2)}{\max(\|\|(u_y^{\mathrm{DNS}},u_z^{\mathrm{DNS}})^T\|\|_2)}$
DNS	2.04 e-2	1
VBNN	2.65 e-2	1.30
Baseline	3.47 e-2	1.70

hand, the VBNN model correctly predicts the secondary motion close to the corners, i.e. for $y/h = 0.25$, where the profiles almost overlaps the DNS ones without losing accuracy on the main flow. Small improvements of the secondary motion are also noticeable for the other curves, in particular near wall for u_y curves and far from wall for the u_z ones.

3. *Role of the Implicit-Explicit treatment*

Figure 7 compares the secondary motion obtained with the Implicit-Explicit treatment of the divergence of the RST and with the totally Explicit one. Even if the latter damps the magnitude of the motion as desired, it unphysically breaks the symmetries. In addition, the Explicit simulation takes an order of magnitude more time steps to reach the steady state. In general, a convergence speed up of implicit treatments is usually observed[35].

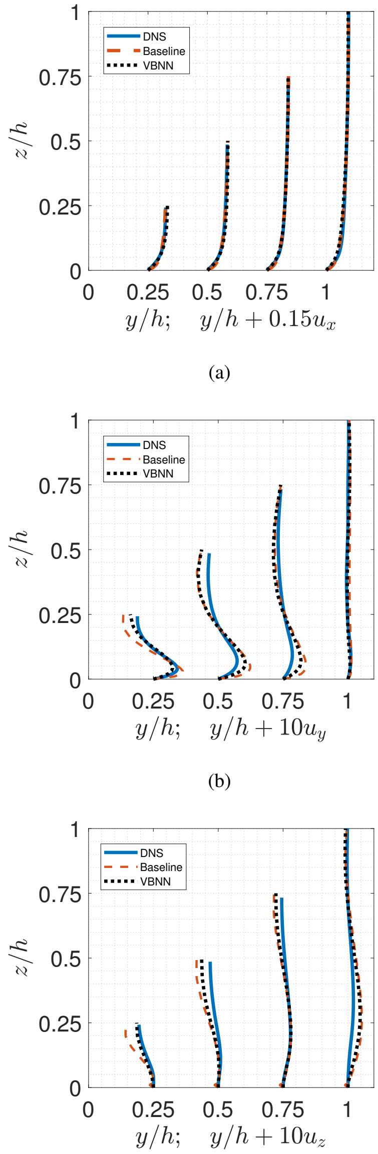
It is important to highlight that differences between the two fields are uniquely due to the treatment of the divergence of the RST into the equations. As a matter of fact, the same $\nabla\cdot\boldsymbol{\tau}$ field is fed into the equations.

4. *Role of the dependencies choice*

In this section we test also the dependences hypothesis (6) by choosing the simpler relation

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \mathbf{f}(\mathbf{s}, \mathbf{w}, \widetilde{\nabla\cdot\mathbf{S}}, Re_d), \tag{20}$$

i.e. by removing the dependence on $\widetilde{\nabla k}$. In this case the basis consists of 6 elements while the invariants are 14. This hypothesis still let possible the Implicit-Explicit treatment of the RANS system because $\widetilde{\nabla\cdot\mathbf{S}}$ is still a basis vector. This would not be the case if $\widetilde{\nabla\cdot\mathbf{S}}$ were excluded from



(a)

(b)

(c)

FIG. 6. Primary (a) and secondary motion (b-c) velocity components along the red lines defined in the square section in Figure 2. The sign of u_y is changed compared to the coordinates defined in Figure 2 to make comparison with literature[26,39] easier.

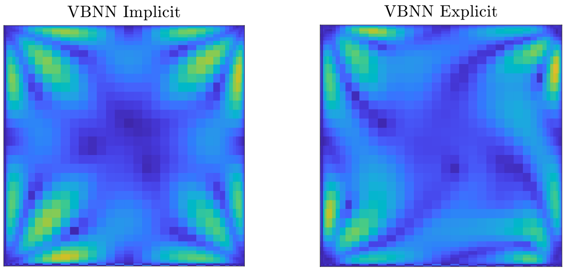


FIG. 7. Comparison of the magnitude of the secondary motion in case of Implicit-Explicit (left) or purely Explicit (right) treatment of $\nabla\cdot\boldsymbol{\tau}$. The same colormap as in Figure 5 is used.

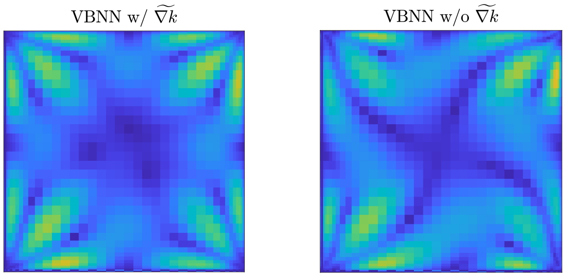


FIG. 8. Magnitude of the secondary motion: on the left the field assuming a dependence on $\widetilde{\nabla k}$, on the right the field not assuming a dependence on $\widetilde{\nabla k}$. The same colormap as in Figure 5 is used.

the dependencies in (6). Considering the results of the previous section, the analysis without $\widetilde{\nabla\cdot\mathbf{S}}$ is not carried out.

The RMSE error in this case is $0.37\cdot10^{-1}$, bigger that the corresponding value in Table I. This behavior is expected because a smaller vector basis and a smaller set of invariants are considered.

Figure 8 shows the secondary motion in the two cases. Even when $\widetilde{\nabla k}$ is not considered, the secondary motion is still correctly damped compared to the Baseline case. However, the field loses its symmetry with respect to the square diagonals.

The lack of symmetry of the secondary motion incurred both in the Explicit treatment and by removing $\widetilde{\nabla k}$ from the dependencies can compromise the symmetry of the main motion by inducing a non-physical rotation. This phenomenon is confirmed in Figure 9.

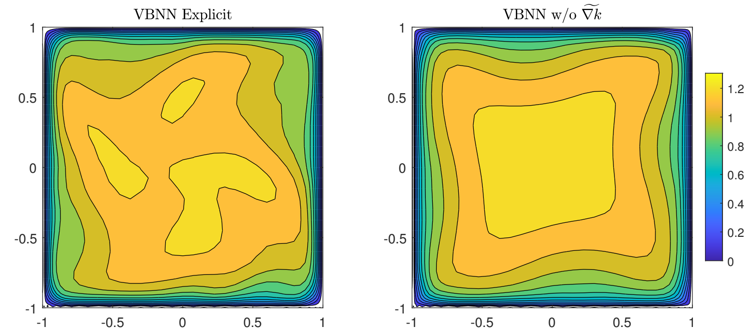


FIG. 9. Main motion u_x/u_b with the Explicit treatment and by removing $\widetilde{\nabla k}$ from the dependencies of $\widetilde{\nabla \cdot \tau}$.

5. *Role of the baseline RANS model*

Until now, the Launder, Reece and Rodi RSTM has been used as starting Baseline RANS model to generate the datased to train the VBNN. While it has been shown that this model overestimates the intensity of the secondary motion, it still predicts this phenomenon. In this section, the Launder and Sharma (LS) $k-\varepsilon$ linear model[42] is used. For the sake of brevity, only the *a posteriori* results concerning the velocity field are shown.

Figure 10 compares the intensity of the secondary motion of the DNS, VBNN and Baseline models. As expected, the Baseline model does not predict secondary motion. On the other hand, the VBNN model predicts it even if underestimated. Analogously to the LRR model case, the intensity of the VBNN model is in between the Baseline and DNS case.

Table III reports the values of $\max(||(u_y,u_z)^T||_2)/u_b$, and the relative attenuation using the DNS value as reference. It confirms the absence of the secondary motion in the Baseline case. The maximum intensity in the VBNN case is 81% the DNS one.

Figure 11 shows the velocity profiles of the primary and secondary flow along the red lines defined in Figure 2. The u_x profiles of both the Baseline and VBNN models are in agreement with the DNS. As already observed, the Baseline model does not predict any secondary flow while the VBNN predicts non null u_y and u_z profiles. Regarding u_y, the VBNN underestimates the maximum values and slightly overestimates the minimum values at $y/h = 0.25$ and $y/h = 0.5$. The

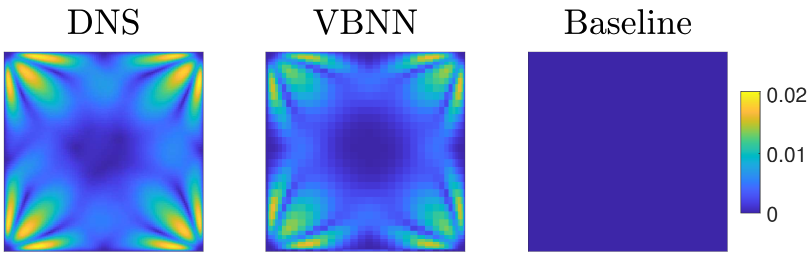


FIG. 10. Magnitude of the secondary motion in the DNS (left), the VBNN (middle) and the LS $k-\varepsilon$ Baseline model (right). Lighter colors correspond to higher magnitudes of the secondary flow.

TABLE III. Maxima of the secondary motion norm and corresponding amplification factor when using the LS $k-\varepsilon$ as baseline model.

model	$\max(\|\|(u_y,u_z)^T\|\|_2)/u_b$	$\frac{\max(\|\|(u_y,u_z)^T\|\|_2)}{\max(\|\|(u_y^{\mathrm{DNS}},u_z^{\mathrm{DNS}})^T\|\|_2)}$
DNS	2.04 e-2	1
VBNN	1.65 e-2	0.81
Baseline	3.38 e-9	1.66 e-7

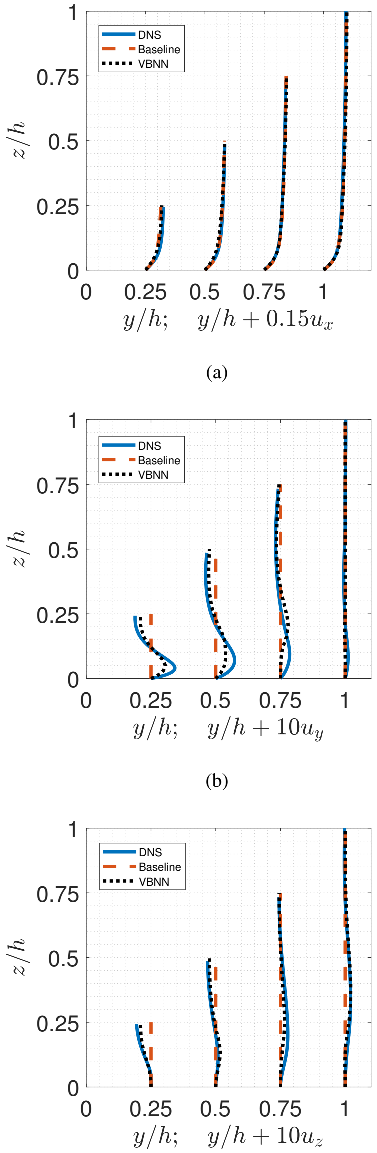
maximum at $y/h = 0.75$ is predicted slightly to far from wall. Regarding u_z, the VBNN profiles are in agreement with the DNS values.

B. Flow over periodic hills

1. *Dataset*

The DNS data[10] have been used with different hills geometries but fixed bulk Reynolds number $Re_b = 5600$ are available. Figure 12 shows the different steepness associated to the parameter α. The smaller is α, the steeper is the hill profile. All lengths are set dimensionless dividing by the hill high h.

The flows with $\alpha = 0.5, 0.8, 1.2, 1.5$ have been employed for the training (with the same split-



(a)

(b)

(c)

FIG. 11. Primary (a) and secondary motion (b-c) velocity components along the red lines defined in the square section in Figure 2. The Baseline model is the LS $k-\varepsilon$ linear model. The sign of u_y is changed compared to the coordinates defined in Figure 2 to make comparison with literature[26,39] easier.

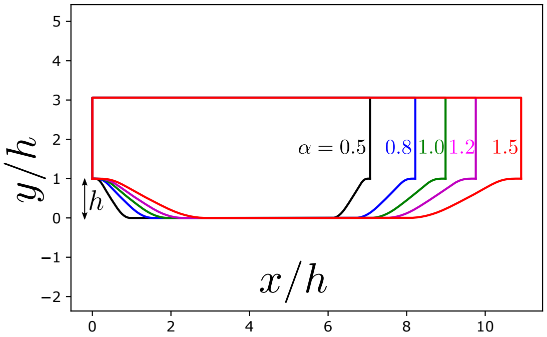


FIG. 12. Periodic hills shapes with respect to the α slope parameter.

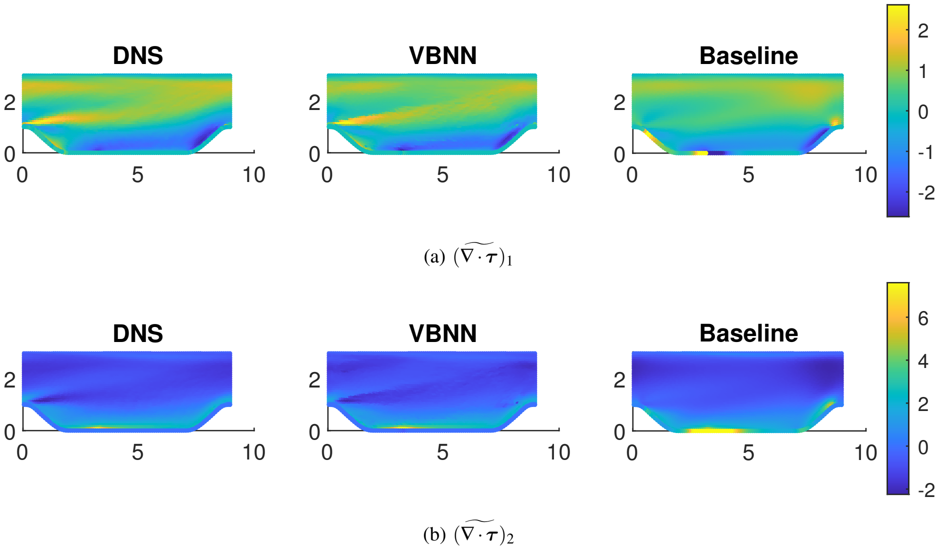
ting $80\% - 20\%$ for training and validation) while the case with $\alpha = 1.0$ is used for testing only. The RANS simulations have been performed with a 2D domain. The cardinality of the training dataset is about $5.9 \cdot 10^4$ cells that is the cells number in the 2D RANS domain. It has been observed that a dropout regularisation method[43] with drop probability of 0.1 was helpful in the training and consequently adopted. The Launder and Sharma $k - \varepsilon$ linear model is employed as Baseline RANS model. The analysis with the LRR model has not been performed due to convergence issues. All computational details can be found in Xiao et al.[10] where the OpenFOAM case is available.

2. *Results analysis*

Figure 13 shows the first two components only of $\widetilde{\nabla \cdot \boldsymbol{\tau}}$ for the DNS, VBNN and Baseline models, being the third component zero (the VBNN correctly predicts it). The VBNN components are in agreement with the reference ones while this is not true for the Baseline model.

For the first component, VBNN provides a correct description of the maxima loci that starts from the crest of the front hill. It also predicts the limited region of local maximum at the middle-end of the first hill and the following minimum. The Baseline case has wrong, both in location and values, maxima and minima in the left part of the domain. It predicts correctly the minima region on the second hill (well predicted by the VBNN model too). However, there is a wrong maximum on the top of the second hill.

Regarding the second component, the VBNN model describes correctly the value and the ex-



(a) $(\widetilde{\nabla \cdot \boldsymbol{\tau}})_1$

(b) $(\widetilde{\nabla \cdot \boldsymbol{\tau}})_2$

FIG. 13. Comparison between first (a) and second (b) components of $\widetilde{\nabla \cdot \boldsymbol{\tau}}$ from the DNS (on the left), the components obtained with the VBNN (in the middle) and the RSTM Baseline one (on the right). The third component is not shown because uniformly zero.

FIG. 14. Ratio between turbulent-like viscosity ν_{tl} and kinematic viscosity ν.

tension of the maximum located at $x/h \approx 3.5,\ y/h = 0$ and the Baseline model overpredicts both aspects. Finally the VBNN case is characterized by the minima region that starts from the first hill crest.

Figure 14 shows ν_{tl}/ν to understand the relevance of the Implicit-Explicit treatment. The ratio is positive in the majority of the domain with the exception of the two region immediately above the hills, predominantly above the rear one. In particular, this quantity assumes values $O(10^2)$ with maximum of about 800. This behavior is helpful for the conditioning of the system. It has been observed that the dropout regularization helps in reducing the regions with negative turbulent-like viscosity.

Figure 15 represents the horizontal velocity profiles along the vertical lines at $x/h = c$ with $c = 0,\dots,8$. The profiles are obtained once the simulation reaches the steady state with $\nabla \cdot \boldsymbol{\tau}$ coming from the VBNN model. Figure 15(a) shows the whole domain while Figure 15(b) focuses on the first hill downstream wall region and Figure 15(c) depicts the middle top wall region.

Generally speaking, the VBNN curves are closer to the DNS ones compared to the Baseline ones. This behaviour is observable in the whole computational domain. It is worth mentioning that the VBNN model predicts, even if underestimated, the local maximum of the horizontal velocity on the crest of the first hill ($x/h = 0$, $y/h = 1$). This behaviour is not captured by the Baseline model for which u_x monotonically increases until $y/h \approx 2.8$.

Figure 15(b) shows that the Baseline model predicts almost null reversal flow downstream the first hill and underestimates the size of the recirculation region. These are expected errors when simulating the flow over periodic hills with linear $k-\varepsilon$ models[44]. On the other hand, the VBNN simulation predicts reversal flow quantitatively close to the DNS case for $x/h = 1$ and $x/h = 2$. VBNN still predicts reversal flow, even if underestimated, for $x/h = 3$. However, at $x/h = 4$ the VBNN horizontal velocity is positive near wall while the DNS one is still negative.

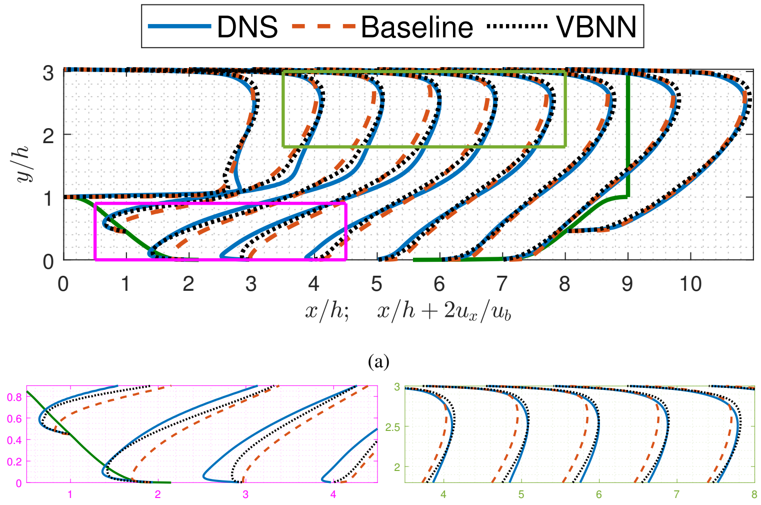
Figure 15(c) represents the maxima of the curves that are located near the upper wall ($y/h = 3.036$). The maxima are slightly but constantly underestimated by the Baseline model. Conversely, the VBNN curves are very close to the DNS ones and do not suffer of the underestimation issue.

Figure 16(a) represents the vertical velocity profiles along the vertical lines at $x/h = c$ with $c = 0,\dots,8$, while Figure 16(b) focuses on the $x/h = 1$ line near the hill. VBNN leads to improvements compared to the Baseline $k-\varepsilon$ model, even if less prominent than the u_x case. The region with best improved accuracy is the $x/h = 1$ line near the hill wall represented in Figure 16(b). The VBNN model predicts a positive velocity region close to the DNS one while the $k-\varepsilon$ turbulence model predicts a positive velocity in a smaller region. In particular, the vertical velocity goes from negative to positive at $y/h \approx 0.93$ in the DNS, $y/h \approx 0.86$ in the VBNN and $y/h \approx 0.70$ in the $k-\varepsilon$ model.

Finally, it is worth noting that an analysis of the effects of the Explicit treatment of $\widetilde{\nabla\cdot\boldsymbol{\tau}}$ on the simulated flow has not been possible due to instability issues.

V. CONCLUSIONS

The present paper proposes a new data-driven turbulence model to close and increase accuracy of the RANS equations. Hence the model predicts the divergence of the Reynolds Stress Tensor, usually called Reynolds Force Vector[27]. This target vectorial quantity is obtained through a Neural Network that predicts the coefficients of a vector basis expansion. These coefficients are



(a)

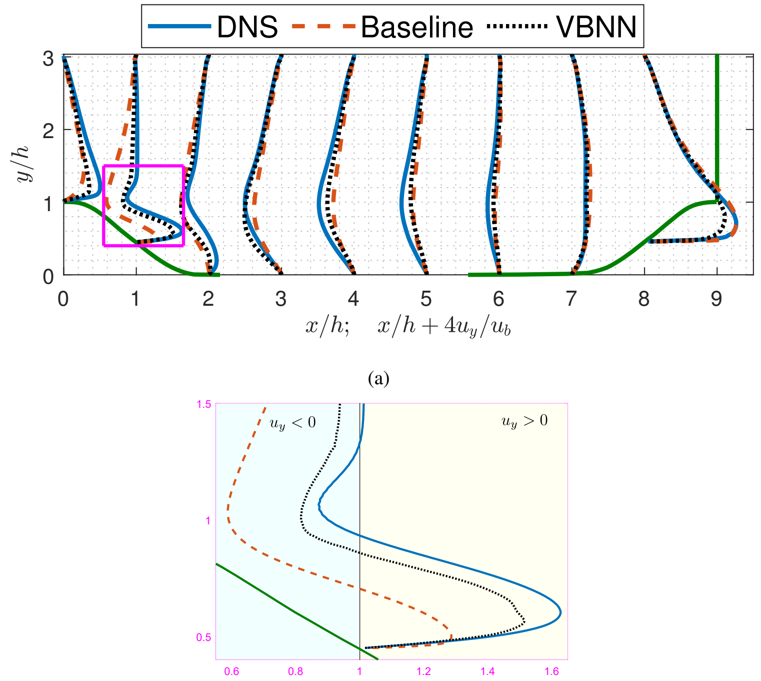
(b)

(c)

FIG. 15. u_x profiles comparison between DNS, VBNN and Baseline cases (a). Zoom downstream the first hill crest in magenta box in figure above (b). Zoom in the middle top wall region in green box in figure above (c).

functions of invariant scalar quantities that depend on the averaged fields. The vector basis and the invariants are uniquely defined once a list of dependencies of the target function is made. A possible dependencies choice is discussed starting from a well known assumption in literature for the Reynolds Stress Tensor anisotropy. Moreover, the architecture of the trained network and the invariants choice guarantee both Galilean and coordinates-frame rotation invariances. In addition, this approach closes directly the RANS system and does not require any coupling with classic turbulence models. An implicit treatment of the first term of the expansion is proposed to increase the conditioning of the RANS system.

The proposed model is tested for the flow in a square duct and the flow over periodic hills. Both flows, despite their geometrical simplicity, present features that classic turbulence models do not describe correctly, in particular the secondary flow for the former and the recirculation flow downstream the first hill in the latter. The duct flow case is tested using a classic Launder and Sharma $k-\varepsilon$ linear closure and the Launder, Reece and Rodi Reynolds Stresses Transport Model. The periodic hills case is tested using the Launder and Sharma $k-\varepsilon$ linear closure. The



(a)

(b)

FIG. 16. u_y profiles comparison between DNS, VBNN and Baseline cases (a). Zoom downstream the first hill crest at $x/h = 1$ in magenta box in figure above (b).

data-driven model qualitatively and quantitatively outperforms classic turbulence models in both scenarios.

ACKNOWLEDGEMENTS

D. Oberto wants to thank also M. Pintore and F. Della Santa for the precious advices on the neural networks implementation and A. Giammarini for the discussion on invariance properties. The authors performed this research in the framework of the Italian MIUR Award "Dipartimenti di Eccellenza 2018-2022" granted to the Department of Mathematical Sciences, Politecnico di Torino (CUP: E11G18000350001). S.B. has been supported by the Italian MIUR PRIN Project 20204LN5N5_003 and D.O. by the Italian MIUR PRIN Project 201752HKH8_003. The authors are members of the Italian INdAM-GNCS research group.

Appendix A: Dependencies of $\widetilde{\nabla\cdot\boldsymbol{\tau}}$

Starting from the definition of RST anisotropy and its expansion in (3), one could write

$$\boldsymbol{\tau} = 2k\Big[\sum_{j=1}^{10} c_j(\lambda_1,\dots,\lambda_6)\ \mathbf{T}_j + \frac{1}{3}\mathbf{I}\Big]. \tag{A1}$$

It is possible to compute the divergence of the above expression as

$$\begin{aligned}\nabla\cdot\boldsymbol{\tau} =& \underbrace{2\sum_{j=1}^{10} c_j(\lambda_1,\dots,\lambda_6)[\nabla k]^T\mathbf{T}_j}_{\alpha} + \underbrace{2\sum_{j=1}^{10} k[\nabla c_j(\lambda_1,\dots,\lambda_6)]^T\mathbf{T}_j}_{\beta} +\\ & \underbrace{2\sum_{j=1}^{10} kc_j(\lambda_1,\dots,\lambda_6)\nabla\cdot\mathbf{T}_j}_{\gamma} + \underbrace{\frac{2}{3}\nabla k}_{\delta}.\end{aligned} \tag{A2}$$

Let us suppose that both the tensor basis expressed in (5) and the turbulent kinetic energy are known (the standard approach is to obtain them from a RANS simulation). While the terms α,γ,δ can be treated by a neural network that obtains scalar coefficients only, the term β contains the gradient of the unknowns that are vector quantities. Thus, the expression (A2) can not be directly used while preserving coordinates-frame rotation invariance. Nonetheless, it can be used as a guideline to write down a new constitutive law for $\nabla\cdot\boldsymbol{\tau}$ or its dimensionless counterpart $\widetilde{\nabla\cdot\boldsymbol{\tau}} = \frac{k^{1/2}}{\varepsilon}\nabla\cdot\boldsymbol{\tau}$. It has been decided to predict the latter to be as close as possible to the TBNN approach in[19] where the dimensionless Reynolds Stress Tensor anisotropy is predicted.

From the above computations, it seems natural to suppose dependences of $\widetilde{\nabla\cdot\boldsymbol{\tau}}$ from $\mathbf{s}$, $\mathbf{w}$, their respective divergences and ∇k. It is worth noting that the divergence of each tensors $\{\mathbf{T}_j\}$ in (A2) involves multiplications of $\mathbf{s}$, $\mathbf{w}$ and their respective divergences.

In order to work with dimensionless quantities only, analogously to Ling et al.[19], it has been decided to make the following assumption:

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \mathbf{f}(\mathbf{s},\mathbf{w},\widetilde{\nabla\cdot\mathbf{S}},\widetilde{\nabla\cdot\mathbf{W}},\widetilde{\nabla k}), \tag{A3}$$

where $\mathbf{s} = \frac{k}{\varepsilon}\mathbf{S}$, $\mathbf{w} = \frac{k}{\varepsilon}\mathbf{W}$, $\widetilde{\nabla\cdot\mathbf{S}} = \frac{k^{5/2}}{\varepsilon^2}\nabla\cdot\mathbf{S}$, $\widetilde{\nabla\cdot\mathbf{W}} = \frac{k^{5/2}}{\varepsilon^2}\nabla\cdot\mathbf{W}$ and $\widetilde{\nabla k} = \frac{k^{1/2}}{\varepsilon}\nabla k$ are the dimensionless counterparts of the symmetric part of the velocity gradient $\mathbf{S}$, the antisymmetric part of the velocity gradient $\mathbf{W}$, the divergence of $\mathbf{S}$, the divergence of $\mathbf{W}$ and the gradient of k, respectively.

The list of dependences (A3) can be simplified because $\nabla\cdot\mathbf{S} = \nabla\cdot\mathbf{W} = \frac{1}{2}\Delta\mathbf{u}$ from the Schwarz theorem and the incompressibility assumption. Indeed, if $\mathbf{u}$ is sufficiently regular, it holds

$$\begin{aligned}\nabla\cdot\mathbf{S} &= \frac{\partial}{\partial x_j}\frac{1}{2}\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right] = \frac{1}{2}\left[\frac{\partial^2 u_i}{\partial x_j^2}+\frac{\partial}{\partial x_i}\frac{\partial u_j}{\partial x_j}\right] = \frac{1}{2}\frac{\partial^2 u_i}{\partial x_j^2}\\ \nabla\cdot\mathbf{W} &= \frac{\partial}{\partial x_j}\frac{1}{2}\left[\frac{\partial u_i}{\partial x_j}-\frac{\partial u_j}{\partial x_i}\right] = \frac{1}{2}\left[\frac{\partial^2 u_i}{\partial x_j^2}-\frac{\partial}{\partial x_i}\frac{\partial u_j}{\partial x_j}\right] = \frac{1}{2}\frac{\partial^2 u_i}{\partial x_j^2}.\end{aligned} \tag{A4}$$

As a consequence, the constitutive assumption can be simplified to

$$\widetilde{\nabla\cdot\boldsymbol{\tau}} = \mathbf{f}(\mathbf{s},\mathbf{w},\widetilde{\nabla\cdot\mathbf{S}},\widetilde{\nabla k}). \tag{A5}$$

In this work, a dependence from $\nabla\cdot\mathbf{S}$ and not from $\nabla\cdot\mathbf{s}$ (both to be made dimentionless) is supposed to make the implicit treatment of the first expansion term straightforwardly.

Finally, as remarked by several authors[26,34], any other scalar quantity can be included in the constitutive assumption without changing the coordinates-frame rotation property. In particular, in our work we assume an additional dependence from the wall-distance based Reynolds number Re_d.

Appendix B: Implicit-Explicit treatment in OpenFOAM

The system (18) does not require any coupling with a turbulence model and can theoretically be solved in OpenFOAM with a laminar solver like `icofoam`. The explicit term $(\nabla\cdot\boldsymbol{\tau})^\dagger$ is easy to implement because it is sufficient to define a new solver starting from an existing one by adding a fixed source term into the momentum equation.

The implicit term is less trivial to implement. The field ν_{tl}^+ cannot be defined as a uniform field as the kinematic viscosity. Therefore, it has been decided to implement a "fake" turbulence model that passes the same ν_{tl}^+ field at each solver iteration. Thus, the `simpleFoam` solver is used. To the best knowledge of the authors, OpenFOAM is coded to deal with classic turbulent viscosities that are inside the divergence operator. In order to modify the code as less as possible, it has been decided to solve for an equivalent momentum equation that reads

$$\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}-\nabla\cdot[(\nu+\nu_{tl}^+)\nabla\mathbf{u}] = -\nabla p-(\nabla\cdot\boldsymbol{\tau})^\dagger-\nabla\mathbf{u}\,\nabla\nu_{tl}^+. \tag{B1}$$

REFERENCES

[1] T. Craft, B. Launder, and K. Suga, “Development and application of a cubic eddy-viscosity model of turbulence,” International Journal of Heat and Fluid Flow **17**, 108–115 (1996).

[2]T. A. Oliver and R. D. Moser, "Bayesian uncertainty quantification applied to RANS turbulence models," Journal of Physics: Conference Series **318**, 042032 (2011).

[3]R. D. Moser, J. Kim, and N. N. Mansour, "Direct numerical simulation of turbulent channel flow up to Reτ=590," Physics of Fluids **11**, 943–945 (1999).

[4]H. Abe, H. Kawamura, and Y. Matsuo, "Direct numerical simulation of a fully developed turbulent channel flow with respect to the reynolds number dependence," Journal of Fluids Engineering **123**, 382–393 (2001).

[5]S. Pirozzoli, J. Romero, M. Fatica, R. Verzicco, and P. Orlandi, "One-point statistics for turbulent pipe flow up to," Journal of Fluid Mechanics **926** (2021), 10.1017/jfm.2021.727.

[6]A. Pinelli, M. Uhlmann, A. Sekimoto, and G. Kawahara, "Reynolds number dependence of mean flow structure in square duct turbulence," Journal of Fluid Mechanics **644**, 107–122 (2010).

[7]H. Zhang, F. X. Trias, A. Gorobets, Y. Tan, and A. Oliva, "Direct numerical simulation of a fully developed turbulent square duct flow up to $Re\tau = 1200$," International Journal of Heat and Fluid Flow **54**, 258–267 (2015).

[8]S. Pirozzoli, D. Modesti, P. Orlandi, and F. Grasso, "Turbulence and secondary motions in square duct flow," Journal of Fluid Mechanics **840**, 631–655 (2018).

[9]M. Breuer, N. Peller, C. Rapp, and M. Manhart, "Flow over periodic hills – numerical and experimental study in a wide range of reynolds numbers," Computers & Fluids **38**, 433–457 (2009).

[10]H. Xiao, J.-L. Wu, S. Laizet, and L. Duan, "Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations," Computers & Fluids **200**, 104431 (2020).

[11]F. X. Trias, A. Gorobets, and A. Oliva, "Turbulent flow around a square cylinder at reynolds number 22,000: A DNS study," Computers & Fluids **123**, 87–98 (2015).

[12]A. Cimarelli, A. Leonforte, and D. Angeli, "Direct numerical simulation of the flow around a rectangular cylinder at a moderately high reynolds number," Journal of Wind Engineering and Industrial Aerodynamics **174**, 39–49 (2018).

[13]A. Chiarini and M. Quadrio, "The turbulent flow over the BARC rectangular cylinder: A DNS study," Flow, Turbulence and Combustion **107**, 875–899 (2021).

[14]J. N. Kutz, "Deep learning in fluid dynamics," Journal of Fluid Mechanics **814**, 1–4 (2017).

[15]K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," Annual

Review of Fluid Mechanics **51**, 357–377 (2019), https://doi.org/10.1146/annurev-fluid-010518-040547.

[16] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annual Review of Fluid Mechanics **52**, 477–508 (2020).

[17] R. Vinuesa and S. L. Brunton, "The potential of machine learning to enhance computational fluid dynamics," (2021), arXiv:2110.02085 [physics.flu-dyn].

[18] J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," Journal of Computational Physics **318**, 22–35 (2016).

[19] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," Journal of Fluid Mechanics **807**, 155–166 (2016).

[20] S. B. Pope, "A more general effective-viscosity hypothesis," Journal of Fluid Mechanics **72**, 331 (1975).

[21] R. Fang, D. Sondak, P. Protopapas, and S. Succi, "Neural network models for the anisotropic reynolds stress tensor in turbulent channel flow," Journal of Turbulence **21**, 525–543 (2019).

[22] H. S. de Ocáriz Borde, D. Sondak, and P. Protopapas, "Convolutional neural network models and interpretability for the anisotropic reynolds stress tensor in turbulent one-dimensional flows," Journal of Turbulence , 1–28 (2021).

[23] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," Physics of Fluids **33**, 055133 (2021).

[24] J. Steiner, R. P. Dwight, and A. Viré, "Data-driven RANS closures for wind turbine wakes under neutral conditions," Computers & Fluids **233**, 105213 (2022).

[25] J.-X. Wang, J.-L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on DNS data," Physical Review Fluids **2**, 034603 (2017).

[26] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," Physical Review Fluids **3**, 074602 (2018).

[27] M. A. Cruz, R. L. Thompson, L. E. Sampaio, and R. D. Bacchi, "The use of the reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling," Computers & Fluids **192**, 104258 (2019).

[28] B. E. Launder, G. J. Reece, and W. Rodi, "Progress in the development of a reynolds-stress turbulence closure," Journal of Fluid Mechanics **68**, 537–566 (1975).

[29]M. M. Gibson and B. E. Launder, "Ground effects on pressure fluctuations in the atmospheric boundary layer," Journal of Fluid Mechanics **86**, 491–511 (1978).

[30]C. G. Speziale, S. Sarkar, and T. B. Gatski, "Modelling the pressure–strain correlation of turbulence: an invariant dynamical systems approach," Journal of Fluid Mechanics **227**, 245–272 (1991).

[31]T. B. Gatski and C. G. Speziale, "On explicit algebraic stress models for complex turbulent flows," Journal of Fluid Mechanics **254**, 59–78 (1993).

[32]B. Basara and S. Jakirlic, "A new hybrid turbulence modelling strategy for industrial CFD," International Journal for Numerical Methods in Fluids **42**, 89–116 (2003).

[33]Q.-S. Zheng, "Theory of representations for tensor functions—a unified invariant approach to constitutive equations," Applied Mechanics Reviews **47**, 545–587 (1994).

[34]P. M. Milani, J. Ling, and J. K. Eaton, "Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling," Journal of Fluid Mechanics **906** (2020), 10.1017/jfm.2020.820.

[35]B. P. Brener, M. A. Cruz, R. L. Thompson, and R. P. Anjos, "Conditioning and accurate solutions of reynolds average navier–stokes equations with data-driven turbulence closures," Journal of Fluid Mechanics **915** (2021), 10.1017/jfm.2021.148.

[36]E. F. Fonseca, V. B. Rangel, B. P. Brener, M. A. Cruz, and R. L. Thompson, "Pre-processing DNS data to improve statistical convergence and accuracy of mean velocity fields in invariant data-driven turbulence models," Theoretical and Computational Fluid Dynamics (2022), 10.1007/s00162-022-00603-4.

[37]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," (2014), arXiv:1412.6980 [cs.LG].

[38]D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," (2015), arXiv:1511.07289 [cs.LG].

[39]J. Wu, H. Xiao, R. Sun, and Q. Wang, "Reynolds-averaged navier–stokes equations with explicit data-driven reynolds stress closure can be ill-conditioned," Journal of Fluid Mechanics **869**, 553–586 (2019).

[40]M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasude-

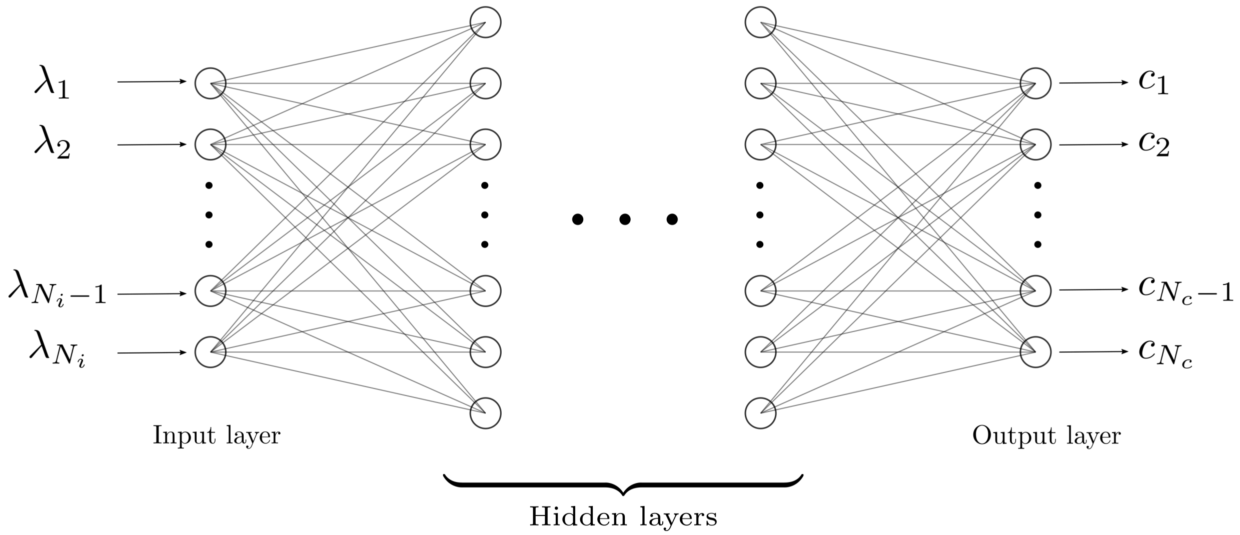
van, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015), software available from tensorflow.org.

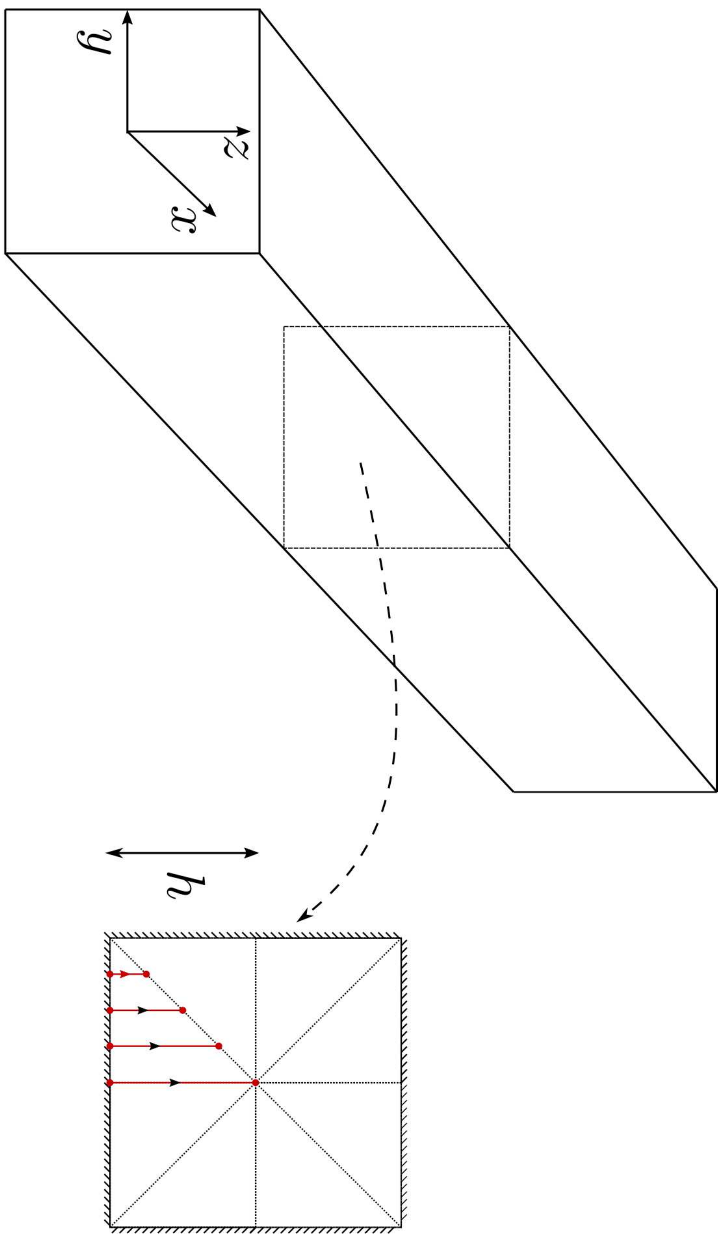
[41] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," Computers in Physics **12**, 620 (1998).

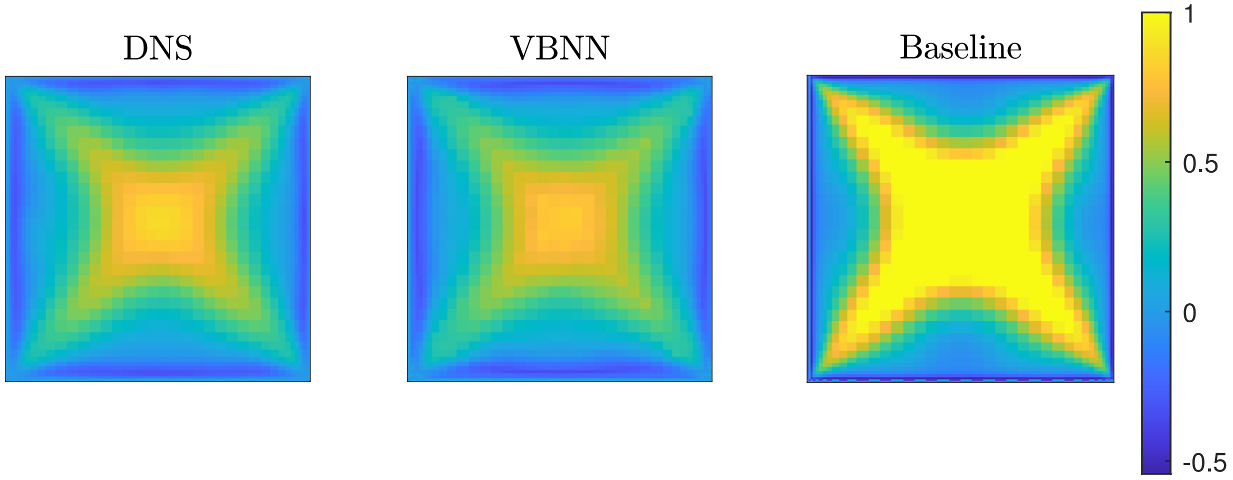
[42] B. Launder and B. Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," Letters in Heat and Mass Transfer **1**, 131–137 (1974).

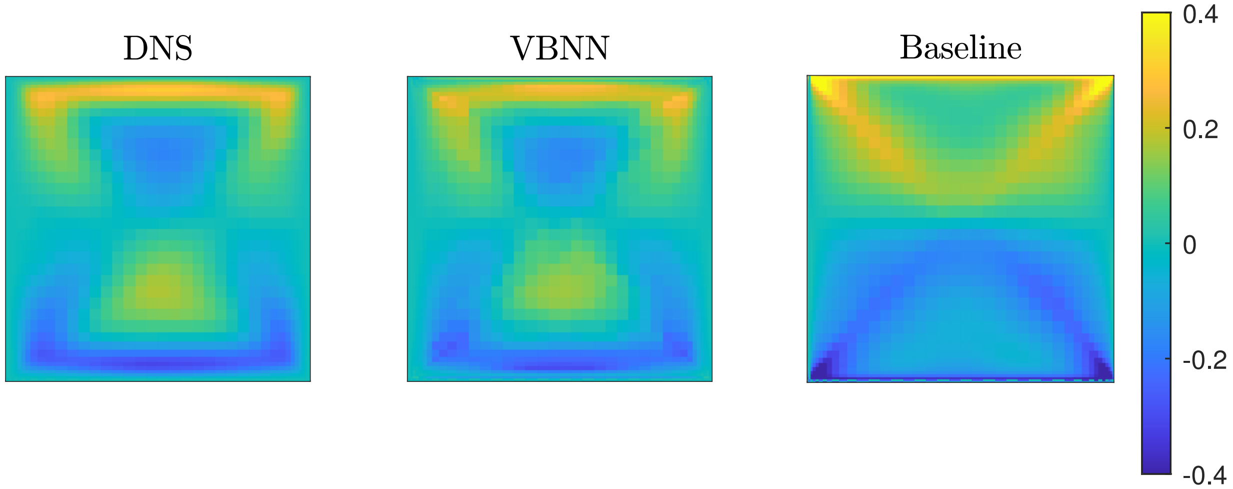
[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. **15**, 1929–1958 (2014).

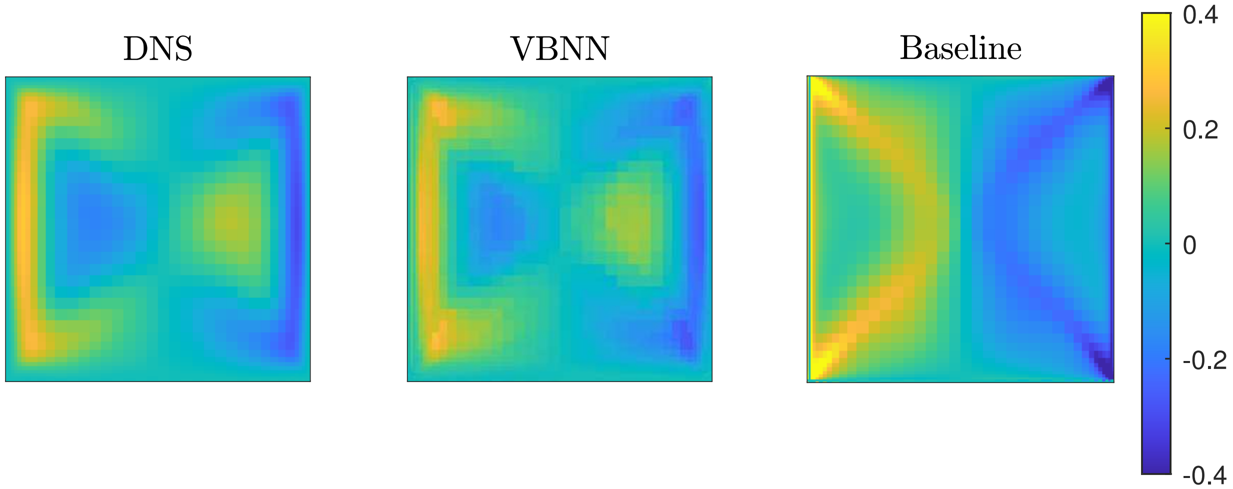
[44] C. Rapp, M. Breuer, M. Manhart, and N. Peller, "UFR 3-30, 2D periodic hill flow,".

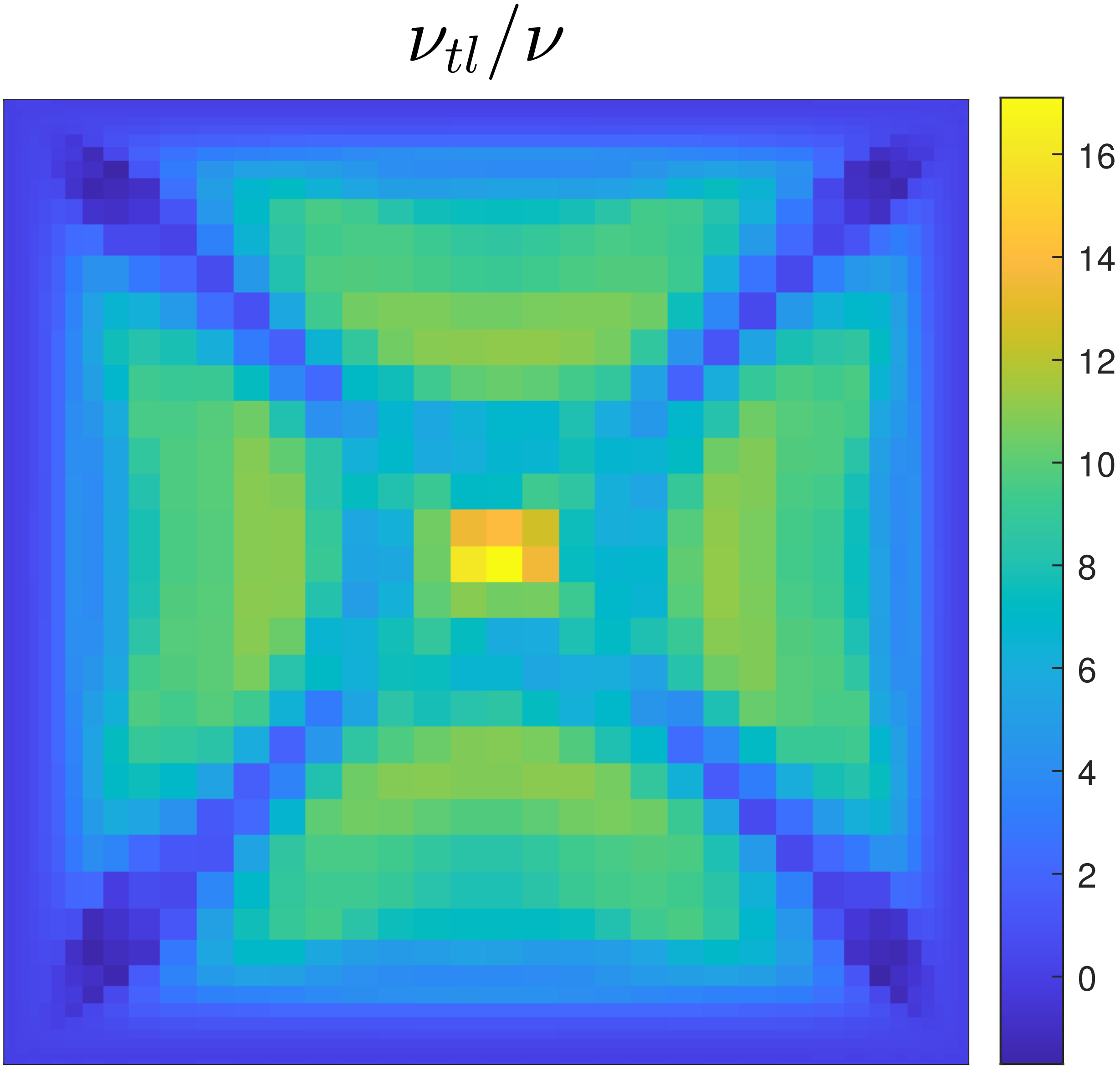

λ_1
λ_2
λ_{N_i-1}
λ_{N_i}
c_1
c_2
c_{N_c-1}
c_{N_c}
Input layer
Output layer
Hidden layers

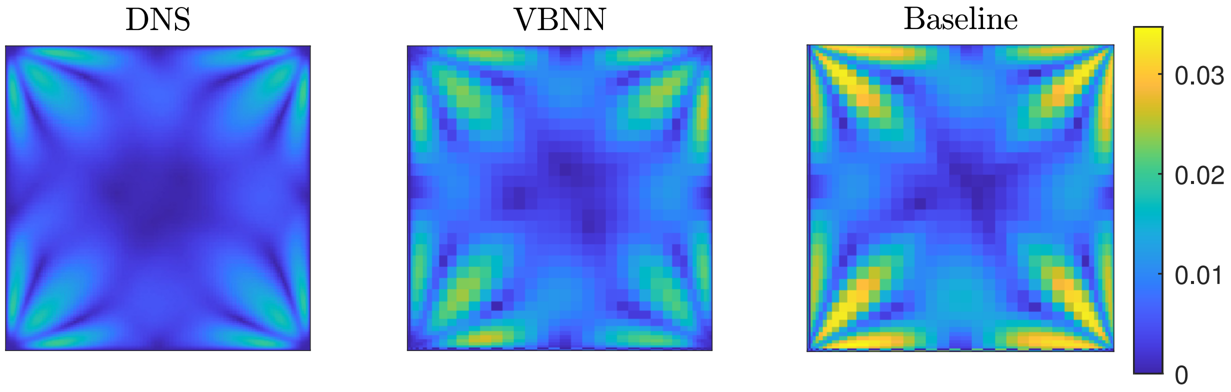

z
x
y
h

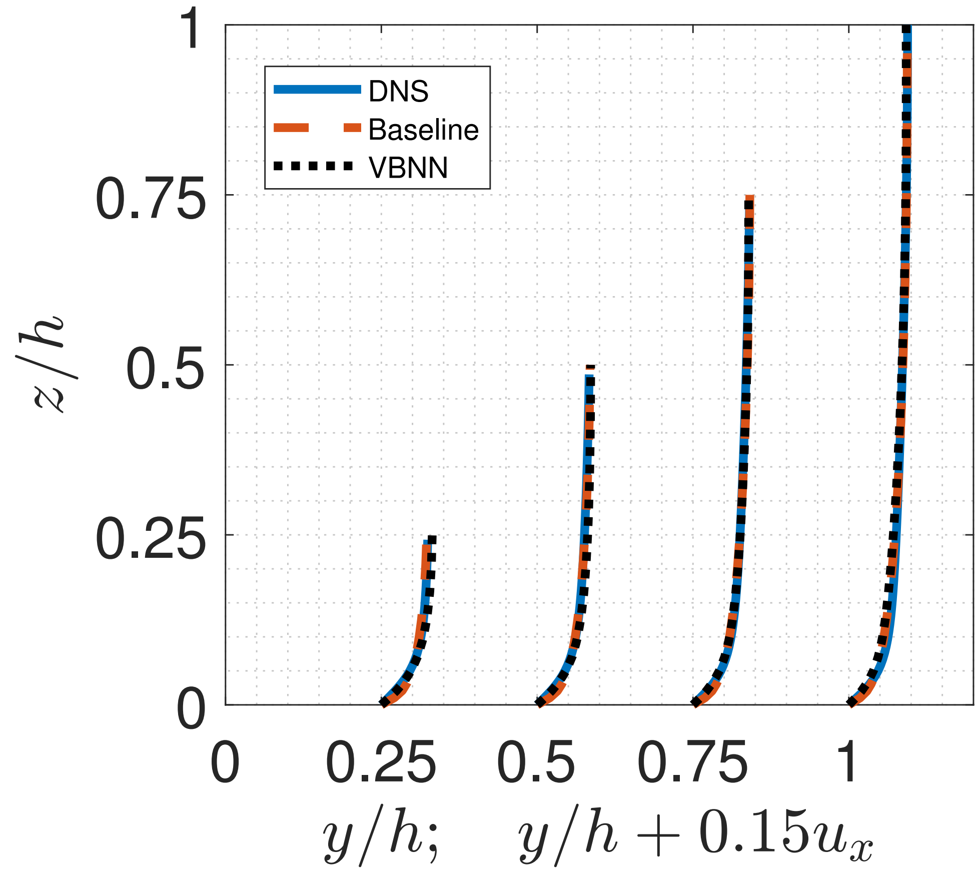

DNS
VBNN
Baseline
1
0.5
0
-0.5

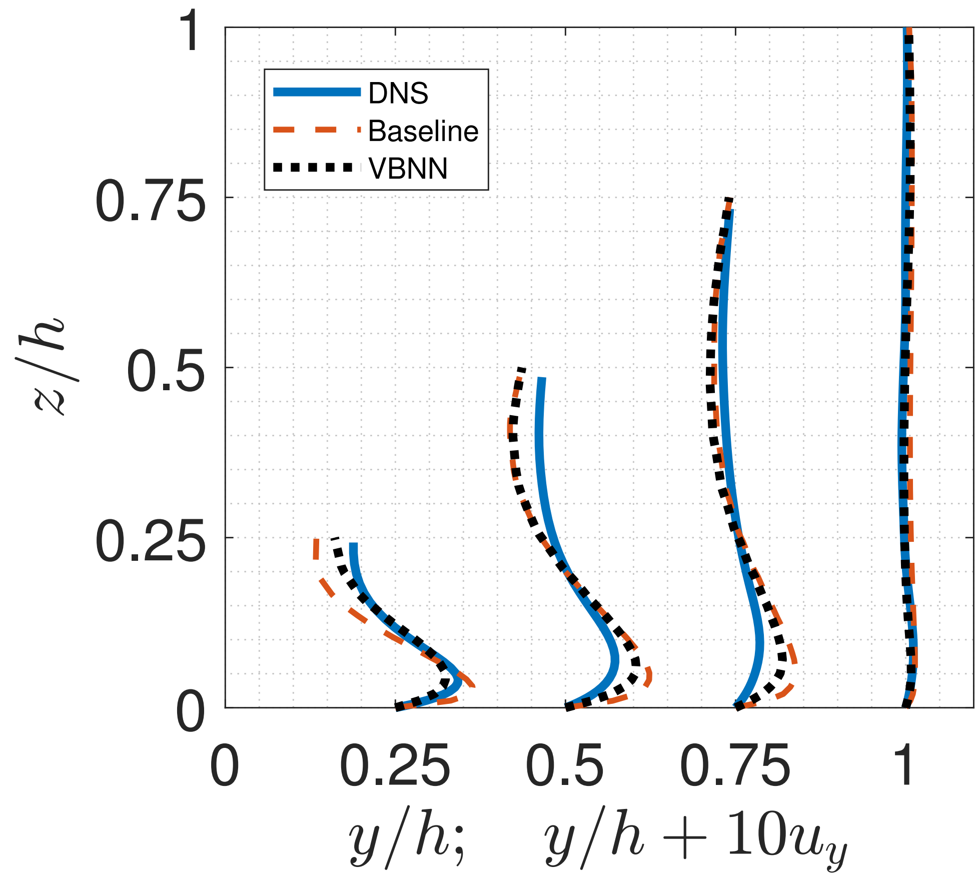

DNS
VBNN
Baseline
0.4
0.2
0
-0.2
-0.4

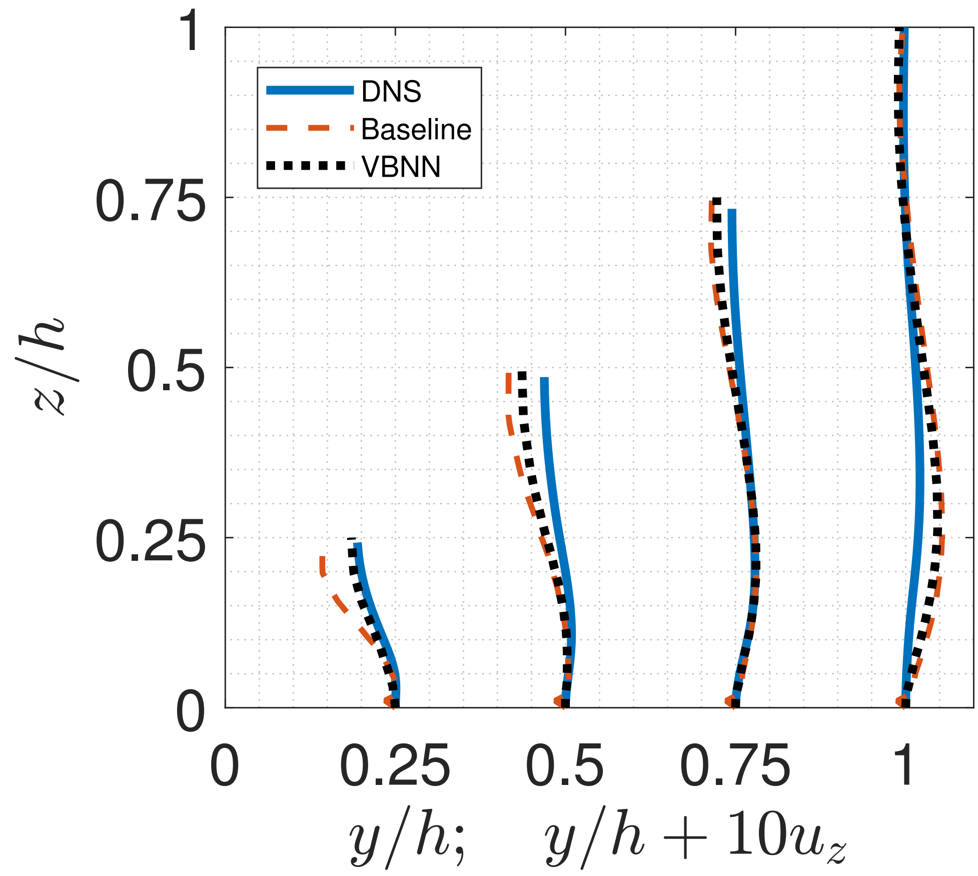

DNS
VBNN
Baseline
0.4
0.2
0
-0.2
-0.4

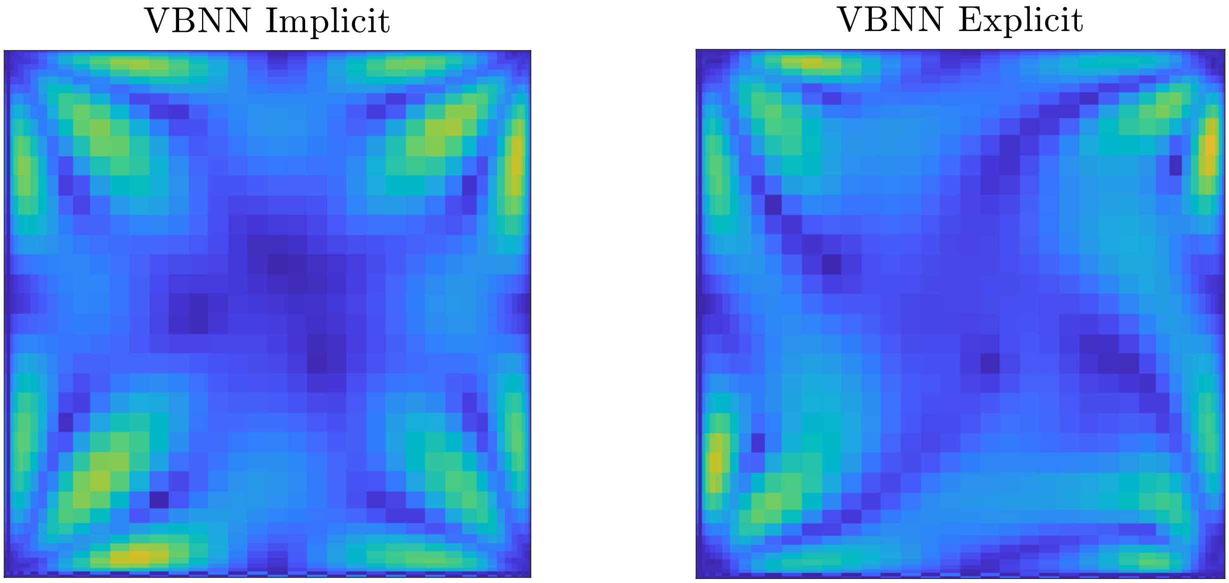

ν_{tl}/ν
16
14
12
10
8
6
4
2
0

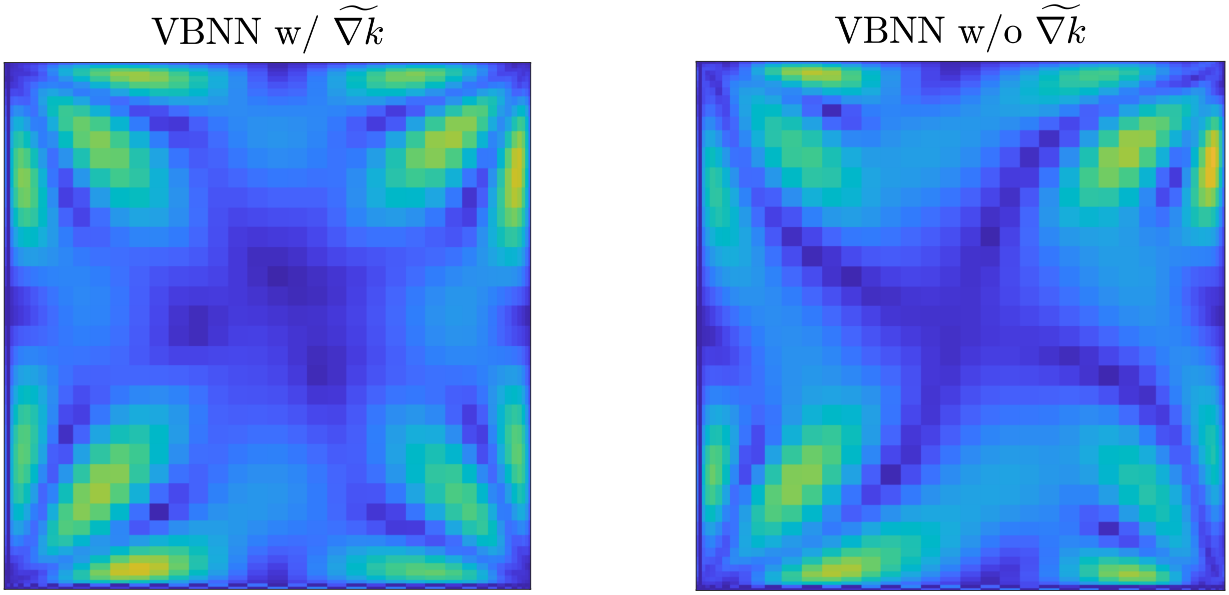

DNS
VBNN
Baseline
0.03
0.02
0.01
0


DNS
Baseline
VBNN
1
0.75
0.5
0.25
0
z/h
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 0.15u_x$


DNS
Baseline
VBNN
1
0.75
0.5
0.25
0
z/h
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 10u_y$

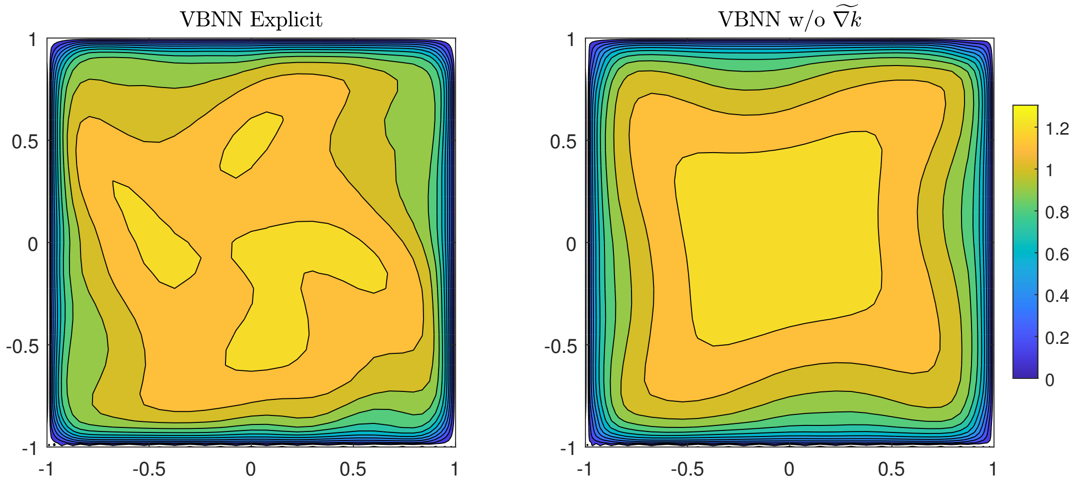

DNS
Baseline
VBNN
z/h
1
0.75
0.5
0.25
0
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 10u_z$

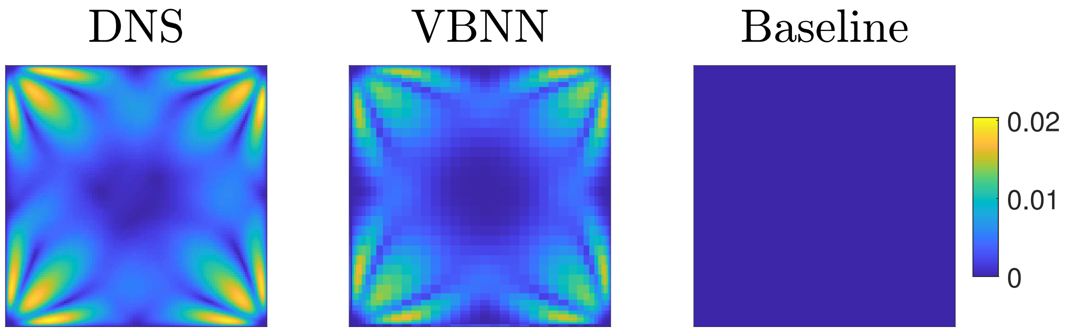

VBNN Implicit
VBNN Explicit

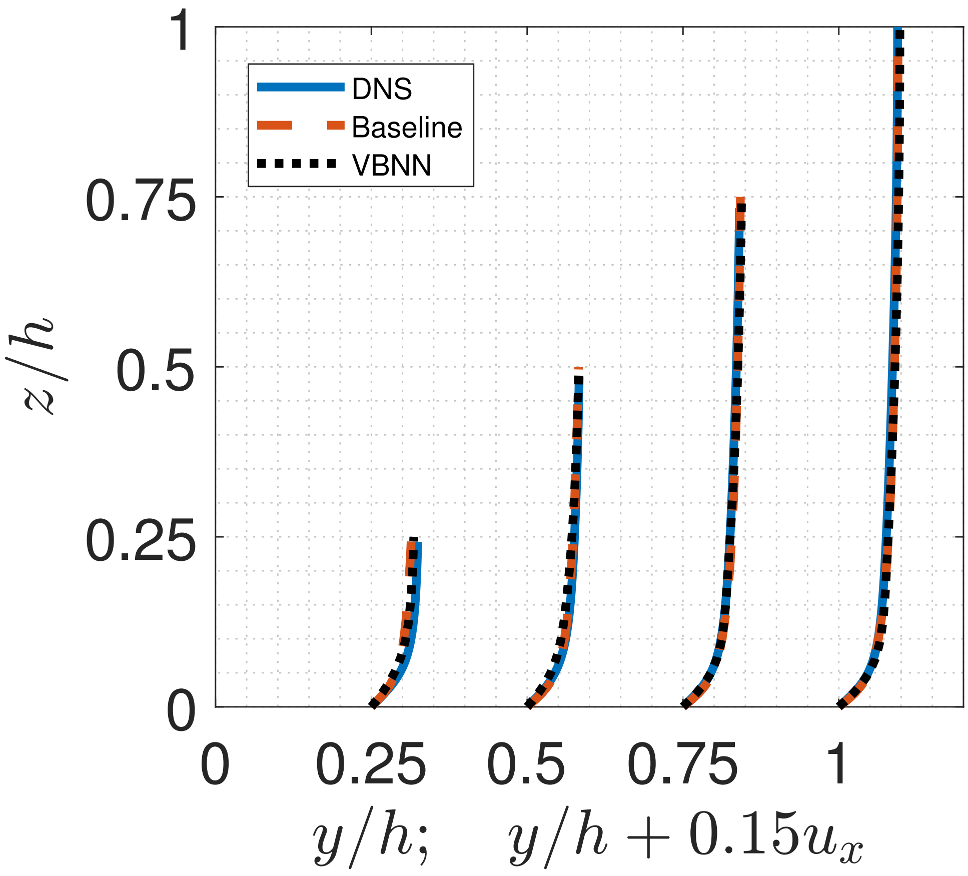


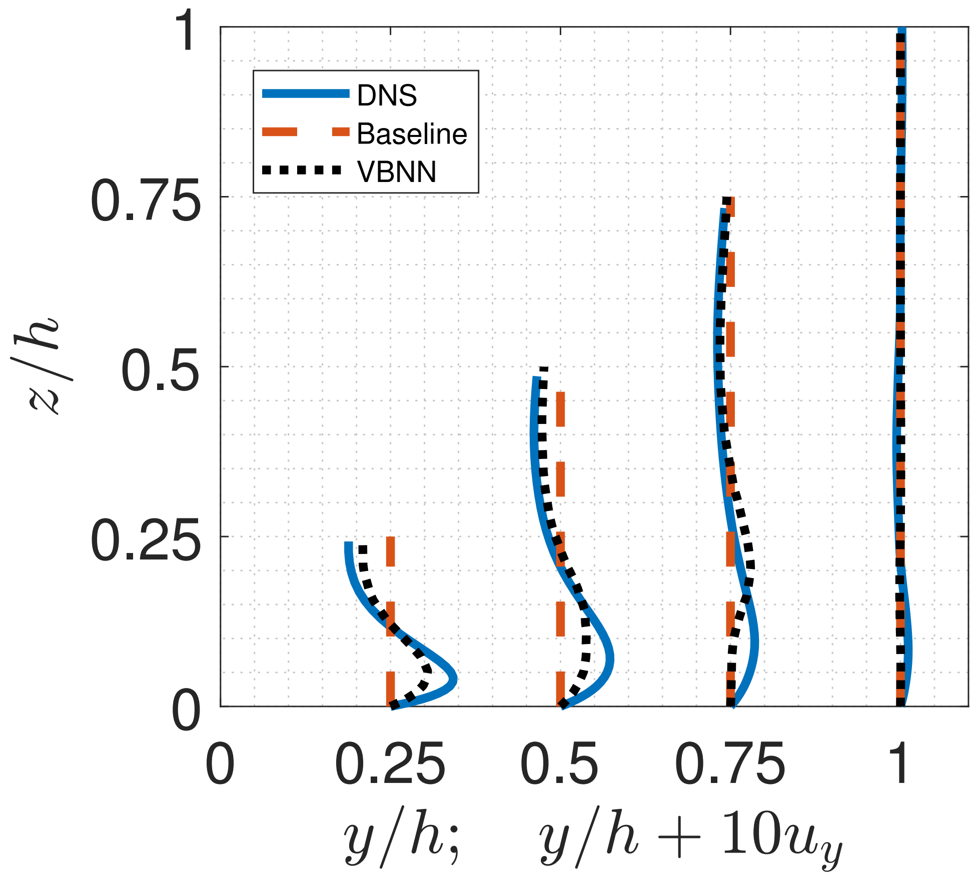
VBNN w/ $\widetilde{\nabla}k$

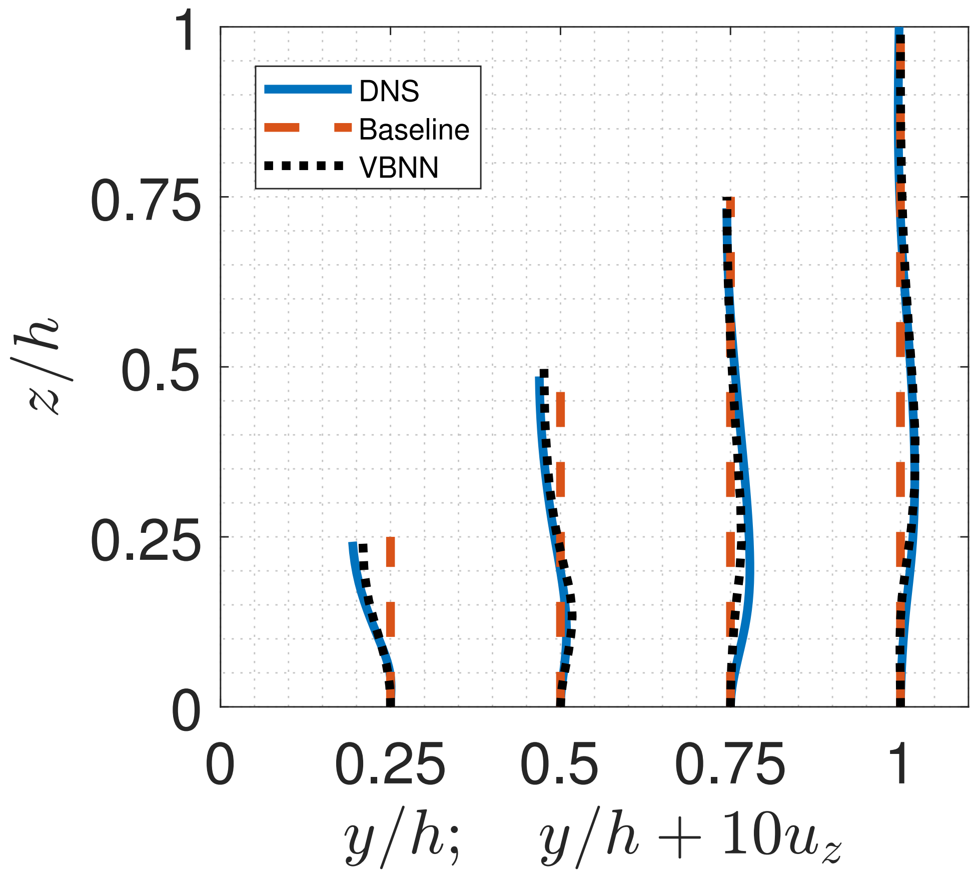
VBNN w/o $\widetilde{\nabla}k$

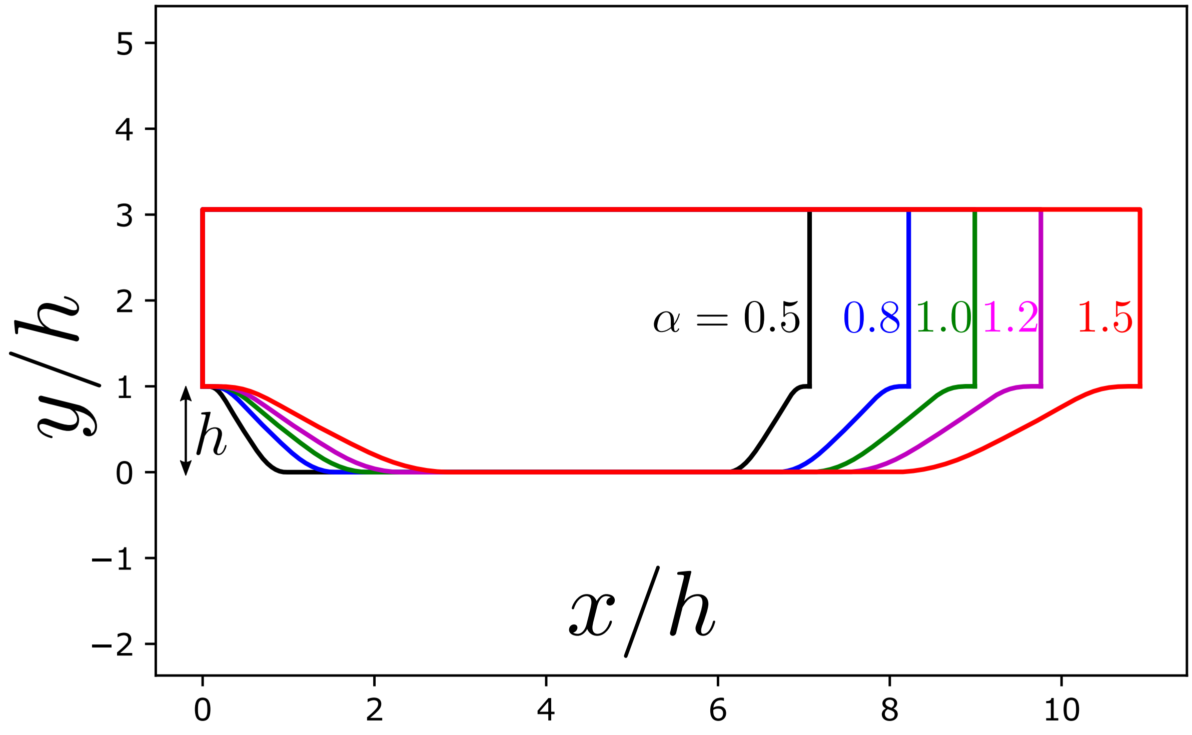

VBNN Explicit
VBNN w/o $\widetilde{\nabla k}$
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.2
1
0.8
0.6
0.4
0.2
0

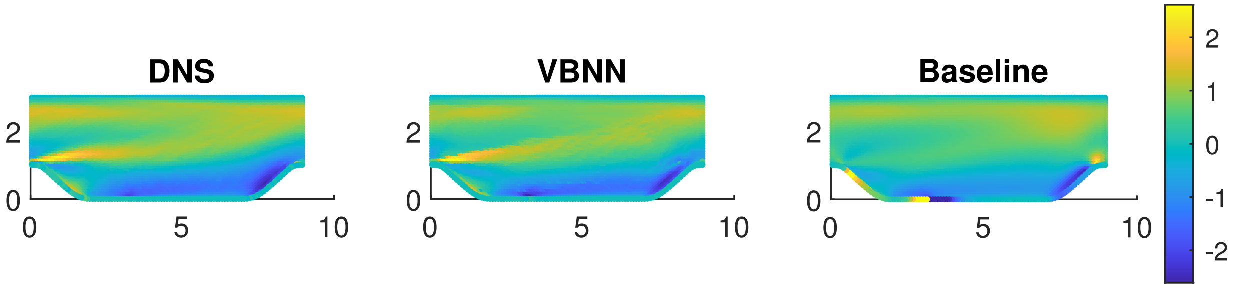

DNS VBNN Baseline

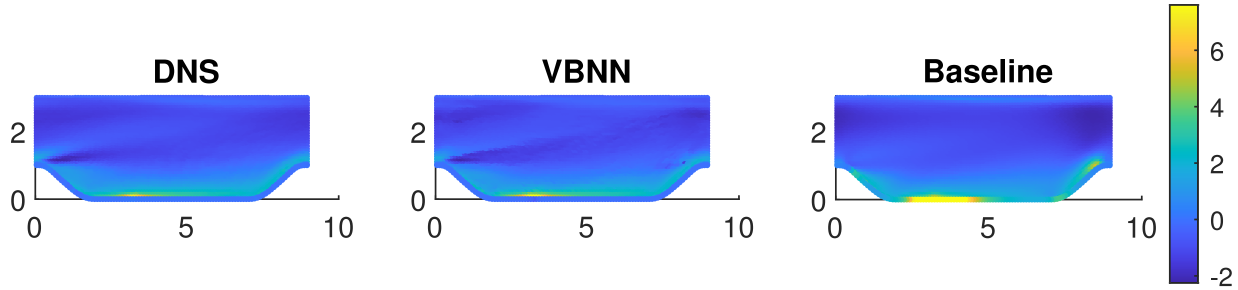

DNS
Baseline
VBNN
1
0.75
0.5
0.25
0
z/h
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 0.15u_x$

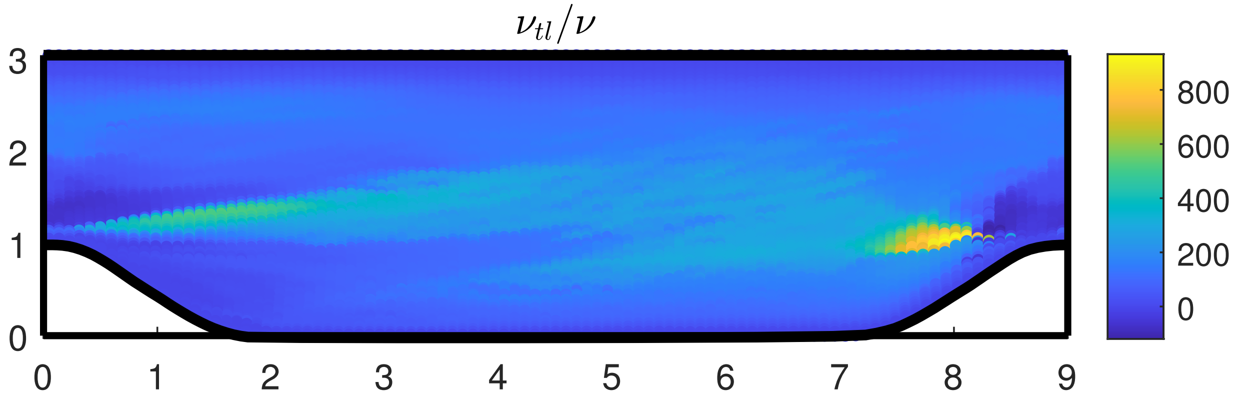

DNS
Baseline
VBNN
z/h
1
0.75
0.5
0.25
0
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 10u_y$

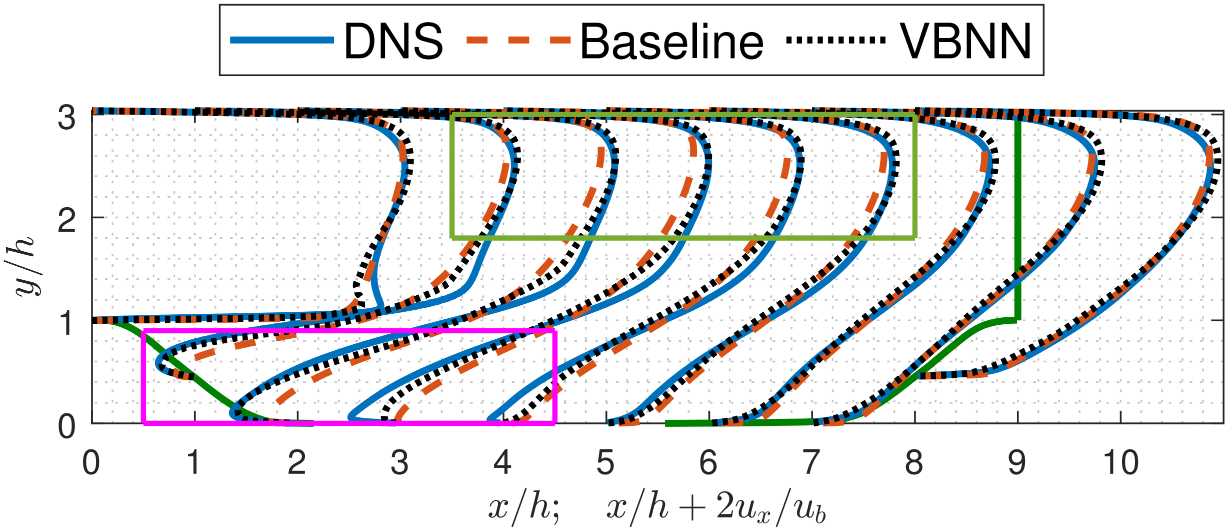

DNS
Baseline
VBNN
z/h
1
0.75
0.5
0.25
0
0
0.25
0.5
0.75
1
$y/h; \quad y/h + 10u_z$

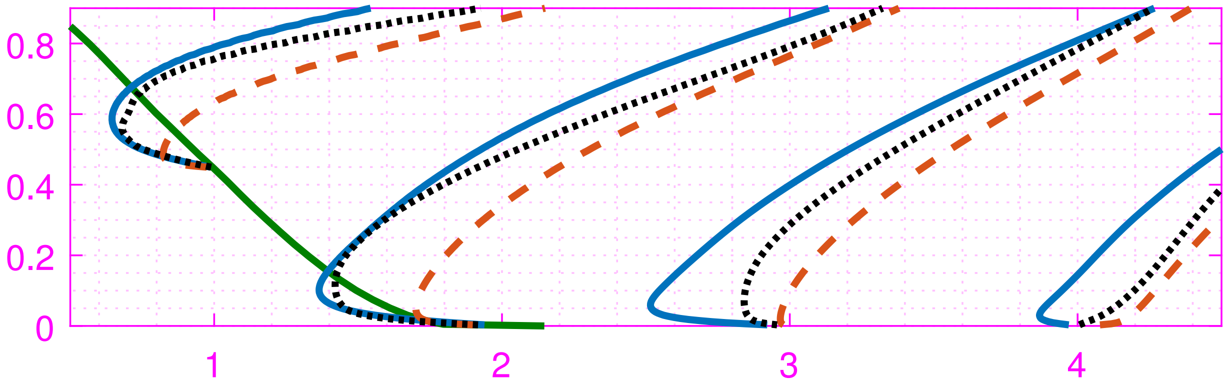

5
4
3
2
1
0
−1
−2
y/h
h
$\alpha = 0.5$
0.8
1.0
1.2
1.5
x/h
0
2
4
6
8
10

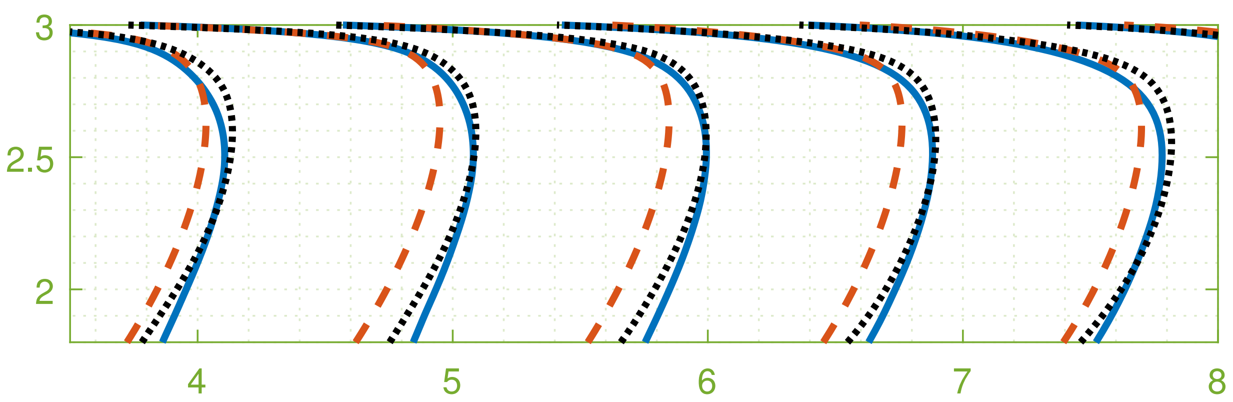

DNS
VBNN
Baseline
2
0
0
5
10
2
0
0
5
10
2
0
0
5
10
2
1
0
-1
-2

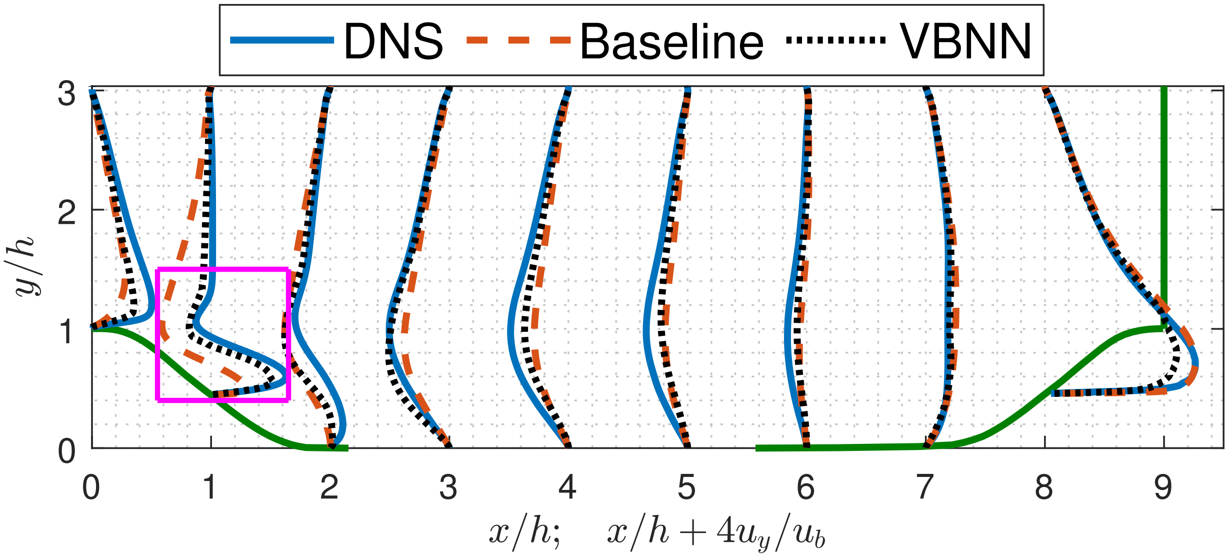

DNS
VBNN
Baseline
2
0
0
5
10
6
4
2
0
-2

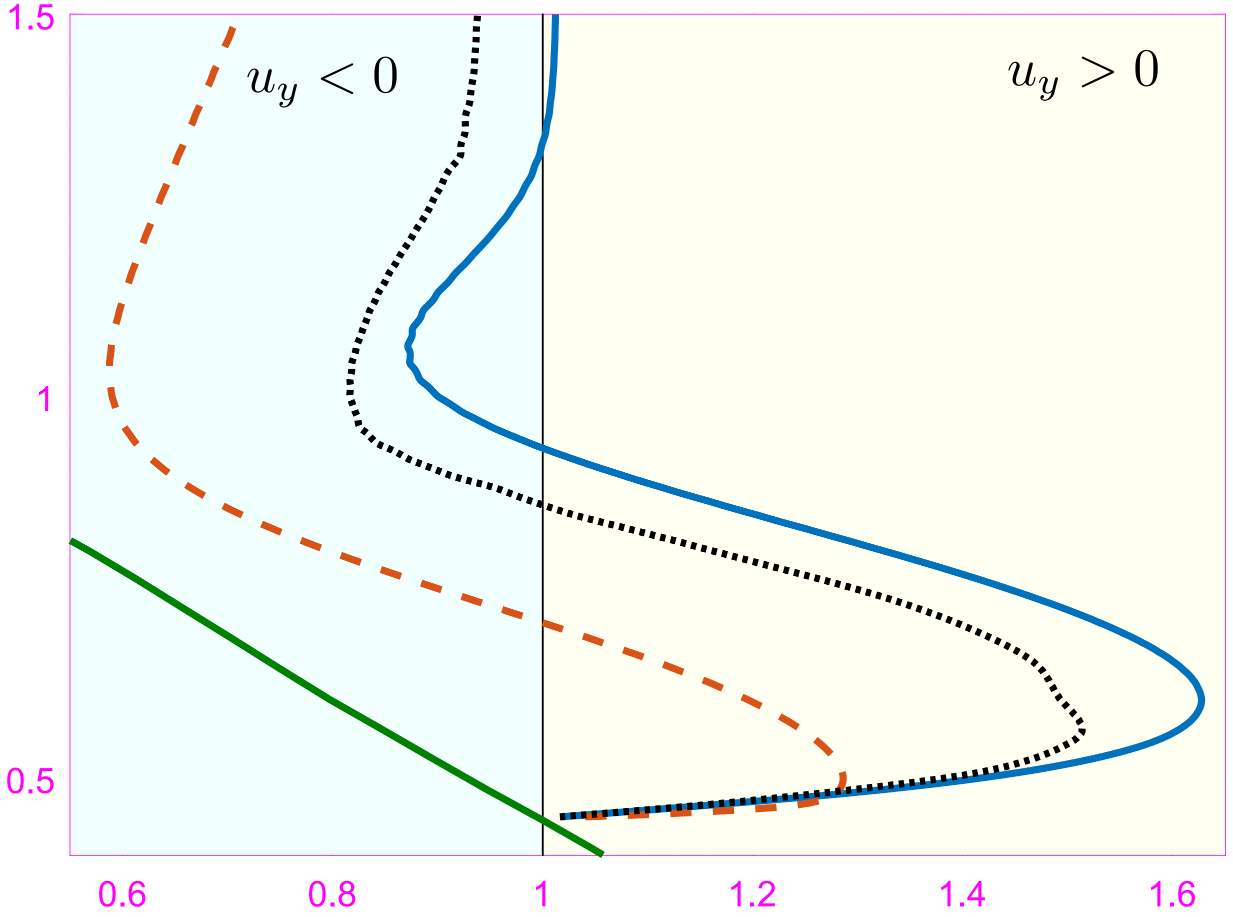

ν_{tl}/ν
3
2
1
0
0
1
2
3
4
5
6
7
8
9
800
600
400
200
0


DNS
Baseline
VBNN
y/h
0
1
2
3
0
1
2
3
4
5
6
7
8
9
10
$x/h; \quad x/h + 2u_x/u_b$


0.8
0.6
0.4
0.2
0
1
2
3
4


3
2.5
2
4
5
6
7
8


DNS
Baseline
VBNN
3
2
1
0
y/h
0
1
2
3
4
5
6
7
8
9
$x/h; \quad x/h + 4u_y/u_b$


$u_y < 0$
$u_y > 0$
1.5
1
0.5
0.6
0.8
1
1.2
1.4
1.6