Data-Driven RANS Turbulence Closures for Forced Convection Flow in Reactor Downcomer Geometry

Arsen S. Iskhakov,[*,a] Cheng-Kai Tai,[a] Igor A. Bolotnov,[a] Tri Nguyen,[b]
Elia Merzari,[b] Dillon R. Shaver,[c] and Nam T. Dinh[a]

[a] *Department of Nuclear Engineering*
North Carolina State University, Raleigh, NC, USA

[b] *Department of Nuclear Engineering*
Penn State University, State College, PA, USA

[c] *Nuclear Science and Engineering Division*
Argonne National Laboratory, Lemont, IL, USA

[*]Email: aiskhak@ncsu.edu

Number of pages:	33
Number of tables:	8
Number of figures:	18

Abstract

Recent progress in data-driven turbulence modeling has shown its potential to enhance or replace traditional equation-based Reynolds-averaged Navier-Stokes (RANS) turbulence models. This work utilizes invariant neural network architectures to model Reynolds stresses and turbulent heat fluxes in forced convection flows (when the models can be decoupled). As the considered flow is statistically 1D, the invariant NN architecture for the Reynolds stress model reduces to the linear eddy viscosity model. To develop the data-driven models, direct numerical and RANS simulations in vertical planar channel geometry mimicking a part of the reactor downcomer are performed. Different conditions and fluids relevant to advanced reactors (sodium, lead, unitary Prandtl number fluid, and molten salt) constitute the training database. The models enabled accurate predictions of velocity and temperature and, compared to the baseline $k - \tau$ turbulence model with the simple gradient diffusion hypothesis, do not require tuning of the turbulent Prandtl number. The data-driven framework is implemented in the open source GPU-accelerated spectral element solver `nekRS` and has shown potential for future developments and consideration of more complex mixed convection flows.

Keywords — Machine learning, Turbulence modeling, Forced convection, Low and high Prandtl fluids, Data-driven modeling

I. INTRODUCTION

The traditional mechanistic modeling framework has allowed considerable progress in computational fluid dynamics (CFD) in the past century. However, despite the continuous growth of computational power and advances in numerical methods, solution of the Navier-Stokes equations (known as DNS, direct numerical simulations) remains overly computationally expensive for modern engineering scale problems. One of the most popular approaches to reduce modeling cost is to perform Reynolds-averaged Navier-Stokes (RANS) simulations. Although numerically efficient, RANS equations require closure models for Reynolds stresses (RS) and turbulent heat fluxes (THF). Furthermore, widely employed traditional two-equation models such as the well-known $k - \varepsilon$ [1], $k - \tau$ [2], *etc.* turbulence models exhibit large model form / coefficient uncertainties [3] especially for non-adiabatic flows with buoyancy effects.

As a result of these bottlenecks, there is a continuous interest in data-driven (DD) modeling approaches. There are several strategies that can be adopted for DD modeling of dynamical systems [4], including, but not limited to:

- DD discovery when ML methods are used to discover unknown governing equations. An example of such framework is SINDy (sparse identification of nonlinear dynamics) [5]. The strategy is useful when the observed physical phenomenon does not have an adequate mathematical description in the form of differential equations.

- DD solution that uses ML to approximate a solution map (the Koopman operator) that predicts the evolution of a dynamical system over time. This avoids a costly solution of the governing equations. For instance, neural operators are gradually gaining attention in this area, *e.g.*, FNO [6] and DeepONet [7].

- Error correction, when ML is used to address inadequacy in the governing equations, closures, or quantities of interest [8, 9]. The strategy directly focuses on inadequate term(s) in the mathematical description in an attempt to reduce the uncertainties in the modeling process.

- DD closure modeling [10], which is the focus of this work. Recent progress in DD turbulence modeling techniques has shown their potential to improve the predictive capability of RANS simulations.

A recently initiated Integrated Research Project of the U.S. Department of Energy's Nuclear Energy Advanced Modeling and Simulation program (IRP-NEAMS-1.1) seeks to deliver improved, fast-running models for complex physical phenomena involving turbulent mixing, thermal stratification, and thermal striping in complex geometries relevant to advanced reactors by addressing several challenge problems. In particular, one of the objectives of Challenge Problem 1 [11] is to develop DD RANS turbulence models for conditions pertinent to advanced nuclear reactors, including laminar-to-turbulent and forced-to-mixed convection transitions in different coolants (sodium (Na), lead (Pb) and molten salt (FliBe)). This paper serves as a first step towards the outlined goal and reports on the development of the DD RS and THF models for forced convection conditions.

Several different methodologies are available for the development of DD turbulence closures, and we refrain from an extensive review, which can be found elsewhere, *e.g.*, [4, 10, 12]. As the literature survey shows, a particular implementation depends on the following:

- Problem of interest (*e.g.*, modeling of near-wall turbulence [13], adiabatic [14] or buoyant [15] flows, *etc.*);
- Chosen ML algorithm (feedforward neural network (NN) [14], convolutional NN [16], gene expression programming [15], *etc.*);
- A number of physical constraints and knowledge introduced (invariance [14], realizability [17], *etc.*);
- Quality and quantity of data and available information on the system; generalizability, computational efficiency, intepretability, explainability, consistency, and other requirements.

This work uses a linear eddy viscosity model (invariant tensor basis neural network (TBNN) in statistically 1D flow) and vector basis neural network (VBNN) architectures for pointwise prediction of RS and THF [14, 18]. Ling et al. [14, 19] firstly proposed TBNN architecture to predict RS in several canonical adiabatic flows. Notable improvements are observed compared to the traditional RANS modeling approaches; moreover, the invariant architecture has demonstrated superiority over an ordinary NN implementation. Geneva et al. [20] used a Bayesian formulation for TBNN to perform an uncertainty analysis of the developed DD turbulence model for a variety of canonical flows. Prof. Sandberg's research group (*e.g.*, [21]) has shown that the invariant formulation can also be adopted for different ML algorithms, such as gene expression programming, and has made significant progress in the area [10]. The similar invariant strategy (VBNN) can be adopted for modeling of THF [17, 18, 22] in incompressible flows. It should be noted that modeling of mixed and natural convection is much more complicated due to the involved coupling of RS and THF (see, *e.g.*, [15, 23]) and will be considered in future endeavors.

Although numerous papers focus on separate DD modeling of RS or THF, we demonstrate their simultaneous work to reduce errors in the simulation results. Another novelty is the attempt to develop a joint THF model for low- and high-Prandtl fluids (liquid metals and molten salt). To train NNs, the direct numerical simulation (DNS) [24, 25] and RANS databases for forced convection flows in a geometry that mimics part of the reactor downcomer are established. The built models are implemented in the spectral element solver `nekRS` [26] and have shown potential for future development and consideration of more complex mixed convection flows [11].

The rest of the paper is organized as follows. Sec. II details the methodology and training database used for the development of DD models. Sec. III provides results of the models' tests. Sec. IV concludes the paper and briefly discusses directions for future work.

II. METHODOLOGY

II.A. Problem Formulation

Consider the non-dimensional RANS equations with the Boussinesq approximation:

$$\begin{gathered}
\frac{\partial \overline{u}_i}{\partial x_i} = 0 \\
\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} + \mathrm{Ri}\, \overline{T} g_i \\
\frac{\partial \overline{T}}{\partial t} + \overline{u}_j \frac{\partial \overline{T}}{\partial x_j} = \frac{1}{\mathrm{RePr}} \frac{\partial^2 \overline{T}}{\partial x_j \partial x_j} - \frac{\partial \overline{u_j' T'}}{\partial x_j}
\end{gathered} \tag{1}$$

where x_i is the coordinate vector, t is time; u_i is the velocity vector, p is pressure, T is temperature, $\overline{u_i' u_j'}$ is the RS tensor, $\overline{u_j' T'}$ is the THF vector, g_i is a unit vector in vertical z direction; Re, Pr, and Ri are Reynolds, Prandtl, and Richardson numbers; $\overline{\cdot}$ and $\cdot'$ denote Reynolds-averaged and fluctuating components, respectively. More details on the non-dimensionalization procedure and reference scales can be found in [11]. The Boussinesq term Ri $\overline{T} g_i$ takes into account buoyancy effects; however, in this work full incompressibility is assumed: Ri $= 0$. As a result, temperature can be treated as a passive scalar (no feedback to velocity from temperature) and RS are decoupled from THF.

The goal of this work is to develop DD closure models for RS $\overline{u_i'u_j'}$ and THF $\overline{u_j'T'}$ to facilitate the modeling of forced convection flows for different coolants (Pr) and conditions (Re) relevant to advanced nuclear reactors. The results reported in this article also serve as a basis for further consideration of mixed convection flows ($\mathrm{Ri} \neq 0$) [11].

II.B. Training Database

II.B.1. DNS setup

The geometry of interest is section of a nuclear reactor downcomer. Due to the relatively large internal radius, this is well approximated as a pair of parallel plates, shown in Figure 1. Table I describes the geometrical parameters of the downcomer in dimensional and non-dimensional units. The overall length ($L_z = 60$) is divided into three parts: an adiabatic inlet ($z < 5$), an active region with applied constant heat flux ($5 \leq z \leq 55$), and an adiabatic outlet ($z \geq 55$) [11, 24, 27].

The channel parameters are chosen according to the recommendations of the industrial project collaborators [11] and reflect the characteristics of the advanced reactor designs [11]. Non-dimensionalization of the problem makes the results applicable to different designs since one can rescale them to particular parameters. This also makes the solution more numerically efficient.

TABLE I
Geometrical parameters of the channel.

Parameter	Dimensional, m	Non-dimensional
Distance between walls L_y	0.034	0.5
Spanwise length L_x	0.034π	0.5π
Adiabatic inlet	0.34	5
Heated / cooled	3.4	50
Adiabatic outlet	0.34	5
Total length L_z	4.08	60

The following boundary conditions (BCs) are enforced. To obtain a fully developed turbulence in the channel, the inlet velocity is recycled from plane $z \lesssim 5$. To avoid backflow, a stabilized outflow condition is set at the outlet (Dong et al. [28]). The walls (planes $y = \pm 0.25$) are no-slip and the sides (planes $x = \pm 0.25\pi$) are periodic. The inlet temperature is $T_{in} = 0$. A constant heat flux is applied beginning at $z = 5$ and ending at $z = 55$. Two cases for the heat flux BCs are considered:

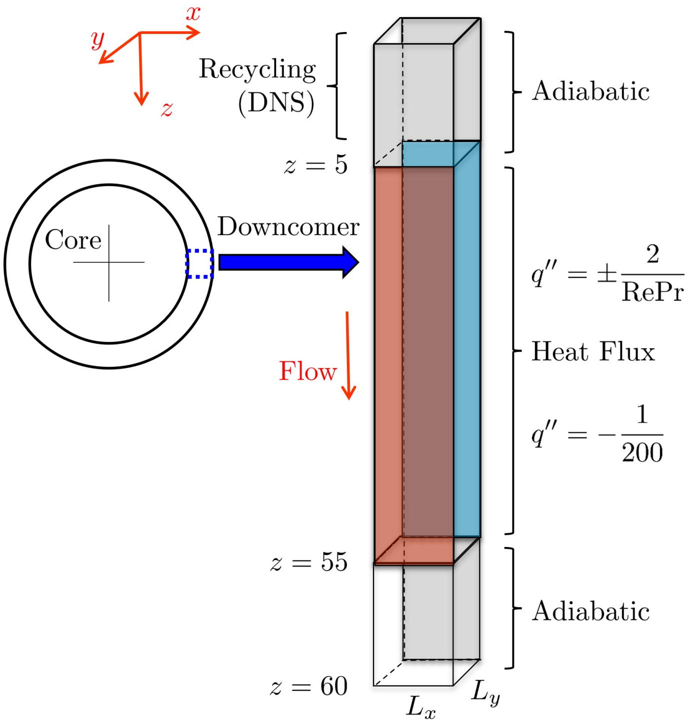


Fig. 1. Reactor downcomer (part) geometry.

- Case 1: The inner wall is heated and the outter wall is cooled with fluxes:

$$q'' = \pm\frac{2}{\mathrm{RePr}} \tag{2}$$

- Case 2: Both walls are cooled with fluxes:

$$q'' = -\frac{1}{200} \tag{3}$$

Constant heat flux BC Eq. (2) aims to establish a database for fundamental phenomena (modeling of the separate effects); non-uniform power shape and other heating phenomena are not considered. The simultaneous cooling case Eq. (3) is of interest for lead-cooled fast reactors: there is a possible scenario in which the reactor vessel has a lower temperature from both sides of the coolant [11]. For the other fluids, it is also important to study case 2 for the generalizability of the data and models.

Table II provides the dataset parameters used for training and testing of the models under development. Overall, there are 20 datasets, 2 of which are used for testing purposes – datasets 5 and 15 as denoted in the table. Datasets 1 − 17 use the wall heating configuration described by Eq. 2 (case 1), while datasets 18 – 20 use the configuration described by Eq. 3 (case 2). Note that for the RS model, there is no difference between the fluids as the momentum equation is decoupled from the energy equation. Thus, there are only 9 different Re conditions. A detailed analysis of the DNS results can be found in [24, 25].

TABLE II
Parameters of the datasets.

No.	Re	Pr	Purpose
1	5000	0.0048 (Na) Case 1	Training
2	6000		
3	7500		
4	10000		
5	15000		Test
6	3500	0.0169 (Pb) Case 1	Training
7	5000		
8	7500		
9	5000	1.0 Case 1	Training
10	7500		
11	8500		
12	10000		
13	4000	12.0 (FliBe) Case 1	Test
14	5000		Training
15	7500		
16	10000		
17	12000		
18	5000	12.0 (FliBe) Case 2	Training
19	7500		
20	10000		

II.B.2. RANS setup

The developed framework requires matching RANS simulations (for more details, see Section II.E), which are performed for the same conditions as DNS (Table II) and using the same mesh to avoid uncertainties associated with the interpolation from one mesh to another. This is particularly important near the walls. To reduce the computational cost, the spanwise x direction is reduced to have only 4 spectral elements. This is the minimum number required in `NekRS` to set periodic BCs for a "pseudo" 2D setup. The simulations are performed using the $k - \tau$ turbulence model [2], while for THF the simple gradient diffusion hypothesis [29] is used with a constant turbulent Prandtl number ($\Pr_t = 1$). Instead of recycling the inlet velocity, the inlet profiles for u_z, k, and τ are applied from the precursor RANS simulations. The main goal is to have fully developed profiles at $z = 5$ where heating / cooling begins. The simulations are performed until reaching steady-state regimes for each flow to extract the necessary quantities.

II.B.3. Data preprocessing

Figure 2 illustrates the data preprocessing procedure. Having performed DNS, which produces temporal statistical data for RS, THF, and other quantities [24, 25], the 3D fields obtained are spatially averaged in the spanwise direction x. Then values on the walls were removed because they do not contain meaningful information. These "reduced" 2D fields are used to train the RS model. Since the velocity is statistically 1D, the reduced 2D fields can also be spatially collapsed in the streamwise z direction to train the RS model. However, it was found that the RS model can be trained more efficiently if averaging of 10 consecutive profiles is performed. The resulting "compressed" 2D fields are used to train the RS model. To train the THF model, the unheated inlet / outlet regions are additionally removed. A similar procedure is adopted for the RANS datasets.

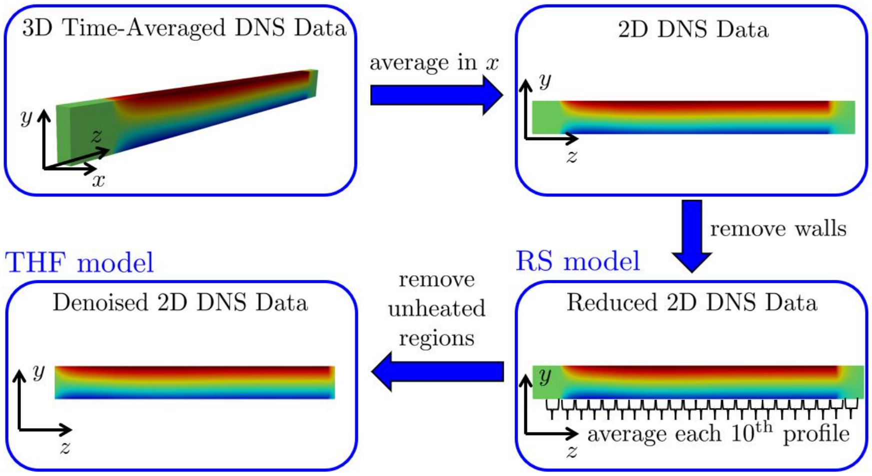


Fig. 2. Data preprocessing schematic.

II.C. Reynolds Stress Model Description

To develop the RS model we begin with the weak equilibrium assumption [29]:

$$\overline{u_i'u_j'} \equiv k\frac{\overline{u_i'u_j'}}{k} = k\left(2b_{ij} + \frac{2}{3}\delta_{ij}\right) \tag{4}$$

where k is the turbulent kinetic energy (TKE) obtained by solving the $k-\tau$ turbulence model, b_{ij} is the scaled anisotropic RS tensor. Eq. (4) implies that spatial and temporal variations in b_{ij} are neglected, but the variations in k are retained [29]. Using the locality assumption for b_{ij} allows it to be predicted point-wise using a feedforward NN, which potentially makes the model applicable to a different geometry.

Furthermore, based on the dimensionality analysis [30], it is assumed:

$$b_{ij} = b_{ij}\left(\frac{\partial \overline{u}_i}{\partial x_j}, k, \varepsilon\right) = b_{ij}\left(s_{ij}, w_{ij}\right)$$

where ε is the TKE dissipation rate, s_{ij} is scaled by turbulent time scale $\left(\tau = \frac{k}{\varepsilon}\right)$ symmetric tensor:

$$s_{ij} = \frac{1}{2}\tau\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{5}$$

w_{ij} is the scaled antisymmetric tensor:

$$w_{ij} = \frac{1}{2}\tau\left(\frac{\partial \overline{u}_i}{\partial x_j} - \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{6}$$

To enforce the Galilean invariance, the tensor basis representation is employed [31]:

$$b_{ij}\left(s_{ij}, w_{ij}\right) = \sum_{n=1}^{N_t} g_n\left(\boldsymbol{\mathcal{I}}\right) t_{ij}^{(n)} \tag{7}$$

where N_t is the number of tensor bases $t_{ij}^{(n)}$ (Table III), g_n are tensor basis functions of invariants $\boldsymbol{\mathcal{I}} = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_m\}$ (Table IV) [31].

TABLE III
Tensor bases for the RS model.

$\mathrm{T}^{(1)} = \mathrm{S}$	$\mathrm{T}^{(2)} = \mathrm{S}^2$	–	–
$\mathrm{T}^{(3)} = \mathrm{W}^2$	–	–	–
$\mathrm{T}^{(4)} = \mathrm{SW} - \mathrm{WS}$	$\mathrm{T}^{(5)} = \mathrm{SW}^2 - \mathrm{W}^2\mathrm{S}$	$\mathrm{T}^{(6)} = \mathrm{WSW}^2 - \mathrm{W}^2\mathrm{SW}$	$\mathrm{T}^{(7)} = \mathrm{S}^2\mathrm{W} - \mathrm{WS}^2$

Note that Pope [30] proposed a slightly different integrity basis. The tensor basis representation suggested in Ref. [31] is more compact and, therefore, adopted in this work. Also, note that, for

TABLE IV
Invariants for the RS model.

$\mathcal{I}_0 = \mathrm{tr}(\mathrm{S})$	$\mathcal{I}_1 = \mathrm{tr}(\mathrm{S}^2)$	$\mathcal{I}_2 = \mathrm{tr}(\mathrm{S}^3)$
$\mathcal{I}_3 = \mathrm{tr}(\mathrm{W}^2)$	–	–
$\mathcal{I}_4 = \mathrm{tr}(\mathrm{SW}^2)$	$\mathcal{I}_5 = \mathrm{tr}(\mathrm{S}^2\mathrm{W}^2)$	$\mathcal{I}_6 = \mathrm{tr}(\mathrm{S}^2\mathrm{W}^2\mathrm{SW})$

convenience, the tensor notation is replaced by matrix one ($t_{ij}^{(n)} = \mathrm{T}^{(n)}$, $s_{ij} = \mathrm{S}$, $w_{ij} = \mathrm{W}$).

One can implement the tensor basis representation Eq. (7) in a NN architecture: a TBNN is shown in Figure 3, Ref. [14]. Using input invariants $\boldsymbol{\mathcal{I}}$ (a set of scalars), the NN learns to predict tensor basis functions g_n, which are then multiplied by the corresponding tensor bases (precomputed) $t_{ij}^{(n)}$ and summed. The obtained using Eq. (7) scaled anisotropic tensor b_{ij} is used to calculate the loss $\mathcal{L}$ and to backpropagate the error to update the NN's weights and biases. We also supply the distance to the nearest wall y_w and Re as input features, as they help to improve the generalizability of the model. The NN's architecture and hyperparameters are described in Table V.

TABLE V
Hyperparameters and architecture of the NN for the RS model.

Parameter	Value / Method
Number of layers and neurons	8 – 20 – 20 – 20 – 20 – 7
Activation	SELU
Optimizer	Adam
Loss function	Mean squared error
Regularization	Dropout 5%
Learning rate	0.01 with decay on plateau
Number of epochs	500
Mini-batch size	32

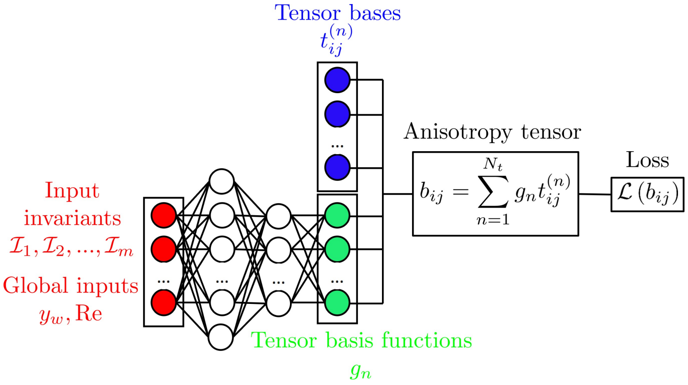


Fig. 3. A tensor basis neural network architecture.

Remark. In the case of incompressible statistically 1D velocity, Eq. (7) shrinks to the the linear eddy viscosity model:

$$b_{ij} = g_1\left(\mathcal{I}_1\right) t_{ij}^{(1)} = g_1\left(\mathcal{I}_1\right) s_{ij} \tag{8}$$

where the only required RS component is:

$$\overline{u_y' u_z'} = 2k b_{yz} = g_1\left(\mathcal{I}_1\right) \frac{k^2}{\varepsilon} \frac{\partial \overline{u}_z}{\partial y} \tag{9}$$

Note that comparing to equation-based models, g_1 is non-constant since it is predicted by the NN. However, since this work aims to establish a basis for more complex flows, the full representation Eq. (7) is kept.

II.D. Turbulent Heat Flux Model Description

To model THF, based on the dimensionality analysis [32], it is first assumed:

$$\overline{u_j' T'} = \overline{u_j' T'}\left(\frac{\partial \overline{u}_i}{\partial x_j}, \frac{\partial \overline{T}}{\partial x_j}, k, \varepsilon, k_T, \varepsilon_T\right)$$

where k_T is the temperature variance, ε_T is the thermal dissipation rate. Note that the dependence of THF on RS is implicitly taken into account through the dependency on $\frac{\partial \overline{u}_i}{\partial x_j}$, k, and τ.

Unfortunately, `nekRS` does not yet have any four equation models [33] implemented and, following [34], equal turbulent time scales $\frac{k}{\varepsilon} \approx \frac{k_T}{\varepsilon_T}$ are assumed, yielding:

$$\overline{u_j' T'} \approx \overline{u_j' T'}\left(\frac{\partial \overline{u}_i}{\partial x_j}, \frac{\partial \overline{T}}{\partial x_j}, k, \varepsilon\right) = \overline{u_j' T'}\left(s_{ij}, w_{ij}, \vartheta_j\right)$$

where temperature gradient is scaled by the turbulent length scale $\tau\sqrt{k}$:

$$\vartheta_j = \tau\sqrt{k}\frac{\partial \overline{T}}{\partial x_j} \tag{10}$$

The above assumption of equal time scales is significantly violated near the walls since temperature fluctuations are possible on the walls with the flux BC, whereas the non-slip BC for the velocity does not allow it. An example of a more appropriate scaling using the four-equation model can be found in [17].

Furthermore, to enforce invariant properties, a vector basis representation is employed [31]:

$$h_i\left(s_{ij}, w_{ij}, \vartheta_j\right) = \sum_{n=1}^{N_v} f_n\left(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{J}}\right) v_i^{(n)} \tag{11}$$

where THF is scaled by turbulent velocity scale $\sqrt{k}$:

$$h_i = \frac{1}{\sqrt{k}} \overline{u_i' T'} \tag{12}$$

N_v is the number of vector bases $v_i^{(n)}$ (Table VI), f_n are vector basis functions, $\boldsymbol{\mathcal{J}} = \{\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_r\}$ are additional invariants due to dependence on $\frac{\partial \overline{T}}{\partial x_j}$ [31] (Table VII).

TABLE VI
Vector bases for the THF model.

$\mathbf{v}^{(1)} = \boldsymbol{\vartheta}$	$-$
$\mathbf{v}^{(2)} = \mathrm{S}\boldsymbol{\vartheta}$	$\mathbf{v}^{(3)} = \mathrm{S}^2\boldsymbol{\vartheta}$
$\mathbf{v}^{(4)} = \mathrm{W}\boldsymbol{\vartheta}$	$\mathbf{v}^{(5)} = \mathrm{W}^2\boldsymbol{\vartheta}$
$\mathbf{v}^{(6)} = (\mathrm{SW} + \mathrm{WS})\boldsymbol{\vartheta}$	$-$

TABLE VII
Additional invariants for the THF model.

$\mathcal{J}_1 = \boldsymbol{\vartheta\vartheta}$	$-$	$-$
$\mathcal{J}_2 = \boldsymbol{\vartheta}\mathrm{S}\boldsymbol{\vartheta}$	$\mathcal{J}_3 = \boldsymbol{\vartheta}\mathrm{S}^2\boldsymbol{\vartheta}$	$-$
$\mathcal{J}_4 = \boldsymbol{\vartheta}\mathrm{W}^2\boldsymbol{\vartheta}$	$-$	$-$
$\mathcal{J}_5 = \boldsymbol{\vartheta}\mathrm{SW}\boldsymbol{\vartheta}$	$\mathcal{J}_6 = \boldsymbol{\vartheta}\mathrm{S}^2\mathrm{W}\boldsymbol{\vartheta}$	$\mathcal{J}_7 = \boldsymbol{\vartheta}\mathrm{WSW}^2\boldsymbol{\vartheta}$

Note that in the case of incompressible statistically 1D velocity and 2D temperature, Eq. (11) shrinks to:

$$h_i = \sum_{n=1}^{3} f_n\left(\mathcal{I}_1, \mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3, \mathcal{J}_7\right) v_i^{(n)} \tag{13}$$

where h_i and v_i are 2D vectors (with wall-normal y and streamwise z components). However, since this work aims to establish the basis for more complex flows, the full representation Eq. (11) is adopted.

Furthermore, the scaled THF h_i is predicted using the diffusivity tensor d_{ij} [18]:

$$h_i = \left(\sum_{n=1}^{N_v} f_n v_{ij}^{(n)}\right) \vartheta_j = d_{ij} \vartheta_j \tag{14}$$

where $v_{ij}^{(n)}$ are tensors. Note that such a representation does not mean that we seek to predict $d_{ij} = d_{ij}\left(s_{ij}, w_{ij}, \vartheta_j\right)$ as it has a different integrity basis [31]. In the representation considered, multiplication by ϑ_j is just an additional (postponed) step in the VBNN architecture.

Prediction of THF through the diffusivity matrix allows enforcement of realizability using the following (alternative) approaches:

- By hardcoding the realizability into the VBNN architecture by decomposing the diffusivity matrix into symmetric and antisymmetric parts [17]:

  $$\mathrm{D} = \mathrm{D}_s + \mathrm{D}_a$$

  where $\mathrm{D}_s = \frac{1}{2}\left(\mathrm{A} + \mathrm{A}^\intercal\right)\left(\mathrm{A} + \mathrm{A}^\intercal\right)$, $\mathrm{D}_a = \frac{1}{2}(\mathrm{B} - \mathrm{B}^\intercal)$, while the matrices A and B are predicted through separate tensor basis functions f_n^{A} and f_n^{B}:

  $$a_{ij} = \sum_{n=1}^{N_v} f_n^{\mathrm{A}} v_{ij}^{(n)}$$

  $$b_{ij} = \sum_{n=1}^{N_v} f_n^{\mathrm{B}} v_{ij}^{(n)}$$

  Note that the symmetric part D_s is positive semidefinite (PSD), which ensures that D is also PSD.

- By *a posteriori* nullifying negative eigenvalues of the symmetric part of D [18]:

  $$\mathrm{D} = \mathrm{D}_s' + \mathrm{D}_a$$

  where D_s' is the symmetric part with nullified negative eigenvalues after prediction.

- By adding a regularization term to the loss function $\mathcal{L}$ that penalizes negative eigenvalues (physics-informed NN [35]):

  $$\mathcal{L}_{\mathrm{PINN}} = \mathrm{MSE}\left(\widehat{h}_i, h_i\right) + \beta \sum_k \mathrm{ReLU}(-\lambda_k)$$

  where λ_k are the eigenvalues of D, β is the weight of regularization, ReLU is the rectified linear unit function, $\widehat{h}_i$ and h_i denote predicted and true quantities, respectively.

The first approach markedly deteriorated the predictive capability of VBNN. The last two strategies have similar performance, but computing eigenvalues during training is an expensive procedure. Therefore, the diffusivity matrix was *a posteriori* corrected (usually, only a few % of the predicted values have negative eigenvalues).

A VBNN is depicted in Figure 4, Ref. [18]. We also supply y_w, Re, and Pr (where different fluids are used for training) as input features. A VBNN predicts the vector basis functions f_n with their subsequent multiplication to bases $v_{ij}^{(n)}$ according to Eq. (14); its architecture and hyperparameters are described in Table VIII.

TABLE VIII
Hyperparameters and architecture of the NN for the THF model.

Parameter	Value / Method
Number of layers and neurons	16 − 200 − 200 − 200 − 200 − 6
Activation	SELU
Optimizer	Adam
Loss function	Mean squared error
Regularization	Dropout 5%
Learning rate	0.02 with decay on plateau
Number of epochs	1000
Mini-batch size	20000

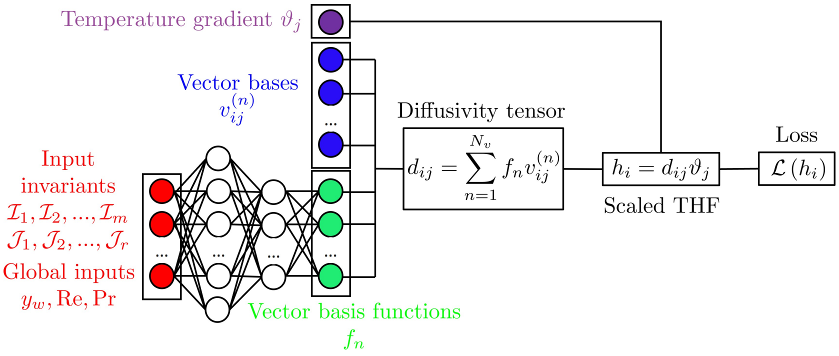

Temperature gradient ϑ_j
Vector bases
$v_{ij}^{(n)}$
Diffusivity tensor
$d_{ij} = \sum_{n=1}^{N_v} f_n v_{ij}^{(n)}$
$h_i = d_{ij}\vartheta_j$
Scaled THF
Loss
$\mathcal{L}(h_i)$
Input
invariants
$\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_m$
$\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_r$
Global inputs
y_w, Re, Pr
Vector basis functions
f_n

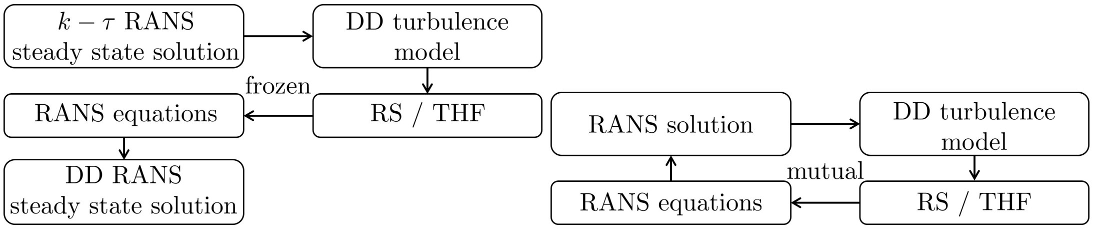
Fig. 4. A vector basis neural network architecture.

II.E. Training and Prediction Workflows

Figure 5 shows the different coupling strategies that can be employed for the DD RANS turbulence models. Figure 5(a) is the "frozen" coupling (adopted throughout this paper): where steady-state RANS quantities are used as inputs to predict RS and THF from DNS only once. Then the RANS Eqs. (1) are relaxed around constant (frozen) RS and THF values to obtain "improved" RANS results.

Figure 5(b) shows the "mutual" coupling strategy, when DD closures are invoked at each time step / iteration during a simulation. Such a framework requires using DNS inputs, which, on the one hand, makes the training easier (no matching RANS database is needed), but, on the other hand, convergence and numerical stability are difficult to achieve.

Mutual coupling is a more general approach, as the relationship between baseline RANS inputs and DNS targets in frozen coupling is not guaranteed. Although mutual coupling worked successfully for modeling of eddy viscosity [36], the prediction of RS was numerically unstable. There are probably two reasons of the failure that are explained in [37]: (1) explicit representation of RS and (2) statistical convergence error in the data. While strategy of Banko et al. [37] to mitigate these issues (semi-implicit treatment and filtering of RS) was successful, it did not work in our case. We also notice that frozen coupling is more numerically efficient as it requires only a single propagation through the NNs.



(a) (b)

Fig. 5. Coupling strategies: (a) frozen; (b) mutual.

The adopted training workflow schematic is shown in Figure 6. It requires a high-fidelity DNS database for its targets (RS $\overline{u_i'T'}$ and THF $\overline{u_i'u_j'}$) and a corresponding RANS database for inputs (spatial velocity derivatives $\frac{\partial \overline{u}_i}{\partial x_j}$ and temperature derivatives $\frac{\partial \overline{T}}{\partial x_j}$) and scaling quantities (TKE k and turbulent timescale τ). "Global" inputs (y_w, Re, and Pr) must also be provided.

The targets are subject to scaling according to Eqs. (4) and (12). The inputs are scaled using Eqs. (5), (6) and (10) with subsequent calculation of the invariants and bases. To equalize the importance of input features (*e.g.*, Re $\sim 10^3 - 10^4$, while $y_w \sim 0 - 10^0$), they are normalized in the ranges $[0, 1]$ (for non-negative quantities) and $[-1, 1]$ for other quantities.

The NNs are separately trained (RS is decoupled from THF in forced convection flows) using the `Pytorch` library [38] for `Python` and then saved as serializable `torchscrpit` models to be efficiently invoked in a `C++` user defined function[a].

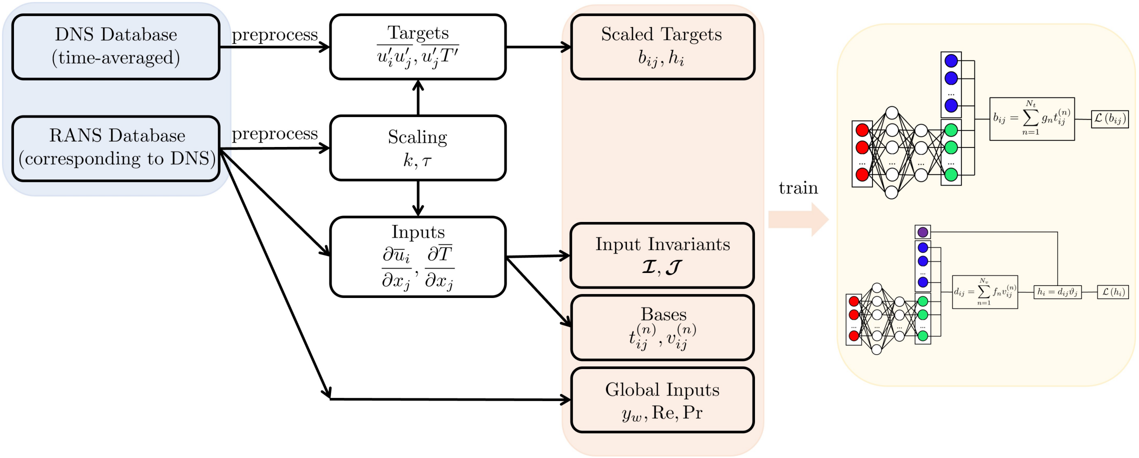

DNS Database
(time-averaged)
RANS Database
(corresponding to DNS)
preprocess
preprocess
Targets
$\overline{u_i'u_j'}, \overline{u_j'T'}$
Scaling
k, τ
Inputs
$\frac{\partial \overline{u}_i}{\partial x_j}, \frac{\partial \overline{T}}{\partial x_j}$
Scaled Targets
b_{ij}, h_i
Input Invariants
$\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{J}}$
Bases
$t_{ij}^{(n)}, v_{ij}^{(n)}$
Global Inputs
y_w, Re, Pr
train
$b_{ij} = \sum_{n=1}^{N_t} g_n t_{ij}^{(n)}$
$\mathcal{L}\left(b_{ij}\right)$
$d_{ij} = \sum_{n=1}^{N_v} f_n v_{ij}^{(n)}$
$h_i = d_{ij}\vartheta_j$
$\mathcal{L}\left(h_i\right)$

Fig. 6. Training workflow.

The prediction workflow is shown in Figure 7. Since NNs are trained to predict DNS quantities based on RANS input features, one first needs to converge a RANS simulation to steady state. After the extraction of the inputs and scaling parameters, invariants and bases are to be computed. Note that the RANS simulations can be performed on any mesh; though a mesh convergence study might be useful to estimate necessary mesh refinement.

After prediction, b_{ij} and h_i are subjects to rescaling by the RANS quantities k and τ. Finally, one needs to compute the spatial derivatives of RS ($\frac{\partial a_{ij}}{\partial x_j}$) and THF ($\frac{\partial \overline{u_j'T'}}{\partial x_j}$) and add them as an explicit body force and heat source terns in the RANS Eqs. (1). Here $a_{ij} = 2kb_{ij}$ is the anisotropic tensor, as the isotropic part $\frac{2}{3}k\delta_{ij}$ does not influence the mean flow quantities (can be absorbed in the modified pressure).

[a] see an example on GitHub

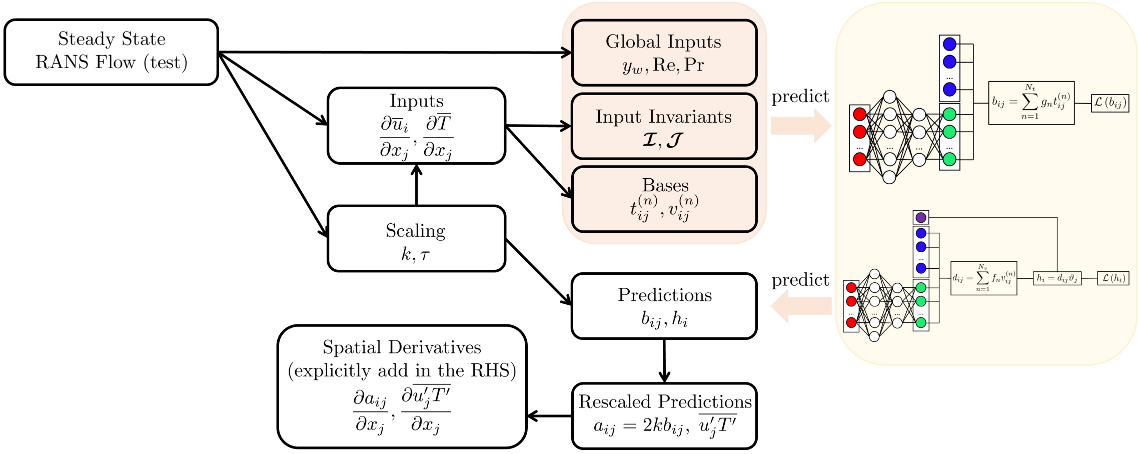

Steady State
RANS Flow (test)
Inputs
$\frac{\partial \overline{u}_i}{\partial x_j}, \frac{\partial \overline{T}}{\partial x_j}$
Scaling
k, τ
Global Inputs
y_w, Re, Pr
Input Invariants
$\mathcal{I}, \mathcal{J}$
Bases
$t_{ij}^{(n)}, v_{ij}^{(n)}$
predict
$b_{ij} = \sum_{n=1}^{N_t} g_n t_{ij}^{(n)}$
$\mathcal{L}(b_{ij})$
predict
$d_{ij} = \sum_{n=1}^{N_v} f_n v_{ij}^{(n)}$
$h_i = d_{ij}\vartheta_j$
$\mathcal{L}(h_i)$
Predictions
b_{ij}, h_i
Rescaled Predictions
$a_{ij} = 2kb_{ij}, \overline{u_j'T'}$
Spatial Derivatives
(explicitly add in the RHS)
$\frac{\partial a_{ij}}{\partial x_j}, \frac{\partial \overline{u_j'T'}}{\partial x_j}$

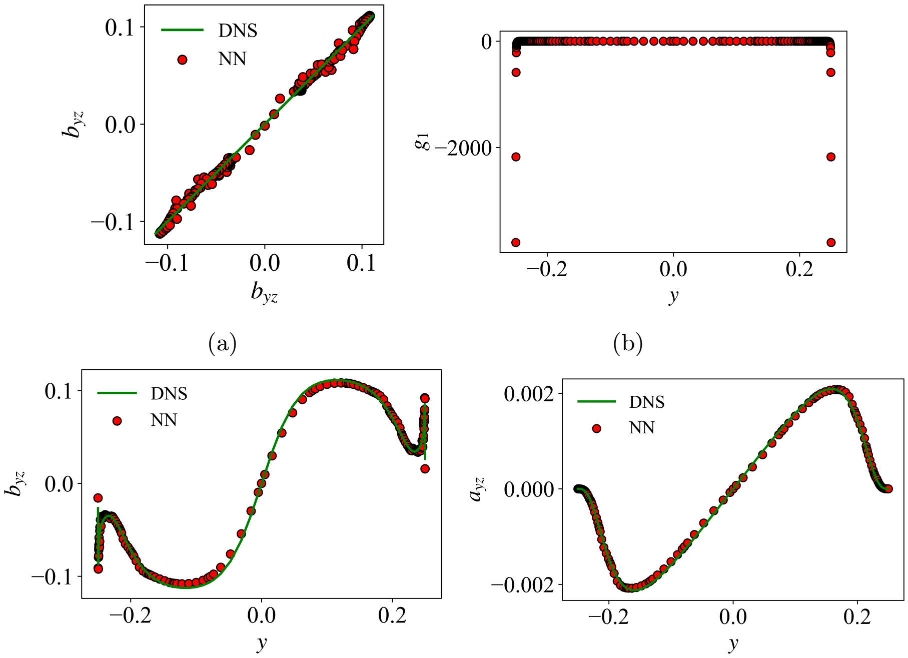
Fig. 7. Prediction workflow.

III. RESULTS

III.A. Reynolds Stress Model Test

A priori test results for the RS model are shown in Figure 8 (for FliBe, Re=4000) and Figure 9 (for Na, Re = 15000). One can see the accuracy is good for both cases. It is interesting to note how function g_1 changes near the walls: recall that traditional RANS models (including the $k-\tau$) assume $g_1 = \text{const}$.

A posteriori test results for the RS model are shown in Figure 10 (for FliBe, Re = 4000) and Figure 11 (for Na, Re = 15000). In Figure 10(a) one can see that the results of the $k-\tau$ model for the anisotropic component a_{yz} are significantly different from the DNS data, while the DD predictions are accurate. At the same time, one of the issues with DD point-wise predictions is the smoothness requirement: in Figure 10(b) one can see that the computed derivative (added as a body force) has some spikes compared to the relatively smooth DNS data. Fortunately, this does not cause any convergence issues and the resulting velocity is smooth (Figures 10(c) and 10(d)). To reduce the non-smoothness, Fiore et al. [17] introduced a regularization term in the loss function that penalizes non-smoothness of finite-difference derivatives of the predicted quantities. This strategy requires keeping track of spatial location during the training (to compute the finite difference approximation) and might be adopted in future work. In Figures 10(c) and 10(d) it can be seen that the resulting velocity of the DD RANS simulation is closer to the DNS data than the results obtained with the $k-\tau$ model. Similar conclusions apply to the Na, Re = 15000 dataset, Figure 11



(a)

(b)

(c)

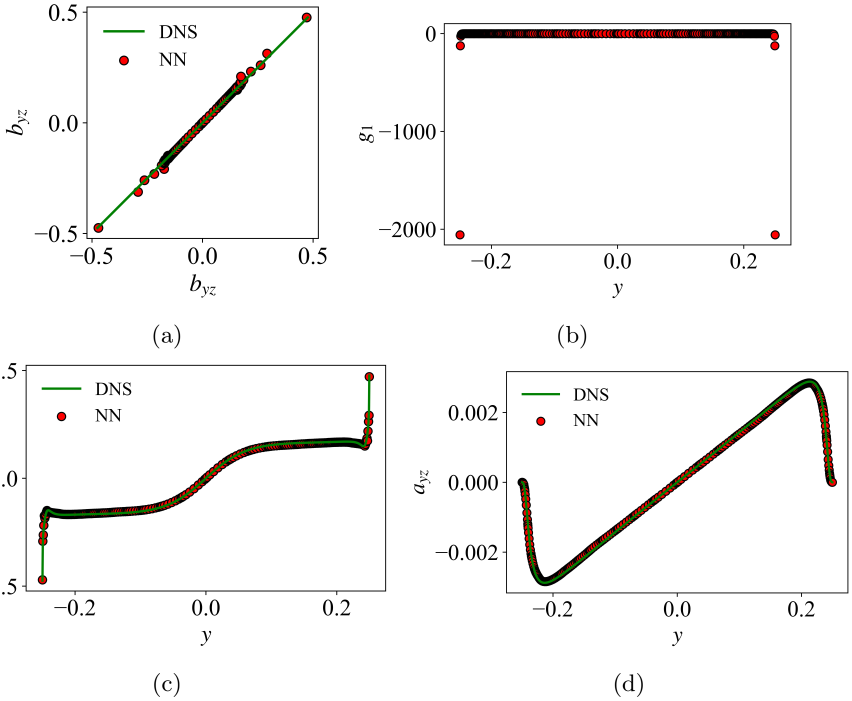
(d)

Fig. 8. *A priori* test of the RS model for FliBe, Re $= 4000$ dataset: (a) b_{yz} (predictions vs. DNS data); (b) profile of g_1; (c) profile of b_{yz}; (d) profile of $a_{yz} = 2kb_{yz}$.

(note that the $k - \tau$ model is more accurate for this dataset with higher Re).

III.B. Turbulent Heat Flux Model Test

A posteriori test results for the THF model are shown in Figures 12 and 13 (for FliBe, Re=4000) and Figures 14 and 15 (for Na, Re $=$ 15000). The upper figures show DNS data; the middle ones are predictions, and the bottom ones are errors in the predictions. The black curves are the local profiles. Note that the channel is stretched vertically for better visualization. The difference in THF for these high- and low-Pr fluids is significant: the developing region for FliBe is much longer ($z \approx 45$) compared to Na ($z \approx 15$) and the THF values for FliBe are an order of magnitude lower. The wall-normal component of THF $\overline{u'_y T'}$ has a qualitatively similar behavior with the $k - \tau$ eddy viscosity ν_t (not shown); at the same time, the derivative of the stream component $\frac{\overline{u'_z T'}}{\partial z}$ vanishes there. This suggests that flow in the developed region can be accurately modeled using the simple gradient diffusion hypothesis given an appropriate turbulent Prandtl number Pr_t. The error is expected to be higher in the developing region.



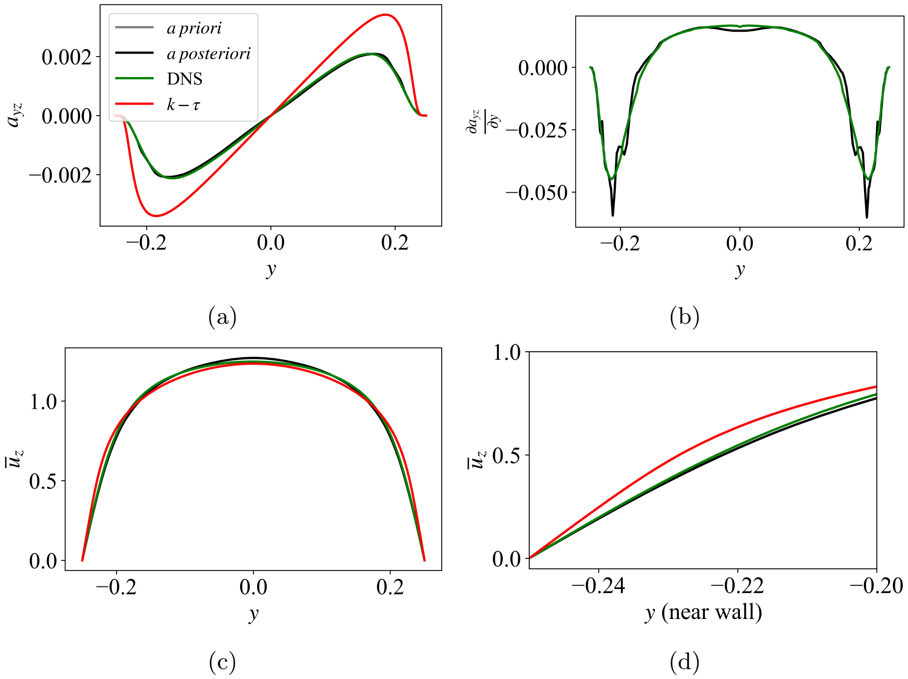
(a)

(b)

(c)

(d)

Fig. 9. *A priori* test of the RS model for Na, Re = 15000 dataset: (a) b_{yz} (predictions versus DNS data); (b) profile of g_1; (c) profile of b_{yz}; (d) profile of $a_{yz} = 2kb_{yz}$.



(a)

(b)

(c)

(d)

Fig. 10. *A posteriori* test of the RS model for FliBe, Re = 4000 dataset. Profiles of: (a) a_{yz}; (b) $\frac{\partial a_{yz}}{\partial y}$; (c) u_z; (d) u_z (near wall).

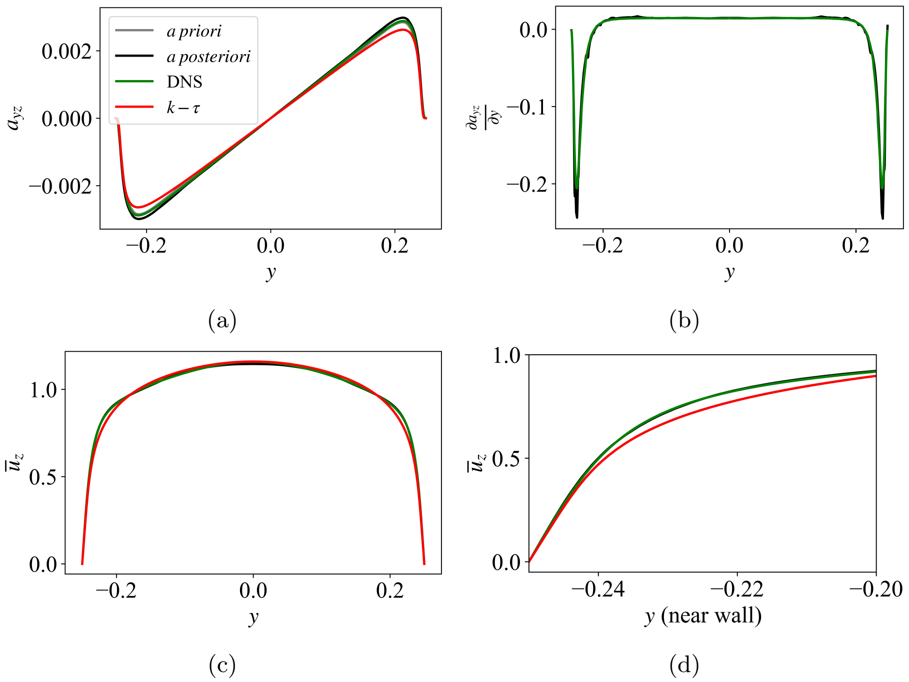


Fig. 11. *A posteriori* test of the RS model for Na, Re $=$ 15000 dataset. Profiles of: (a) a_{yz}; (b) $\frac{\partial a_{yz}}{\partial y}$; (c) u_z; (d) u_z (near wall).

One can see that the THF predictions are accurate enough to obtain a temperature closer to the DNS data than the $k-\tau$ model with $\mathrm{Pr}_t = 1$, Figures 16 and 17. By tuning Pr_t one can get accurate temperature predictions (RANS simulations were also performed for $\mathrm{Pr}_t = 0.85$ and $\mathrm{Pr}_t = 1.2$). One can see that a higher Pr_t allows for more accurate predictions to be obtained: a higher $\mathrm{Pr}_t = \frac{\nu_t}{\alpha_t}$ means a smaller turbulent diffusivity α_t (given a fixed turbulent viscosity ν_t obtained using the $k-\tau$ turbulence model). This increases the resistance to heat transfer; i.e. the temperature gradients increase and the difference between the wall temperature and the bulk temperature increases. Therefore, one will observe an increase in temperature near the wall (currently, it is underpredicted). This discussion demonstrates the difficulty one faces when models such flows using the gradient diffusion hypothesis: Pr_t is not known *a priori*.

The lack of information on the values of Pr_t was one of the motivations to set $\mathrm{Pr}_t = 1$ in the baseline RANS simulations: having a complete Reynolds analogy to relate turbulent momentum and heat transfer for fluids such as Na, Pb, unitary-Pr, and FliBe.

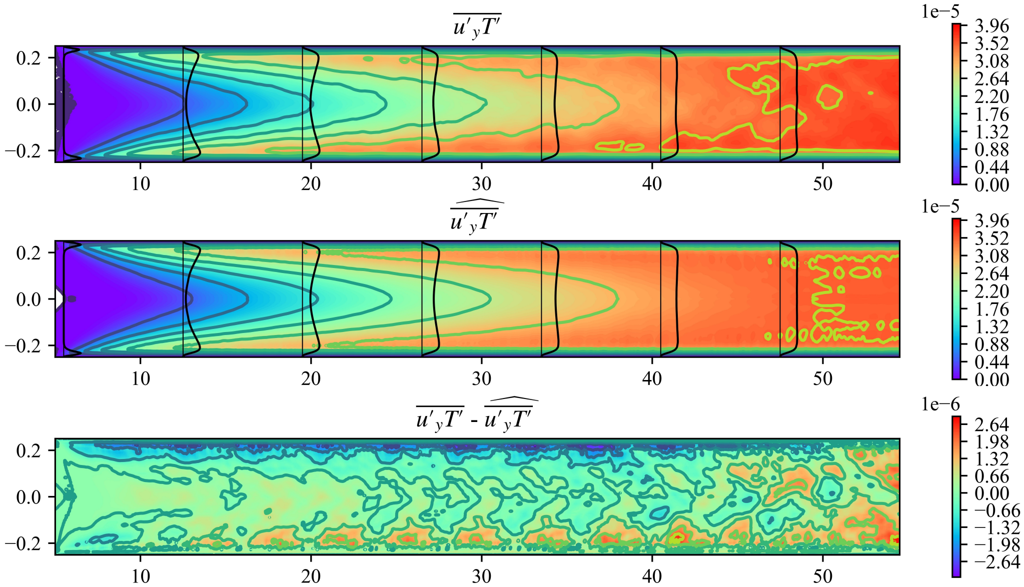


Fig. 12. *A posteriori* test results for the THF model for FliBe, Re = 4000 dataset ($\overline{u'_y T'}$).

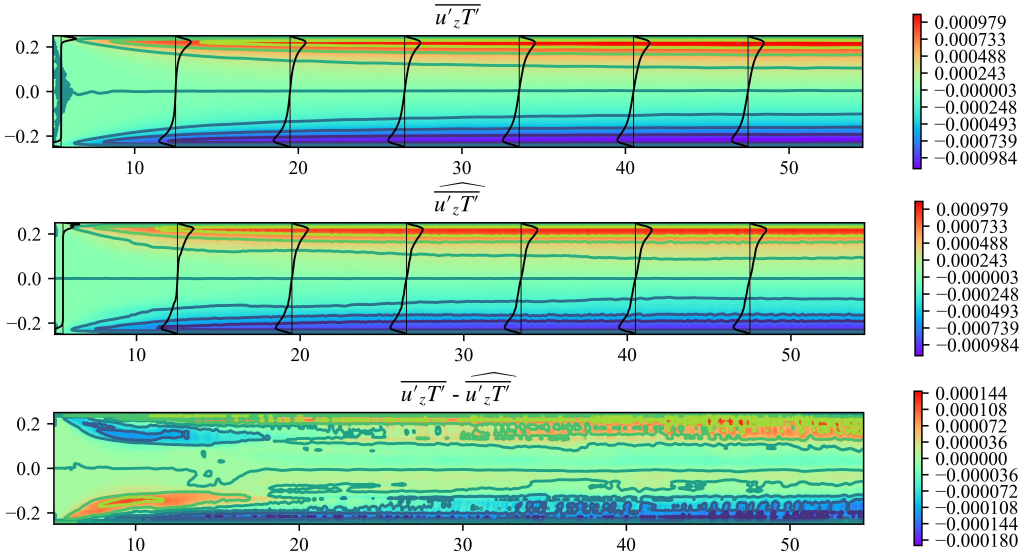


Fig. 13. *A posteriori* test results for the THF model for FliBe, Re = 4000 dataset ($\overline{u'_z T'}$).

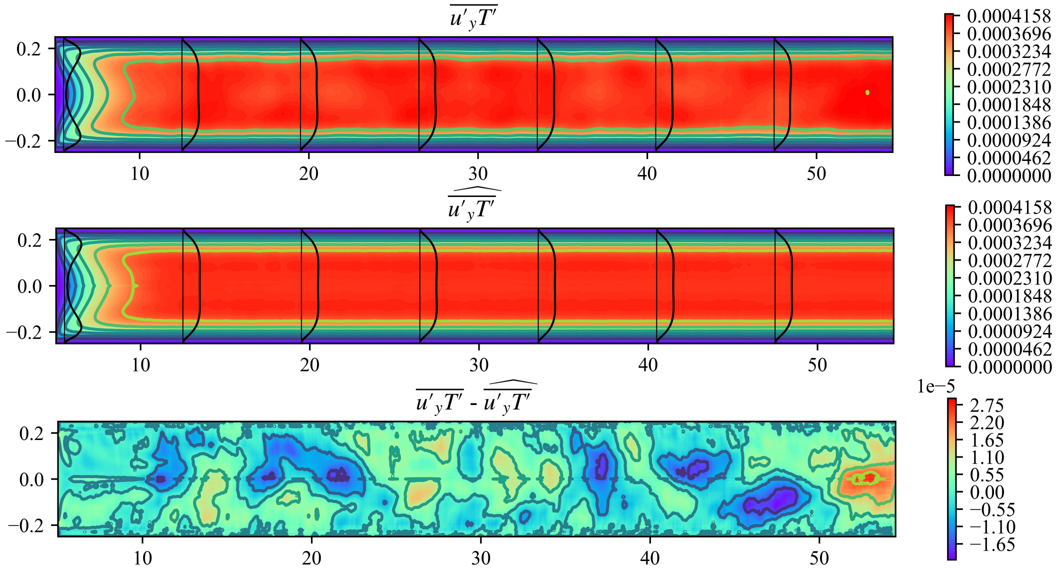


Fig. 14. *A posteriori* test results for the THF model for Na, Re = 15000 dataset ($\overline{u'_y T'}$).

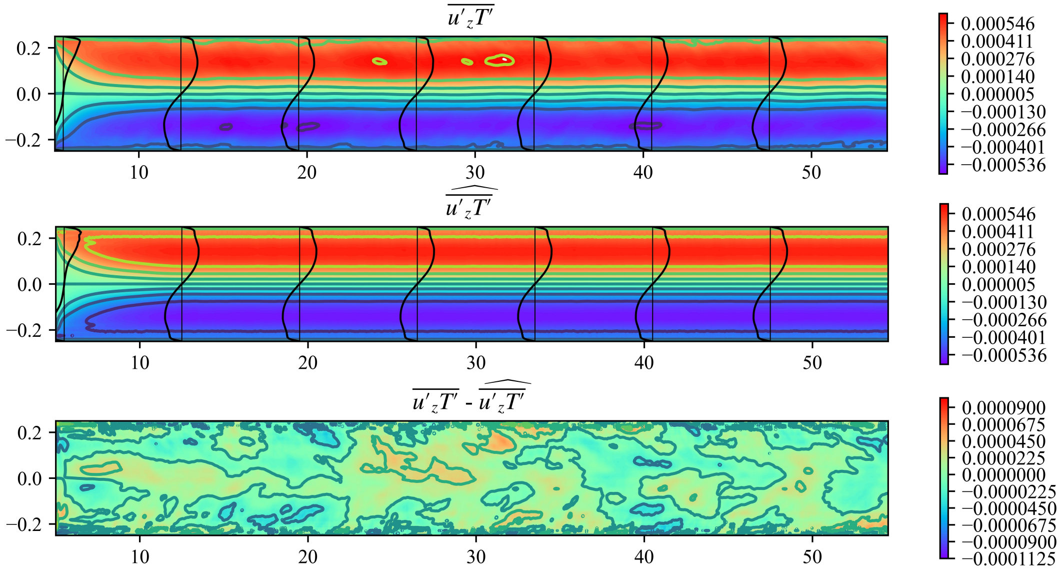


Fig. 15. *A posteriori* test results for the THF model for Na, Re = 15000 dataset ($\overline{u'_z T'}$).

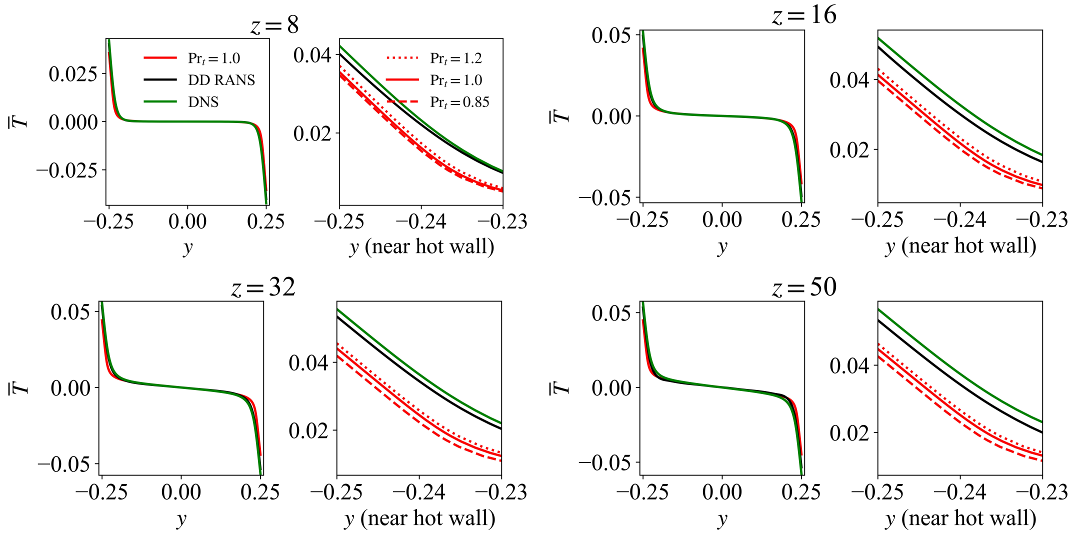


Fig. 16. Temperature profiles at different z for FliBe, Re = 4000 dataset.

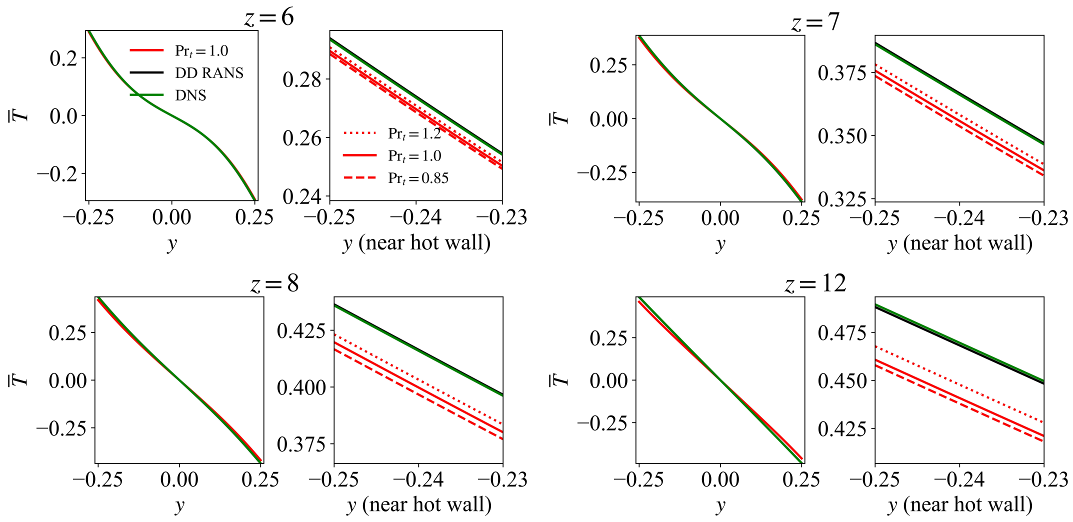
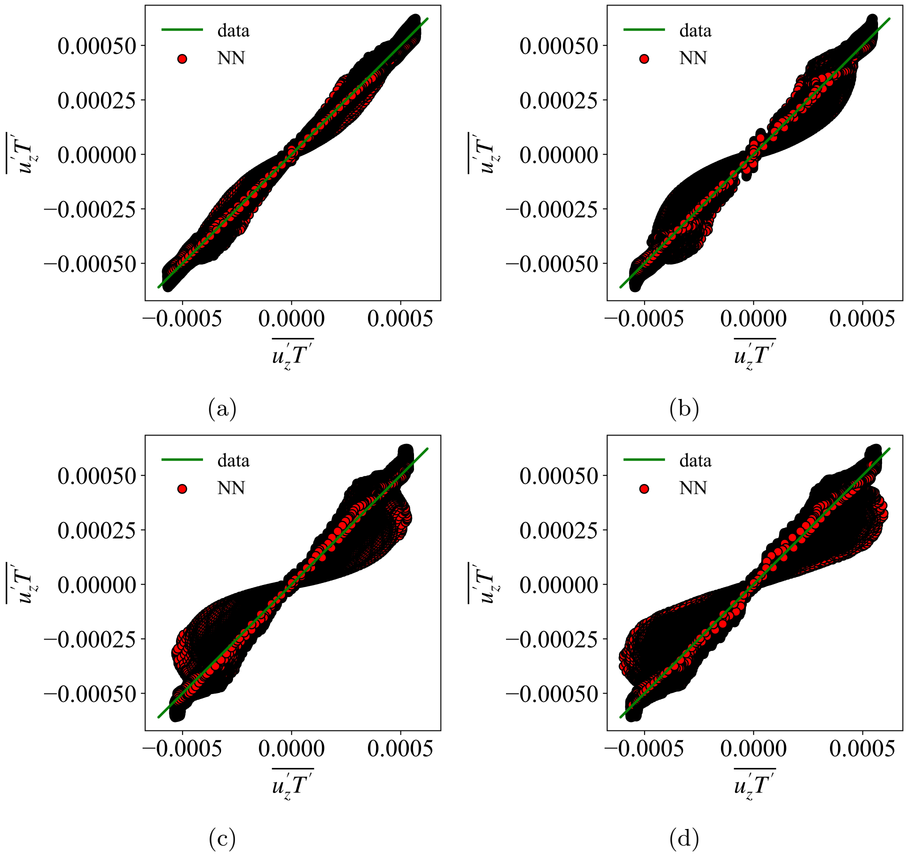


Fig. 17. Temperature profiles at different z for Na, Re = 15000 dataset.

III.C. On the development of a multi-fluid THF model

In the previous subsection, the THF models were trained separately using only FliBe or Na datasets. This section briefly describes the results when the datasets of different fluids were combined for training. From Section III.B it is obvious that the thermal behavior of high- and low-Pr fluids is quite different, which is a challenge for the development of multi-fluid models.

A priori test results for the THF models are shown in Figure 18. Although the accuracy of the wall-normal THF $\overline{u_y'T'}$ does not deteriorate (not shown), the accuracy of the streamwise THF $\overline{u_z'T'}$ deteriorates significantly if different fluids are used for training. Figure 18(a) shows the accuracy for Na, Re = 15000 dataset if only Na datasets are used for training; Figure 18(b) shows the accuracy if the Na and Pb datasets are used; Figure 18(c) shows the accuracy if the Na, Pb, and Un datasets are used; Figure 18(d) shows the accuracy if the Na, Pb, Un and FliBe datasets are used for training. Therefore, we conclude that development of a multi-fluid model is currently possible only at the cost of accuracy for the streamwise THF component. Fiore et al. [17] reported good accuracy for different fluids (Pr = 0.01-0.71); however, a four-equation model was used for scaling purposes.



(a)

(b)

(c)

(d)

Fig. 18. *A priori* THF model test for Na, Re = 15000 dataset (using different fluids for training): (a) only Na datasets; (b) Na and Pb datasets; (c) Na, Pb, and Un datasets; (d) Na, Pb, Un, and FliBe.

IV. CONCLUSION

In this paper, the DD RS and THF turbulence closures were developed. DNS and RANS data for forced convection flows in a vertical planar channel mimicking a reactor downcomer geometry are used for training and testing purposes. Different coolants relevant to advanced nuclear reactors are considered: sodium, lead, unitary Pr number fluid, and molten salt. Both the RS and THF models are based on invariant NN architectures (though, in the considered statistically 1D flow the RS model shrinks to a linear eddy viscosity formulation). It is shown that the models yield more accurate temperature predictions without *a priori* knowledge of the turbulent Prandtl number. The models are compared with the reference DNS data; more extensive VV&UQ activities will follow.

The possibility of developing a multi-fluid THF model is also analyzed; it is shown that the use of different fluids for training of a single model deteriorates the accuracy of the streamwise THF. Perhaps a more appropriate scaling using a four-equation model can improve this. Future work will focus on mixed convection flow conditions more directly relevant to nuclear reactor geometries.

NOMENCLATURE

Latin symbols	
a_{ij}	anisotropic tensor
b_{ij}	scaled anisotropic tensor
$d_{ij} = \mathrm{D}$	diffusivity tensor
f_n	vector basis functions
g_n	tensor basis functions
h_i	scaled turbulent heat flux
$\boldsymbol{\mathcal{I}}$	invariants for TBNN
$\boldsymbol{\mathcal{J}}$	additional invariants for VBNN
k	turbulent kinetic energy
k_T	temperature variance
$L_i = \{L_x, L_y, L_z\}$	channel dimensions
$\mathcal{L}$	loss
N_t	number of tensor bases
N_v	number of vector bases
p	pressure
q''	wall heat flux
$s_{ij} = \mathrm{S}$	scaled symmetric tensor
t	time
$t_{ij}^{(n)} = \mathrm{T}^{(n)}$	tensor bases
T	temperature
$u_i = \{u_x, u_y, u_z\}$	velocity vector
$\overline{u_j' T'}$	turbulent heat flux vector
$\overline{u_i' u_j'}$	Reynolds stress tensor
$v_i^{(n)}$	vector bases
$v_{ij}^{(n)}$	vector bases (with postponed multiplication by ϑ_j)
$w_{ij} = \mathrm{W}$	scaled antisymmetric tensor
$x_i = \{x, y, z\}$	coordinate vector
y_w	distance to the wall
Greek symbols	
δ_{ij}	Kronecker delta
ε	rate of dissipation of TKE
ε_T	thermal dissipation rate
$\vartheta_j = \boldsymbol{\vartheta}$	scaled temperature gradient
τ	turbulent time scale
Non-dimensional criteria	
Re	Reynolds number
Pr	Prandtl number
Ri	Richardson number
Subscripts	
a	antisymmetric
in	inlet
s	symmetric
t	turbulent
Other notations	
$\mathrm{tr}(\,\cdot\,)$	trace of matrix
$\cdot^{\intercal}$	transposed matrix
$\overline{\cdot}$	Reynolds-averaged component
$\cdot'$	fluctuating components
$\widehat{\cdot}$	predicted quantity

ACKNOWLEDGMENTS

This work was funded by a U.S. Department of Energy Integrated Research Project entitled "Center of Excellence for Thermal-Fluids Applications in Nuclear Energy: Establishing the knowledgebase for thermal-hydraulic multiscale simulation to accelerate the deployment of advanced reactors – IRP-NEAMS-1.1: Thermal-Fluids Applications in Nuclear Energy". The authors are also grateful to all IRP-NEAMS-1.1 Challenge Problem 1 participants for valuable discussion and recommendations.

REFERENCES

[1] B. E. Launder and D. B. Spalding, “The numerical computation of turbulent flows,” *Computer Methods in Applied Mechanics and Engineering*, **3**, *2*, 269 (1974); https://doi.org/10.1016/0045-7825(74)90029-2.

[2] S. Thangam, R. Abid, and C. G. Speziale, “Application of a new k-tau model to near wall turbulent flows,” *AIAA Journal*, **30**, *2*, 552 (1992); https://doi.org/10.2514/3.10952.

[3] H. Xiao and P. Cinnella, “Quantification of model uncertainty in RANS simulations: A review,” *Progress in Aerospace Sciences*, **108**, 1 (2019); https://doi.org/10.1016/j.paerosci.2018.10.001.

[4] A. S. Iskhakov, C.-K. Tai, I. Bolotnov, and N. T. Dinh, “A perspective on data-driven coarse grid modeling for system level thermal hydraulics,” *Nuclear Science and Engineering*, 1–16 (2022); https://doi.org/10.1080/00295639.2022.2107864.

[5] S. Brunton, J. Proctor, and N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *The Proceedings of the National Academy of Sciences*, **113**, *15*, 3932 (2016); https://doi.org/10.1073/pnas.15173841.

[6] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” (2020); https://doi.org/10.48550/arXiv.2107.07562.

[7] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. Karniadakis, “Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators,” *Nature Machine Intelligence*, **3**, 218 (2021); https://doi.org/10.1038/s42256-021-00302-5.

[8] E. J. Parish and K. Duraisamy, “A paradigm for data-driven predictive modeling using field inversion and machine learning,” *Journal of Computational Physics*, **305**, 758 (2016); https://doi.org/10.1016/j.jcp.2015.11.012.

[9] K. Duraisamy, “Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence,” *Physical Review Fluids*, **6**, 050504 (2021); 10.1103/PhysRevFluids.6.050504.

[10] R. D. Sandberg and Y. Zhao, “Machine-learning for turbulence and heat-flux model development: A review of challenges associated with distinct physical phenomena and progress to date,” *International Journal of Heat and Fluid Flow*, **95**, 108983 (2022); https://doi.org/10.1016/j.ijheatfluidflow.2022.108983.

[11] I. A. Bolotnov, N. T. Dinh, A. S. Iskhakov, C.-K. Tai, E. Merzari, T. Nguyen, E. Baglietto, R. Wiser, Y. Hassan, R. Hu, D. Shaver, S. Schunert, H. Choi, B. Jackson, M. Martin, and M. Dzodzo, “Challenge problem 1: benchmark specifications for the direct numerical simulation of canonical flows,” ANL/NSE-21/11, Argonne National Laboratory (2021); https://doi.org/10.2172/1873405.

[12] K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annual Review of Fluid Mechanics*, **51**, *1*, 357 (2019); 10.1146/annurev-fluid-010518-040547.

[13] J. Ling, M. Barone, W. Davis, K. Chowdhary, and J. Fike, “Development of machine learning models for turbulent wall pressure fluctuations,” *55th AIAA Aerospace Sciences Meeting*, 0755 (2017); https://doi.org/10.2514/6.2017-0755.

[14] 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); https://doi.org/10.1017/jfm.2016.615.

[15] X. Xu, A. S. H. Ooi, and R. D. Sandberg, “Reynolds-averaged stress and scalar-flux closures via symbolic regression for vertical natural convection,” *International Journal of Heat and Fluid Flow*, **96**, 108981 (2022); https://doi.org/10.1016/j.ijheatfluidflow.2022.108981.

[16] Y. Liu, R. Hu, L. Zou, and D. Nunez, “SAM-ML: Integrating data-driven closure with nuclear system code SAM for improved modeling capability,” *Nuclear Engineering and Design*, **400**, 112059 (2022); https://doi.org/10.1016/j.nucengdes.2022.112059.

[17] M. Fiore, L. Koloszar, M. Miguel Alfonso, M. D., and B. Yann, “Physics-constrained machine learning for thermal turbulence modelling at low Prandtl numbers,” (2022); https://arxiv.org/abs/2201.06301.

[18] 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**, A27 (2021); https://doi.org/10.1017/jfm.2020.820.

[19] J. Ling, R. Jones, and J. Templeton, “Machine learning strategies for systems with invariance properties,” *Journal of Computational Physics*, **318**, 22 (2016); https://doi.org/10.1016/j.jcp.2016.05.003.

[20] N. Geneva and N. Zabaras, “Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks,” *Journal of Computational Physics*, **383**, 125 (2019); https://doi.org/10.1016/j.jcp.2019.01.021.

[21] H. D. Akolekar, J. Weatheritt, N. Hutchins, R. D. Sandberg, G. Laskowski, and V. Michelassi, “Development and use of machine-learnt algebraic Reynolds stress models for enhanced prediction of wake mixing in low-pressure turbines,” *Journal of Turbomachinery*, **141**, *4* (2019); https://doi.org/10.1115/1.4041753.

[22] P. M. Milani, J. Ling, and J. K. Eaton, “On the generality of tensor basis neural networks for turbulent scalar flux modeling,” *International Communications in Heat and Mass Transfer*, **128**, 105626 (2021); https://doi.org/10.1016/j.icheatmasstransfer.2021.105626.

[23] X. Xu, F. Waschkowski, A. S. H. Ooi, and R. D. Sandberg, “Towards robust and accurate Reynolds-averaged closures for natural convection via multi-objective CFD-driven machine learning,” *International Journal of Heat and Mass Transfer*, **187**, 122557 (2022); https://doi.org/10.1016/j.ijheatmasstransfer.2022.122557.

[24] C.-K. Tai, T. Nguyen, A. Iskhakov, E. Merzari, N. Dinh, and I. Bolotnov, “Direct Numerical Simulation of Low and Unitary Prandtl Number Fluids in Reactor Downcomer Geometry,” *Nuclear Technology (submitted for publication)* (2023); https://arxiv.org/abs/2204.00430.

[25] T. Nguyen, E. Merzari, C.-K. Tai, and I. A. Bolotnov, “Direct numerical simulation of high Prandtl number fluid flow in the downcomer of an advanced reactor,” (2022); https://arxiv.org/abs/2203.14157.

[26] P. Fischer, S. Kerkemeier, M. Min, Y.-H. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, and T. Warburton, “NekRS, a GPU-accelerated spectral element Navier-Stokes solver,” (2021); https://arxiv.org/abs/2104.05829.

[27] T. Nguyen, E. Merzari, C.-K. Tai, I. Bolotnov, and B. Jackson, “Toward improved correlations for mixed convection in the downcomer of molten salt reactors,” *Nuclear Technology (submitted for publication)* (2023).

[28] S. Dong, G. E. Karniadakis, and C. Chryssostomidis, “A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains,” *Journal of Computational Physics*, **261**, 83 (2014); https://doi.org/10.1016/j.jcp.2013.12.042.

[29] S. B. Pope, *Turbulent Flows*, Cambridge University Press (2000); https://doi.org/10.1017/CBO9780511840531.

[30] S. B. Pope, “A more general effective-viscosity hypothesis,” *Journal of Fluid Mechanics*, **72**, *2*, 331–340 (1975); https://doi.org/10.1017/S0022112075003382.

[31] Q.-S. Zheng, “Theory of representations for tensor functions - a unified invariant approach to constitutive equations,” *Applied Mechanics Reviews*, **47**, *11*, 545 (1994); https://doi.org/10.1115/1.3111066.

[32] T.-H. Shih and J. L. Lumley, “Remarks on turbulent constitutive relations,” *Mathematical and Computer Modelling*, **18**, *2*, 9 (1993); https://doi.org/10.1016/0895-7177(93)90002-G.

[33] S. Manservisi and F. Menghini, “A CFD four parameter heat transfer turbulence model for engineering applications in heavy liquid metals,” *International Journal of Heat and Mass Transfer*, **69**, 312 (2014); https://doi.org/10.1016/j.ijheatmasstransfer.2013.10.017.

[34] J. Weatheritt, Y. Zhao, R. D. Sandberg, S. Mizukami, and K. Tanimoto, “Data-driven scalar-flux model development with application to jet in cross flow,” *International Journal of Heat and Mass Transfer*, **147**, 118931 (2020); https://doi.org/10.1016/j.ijheatmasstransfer.2019.118931.

[35] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, **3**, 422–440 (2021); https://doi.org/10.1038/s42254-021-00314-5.

[36] C.-K. Tai, A. S. Iskhakov, N. T. Dinh, and I. A. Bolotnov, “Towards data-driven turbulence modeling of mixed convection in advanced reactors using DNS data,” *The 19th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-19)*, 36432, Brussels, Belgium (2022).

[37] Banko, A.J. and Eaton J.K., “Estimating performance bounds of machine-learning Reynolds-stress models via optimal tensor basis expansions,” *Center for Turbulence Research Annual Research Briefs* (2020)URL http://web.stanford.edu/group/ctr/ResBriefs/2020/19_Banko.pdf.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” (2019); https://arxiv.org/abs/1912.01703v1.