Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows

Pedro Stefanin Volpiani,[1,*] Morten Meyer,[1] Lucas Franceschini,[1] Julien Dandois,[1]
Florent Renac,[2] Emeric Martin,[2] Olivier Marquet,[1] and Denis Sipp[1]
[1]*ONERA, The French Aerospace Lab, 8 Rue des Vertugadins, 92190 Meudon, France*
[2]*ONERA, The French Aerospace Lab, 29 Avenue de la Division Leclerc, 92320 Chatillon, France*

(Received 10 December 2020; accepted 20 May 2021; published 16 June 2021)

Most widely used Reynolds-averaged Navier-Stokes (RANS) models employ the Boussinesq approximation, which assumes a linear relationship between the turbulent Reynolds stresses and the mean-velocity gradient tensor. This assumption, which can be very stringent, is more suited for simple shear-flows and is regarded as an important shortcoming for the improvement of the representation of turbulence in complex geometries. Correction of the local turbulence length scales, as achieved, for example, by the introduction of a correction term in the eddy-viscosity equation of a Spalart-Allmaras model, then indeed only allows relatively small corrections to the base-line model and may not be sufficiently flexible to adapt to common situations where the Reynolds stresses are not aligned with the velocity gradient tensor. For the variational data-assimilation step, we consider a vectorial source correction term in the momentum equations (output quantity) together with the Spalart-Allmaras model, which allows full flexibility to adapt to any mean-flow topology. We show how the machine-learning framework should be adapted, in particular with respect to the vectorial nature of the correction term and given invariance properties. As for the input quantities, we discuss the impact of considering either local quantities for the nondimensionalization or global ones that characterize the configuration. We showcase the procedure on the periodic hill configuration, for which a rich direct numerical simulation database is available in the literature: in particular, the availability of the full mean-flow solution for a range of Reynolds numbers and geometries ensures identification of "physical" correction terms and allows learning of turbulence models that accurately extrapolate to new Reynolds numbers and geometries.

DOI: 10.1103/PhysRevFluids.6.064607

I. INTRODUCTION

With the accelerating developments and the availability of modern machine-learning tools, the number of computational fluid dynamics (CFD) applications involving data-driven methods has skyrocketed in recent years. Machine-learning techniques have been used on multiple occasions [1–9], especially to provide turbulence modeling for Reynolds-averaged Navier-Stokes (RANS) computations (see [10] and further discussion). RANS simulations remain an invaluable tool in engineering design, analysis, and optimization for high-Reynolds-number turbulent flows thanks to their low computational cost. In this approach, the mean statistical operator (also called the Reynolds operator) is applied to the governing equations of fluid dynamics, which results in the mean-flow equations. A mathematical model is then essential to close these equations and to describe the effect

[*] pedro.stefanin_volpiani@onera.fr

of unresolved scales (the Reynolds stresses) on the mean-flow solution. Such models generally involve a constitutive relation that connects the Reynolds stresses to the mean-flow field and one or two equations that govern the turbulent scales. The most common constitutive relation, called the Boussinesq approximation, assumes a linear relationship between the Reynolds stress tensor and the symmetric part of the mean-velocity gradient tensor. The Boussinesq approximation is most successful in simple shear flows [11]. However, in more general configurations, where flows are dominated by streamline curvature, one should resort to a more complex modeling of the Reynolds stresses, for example the explicit algebraic Reynolds stress model [12], or the more complex Reynolds stress model [13].

Data-augmented turbulence models may be obtained by compensating model form errors by training correction terms with high-fidelity data [14]. Ling and Templeton [15] introduced classifiers to detect regions of low turbulence model accuracy or model assumption violations using direct numerical simulation (DNS) and large eddy simulation (LES) solutions. Wu *et al.* [16] used a random forest model to estimate *a priori* the prediction confidence of RANS solutions, while Wang *et al.* [17] corrected the Reynolds stress tensor directly from mean flow features using machine learning (ML), which increased the accuracy of the computation even on other test cases. In a similar way, Ling *et al.* [18] introduced a method to correct the Reynolds stress by a neural network (NN) structure with embedded invariance properties of the tensor. The same concept was later extended to a more general, although expensive, framework by Wu *et al.* [19]. However, the above methods directly learn a correction from high-fidelity data. The constructed functions can therefore be inconsistent with RANS model structures. To ensure this consistency, data-assimilation (DA) techniques based on inverse methods [20] were used for the inference of model parameters, such as the field inversion approach developed by Parish and Duraisamy [21]. The method uses Bayesian inference and ML to improve RANS model predictions, and it was successfully applied to airfoils with separation [22]. It was more recently improved with respect to the numerical implementation from a two-stage, so called off-line, approach in which the DA is conducted in a first step and the ML in a second step, to an on-line method in several versions, where these stages are coupled [23].

The present article deals with an in-depth discussion on the choice of inputs and outputs that are machine-learned. The inputs are normalized quantities related to the state variable in the RANS model, and the outputs are correction fields that are applied to the baseline model.

The choice of the input features is crucial. Ling and Templeton [15] and Wang *et al.* [17] have introduced various quantities and normalization schemes to avoid feature domination and to improve the training behavior of the neural network. Normalization schemes allow all features to exhibit a maximum value of approximately 1 and a minimum value of around 0 if the feature is positive (or -1 if the feature is both positive and negative). In contrast to previous works, we introduce global parameters (a characteristic length and velocity scale of the configuration, for example) to achieve this normalization instead of local features (local velocities and theirs gradients, etc.). This will of course limit the generality of the data-driven model to similar configurations, characterized by the same global parameters. Yet, we believe that designing a model for such classes of flows is still useful, especially if the model performs significantly better when compared to the results with a normalization scheme based on local features. We aim in this article to compare a global and a local normalization scheme.

For the outputs, we start by noting that industrial applications have favored up to now the use of a simple linear modeling of the Reynolds stresses, the Boussinesq approximation, due to its low induced-cost and increased robustness. Yet, Franceschini *et al.* [24] recently showed, in the context of mean-flow data assimilation, that the Boussinesq approximation could induce rigid models that were able to only approximately reconstruct, through DA, simple mean flows, such as the one obtained in backward-facing steps that exhibit separation and pressure gradients. In simple shear-dominated flows, the approximation is in fact known to be "effective" (turbulence essentially enhances mixing in an isotropic way, which is compatible with a linear constitutive relation), but actually not even strictly valid (Monier *et al.* [25] showed that the Reynolds stresses were only roughly aligned with the local velocity gradient in turbulent boundary layers). In view of drastically improving the fidelity of RANS computations, it therefore seems reasonable to lift this constraint in the modeling of the Reynolds stresses. Yet, to maintain the cost and robustness of the data-augmented model at a reasonable level, a first step could be to consider a simple base-line linear constitutive relation and to machine-learn the part that is not linear with high-fidelity data.

We will consider, as in the data-assimilation studies of Foures *et al.* [20] and Symon *et al.* [26], a vectorial source correction term in the momentum equations of the Navier-Stokes equations. In such a situation, Franceschini *et al.* [24] showed that the data-assimilation procedure allowed an almost exact recovery of the mean-flow over a backward-facing step. Contrary to a scalar correction source term in the equations governing the turbulent scales, the vectorial momentum correction term allows for a much more flexible model to accommodate the prescribed mean-flow features. In the present work, we aim at demonstrating an off-line DA and ML procedure that is based on this vectorial momentum correction term. In this context, variational data assimilation is used to infer the vectorial source correction from high-fidelity numerical data. The machine-learning method is then applied to reconstruct this quantity from the local mean-flow features. It is worth noting that the quality of the DA reconstruction provides an upper bound for the quality of the resulting data-augmented models: if the quality of the assimilated solution is poor, so is the resulting augmented model prediction. We therefore hope that by considering a more flexible correction term, we may improve the quality of the resulting models. Also, we will solely use complete velocity field information for the data-assimilation procedure: in Refs. [20,24,26] it was indeed found that, only in such situations, the correction term had a physical meaning because it could favorably be compared to actual features of the Reynolds-Stress tensor. In the case of sparse measurements (such as velocity measurements along lines or global measurements such as lift or drag forces), the data-assimilation process is strongly underdetermined, and many correction terms are possible. Regularization terms which are similar to considering state-covariance information then need to be introduced.

As for the configurations, we consider the periodic-hill test case for which plenty of data from DNSs and/or LESs are available [27,28]. These configurations, which are characterized by Reynolds numbers (based on the height of the hills and the mean streamwise velocity atop the hills) between 2800 and 19 000, present a turbulent flow with free separation and reattachment points along with a significant recirculation region—a flow topology for which linear eddy viscosity models suffer from inaccuracies. Also, note that the usual turbulence models are calibrated at much higher Reynolds numbers, which by itself is a reason for inaccuracy. The databases offer well-defined flow conditions, which makes it especially suited as a benchmark case for learning and testing data-augmented turbulence models.

The outline of the paper is the following. First (Sec. II), we present the periodic hill configuration and the available simulation results. Then (Sec. III), we present the base-line turbulence model, here the Spalart-Allmaras model, and a comparison between predicted mean flows and reference solutions obtained in the literature. In Sec. IV, we introduce the vectorial source correction term in the RANS equations, and we present the variational data-assimilation framework that allows tuning this term to minimize the gap between the predicted field and reference fields given in the literature. We will check in particular the physical relevance of the tuned correction term by comparison with reference data from the high-fidelity solutions. In Sec. V, we introduce the neural networks, whose coefficients will be machine-learned to recover the data-assimilated vectorial source correction terms from local input features of the mean flow. We will describe these features in particular and discuss how to prescribe invariance of the predicted vectorial correction term when an arbitrary rotation of the base unit vectors is considered. Varying the Reynolds numbers and the shape of the hills, we analyze in detail the learning/validation/interpolation/extrapolation steps of the data-augmented turbulence models. In Sec. VI, we compare two sets of input features, the first based on global quantities and the second based on local ones. In Sec. VII, we offer a summary of the results and an outlook.

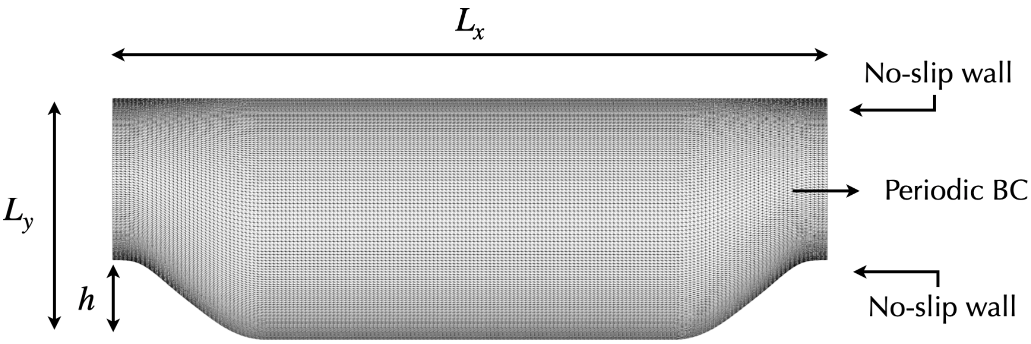

FIG. 1. Computational domain for the flow over the baseline geometry of periodic hills.

II. CONFIGURATION AND REFERENCE FLOW FIELD

The flow over periodically arranged hills in a channel has become a very useful benchmark test case to evaluate numerical methods, meshing strategies, wall modeling assumptions, and turbulence closures for RANS and LES [29–34]. High-fidelity simulations [35,36] brought out the complex flow features typical of this configuration. The flow exhibits separation from a curved surface and reattachment at the flat surface between the hills, unsteady shear layer surrounding the main recirculation bubble, and strong acceleration due to the geometry obstruction. Moreover, the extent of the recirculating bubble, dominated by large-scale energetic eddies with strong deformation and dynamics, is extremely difficult to describe within the RANS-modeling framework [37]. This configuration was also chosen thanks to the freely available database provided in the literature [27,28]. Recently, Gloerfelt and Cinnella [27] performed direct and large-eddy simulations for flows at Reynolds numbers varying from 2800 up to 37 000. Xiao *et al.* [28] extended the database by changing the geometry and keeping the Reynolds number at 5600.

The base periodic-hill geometry shown in Fig. 1 is represented as piecewise third-order polynomial functions [28]. It consists of a plane channel with curved constrictions of height h. The channel height is $L_y = 3.035h$. To take into account the new geometries simulated by Xiao *et al.* [28], the horizontal length is parametrized by a as $L_x = (3.858a + 5.142)h$, a varying from 0.8 to 1.2, with $a = 1$ representing the original geometry. Periodicity of the flow (velocity and pressure components) is assumed in the streamwise direction, in which the flow is driven by a constant body force. This body force is implemented as a uniform forcing term of strength f_1 in the x-momentum equation to ensure the specified bulk Reynolds number Re_b:

$$\mathrm{Re}_b = \frac{u_b h}{\nu} \quad \text{with} \quad u_b = \frac{1}{2.035h} \int_h^{3.035h} u_1(y)dy. \tag{1}$$

Each Reynolds number is characterized by a unique value of f_1, which is directly related to the mean streamwise force that is exerted on the upper and lower walls, $F_1 := \int_{\partial\Omega_{u,l}} (-pn_1 + \tau_{1j}n_j)dS$, through

$$f_1\Omega = F_1, \tag{2}$$

with Ω being the volume of the periodic region. Four different values of Reynolds numbers have been considered: $\mathrm{Re}_b = 2800$, 5600, 10 595, and 19 000, as well as three geometries: $a = 1$, 0.8, and 1.2. Table I summarizes all cases investigated in this study.

TABLE I. Summary of configurations studied in this work.

Case	Re_b	a	Simulations performed	High-fidelity data
PH-2800	2800	1	RANS, DA, NN-based RANS	[27]
PH-5600	5600	1	RANS, DA, NN-based RANS	[28]
PH-5600-0.8	5600	0.8	RANS, NN-based RANS	[28]
PH-5600-1.2	5600	1.2	RANS, NN-based RANS	[28]
PH-10595	10 595	1	RANS, DA, NN-based RANS	[27]
PH-19000	19 000	1	RANS, DA, NN-based RANS	[27]

III. REYNOLDS-AVERAGED NAVIER-STOKES EQUATIONS AND ASSESSMENT OF THE BOUSSINESQ HYPOTHESIS

The incompressible steady-state RANS equations for the mean velocity u_i and the mean pressure p are

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{3}$$

$$\frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial (2\nu S_{ij})}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \tag{4}$$

where $S_{ij} = (u_{i,j} + u_{j,i})/2$ is the mean strain rate tensor, ν is the molecular viscosity, and $f_i = f_i^{\mathrm{DNS}}$ is the constant streamwise body force. The term $\tau_{ij} = -\overline{u_i' u_j'}$ is the Reynolds stress tensor.

To obtain a predictive RANS model, we introduce a linear constitutive relation linking the Reynolds stress tensor to the symmetric mean-velocity gradient tensor:

$$\tau_{ij} = 2\nu_t S_{ij} - \tfrac{2}{3}k\delta_{ij} \quad \text{with} \quad k = \tfrac{1}{2}\overline{u_i' u_i'}. \tag{5}$$

The turbulence kinetic energy k is usually incorporated in the pressure p. The kinetic eddy viscosity ν_t is determined by solving the one-equation Spalart-Allmaras (SA) turbulence model [38] for quantity $\widetilde{\nu}$ as

$$\nu_t = \widetilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\widetilde{\nu}}{\nu}, \tag{6}$$

where $\widetilde{\nu}$ obeys the following transport equation:

$$u_j \frac{\partial \widetilde{\nu}}{\partial x_j} = c_{b1}\widetilde{S}\widetilde{\nu} + \frac{1}{\sigma}\frac{\partial}{\partial x_k}\left[(\nu + \widetilde{\nu})\frac{\partial \widetilde{\nu}}{\partial x_k}\right] + \frac{c_{b2}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_k}\frac{\partial \widetilde{\nu}}{\partial x_k} - c_{w1} f_w \left[\frac{\widetilde{\nu}}{d}\right]^2 \tag{7}$$

with

$$c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad c_{v1} = 7.1, \quad \sigma = 2/3,$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad \kappa = 0.41,$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}; \quad g = r + c_{w2}(r^6 - r),$$

$$r = \frac{\widetilde{\nu}}{\widetilde{S}\kappa^2 d^2}, \quad \widetilde{S} = S + \frac{\widetilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad S = \sqrt{2\Omega_{ij}\Omega_{ij}}.$$

The tensor $\Omega_{ij} = (u_{i,j} - u_{j,i})/2$ is the rotation tensor, and d is the distance from the closest wall. Thus, the solution of the direct problem gives us the field $\mathbf{q} = (\mathbf{u}, p, \widetilde{\nu})^T$. To achieve a given Reynolds number Re_b, it is mandatory to tune again the constant body force $f_1 = f_1^{\mathrm{SA}}$ in the RANS

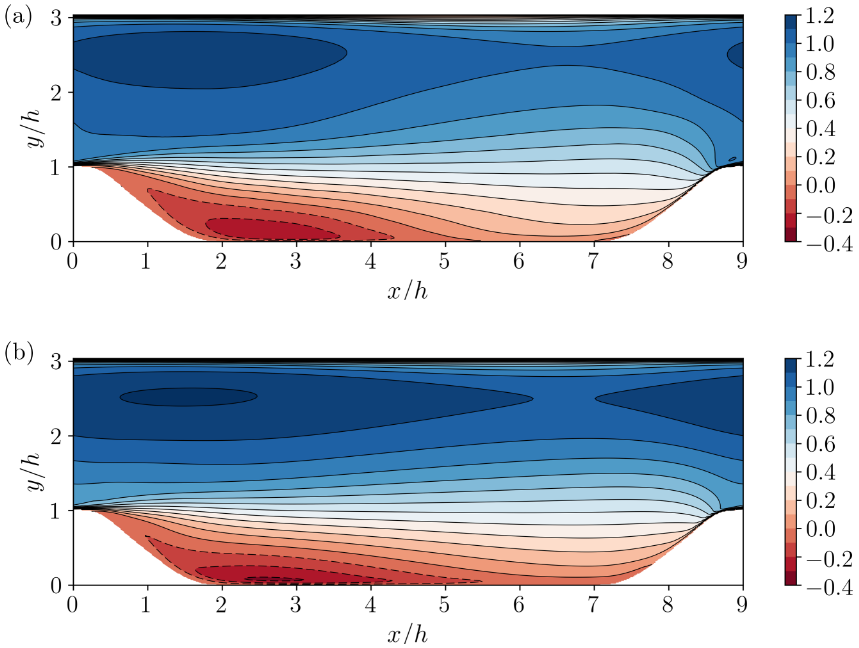

FIG. 2. Mean streamwise velocity field for the case with $\mathrm{Re}_b = 2800$. (a) DNS from Gloerfelt and Cinnella [27] and (b) RANS simulation using the original SA model.

model: since the (uncorrected) model does not recover the exact reference mean-flow fields, the predicted mean drag coefficient on the upper and lower walls is obviously (slightly) different from the reference values of the high-fidelity simulations, and the constant body force needs to be adjusted. The open-source finite-element software FREEFEM [39] is employed to solve the direct problem (see the Appendix for more details).

The streamwise velocity field from the original SA-RANS simulation is shown in Fig. 2 together with the reference solution. As expected, the discrepancies are striking, especially in terms of the extent of the mean recirculation bubble. As already mentioned, the fact that the separation process is highly time- and space-dependent, occurring over a large portion of the surface on the leeward hill side, makes it extremely difficult (if not impossible) to describe the flow within any existing RANS eddy-viscosity framework [37]. As a matter of fact, if we introduce an indicator defined through the inner product between the traceless stress tensor (also called the anisotropic stress tensor) $\mathbf{R} = \overline{u_i'u_j'} - \frac{2}{3}k\delta_{ij}$ and the mean strain rate tensor $\mathbf{S}$ as done in [11]

$$\rho_{\mathrm{RS}} = \frac{|\mathbf{R} : \mathbf{S}|}{\|\mathbf{R}\|\,\|\mathbf{S}\|}, \tag{8}$$

it is possible to test the validity of Boussinesq's hypothesis. This indicator measures the proportionality between these two tensors and is analogous to the cosine of the angle between vectors. Quantity ρ_{RS} varies between 0 and 1, and if $\rho_{\mathrm{RS}} = 1$, then Boussinesq's approximation is fully respected. Schmitt [11] established (through geometrical reasoning) that if $\rho_{\mathrm{RS}} > 0.86$, then it would make sense to use a linear eddy-viscosity approximation. In Fig. 3, we note that the indicator ρ_{RS} for case PH-2800 is lower than 0.8 in most of the domain, especially close to the walls, proving that RANS models based on such an approximation need to be corrected (at least for this type of flow). The objective of this work is to improve or even fix this issue by relying on data assimilation coupled with machine-learning techniques.

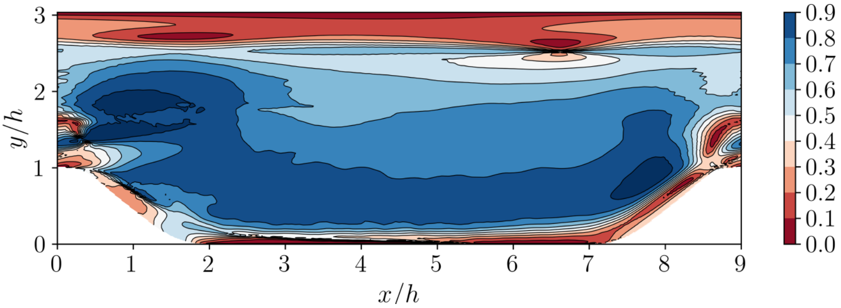

FIG. 3. A map of ρ_{RS} for the DNS of case PH-2800.

IV. MINIMAL MOMENTUM FORCE CORRECTION (OUTPUT) OF THE RANS MODEL BY VARIATIONAL DATA ASSIMILATION

In the present section, we consider an additional vectorial source term f_i^c in the momentum equations of the RANS model (4):

$$\frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial (2\nu S_{ij})}{\partial x_j} + f_i^{\nu_t} + f_i + f_i^c, \tag{9}$$

where $f_i^{\nu_t} = \frac{\partial(2\nu_t S_{ij})}{\partial x_j}$ are the Reynolds stresses due to the eddy-viscosity.

We are now interested in finding correction terms f_i^c that make the predicted mean-flow $\mathbf{u}(\mathbf{f}^c)$ match the reference mean-flow $\mathbf{u}_{\mathrm{ref}}$. Similarly to [24], the forcing term f_i^c only accounts for a small part of the full Reynolds stresses, since the eddy-viscosity force $f_i^{\nu_t}$ already models most of it. We are interested in particular in finding, among the many possible solutions, the specific f_i^c that exhibits the smallest amplitude. Doing so, the amplitude of f_i^c is minimal, which should facilitate the machine-learning step of the next section (by focusing the learning process on relevant and important data). It is actually the flexibility of the vectorial source correction term f_i^c that allows us to find a solution for which the predicted mean-flow field closely matches the reference solution [24], $\mathbf{u}(\mathbf{f}^c) \approx \mathbf{u}_{\mathrm{ref}}$.

This correction procedure can be seen as an optimization problem where the source term, modeling the Reynolds-stress tensor, $\mathbf{f}^c$, is tuned to minimize the objective function $\mathscr{J}$:

$$\mathscr{J} = \frac{1}{2}\|\mathbf{u}(\mathbf{f}^c) - \mathbf{u}_{\mathrm{ref}}\|_{L_2}^2 + \frac{l^2}{2}\|\mathbf{f}^{\mathbf{c}}\|_{L_2}^2, \tag{10}$$

where $\|\cdot\|_{L_2}$ is the functional L_2 norm. The first term represents the discrepancy between the predicted velocity field $\mathbf{u}(\mathbf{f}^c)$ and the reference state $\mathbf{u}_{\mathrm{ref}}$ (from DNS simulations), while the second term promotes small-amplitude correction terms. The parameter l^2 is chosen sufficiently small to allow accurate reconstruction of the reference flow and sufficiently large to ensure a small amplitude of f_i^c.

The governing equations may be written in a compact form as

$$\mathcal{R}(\mathbf{q}) = \mathcal{P}\mathbf{f}^c, \tag{11}$$

where $\mathbf{q} = (\mathbf{u}, p, \tilde{\nu})^T$ is the state vector, and $\mathcal{P}$ is the prolongation operator such that $\mathcal{P}\mathbf{f} = (\mathbf{f}, 0, 0)^T$. Due to the size of the control space, gradient-based techniques are necessary to minimize the objective functional. We will use the low-memory BFGS algorithm [40], which reconstructs the Hessian matrix from finite-difference approximations and captures the second-derivative behavior of the cost functional $\mathscr{J}$. Following Franceschini *et al.* [24], the gradient of the objective functional

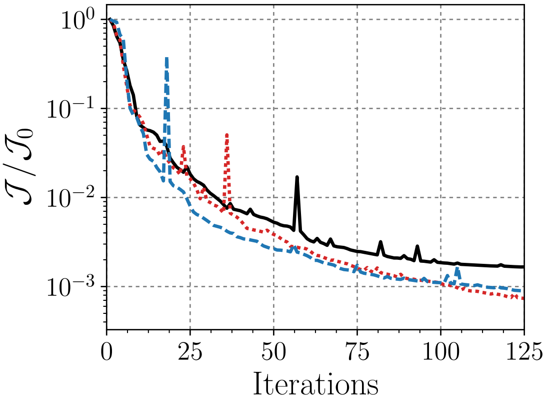

FIG. 4. Objective function convergence during data assimilation with L-BFGS. Case PH-2800 (solid line), PH-5600 (dotted line), and PH-10595 (dashed line).

reads

$$\frac{d\mathscr{J}}{d\mathbf{f}^c} = -\mathbf{u}^\dagger + l^2\mathbf{f}^c, \tag{12}$$

where the adjoint state $\mathbf{q}^\dagger = (\mathbf{u}^\dagger, p^\dagger, \tilde{\nu}^\dagger)^T$ is a solution of

$$\left(\frac{\partial\mathcal{R}}{\partial\mathbf{q}}\right)^\dagger \mathbf{q}^\dagger = -\mathcal{P}(\mathbf{u}-\mathbf{u}_{\text{ref}}). \tag{13}$$

The notation $(\frac{\partial\mathcal{R}}{\partial\mathbf{q}})^\dagger$ corresponds to the operator adjoint to $\frac{\partial\mathcal{R}}{\partial\mathbf{q}}$ with respect to the L_2 functional scalar-product. We remark that this method is based on the solution of adjoint problems to compute the gradient, which may not be available in any CFD code (even though automatic difference tools become more and more standard nowadays; see, for example, Economon *et al.* [41]). For this reason, we point out that regularized ensemble-based data-assimilation methods may also be used for solving inverse problems [42,43].

From a numerical point of view, the optimization process was conducted starting from a RANS solution. Note that the driving force f_i may be different when performing DNS or RANS computations. Therefore, in the DA procedure the assimilated force f_1^c also contains a constant spatial correction for the f_1 term. Thus, to separate the turbulent stress tensor contribution from the driving force, we can simply remove its streamwise integral $\int_\Omega f_i^c d\Omega$ from the converged f_i^c.

At the end, the full predicted Reynolds stress term of the corrected RANS model reads

$$f_i^{\text{RS,pred}} = f_i^{\nu_t} + f_i^c. \tag{14}$$

This quantity should be comparable to the reference Reynolds stresses of the DNS:

$$f_i^{\text{RS,DNS}} = \frac{\partial\tau_{ij}^{\text{DNS}}}{\partial x_j}, \tag{15}$$

where $\tau_{ij}^{\text{DNS}} = -\overline{u_i'u_j'}$.

The performance of the data-assimilation methodology is evaluated in the following. Figure 4 shows the convergence of the optimization problem for cases PH-2800, PH-5600, and PH-10595. The L-BFGS method was able to decrease the cost function by three orders of magnitude. The corresponding velocity field for configuration PH-2800 is plotted in Fig. 5 together with the DNS solution. An excellent match is observed between the DA solution and the averaged DNS field,

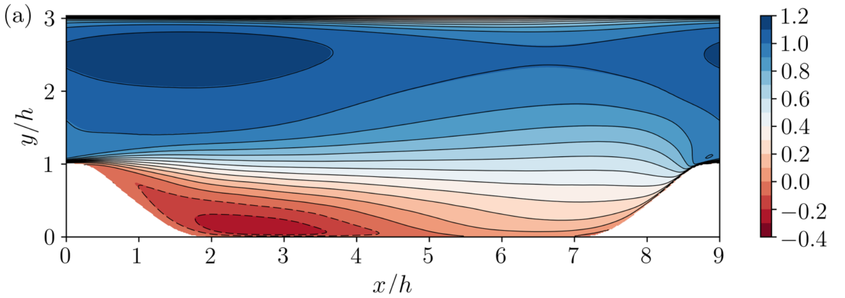
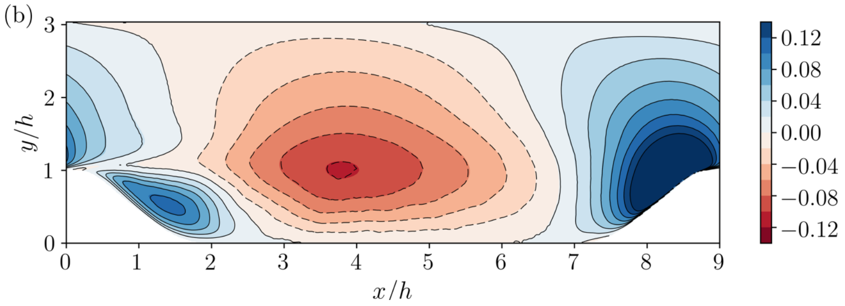

FIG. 5. Mean velocity field for the case with $\mathrm{Re}_b = 2800$. (a) Streamwise velocity field and (b) vertical velocity field. Contour plots: DA solution. Lines: DNS solution.

proving the efficiency of the present methodology. Note that this good match cannot be obtained by tuning the eddy-viscosity or a forcing term in the eddy-viscosity equation.

Next, we investigate the predicted full forcing term ($f_1^{\mathrm{RS,pred}}$) defined in (14) and compare it to the reference field from the DNS provided in (15). Figure 6 visualizes this comparison. From the DNS data [Fig. 6(a)], we can see that $f_1^{\mathrm{RS,DNS}}$ exhibits a strong negative force acting on the high velocity side of the shear layer, a positive one on the recirculation side, with large values around the separation point. The positive force declines across the separation bubble and increases again in the plateau region. A positive volume force is also observed close to the second hill wall indicating a flow acceleration in that region due to the channel geometry. The total predicted force $f_1^{\mathrm{RS,pred}}$ [Fig. 6(b)] shows a similar pattern, also in magnitude, with small differences in the

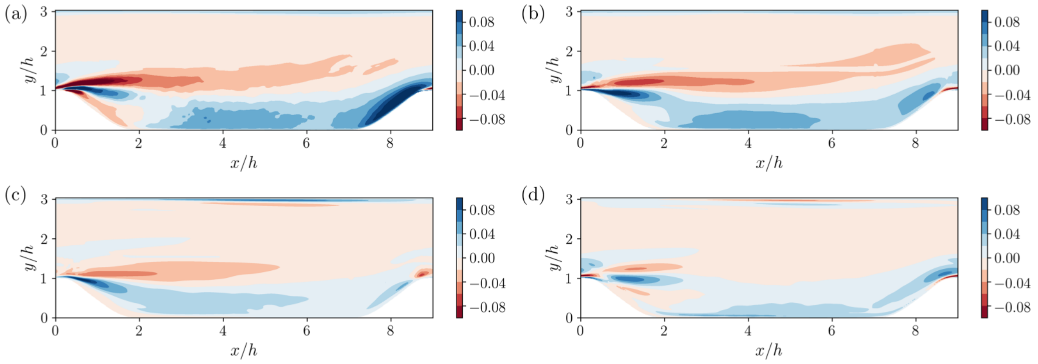

FIG. 6. Dominant component of force by turbulent fluctuations for the case with $\mathrm{Re}_b = 2800$. Red indicates -0.1 and blue $+0.1$. (a) Total force computed from the DNS: $f_1^{\mathrm{rs,DNS}}$, (b) Total force computed from RANS: $f_1^{\mathrm{rs,pred}}$, (c) Component given by the SA model: $f_1^{\nu_t}$, and (d) Correction term given by the DA: f_1^{c}.

TABLE II. Database of training flows to predict flow past periodic hills.

Training flow scenario	Selected cases	Fit on training data only	Fit on entire data without rotation
Scenario I	PH-2800, PH-10595, PH-19000	96.5%	95.6%
Scenario II	PH-2800, PH-5600, PH-10595	96.2%	93.5%

diverging/converging regions. The SA-contribution, $f_1^{\nu_t}$, shown in Fig. 6(c) is not the same one as in the original SA-RANS simulation, since the new term takes into account the modified flow field. The remaining error in $f_1^{\nu_t}$ is compensated by the correction term, i.e., f_1^c, shown in Fig. 6(d). It is striking to note that the predicted Reynolds force is very close to the DNS results, which shows that the data-assimilation process generates physical vectorial source correction terms f_i^c.

In the next step, a mapping function $f(x) \rightarrow f(\eta)$ is constructed, where $\eta(x)$ are local flow features that are readily available from any RANS state.

V. MACHINE LEARNING WITH INPUT PARAMETERS NORMALIZED ON GLOBAL QUANTITIES

In this section, the parameter fields obtained by the data-assimilation step are used to train and test feed-forward neural networks for the reconstruction of a flow field. Input parameters based on global features of the flow are presented in Sec. V A. The network architecture and treatment of rotational invariance are described in Secs. V B and V C. We train our neural network in two different scenarios (Table II). While in the first scenario training flows at Reynolds numbers $\mathrm{Re}_b = 2800,\ 10\,595$, and $19\,000$ were chosen, in the second scenario we have $\mathrm{Re}_b = 2800,\ 5600$, and $10\,595$. The same base geometry ($a = 1$) was used in both scenarios. In this way, the NN-based model obtained from Scenario I can be used in Case PH-5600 to test how well it interpolates (Sec. V D) to flows at intermediary Reynolds number, and the one from Scenario II can be used in Case PH-19000 to test how well it extrapolates (Sec. V E) to flows at higher Reynolds number. In Sec. V F, both scenarios were also employed to perform tests in cases presenting different geometries (PH-5600-0.8 and PH-5600-1.2).

A. Network inputs

We base the choice of input features η on the reported work of Ling and Templeton [15] and Wang *et al.* [17] as summarized in Table III. Here, u_i are the components of the velocity vector, $\mathbf{S}$ is the strain rate, and $\mathbf{\Omega}$ is the rotation rate tensor. We consider here a normalization based on global variables. We use, for example, the strain rate (q_1) and rotation rate (q_2) as separate inputs (and not the ratio between these two schemes), and we consider u_b/h as a reference value. Doing so, we increase the number of input parameters, which introduces more liberty for the network to adapt to the features. Of course overlearning should be avoided, which will be tested in the following. Along this same line of thought, we consider pressure normal stresses (q_3) and shear stresses (q_4) as different inputs. Feature q_5 defines the deviation from orthogonality between the velocity and its gradient and indicates the deviation from parallel shear flows [44]. Features q_6 is the pressure gradient along streamline. We also included in the list quantities related to the SA model ($q_7 - q_{11}$). Differently from Singh *et al.* [22], who considered as input the ratio of production and destruction terms (P/D) in the turbulence transport equations, in this study we opted to include other key quantities and use a different normalization. To reconstruct a turbulent kinetic energy term (q_{12}) in default of a corresponding transport equation, the 2013 version of the quadratic constitutive relation (QCR) [45] is used, with $k_{\mathrm{QCR}} = \frac{3}{2} c_{\mathrm{CR2}} \nu_t \sqrt{2 S_{ij} S_{ij}}$. The streamline curvature (q_{13}) makes use of the direction vector $\Gamma = \mathbf{u}/\|\mathbf{u}\|$ and the streamline path $Ds = \|\mathbf{u}\| Dt$. Thus, except for features q_1 and q_2, each component in the input vector is normalized in the classical way: $q_i = \hat{q}_i/(|\hat{q}_i| + \epsilon)$. The normalization used in this work may be less general than the ones previously used, but as shown

TABLE III. Input features to a neural network.

Feature	Description	Formula	Normalization factor
q_1	Strain rate	$\tanh(\frac{\lVert \mathbf{S} \rVert}{\epsilon_1})$	$\frac{u_b}{h}$
q_2	Rotation rate	$\tanh(\frac{\lVert \mathbf{\Omega} \rVert}{\epsilon_2})$	$\frac{u_b}{h}$
q_3	Normal stress	$\frac{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}}{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}+\epsilon_3}$	$\frac{0.1u_b^2}{h}$
q_4	Normal shear stress	$\frac{\frac{1}{2}\frac{\partial u_k^2}{\partial x_k}}{\lvert\frac{1}{2}\frac{\partial u_k^2}{\partial x_k}\rvert+\epsilon_4}$	$\frac{0.1u_b^2}{h}$
q_5	Gorlé *et al.* [44] marker	$\frac{\lVert\frac{u_i u_j}{u_k u_k}\frac{\partial u_i}{\partial x_j}\rVert}{\lVert\frac{u_i u_j}{u_k u_k}\frac{\partial u_i}{\partial x_j}\rVert+\epsilon_5}$	$\frac{0.1u_b}{h}$
q_6	Streamline pressure gradient	$\frac{u_k\frac{\partial p}{\partial x_k}}{\lvert u_k\frac{\partial p}{\partial x_k}\rvert+\epsilon_6}$	$\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}u_i u_i}$
q_7	Viscosity ratio	$\frac{\nu_t}{\nu_t+\epsilon_7}$	100ν
q_8	SA wall influence	$\frac{1}{1+r}$	
q_9	SA convection	$\frac{u_i\frac{\partial\widetilde{\nu}}{\partial x_i}}{\lvert u_i\frac{\partial\widetilde{\nu}}{\partial x_i}\rvert+\epsilon_9}$	$0.001u_b^2$
q_{10}	SA production	$\frac{c_{b1}\widetilde{S}\widetilde{\nu}}{\lvert c_{b1}\widetilde{S}\widetilde{\nu}\rvert+\epsilon_{10}}$	$0.001u_b^2$
q_{11}	SA destruction	$\frac{c_{w1}f_w(\frac{\widetilde{\nu}}{d})^2}{\lvert c_{w1}f_w(\frac{\widetilde{\nu}}{d})^2\rvert+\epsilon_{11}}$	$0.001u_b^2$
q_{12}	Turbulence intensity	$\frac{k_{qcr}}{k_{qcr}+\epsilon_{12}}$	$\frac{1}{2}u_i^2$
q_{13}	Streamline curvature	$\frac{\lVert D\Gamma/Ds\rVert}{\lVert D\Gamma/Ds\rVert+\epsilon_{13}}$	$\frac{1}{h}$
q_{14}	Cosine of rotation angle	$\cos(\phi)$	
q_{15}	Sine of rotation angle	$\sin(\phi)$	

later, the improvement in the model prediction justifies its use. It is worth mentioning that in practice our code solves the nondimensional RANS equations and therefore quantities and normalization factors are both dimensionless in our case.

B. Network architecture and training

The network structure is a common feed-forward neural network, using rectified linear unit (ReLU) activation functions and a mean-squared error as the network prediction error, i.e., the loss function. It uses a back-propagation algorithm to calculate the derivatives of the loss function with respect to the network weights and biases, which is efficiently done by automatic differentiation. The training procedure effectively represents a second minimization problem, separate from the data assimilation. In this optimization, the network loss is minimized by changing the parameters of the network using, again, an L-BFGS optimizer. For the data at hand, this algorithm is found to be more robust and efficient compared to batch- and stochastic-gradient-descent methods. The open-source library *PyTorch* is employed in order to reuse well-implemented methods for these procedures. Appropriate hyperparameters for the network training are found by a grid search implementation. The learn rate is consequently kept at 0.1 and the final network comprises four hidden layers with 80 neurons per layer, for training on multiple test cases. The input layer represents a vector that contains the input features from Table III. These features are constructed on every mesh vertex, forming the complete dataset for a given flow field. The dataset is then shuffled and divided in a training and a validation set, the first comprising 75% of the data and the second the other 25%. The validation set is used to monitor indications of overfitting and to abort the training in case of marginal improvement. We test on the entire dataset after training has finished and quantify the predictive capability by a fit-function relative to the standard deviation of the dataset:

$$\text{fit} = 1 - \frac{\sum (f_{\text{pred}} - f)^2}{\sum (\overline{f} - f)^2}.$$

C. Accounting for rotational invariance

An additional note should be given on the rotational invariance of the correction parameter produced by the neural network, which is required for the model to be independent of the reference frame. In previous work, such as that by Wu *et al.* [19], the goal was to obtain rotational invariance for a tensorial function that models the anisotropy stress tensor. For this, one must formulate the tensor in terms of an invariance basis. The theory of Pope [46] shows that such a basis can be found for the Reynolds tensor and derives the basis and the corresponding tensor invariants for the incompressible case. The coefficients that expand the basis to any specific tensor are now (unknown) scalar-valued functions of these basis invariants and therefore also invariant themselves. The first-order tensor of this basis is the strain rate, and its coefficient can be interpreted as the linear eddy viscosity. In our work, however, we construct a data model for a vector, not a tensor quantity. In fact, our volume force represents the nonlinear part of the divergence of the stress tensor, as per Eq. (14). The operation, therefore, reduces the rank of the output similar to the divergence operator, and a unique invariance basis cannot generally be found for functions with vector-valued outputs from tensorial inputs [47]. The divergence operator is a good example for this statement, since one can act with it on a general expansion of the tensor basis for the Reynolds stresses. The result will be a relation that depends on the spatial derivatives of the expansion coefficients, which is not per se a rotationally invariant formulation. Without a rigorous mathematical frame, it is a particular challenge to embed the rotational invariance into the neural network for this vectorial output.

Consequently, we decide to use data augmentation to experiment with rotational invariance [48]: rotated versions of the flow are constructed in 10° steps and additionally supplied as input data to the network, on which training and testing can be performed. The orientation of the flow is in this case included in the input feature list, defined as a unit vector of the main flow direction, i.e., identical to the normalised free-stream velocity, (q_{14}) and (q_{15}), where ϕ is the angle between the unit vector and the x-axis. Every rotation operation produces a new dataset with rotated versions of the volume force. Rotation of the remaining local flow quantities, e.g., the strain rate, can be used to verify the invariance of the input features (q_1)–(q_{13}); however, this computation is otherwise obsolete: the results must be identical for all frames. Translational invariance is guaranteed because we use a local formulation in which neither inputs nor outputs depend structurally on location. It should be noted that equal behavior in a uniformly moving reference frame is not yet investigated, although this is also required for a strictly Galilean-invariant model, and left for future work.

One may be surprised why the desired function requires the direction vector input as information about the reference frame, if we want to compose a function that does not depend on the frame. To clarify this: we desire the function *output* to be invariant. In other words, the aim is to obtain a 10°-rotated volume force for a 10°-rotated coordinate frame. If the anisotropic stress tensor was modeled [19], the reference frame was implicitly passed in the form of the local, frame-dependent basis tensor components, while a neural network produced frame-invariant coefficients. Using a volume force vector and data augmentation, a different way is needed to pass information about the reference frame in some other form. This can be understood intuitively if we imagine that we simply do not pass (q_{14}) and (q_{15}). In this case, the network would receive identical data for all rotations of the coordinate system. At the same time, we would require it to return rotated versions of the force. Such a network would not be able to learn, since there would be no injectivity in the training data: these data could not be represented by any function. The choice to pass the direction vector is straightforward and the only possible solution. In fact, it would be interesting to investigate different architectures for this purpose in future work.

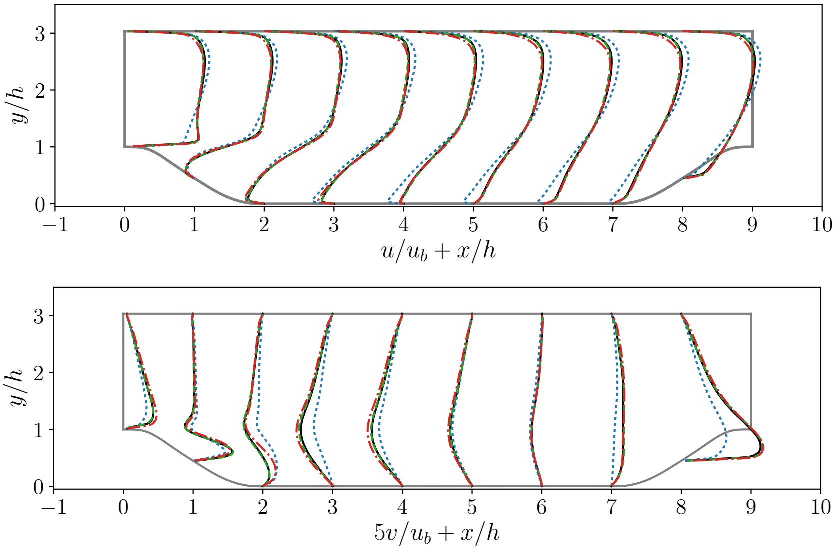

FIG. 7. Mean velocity profiles for case PH-5600. Streamwise velocity (top) and vertical velocity (bottom). DNS (solid lines), DA (dashed lines), SA-RANS (dotted lines), and NN-RANS based on Scenario I (dash-dotted lines).

It is worth noting that if the neural network is not trained on rotationally augmented data, one may observe a deteriorated flow field compared to the baseline RANS solution and/or even encounter converging issues to the specified problem. Poor predictions due to the lack of rotational invariance have also been observed in [49]. Therefore, we strongly encourage the rotation of the dataset before training to increase robustness. Since neural-network training is a statistical procedure, we repeat all training runs three times and retain the best result. Training usually terminates after 700–800 epochs with fit coefficients of approximately 96% on all rotated datasets (data augmentation; see Table II).

D. Interpolation in Reynolds number

We first analyze how the NN-based model from Scenario I (trained at $\mathrm{Re}_b = 2800,\ 10\,595$, and $19\,000$) performs in a flow at an intermediate Reynolds number (Case PH-5600). Figure 7 shows mean velocity profiles at stations $x/h \in \{0.05; 1; 2; 3; 4; 5; 6; 7; 8\}$ for the reference solution (DNS), the data assimilation, the baseline SA-RANS, and the NN-RANS. The results between DA and

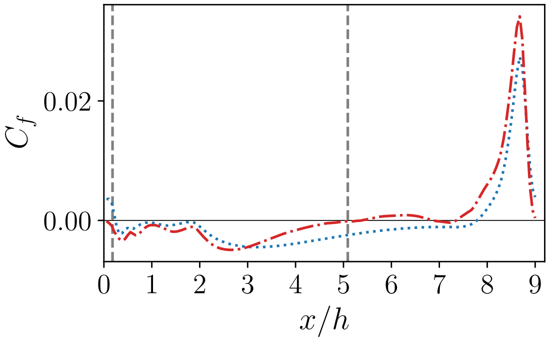

FIG. 8. Skin friction distribution for case PH-5600: SA-RANS (dotted line), NN-RANS (dash-dotted line), and separation and reattachment locations (experiment from Breuer *et al.* [36], dashed lines).

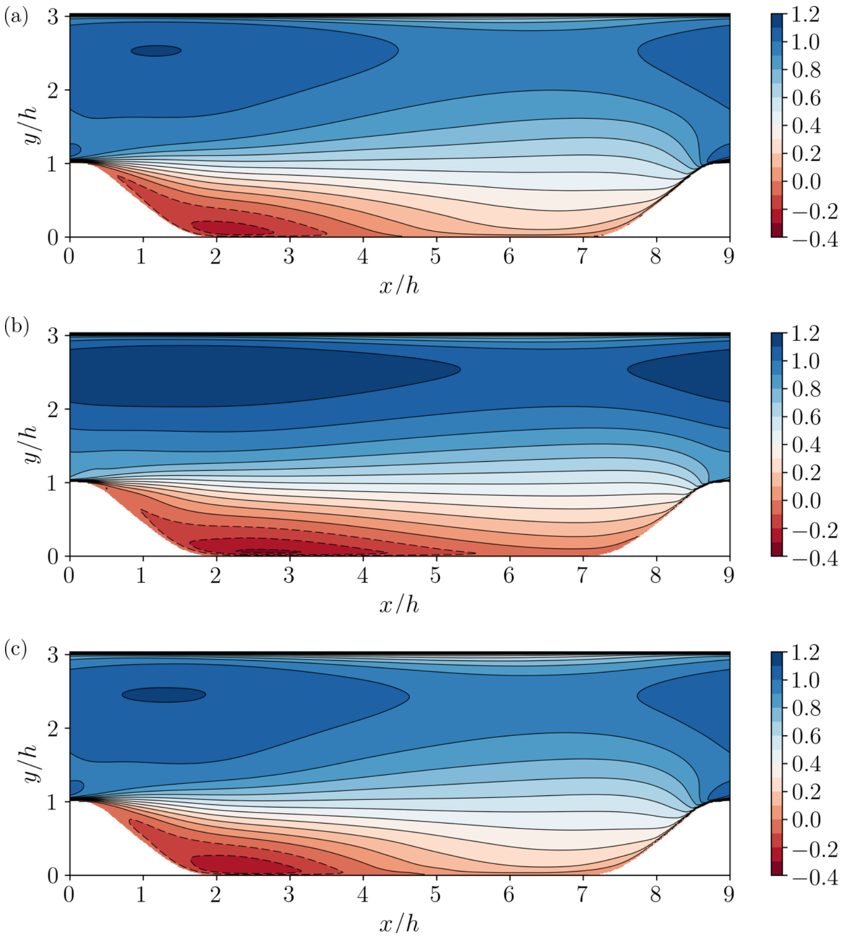

FIG. 9. Mean velocity fields for case PH-5600. (a) DNS, (b) SA-RANS, and (c) NN-RANS based on Scenario I.

DNS are practically indistinguishable, proving the robustness and accuracy of the field-inversion methodology. The velocity profiles predicted by the baseline RANS model are significantly different from the DNS solution, particularly in terms of the recirculation zone. This is also verified by analyzing Fig. 8, which depicts the distribution of the averaged skin friction coefficient along the lower wall. In fact, the separation bubble is drastically increased, ranging through the entire lower-wall plateau when the classic SA-model is employed (Fig. 9). On the other hand, the solution of the NN-RANS simulation, which is trained on the rotationally augmented dataset, agrees incredibly well with the reference profiles.

Figure 9 shows contour plots of the streamwise velocity field for the DNS, the baseline RANS, and the NN-RANS. Despite its qualitative character, it gives us a good description of the general flow field. The baseline RANS simulation predicts a much larger separation region compared to the reference database. It is well known that the turbulence level in the separated shear layer plays an important role in the size of the recirculation zone. For example, a higher level of shear stress implies an enhancement of the fluid entrainment into the shear layer and consequently a shorter recirculation bubble. Contrary to that, a lower turbulence level is consistent with a longer separation zone [37]. Therefore, if the Reynolds stresses (or their divergence, i.e., the forces in the momentum equations) are correctly predicted, the extent of the recirculation bubble can be correctly estimated

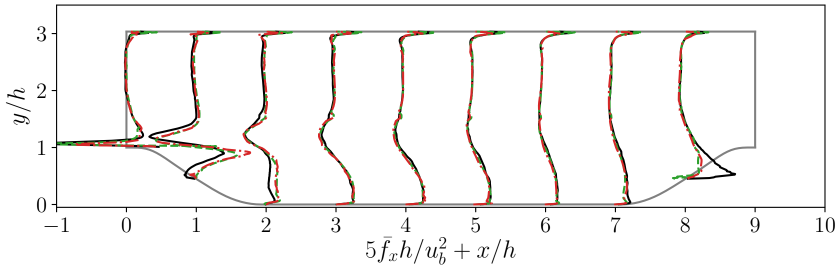

FIG. 10. Mean force in the streamwise direction for case PH-5600. DNS (solid lines), DA (dashed lines), and NN-RANS based on Scenario I (dash-dotted lines).

as in Fig. 9(c). These forces resulting from the data-assimilation step are very close to the ones obtained by the DNS as already discussed in the previous section. We note in Fig. 10 that the forces

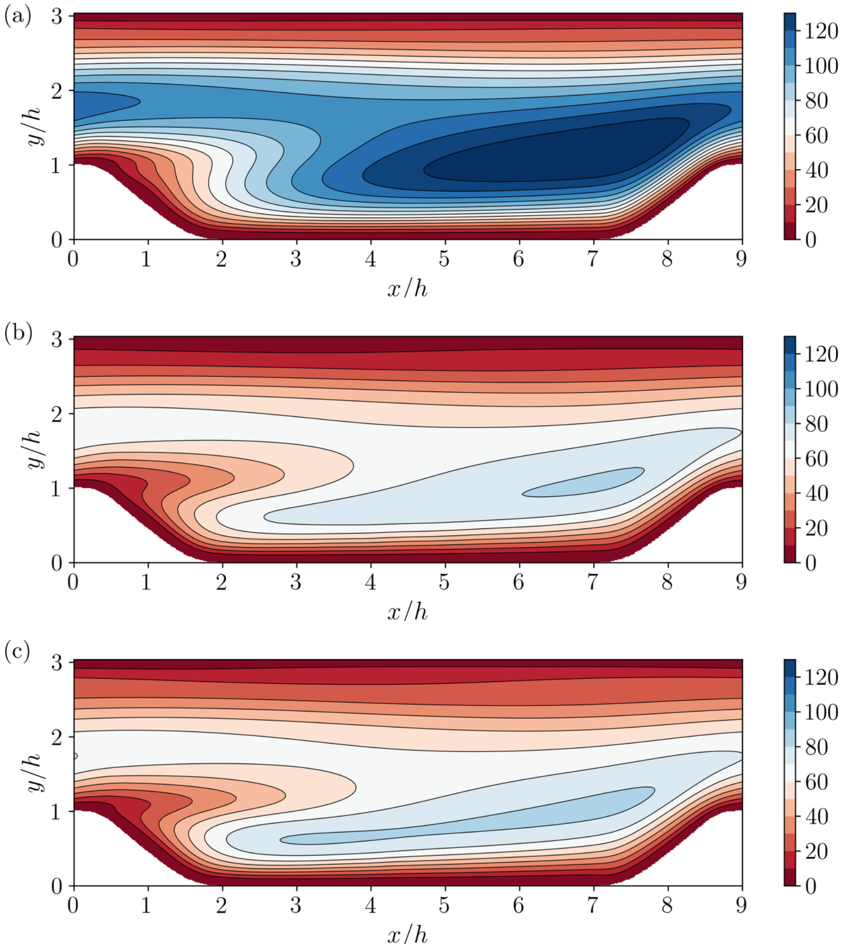

FIG. 11. Ratio between turbulent and laminar viscosities for case PH-5600. (a) SA-RANS, (b) DA, and (c) NN-RANS based on Scenario I.

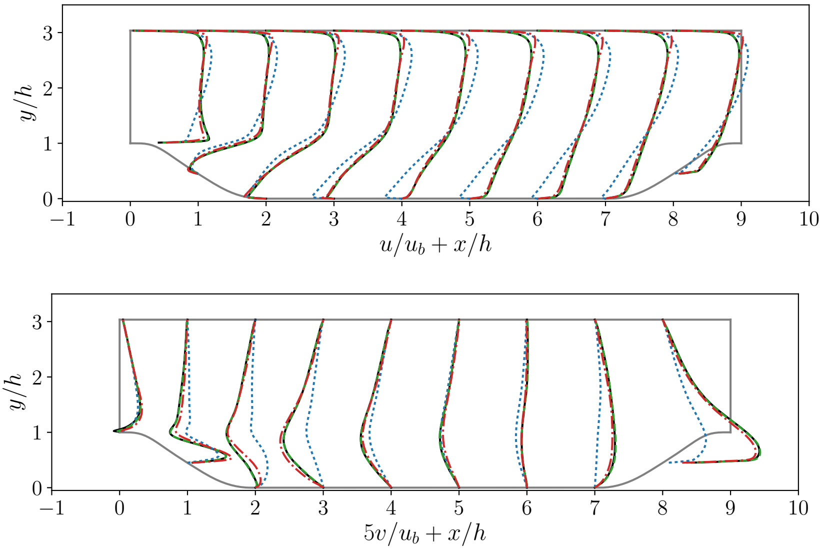

FIG. 12. Mean velocity profiles for case PH-19000. Streamwise velocity (top) and vertical velocity (bottom). DNS (solid lines), DA (dashed lines), SA-RANS (dotted lines), and NN-RANS based on Scenario II (dash-dotted lines).

predicted by the NN model are also very close to the DA/DNS forces (note the scaling factor in Fig. 10), which explain the improvement when using the NN-RANS model.

In Fig. 11, the ratio between turbulent and laminar viscosities is plotted for the baseline RANS simulation, the data assimilation, and the improved NN-RANS simulation. The classical SA-model predicts a much larger ratio of viscosities and a completely different distribution. The NN-RANS simulation compares well with the assimilated results.

E. Extrapolation in Reynolds number

Feed-forward neural networks may not perform well when extrapolating from the dataset. Since we train the network on a very particular configuration, only a specific set of input and output feature combinations occurs in the training set. We are interested now in testing the performance of the NN-model from Scenario II (trained at $\mathrm{Re}_b = 2800,\ 5600$, and 10 595) in a flow at a higher Reynolds number (Case PH-19000). Figure 12 shows profiles for the mean velocity components for the reference solution (LES), the data assimilation, the baseline SA-RANS, and the NN-RANS. Here again, when the NN model is employed, significant improvements are observed in the flow field as compared to the RANS with SA-model. In the latter case, results indicate that the reverse flow ends approximately around $x/h \approx 7.8$, overestimating significantly the size of the recirculating bubble. In the NN-RANS simulation, some discrepancies can be found in the upper channel region, where the extrapolation overestimated the volume forces. Nevertheless, in terms of the mean separation region the agreement is remarkable [Fig. 12(a)]. The machine-learning prediction also provides more accurate results in terms of mean vertical velocity profiles [Fig. 12(b)].

The profiles of the mean skin friction coefficient at the bottom wall for simulations using the standard SA- and the improved NN-RANS models are depicted in Fig. 13 together with the LES result from Gloerfelt and Cinnella [27]. The NN-based RANS simulation successfully captures not only the positions of flow separation and reattachment but also the main trend of the LES data with much better accuracy than the SA-model without correction. Nonetheless, the magnitude of the friction coefficient is underestimated between $8 < x/h < 9$.

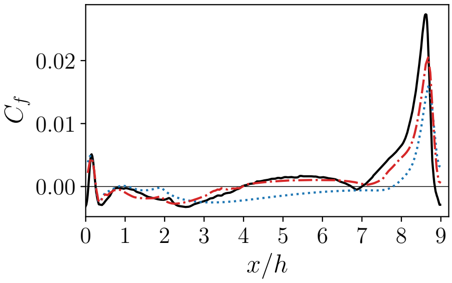

FIG. 13. Skin friction distribution for case PH-5600: SA-RANS (dotted line), NN-RANS (dash-dotted line), and LES from Gloerfelt and Cinnella [27] (solid lines).

F. Testing on different geometries

To demonstrate the capability of the proposed framework to predict flows with different geometries, we also investigated flows over periodic hills with different shape of hill profiles. This is an imperative step to underline the strength of the methodology, since geometry variations are a common scenario in engineering analysis using RANS simulations. More specifically, high-fidelity experimental and/or numerical data might be available only for a few flow conditions with specific Reynolds numbers and geometries, but predictions are needed for similar flows but at somewhat different conditions. Moreover, a more generally adequate methodology for turbulent flow calculations is certainly needed. Although this requires a transfer to entirely new configurations, the first step is a careful analysis of similar geometries.

In this section, both scenarios (I and II) are tested and their influence is discussed. The geometries of the new cases PH-5600-0.8 and PH-5600-1.2 are characterized, respectively, by a steeper and a smoother hill profile than the original case, as shown in Fig. 14. While the flows in Scenario II were trained using different geometries, in Scenario I they also present distinct Reynolds numbers.

Figures 15 and 16 show profiles at different streamwise stations for the mean velocity components for case PH-5600-0.8 and PH-5600-1.2, respectively. The comparison shows that both scenarios can be used to predict the mean velocity field of the periodic hill flow even on a modified geometry. Interestingly, Scenario I (trained using different geometries and Reynolds numbers) performs slightly better in case PH-5600-0.8, but slightly underperforms Scenario II in case PH-5600-1.2. Yet, regardless of the chosen scenario, the proposed framework gives excellent results when compared to the baseline RANS simulation.

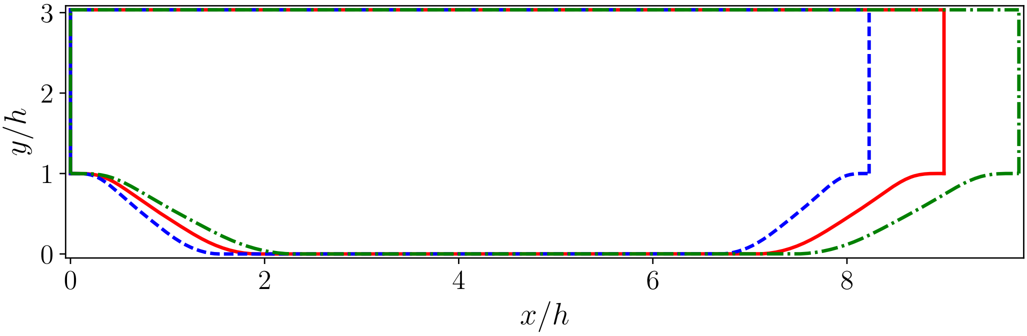

FIG. 14. New geometries by systematically varying the steepness of the hill. Solid line: original geometry ($a = 1$); dashed line: $a = 0.8$; and dash-dotted line: $a = 1.2$.

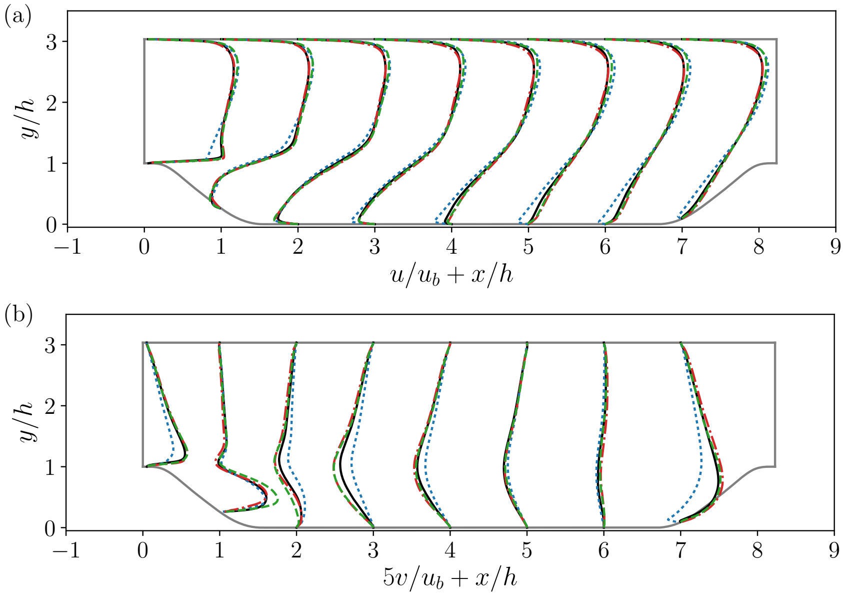

FIG. 15. Mean velocity profiles for case PH-5600-0.8. Streamwise velocity (top) and vertical velocity (bottom). DNS (solid lines), SA-RANS (dotted lines), NN-RANS based on Scenario I (dash-dotted lines), and NN-RANS based on Scenario II (dashed lines).

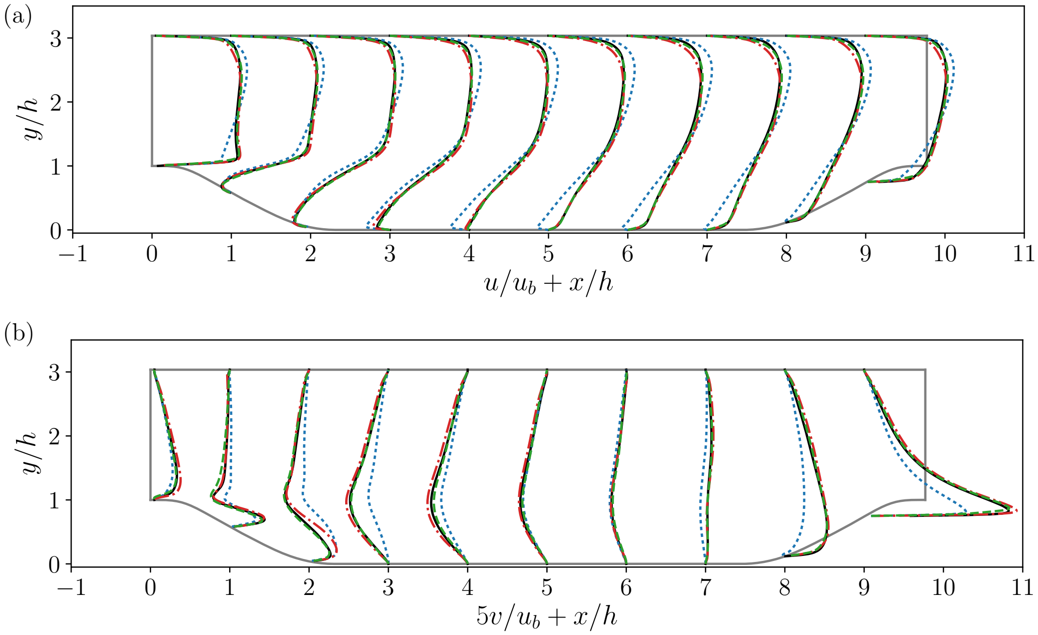

FIG. 16. Mean velocity profiles for case PH-5600-1.2. Streamwise velocity (top) and vertical velocity (bottom). DNS (solid lines), SA-RANS (dotted lines), NN-RANS based on Scenario I (dash-dotted lines), and NN-RANS based on Scenario II (dashed lines).

TABLE IV. New input vector for neural network (set 2).

Feature	Description	Formula	Normalization factor
q_{1-2}	Q-criterion	$\frac{\|\boldsymbol{\Omega}\|^2-\|\mathbf{S}\|^2}{\|\boldsymbol{\Omega}\|^2+\|\mathbf{S}\|^2}$	
q_{3-4}	Ratio of pressure normal stresses to shear stresses	$\frac{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}}{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}+\epsilon_{3-4}}$	$\frac{1}{2}\frac{\partial u_k^2}{\partial x_k}$
$q_{5'}$	Gorlé *et al.* [44] marker	$\frac{\|u_i u_j \frac{\partial u_i}{\partial x_j}\|}{\|u_i u_j \frac{\partial u_i}{\partial x_j}\|+\epsilon_{5'}}$	$\sqrt{u_l u_l u_i \frac{\partial u_i}{\partial x_j} u_k \frac{\partial u_k}{\partial x_j}}$
q_6	Streamline pressure gradient	$\frac{u_k \frac{\partial p}{\partial x_k}}{\lvert u_k \frac{\partial p}{\partial x_k}\rvert+\epsilon_6}$	$\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}u_i u_i}$
q_7	Viscosity ratio	$\frac{\nu_t}{\nu_t+\epsilon_7}$	100ν
q_8	SA wall influence	$\frac{1}{1+r}$	
$q_{10'}$	SA ratio of production to diffusion	$\frac{c_{b1}\widetilde{S}\widetilde{\nu}}{\lvert c_{b1}\widetilde{S}\widetilde{\nu}\rvert+\epsilon_{10'}}$	$\frac{c_{b2}}{\sigma}\frac{\partial\widetilde{\nu}}{\partial x_k}\frac{\partial\widetilde{\nu}}{\partial x_k}$
$q_{11'}$	SA ratio of destruction to diffusion	$\frac{c_{w1}f_w(\frac{\widetilde{\nu}}{d})^2}{\lvert c_{w1}f_w(\frac{\widetilde{\nu}}{d})^2\rvert+\epsilon_{11'}}$	$\frac{c_{b2}}{\sigma}\frac{\partial\widetilde{\nu}}{\partial x_k}\frac{\partial\widetilde{\nu}}{\partial x_k}$
q_{12}	Turbulence intensity	$\frac{k_{qcr}}{k_{qcr}+\epsilon_{12}}$	$\frac{1}{2}u_i^2$
$q_{13'}$	Streamline curvature	$\frac{\|D\Gamma/Ds\|}{\|D\Gamma/Ds\|+\epsilon_{13'}}$	$\frac{\sqrt{k_{qcr}}}{\nu_t}$
q_{14}	Cosine of rotation angle	$\cos(\phi)$	
q_{15}	Sine of rotation angle	$\sin(\phi)$	

VI. MACHINE LEARNING WITH INPUT PARAMETERS NORMALIZED ON LOCAL QUANTITIES

In this section, we evaluate the above results with the more classical local normalization scheme. The new set of input features is designated as set 2 in the following. Features from the new set are summarized in Table IV. We combined some features with the same dimensions from set 1 to make the model portable to other configurations. For example, we combine quantities q_1 and q_2 in Table III as $q_{1-2} = \frac{\|\boldsymbol{\Omega}\|^2-\|\mathbf{S}\|^2}{\|\boldsymbol{\Omega}\|^2+\|\mathbf{S}\|^2}$ and quantities q_3 and q_4 as $q_{3-4} = \frac{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}}{\sqrt{\frac{\partial p}{\partial x_k}\frac{\partial p}{\partial x_k}}+\lvert\frac{1}{2}\frac{\partial u_k^2}{\partial x_k}\rvert}$. Similarly, for quantity $q_{5'}$ in set 2, we use the following normalization: $q_{5'} = \frac{\|u_i u_j \frac{\partial u_i}{\partial x_j}\|}{\|u_i u_j \frac{\partial u_i}{\partial x_j}\|+\sqrt{u_l u_l u_i \frac{\partial u_i}{\partial x_j} u_k \frac{\partial u_k}{\partial x_j}}}$. Similar adjustments can be made for the turbulence variables or the streamline curvature. In Fig. 17 we plot the streamwise velocity profiles using the NN models based on these two sets, and we evaluate how well they extrapolate to flows at higher Reynolds number. This configuration was found to be the most critical case. We note in Fig. 17(a) that the NN generated using set 2 improves the solution in the middle of the channel and also reduces the separation bubble, but globally it underperforms the previously established model. The skin friction coefficient obtained with the simulation using set 2 is also shown in Fig. 17(b). It is important to note that both models improve the RANS solutions substantially, but results obtained with the first set are the closest to the reference data. Note, however, that the second set of input features has the advantage of being more general than the first one. Even if global quantities can easily be quantified for the present family of configurations, when dealing with multiple and more complex scenarios, defining such quantities can be tricky. Moreover, in the first set, note that several input features were normalized using the bulk velocity, thus violating the Galilean invariance of the NN turbulence model. This lack of translational invariance is not detrimental to wall-bounded flow configurations, as the periodic hills, since the wall velocity can be considered as the zero-velocity reference. However, the NN turbulence model obtained using

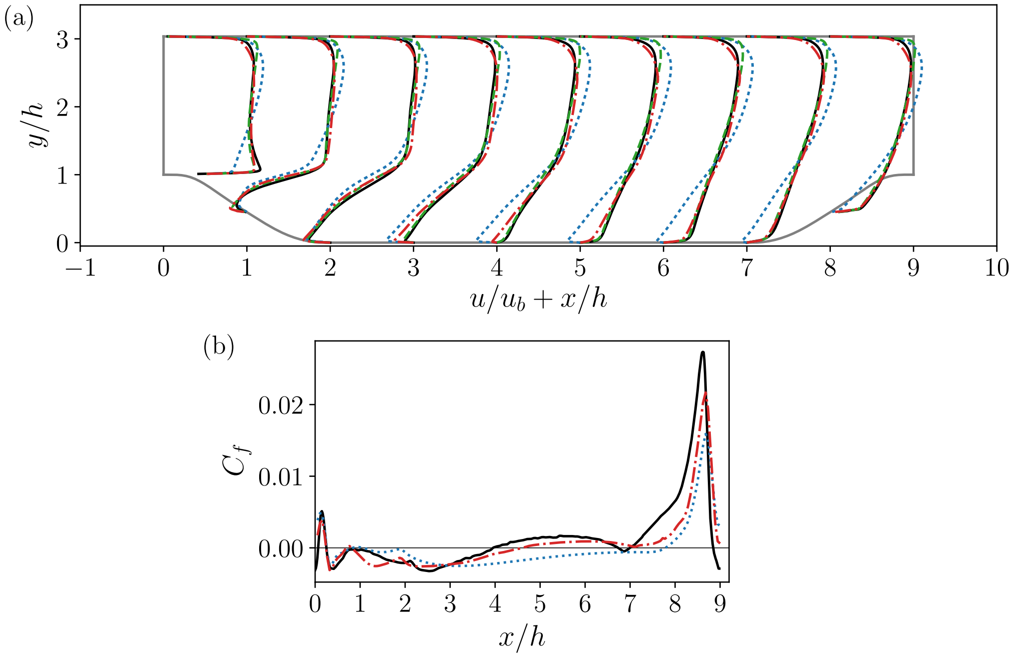

FIG. 17. Mean profiles for case PH-19000. DNS (solid lines), SA-RANS (dotted lines), original NN model using set 1 (dashed lines), and NN-RANS (dash-dotted lines). (a) Streamwise velocity and (b) skin friction coefficient.

the first set might not be applicable to flow configurations involving multiple moving bodies (e.g., turbines) or free-shear mixing layers for which a zero-velocity reference is not easily identified. The choice of input features to improve NN-based RANS models is still a subject that needs further investigation. The lesson here is that machine learning-augmented turbulence models can be used to improve CFD capabilities.

VII. CONCLUSION

Data-driven turbulence modeling has been a central topic of research in recent years. In fact, machine learning has been used to correct the source terms in turbulence transport equations [21,22], or to directly predict the Reynolds stresses or their discrepancies compared to the truth [17–19]. In this work, we demonstrated an off-line method for parameter inference and machine learning for turbulence modeling. The off-line regression strategy to construct the correction function consists of three consecutive steps: (i) variational data assimilation is used to infer a modeling correction from high-fidelity data, (ii) machine learning is used by means of neural-network training to reconstruct the volume force correction to the Spalart-Allmaras turbulence model as a function of available flow quantities, and (iii) RANS computations are performed using the augmented neural-network turbulence model. The novelty of the present work consists of introducing a correction to the Boussinesq hypothesis by adding a volume forcing term in the momentum equations. The correction term is obtained by field inversion based on multiple high-fidelity data and then generalized using neural networks for the same class of flows. The present formulation is more general than the one described in [21,22], since it is not restricted to traditional linear or quadratic eddy viscosity models. The methodology was validated on the periodic-hill configuration at different geometries and Reynolds numbers. This case presents strong flow separation and well-defined flow conditions, making it especially suited as a benchmark case for testing machine-learning-assisted turbulence modeling approaches. Two flow scenarios were used to train the neural-network RANS model. Each scenario was used to test how well the model is able to interpolate/extrapolate to flows at different Reynolds numbers. The neural-network-based model correctly predicts the mean velocity fields and the size of the separation region, in contrast with the baseline RANS model. The proposed framework is also capable of predicting flows over periodic hills with different shapes of hill profiles by employing data from training flows with similar characteristics to those of the target flow. This is extremely important from an engineering point of view, since high-fidelity experimental and/or numerical data might be available only for a few flow conditions with specific Reynolds numbers and geometries, but predictions are needed for similar flows but at somewhat different conditions. Future efforts will enlarge the space of represented configurations and therefore transfer the correction term to largely different flow scenarios. Moreover, a careful analysis investigating the influence of each input feature in the prediction of the output quantities needs to be carried out for both local and global sets. This is important since neural networks with a lower-dimensional feature space would perform better from a computational point of view. The final objective being a widely applicable methodology for turbulent flows, we can conclude that machine learning combined with RANS equations is a powerful tool that improves available mature turbulence models.

ACKNOWLEDGMENTS

This work was supported by the European project HiFi-TURB (High Fidelity LES/DNS Data for Innovative Turbulence Models)—UE H2020 under Grant Agreement No. 814837 and the internal project DeLTA (Deep Learning: Théorie et Applications). It was also partially supported by the European project HOMER (Holistic Optical Metrology for Aero-Elastic Research)—UE H2020 under Grant Agreement No. 769237.

APPENDIX: SUPG IMPLEMENTATION

The numerical implementation of the RANS-SA equations is based on the finite-element method (FEM), available in the FreeFem++ code (see Ref. [39]). Since FEM is naturally numerically unstable at high Reynolds numbers, some stabilization schemes need to be employed. Here, we choose the streamline-upwind Petrov-Galerkin (SUPG) formulation, as proposed by Brooks and Hughes [50]. In this formulation, the test function is advected with the local velocity field, giving an upwind effect, stabilizing the scheme. Several different formulations have been proposed in the literature (see Refs. [51,52]) for various different equations. Here, we employ a simplified version of it, common for unsteady problems [53], where only the advection terms are treated. In a simplified notation, we write the nonlinear residual of the RANS-SA equations in the weak form

$$\begin{aligned} R([\mathbf{u}, p, \tilde{\nu}], [\mathbf{v}, q, \check{\nu}]) = & \int_{\Omega} \big(u_k \partial_{x_k} u_i + f_i\big) v_i + \int_{\Omega} \big(-pI + (\nu + \nu_t)\big(\partial_{x_i} u_j + \partial_{x_j} u_i\big)\big)\big(\partial_{x_j} v_i\big) \\ & - \int_{\Omega} \big(\partial_{x_i} u_i\big) q + \int_{\Omega} \big(u_k \partial_{x_k} \tilde{\nu} - s\big)\check{\nu} + \int_{\Omega} \eta \partial_{x_k} \tilde{\nu} \partial_{x_k} \check{\nu} \\ & + \sum_{\Omega_k} \int_{\Omega_k} \tau_{\mathrm{SUPG}} \big(u_i \partial_{x_i} v_k\big)\big(u_j \partial_{x_j} u_k\big) + \sum_{\Omega_k} \int_{\Omega_k} \tau_{\mathrm{SUPG}} \big(u_i \partial_{x_i} \check{\nu}\big)\big(u_j \partial_{x_k} \tilde{\nu}\big), \end{aligned} \tag{A1}$$

where the last two terms correspond to the SUPG formalism, and the remaining terms are due to the classical (unstable) finite-element formulation. The functions s and η correspond to the source term in the Spalart-Allmaras model and its diffusivity, given by

$$\eta = \sigma^{-1}(\nu + \widetilde{\nu}), \quad s = c_{b1} \widetilde{S} \widetilde{\nu} + \frac{c_{b2}}{\sigma} \frac{\partial \widetilde{\nu}}{\partial x_k} \frac{\partial \widetilde{\nu}}{\partial x_k} - c_{w1} f_w \left[\frac{\widetilde{\nu}}{d}\right]^2. \tag{A2}$$

The function τ_{SUPG} regulates the amount of numerical diffusivity and depends on the local Reynolds (Re_h) number as

$$\tau_{\mathrm{SUPG}} = \frac{\xi(\mathrm{Re}_h)h_T}{2|\mathbf{u}|}, \quad \xi(\mathrm{Re}_h) = \begin{cases} \mathrm{Re}_h/3, & \mathrm{Re}_h \leqslant 3, \\ 1, & \mathrm{Re}_h > 3, \end{cases} \quad \mathrm{Re}_h = \frac{|\mathbf{u}|h_T}{2\nu}, \tag{A3}$$

where the function $\xi(\mathrm{Re}_h)$ is constant for high Reynolds number, saturating in this way the amount of numerical dissipation introduced. The parameter h_T indicates the local element size and is taken here as $h_T = \sqrt{2}A/h_T^{\max}$ (where A is the area of the element), minimizing the numerical dissipation for highly elongated mesh elements (see Ref. [54]). All those integrals are computed with a quadrature that exactly integrates a polynomial of three times the order of the degree considered.

[1] F. Sarghini, G. De Felice, and S. Santini, Neural networks based subgrid scale modeling in large eddy simulations, Comput. Fluids **32**, 97 (2003).

[2] A. Vollant, G. Balarac, and C. Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, J. Turbulence **18**, 854 (2017).

[3] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, Combust. Flame **203**, 255 (2019).

[4] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system, Phys. Fluids **27**, 092101 (2015).

[5] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels, Int. J. Multiphase Flow **85**, 336 (2016).

[6] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, Synthetic turbulent inflow generator using machine learning, Phys. Rev. Fluids **4**, 064603 (2019).

[7] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control, J. Fluid Mech. **865**, 281 (2019).

[8] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz, Deep model predictive flow control with limited sensor data and online learning, Theor. Comput. Fluid Dyn. **34**, 577 (2020).

[9] A. N. Marques, Q. Wang, and Y. Marzouk, Data-driven integral boundary-layer modeling for airfoil performance prediction in laminar regime, AIAA J. **56**, 482 (2018).

[10] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. **51**, 357 (2019).

[11] F. G. Schmitt, About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity, C. R. Méc. **335**, 617 (2007).

[12] S. Wallin and A. V. Johansson, An explicit algebraic reynolds stress model for incompressible and compressible turbulent flows, J. Fluid Mech. **403**, 89 (2000).

[13] B. Eisfeld, C. Rumsey, and V. Togiti, Verification and validation of a second-moment-closure model, AIAA J. **54**, 1524 (2016).

[14] H. Xiao and P. Cinnella, Quantification of model uncertainty in RANS simulations: A review, Progr. Aerospace Sci. **108**, 1 (2019).

[15] J. Ling and J. Templeton, Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged Navier Stokes uncertainty, Phys. Fluids **27**, 085103 (2015).

[16] J.-L. Wu, J.-X. Wang, H. Xiao, and J. Ling, A priori assessment of prediction confidence for data-driven turbulence modeling, Flow, Turbulence Combustion **99**, 25 (2017).

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

[18] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. **807**, 155 (2016).

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

[20] D. P. G. Foures, N. Dovetta, D. Sipp, and P. J. Schmid, A data-assimilation method for reynolds-averaged navier-stokes-driven mean flow reconstruction, J. Fluid Mech. **759**, 404 (2014).

[21] E. J. Parish and K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, J. Comput. Phys. **305**, 758 (2016).

[22] A. P. Singh, S. Medida, and K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA J. **55**, 2215 (2017).

[23] J. R. Holland, J. D. Baeder, and K. Duraisamy, Towards integrated field inversion and machine learning with embedded neural networks for RANS modeling, in *AIAA Scitech 2019 Forum* (AIAA, 2019), p. 1884.

[24] L. Franceschini, D. Sipp, and O. Marquet, Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high reynolds number backward-facing step, Phys. Rev. Fluids **5**, 094603 (2020).

[25] J.-F. Monier, N. Poujol, M. Laurent, F. Gao, J. Boudet, S. Aubert, and L. Shao, Les investigation of boussinesq constitutive relation validity in a corner separation flow, in *Turbo Expo: Power for Land, Sea, and Air* (American Society of Mechanical Engineers, 2018), Vol. 51012, p. V02CT42A023.

[26] S. Symon, N. Dovetta, B. J. McKeon, D. Sipp, and P. J. Schmid, Data assimilation of mean velocity from 2d piv measurements of flow over an idealized airfoil, Exp. Fluids **58**, 61 (2017).

[27] X. Gloerfelt and P. Cinnella, Large eddy simulation requirements for the flow over periodic hills, Flow, Turbulence Combustion **103**, 55 (2019).

[28] 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, Comput. Fluids **200**, 104431 (2020).

[29] L. Davidson and S.-H. Peng, Hybrid LES-RANS modelling: a one-equation SGS model combined with a k–ω model for predicting recirculating flows, Int. J. Numer. Methods Fluids **43**, 1003 (2003).

[30] L. Temmerman, M. A. Leschziner, C. P. Mellen, and J. Fröhlich, Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions, Int. J. Heat Fluid Flow **24**, 157 (2003).

[31] L. Temmerman, M. Hadžiabdić, M. A. Leschziner, and K. Hanjalić, A hybrid two-layer URANS–LES approach for large eddy simulation at high Reynolds numbers, Int. J. Heat Fluid Flow **26**, 173 (2005).

[32] F. Tessicini, L. Temmerman, and M. A. Leschziner, Approximate near-wall treatments based on zonal and hybrid RANS–LES methods for LES at high Reynolds numbers, Int. J. Heat Fluid Flow **27**, 789 (2006).

[33] C. Duprat, G. Balarac, O. Métais, P. M. Congedo, and O. Brugière, A wall-layer model for large-eddy simulations of turbulent flows with/out pressure gradient, Phys. Fluids **23**, 015101 (2011).

[34] M. de la Llave Plata, V. Couaillier, and M.-C. Le Pape, On the use of a high-order discontinuous galerkin method for DNS and LES of wall-bounded turbulence, Comput. Fluids **176**, 320 (2018).

[35] J. Fröhlich, C. P. Mellen, W. Rodi, L. Temmerman, and M. A. Leschziner, Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions, J. Fluid Mech. **526**, 19 (2005).

[36] M. Breuer, N. Peller, Ch. Rapp, and M. Manhart, Flow over periodic hills–numerical and experimental study in a wide range of reynolds numbers, Comput. Fluids **38**, 433 (2009).

[37] S. Jakirlic, Extended excerpt related to the test case: Flow over a periodical arrangement of 2D hills, Tech. Rep. (2012).

[38] P. R. Spalart and S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, La Rech. Aérospatiale **1**, 5 (1994).

[39] F. Hecht, New development in freefem++, J. Numer. Math. **20**, 251 (2012).

[40] J. Nocedal and S. J. Wright, in *Numerical Optimization* (Springer-Verlag, New York, 1999), pp. 490–527.

[41] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, Su2: An open-source suite for multiphysics simulation and design, AIAA J. **54**, 828 (2016).

[42] M. A. Iglesias, K. J. H. Law, and A. M. Stuart, Ensemble kalman methods for inverse problems, Inverse Probl. **29**, 045001 (2013).

[43] X.-L. Zhang, C. Michelén-Ströfer, and H. Xiao, Regularized ensemble kalman methods for inverse problems, J. Comput. Phys. **416**, 109517 (2020).

[44] C. Gorlé, J. Larsson, M. Emory, and G. Iaccarino, The deviation from parallel shear flow as an indicator of linear eddy-viscosity model inaccuracy, Phys. Fluids **26**, 055105 (2014).

[45] M. Mani, D. Babcock, C. Winkler, and P. Spalart, Predictions of a supersonic turbulent flow in a square duct, in *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (AIAA, 2013), p. 860.

[46] S. B. Pope, A more general effective-viscosity hypothesis, J. Fluid Mech. **72**, 331 (1975).

[47] M. Frewer, Covariance and objectivity in mechanics and turbulence, arXiv:1611.07002.

[48] J. Ling, R. Jones, and J. Templeton, Machine learning strategies for systems with invariance properties, J. Comput. Phys. **318**, 22 (2016).

[49] J.-L. Wu, R. Sun, S. Laizet, and H. Xiao, Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling, Comput. Methods Appl. Mech. Eng. **346**, 707 (2019).

[50] A. N. Brooks and T. J. R. Hughes, Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations, Comput. Method. Appl. M. **32**, 199 (1982).

[51] L. P. Franca, S. L. Frey, and T. J. R. Hughes, Stabilized finite element methods: I. application to the advective-diffusive model, Comput. Method. Appl. M. **95**, 253 (1992).

[52] L. P. Franca and S. L. Frey, Stabilized finite element methods: Ii. the incompressible navier-stokes equations, Comput. Method. Appl. M. **99**, 209 (1992).

[53] Y. Bao, D. Zhou, C. Huang, Q. Wu, and X.-q. Chen, Numerical prediction of aerodynamic characteristics of prismatic cylinder by finite element method with spalart–allmaras turbulence model, Comput. Struc. **89**, 325 (2011).

[54] S. Mittal, On the performance of high aspect ratio elements for incompressible flows, Comput. Method. Appl. M. **188**, 269 (2000).