Assessment of data-driven Reynolds stress tensor perturbations for uncertainty quantification of RANS turbulence models

M. Matha *, K. Kucharczyk † and C. Morsbach ‡
German Aerospace Center (DLR), Cologne, Germany

In order to achieve a more simulation-based design and certification process of jet engines in the aviation industry, the uncertainty bounds for computational fluid dynamics have to be known. This work shows the application of machine learning to support the quantification of epistemic uncertainties of turbulence models. The underlying method in order to estimate the uncertainty bounds is based on eigenspace perturbations of the Reynolds stress tensor in combination with random forests.

I. Introduction

As a compromise between computational time and accuracy, Reynolds-averaged Navier-Stokes (RANS) simulation is still the workhorse in industrial design of turbomachinery, since the replacement of RANS by scale-resolving simulations, e.g. direct numerical simulations (DNS) or large eddy simulations (LES) cannot be expected for design optimization simulations in the next years. The derivation of the RANS equations reveals an unclosed term, called the Reynolds stress tensor. This tensor has to be approximated in Computational-Fluid-Dynamics (CFD) simulations by applying turbulence models. The prediction accuracy of the simulation is highly dependent on these kinds of models, which compute the effect of turbulence flow physics on mean flow quantities. Although RANS-based models, such as linear eddy viscosity models (LEVM), are widely used for complex engineering flows, they suffer from the inability to replicate fundamental turbulent processes. Turbulence models are one of the main limitations in striving for reliable, environmentally friendly designs, due to general simplifying assumptions during formulation of closure models. These simplifications are the result of data observation and physical intuition, leading to a significant degree of epistemic uncertainty.

In recent years, the interest in uncertainty quantification (UQ), leading to more reliable simulation results, has grown. Different approaches try to account for the uncertainty of the turbulence model at different modelling levels [1]. Generally, one distinguishes between parametric and non-parametric approaches. While the parametric uncertainties arise from the chosen closure coefficients and their calibration process, non-parametric methodologies directly investigate the uncertainties on modeled terms in the transport equations of the turbulence model. It is expected, that the possible solution space, with respect to the uncertainty of the turbulence model, is larger for the non-parametric approaches [2]. The group of Professor Iaccarino proposed an eigenspace perturbation framework, which is based on the inability of common LEVM to deal with Reynolds stress tensor anisotropy [3, 4]. This methodology belongs to the non-parametric approaches, since it tries to account for the uncertainty due to the closure model form itself. The emergence of machine learning strategies guided the path towards data driven approaches also for the turbulence modelling community [1]. Heyse et al. enhanced the uncertainty estimation based on the eigenspace perturbation approach by adding a data-driven method [5]. In our previous work, we already demonstrated the possibility to use this data-driven perturbation approach for the eigenvalues of the Reynolds stress tensor and validated our implementation [6]. DNS data of a turbulent channel flow was used to train a machine learning model and its enhancement in relation to a data-free method for the quantification of uncertainties was evaluated. As soon as it gets to industrial design applications, more complex flow phenomena, such as flow separation due to adverse pressure gradient and reattachment, emerge. Additionally, the prediction accuracy of RANS turbulence models is significantly reduced in the presence of these flow phenomena.

In order to obtain a more robust and trustworthy quantification of uncertainties for future design application, we investigate this recent data-driven extension for flow configurations featuring flow separation with DLR's CFD solver suite TRACE . TRACE is being developed by the Institute of Propulsion Technology with focus on turbomachinery flows and offers a parallelized, multi-block CFD solver for the compressible RANS equations [7]. We present the functionality of the eigenspace perturbation methodology and present its implementation.

*Research scientist, German Aerospace Center, Institute of Propulsion Technology, Linder Höhe, 51147 Cologne, Germany.
†Research scientist, German Aerospace Center, Institute of Propulsion Technology, Linder Höhe, 51147 Cologne, Germany.
‡Team leader, German Aerospace Center, Institute of Propulsion Technology, Linder Höhe, 51147 Cologne, Germany.

To further advance the data-driven methodology towards relevant turbomachinery flows, additional data sets from scale-resolving simulations of relevant test cases will serve as training a machine learning model. These test cases include flow physics, such as adverse pressure gradient, separation and reattachment. We will explain the machine learning approach and the verification of the trained model in detail. Finally, the data-driven perturbation approach to estimate the epistemic uncertainty of turbulence models is applied for an airfoil test case featuring a small separation zone on the suction surface.

II. Eigenspace perturbation framework

A. Motivation and goal

RANS turbulence models are utilized in order to determine the Reynolds stress tensor $\tau_{ij} = \overline{u_i' u_j'}$ in terms of mean flow quantities $\overline{u}_i = U_i = u_i - u_i'$. The motivation for injecting perturbations to the eigenspace of the Reynolds stress is the inability of LEVM to account correctly for the anisotropy of Reynolds stresses. This is due to the Boussinesq assumption, approximating the turbulent stresses in similar manner to the molecular viscous stresses. The Boussinesq approximations reads

$$\tau_{ij} = -2\mu_t \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) + \frac{2}{3} k \delta_{ij} \,, \tag{1}$$

where the turbulent kinetic energy is defined as $k = \frac{1}{2}\tau_{kk}$ and summation over recurring indices within a product is implied. The strain-rate tensor is denoted as S_{ij} and the eddy viscosity, which derived from the transport equations of the turbulence model, is represented by μ_t.

Based on the epistemic uncertainty, which is introduced into turbulence models by choosing the actual closure model [1], the perturbation approach tries to derive and quantify the effects on quantities of interest (QoI), e.g. the pressure field, by modifying the anisotropy of turbulence within physical limitations. The implemented framework for uncertainty quantification of turbulence models seeks to sample from solutions, which are the result of perturbed states of the Reynolds stress tensor. In this manner a CFD practitioner may get the chance to estimate the sensitivity of some QoI regarding the uncertainty bounds of the turbulence model. In the following section we explain how to obtain a perturbed state of the Reynolds stress tensor and how to apply machine learning in order to get a better uncertainty prediction.

B. Data-free approach

The symmetric Reynolds stress tensor can be expressed by applying an eigenspace decomposition as

$$\tau_{ij} = k \left(a_{ij} + \frac{2}{3}\delta_{ij} \right) = k \left(v_{in} \Lambda_{nl} v_{jl} + \frac{2}{3}\delta_{ij} \right) \,. \tag{2}$$

Equation (2) includes the split into the anisotropy tensor a_{ij} and the isotopic part of τ_{ij}. The eigenspace decomposition provides the eigenvector matrix v and the diagonal eigenvalue matrix Λ, where the eigenvalues represent the shape and the eigenvectors the orientation of the tensor. Emory et al. [3] propose a strategy to perturb the eigenvalues and eigenvectors in equation (2), resulting in a perturbed state of the Reynolds stress tensor

$$\tau_{ij}^* = k \left(v_{in}^* \Lambda_{nl}^* v_{jl}^* + \frac{2}{3}\delta_{ij} \right) \,. \tag{3}$$

The eigenvalue perturbation (determining Λ^*) makes use of the fact, that every physical, realizable state of the Reynolds stress tensor can be mapped onto barycentric coordinates

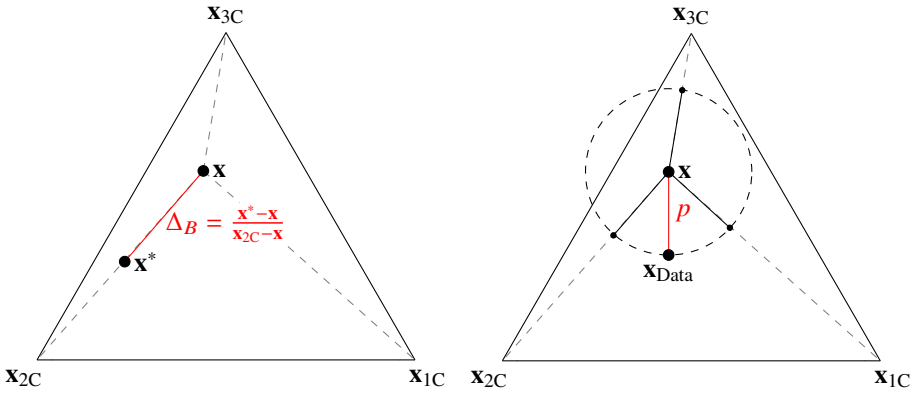
$$\mathbf{x} = \mathbf{x}_{1C} \frac{1}{2} (\lambda_1 - \lambda_2) + \mathbf{x}_2 (\lambda_2 - \lambda_3) + \mathbf{x}_{3C} \frac{1}{2} (3\lambda_3 + 1) \quad \text{with} \quad \lambda_1 \geq \lambda_2 \geq \lambda_3 \,, \tag{4}$$

which is essentially a linear transform according to $\mathbf{x} = \mathbf{B}\boldsymbol{\lambda}$ ($\boldsymbol{\lambda}$ being a vector containing the three eigenvalues λ_i) [8]. Figure 1(a) shows the three limiting states of the Reynolds stress tensor, represented by the one-, two- and three-component (isotropic) turbulent state (1C, 2C and 3C) in the corners of the triangle ($\mathbf{x}_{1C}, \mathbf{x}_{2C}, \mathbf{x}_{3C}$). Thus, Professor Iaccarino's group [4] defined the eigenvalue perturbation as a shift in barycentric coordinates towards each of the limiting states to location $\mathbf{x}^*$, according to

$$\mathbf{x}^* = \mathbf{x} + \Delta_B \left(\mathbf{x}_{(t)} - \mathbf{x} \right) \,. \tag{5}$$

The relative distance $\Delta_B \in [0, 1]$ controls the magnitude of eigenvalue perturbation towards the corner state $\mathbf{x}_{(t)} \in \{\mathbf{x}_{1C}, \mathbf{x}_{2C}, \mathbf{x}_{3C}\}$. The perturbed eigenvalues λ_i^* can be remapped by

$$\boldsymbol{\lambda}^* = \mathbf{B}^{-1}\mathbf{x}^* \,. \tag{6}$$



(a) Relation between unperturbed and perturbed state (b) Definition and usage of perturbation strength

Fig. 1 Schematic representation of the eigenvalue perturbation approach

The creation of the perturbed eigenvector matrix v^* is purely motivated by manipulating the turbulent production term $P_k = -\tau_{ij}\frac{\partial u_i}{\partial x_j}$. Changing the alignment of the eigenvectors of the Reynolds stress tensor and the strain rate tensor S_{ij} limits the production term to a maximum and minimum value [4]. Maximum turbulent production is obtained by not changing the eigenvectors of the Reynolds stress tensor, meaning that they are identical to the eigenvectors of the strain rate tensor $v_{k_{S_{ij}}}$ due to the Boussinesq assumption in Equation (1). Commuting the first and the last eigenvector of the Reynolds stress tensor leads to minimum turbulent production:

$$\begin{aligned} v_{\text{max}} &= \begin{pmatrix} v_{1_{S_{ij}}} & v_{2_{S_{ij}}} & v_{3_{S_{ij}}} \end{pmatrix} \rightarrow P_{k_{\text{max}}} \\ v_{\text{min}} &= \begin{pmatrix} v_{3_{S_{ij}}} & v_{2_{S_{ij}}} & v_{1_{S_{ij}}} \end{pmatrix} \rightarrow P_{k_{\text{min}}} \end{aligned} \tag{7}$$

When combining the eigenvalue and eigenvector perturbation, not only the shape of the Reynolds stress ellipsoid is modified but also the relative alignment with the principle axes of the mean rate of the strain rate tensor is changed (orientation). In should be noted, that targeting the 3C turbulent state with $\Delta_B = 1$ results in identical eigenvalues and consequently the eigenvector matrix cancels out with its inverse. That is the reason, why there will be no distinction between minimized and maximized turbulent production.
To sum up, the perturbation framework promises to only need five distinct simulations $\in \{(1C, P_{k_{\text{max}}}), (1C, P_{k_{\text{min}}}), (2C, P_{k_{\text{max}}}), (2C, P_{k_{\text{min}}})$ and $3C\}$, in order to get the entire information with reference to the epistemic uncertainty of the underlying turbulence model, if $\Delta_B = 1$ is chosen.

C. Data-driven approach

The data-free approach is a purely physics-based methodology, aiming for understandable uncertainty bounds for the turbulence modelling community. One of the major drawbacks of the proposed method is the fact, that a user has to choose the degree of uncertainty by selecting Δ_B before each investigation. The perturbation amplitude, closely related to the degree of uncertainty, has only to become significant in flow regimes, which contravene the assumptions made in the formulation of the turbulence model. Thus, enabling a spatially varying perturbation of the Reynolds stress tensor seems to be worthwhile.

Heyse et al. propose a strategy to combine a random forest model with the forward eigenvalue perturbation approach [5]. Physical flow features are extracted to train a machine learning model in order to predict the local perturbation strength

$$p = |\mathbf{x}_{\text{Data}} - \mathbf{x}_{\text{RANS}}| = |\mathbf{x}^*_{\text{RANS}} - \mathbf{x}_{\text{RANS}}| \,, \tag{8}$$

as illustrated in Figure 1(b). Forward propagating CFD simulations follow training the model, where the predicted perturbation strength is used to modify the Reynolds stress towards the same three limiting states as in the data-free approach. We combine the data-driven enhancement of the anisotropy tensor perturbation with the data-free manipulation of the eigenvectors of the Reynolds stress tensor, as already described above.

1. Choice of machine learning model

The general concept of machine learning is to approximate the relationship between input quantities (features) and output quantities (targets) to make prediction under similar conditions. There are multiple approaches to approximate these relationships. For the sake of interpretability and usability, decision trees are chosen to be the machine learning model in the presented work. Decision trees (also called regression trees for solving regression problems) learn binary rules (if/else decision rules) to predict target values based on given features [9]. Decision trees tend to overfitting, which means that the model is not able to generalize. A machine learning model is able to generalize, if it performs adequate predictions based on a feature space, that is different than the feature space of the training data. Machine learning models, which are prone to overfitting feature a high variance. The potential accuracy of a machine learning model is also dependent on its bias, which is characterized by the difference between the averages of the predictions and the true values. An inflexible model is not capable to fit the total number of data sufficiently, which is determined as a high bias of the model. The fact, that an increasing flexibility (lower bias) comes along with worse generalization (high variance), is known as the bias-variance trade-off. This trade-off describes the aim to choose a machine learning model, that simultaneously has low variance and low bias [10, 11]. Random forests are based on a number of uncorrelated regression trees and offer the possibility to handle the bias-variance trade-off, while enabling powerful predictions [11]. For this reason, we have chosen to use this ensemble learning technique. Instead of just averaging the prediction of individual regression trees, a random forest makes use of two essential key concepts:

- Bootstrapping: Random sampling (with replacement) of the training data for each individual tree, i.e. each tree is trained on a different data set with equal size.
- Feature Bagging: Random subsampling of features at each decision point (also known as split) for each tree, i.e. every tree uses a different feature space at each binary decision.

In combination with bootstrap aggregation (bagging), which implies averaging the prediction of a number of bootstrapped regression trees, the variance is reduced and overfitting is avoided [12]. In this work, the python library *scikit-learn* [13] is used to train the random forests and evaluate their predictions.

2. Choice of flow features

Selecting of input features, which are relevant for predicting more accurate perturbation magnitude p, is critical for turbulence modelling purposes. It has to be ensured, that the chosen feature list represents physical significance in terms of the desired target quantity. Wang et al. [14] identified four raw quantities

$$Q = (\mathbf{S}, \mathbf{\Omega}, \nabla p, \nabla k) \tag{9}$$

to be a reasonable choice as input data for conducting machine learning based on the LEVM. The two raw input tensors $\mathbf{S}, \mathbf{\Omega}$ represent the strain rate and the rotation rate, while ∇p and ∇k are the gradients of pressure and turbulent kinetic energy. In our work, we apply Q as well, and make use of the normalization scheme, derived by Ling et al. [15]. A normalization factor β and the absolute value of each element α of Q according to

$$\hat{\alpha} = \frac{\alpha}{|\alpha| + |\beta|} , \tag{10}$$

lead to the determination of the non-dimensional raw flow features, which are presented in Table 1.

In order to determine the invariant feature basis of the raw flow features, Wang et al. [14] make use of the Hilbert basis theorem. This theorem states, that a finite number of invariants belongs to each minimal integrity basis for a finite tensorial set [16]. In this manner the minimal integrity basis amount to 47 invariants, which are in the following used as input features for training and evaluating the random forest. We add additional physical meaningful flow features to this exhaustive list of features based on domain knowledge and physical intuition. The additional raw input features, which are presented in Table 2, can be computed by providing the turbulent kinetic energy k, the specific turbulent dissipation rate ω, the molecular viscosity μ, the eddy viscosity μ_t, the distance to the nearest wall d, the local Mach number Ma, the mean velocity U_i and its gradient tensor and the mean pressure p and its gradient vector. The normalization procedure is retained in accordance to Equation (10). Thus, a total number of 56 input features is used for training and

Table 1 Raw flow features for constructing the invariant basis

Description	Normalized input $\hat{\alpha}$	raw input α	normalization factor β
Strain rate	$\hat{\mathbf{S}}$	$\mathbf{S}$	ω
Rotation rate	$\hat{\mathbf{\Omega}}$	$\mathbf{\Omega}$	$\|\|\mathbf{\Omega}\|\|$
Pressure gradient	$\widehat{\nabla p}$	∇p	$\rho\|\|\mathbf{U}\cdot\nabla\mathbf{U}\|\|$
Turbulent kinetic energy gradient	$\widehat{\nabla k}$	∇k	$\omega\sqrt{k}$

Table 2 Physical flow features

Numbering	Description	raw input α	normalization factor β
q_1	Q-criterion	$\frac{1}{2}\left(\|\|\mathbf{\Omega}\|\|^2-\|\|\mathbf{S}\|\|^2\right)$	$\|\|\mathbf{S}\|\|^2$
q_2	Turbulent kinetic energy	k	$\frac{1}{2}U_iU_i$
q_3	Wall-distance based Reynolds Number	$\min\left(\frac{\sqrt{k}d}{50\nu},2\right)$	-
q_4	Pressure gradient along streamline	$U_k\frac{\partial p}{\partial x_k}$	$\sqrt{\frac{\partial p}{\partial x_j}\frac{\partial p}{\partial x_j}U_iU_i}$
q_5	Ratio of turbulent time scale to mean strain time scale	$\frac{1}{\omega}$	$\frac{1}{\|\|\mathbf{S}\|\|}$
q_6	Production term	P_k	$k\omega$
q_7	Mach number	Ma	-
q_8	Eddy viscosity	μ_t	μ
q_9	Ratio of total to normal Reynolds stresses	$\|\|\overline{u_i'u_j'}\|\|$	k

evaluating the random forests. Lastly, when adjusting the input features for training and evaluating the machine learning model, each feature is standardized by removing the mean and scaling to unit variance by applying a standard scaler preprocessing functionality of *scikit-learn* [13].

D. Integration of UQ computation in CFD solver suite TRACE

1. Implementation

The aim of running a CFD simulation with a perturbed Reynolds stress tensor is to obtain a new perturbed solution. For smooth and time-efficient simulations, it is advisable to start the perturbation from a sufficiently converged baseline RANS simulation (baseline means standard unmodified turbulence model). Mishra et al. [17] apply a factor to march the solution based on the perturbed Reynolds stress tensor to a fully converged state. In our implementation, we stick to the notation and use a factor f for the reconstruction of Reynolds stresses in order to be able to achieve fully converged perturbed solutions. We discuss the necessity and the effect of this factor in II.D.2. The perturbation of the Reynolds stress tensor was implemented to the existing C code of TRACE and can be subdivided in several steps within each pseudo-time step of steady simulations:

1) Calculation of Reynolds stresses based on Boussinesq approximation in Equation (1)
2) Determination of anisotropy tensor (see Equation (2))
3) Perturbation of anisotropy tensor within physical realizable limits by selecting Δ_B (see Equation (5)) and whether the turbulent production term should be minimized or maximized (see Equation (7)).
4) Reconstruction of perturbed Reynolds stress tensor according to

$$\tau^*_{ij_f}=\tau_{ij}+f\left[k\left(a^*+\frac{2}{3}\delta_{ij}\right)-\tau_{ij}\right], \tag{11}$$

where $f\in[0,1]$ is the described moderation factor, adjusting the total amount of newly perturbed anisotropy

tensor to be considered.

5) Update of the viscous fluxes using perturbed Reynolds stresses explicitly
6) Update of the turbulent production term $P_k = -\tau_{ij}\frac{\partial u_i}{\partial x_j}$ using the perturbed Reynolds stresses explicitly

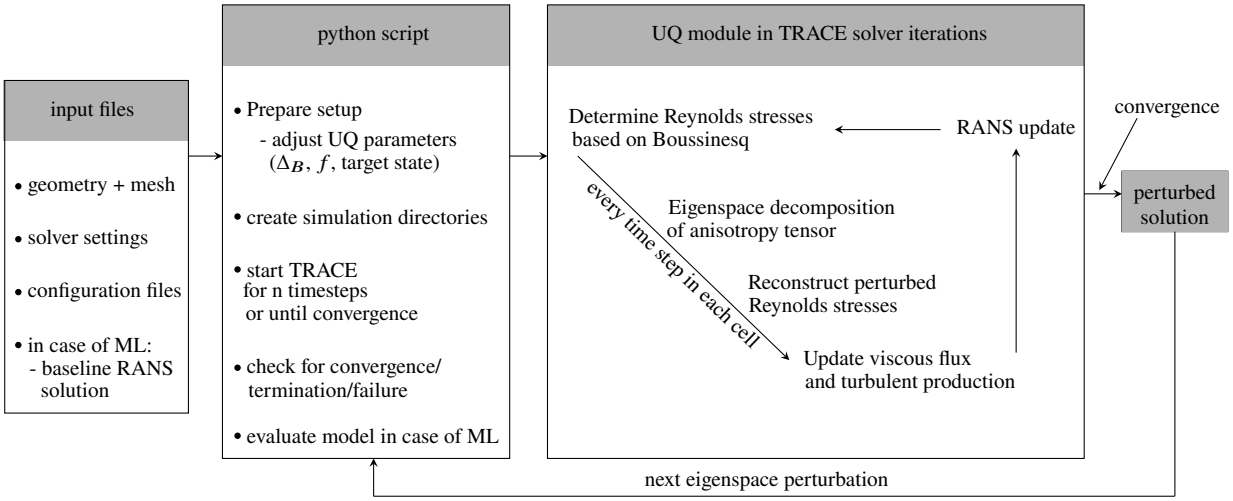


Fig. 2 Implementation of the UQ framework within CFD solver suite TRACE

TRACE has a python interface available, called pyTRACE [18], which can be used to conduct a full set of perturbed simulations and sample the results of QoI. In case of applying a data-driven perturbation of the Reynolds stresses, the python script takes also charge of evaluating a previously in preprocessing trained machine learning model based on extracted mean flow quantities. The high-level python script takes input parameters, containing information regarding the geometry, mesh resolution and additional solver settings. Furthermore, the set of intended perturbed simulations is set up, including selecting the turbulence limit state ($\mathbf{x}_{(t)} \in \{\mathbf{x}_{1C}, \mathbf{x}_{2C}, \mathbf{x}_{3C}\}$), the relative distance $\Delta_B \in [0, 1]$, the alignment of the Reynolds stress eigenvectors with the strain rate tensor ($v_{\min}$ or $v_{\max}$) and the moderation factor $f \in [0, 1]$, as described earlier. As illustrated in Figure 2 the integration of the UQ module in the TRACE simulation run is conducted every time step (steady simulation) in each cell of the computational domain. When the converged perturbed solution is reached, the python script takes charge of setting up the next desired perturbation.

2. Discussion about restriction of perturbation framework

The eigenspace perturbation methodology, presented in II, is solely motivated on quantifying the epistemic uncertainties of LEVM due to the inaccurate account for anisotropic flow phenomena. Professor Iaccarino's group [3, 4, 17] designed this method based on the mathematical derivations of Lumley [19] and Banerjee [8]. These derivations map the states of the Reynolds stress tensor, when it features one, two or three non-zero eigenvalues, onto corners of a constructed triangle, called the barycentric triangle. These states are described to be the extremal states of the Reynolds stress tensor, since the turbulence is only present in one, two or three directions - the corresponding directions of the eigenvectors.

A CFD practitioner is interested in getting to know the effect of the turbulence model uncertainty on certain QoI, which is relevant for the design. However, the relation between the one-, two-, and three-component corner of the Reynolds stress tensor and some QoI is anything but linear. As a consequence, modifying/perturbing the turbulent state of the Reynolds stress tensor towards the corners does not necessarily lead to an extremal state of QoI. Thus, one can only state that the perturbation framework seeks to estimate the uncertainty bounds of some QoI.

Nonetheless, we analyzed the relation of barycentric coordinates and QoI for selected flow cases by sampling points inside the barycentric triangle and propagating the perturbed Reynolds stress tensor in an earlier preinvestigation (not shown here). Therefore, assessed against currently available data, we agree on the fact, that the corners of the barycentric coordinate produce adequate estimate of the uncertainty bounds in most of the flow regions. Although, there might be areas of the flow solution, where the extremal state of turbulence is not corresponding to the extremal state of some QoI. This observation will be also discussed in the application section (see V).

Additionally, we would like to discuss the effect of the moderation factor f, which was already introduced by Mishra et al. [17]. The goal of f is to achieve a converged simulation based on the perturbation approach. We can agree on the fact, that this factor is needed for convergence issues, since some perturbed states tend to be unstable. Understandably, this is the case, for perturbations seeking to decrease the turbulent production term, when aiming for $P_{k_{\text{min}}}$ and/or 3C. Nevertheless, it is shown in Appendix, that the moderation factor is actually a moderation factor for the prescribed Δ_B in case of purely perturbing the eigenvalues of the anisotropy tensor. Generally speaking, Equation (11) reveals, that using $f \leq 1$, helps to stabilize the CFD-simulation, by weakening the effect of perturbation (whether or not eigenvectors are modified).
Accordingly, users are encouraged to not only stating the prescribed Δ_B but also the factor f. Moreover, regardless of the fact, that damping the effects of the actual perturbation (eigenvalues and eigenvectors) is necessary in certain cases, the interpretability of the limiting states of turbulence, represented by the corners of the barycentric triangle, gets weaker.

E. Data sets for training, testing and applying the machine learning model

As already described in I, the final well-trained machine learning model should be sensitive to flow phenomena such as adverse-pressure gradient, separation and reattachment due to the known shortcomings of the LEVM in these regions. Consequently, it is reasonable to use data sets for training, which contain these flow situations as well. We are continuously striving for extending our database, which contains various flow cases for machine learning. For the presented worked, we used the following flow cases:

- DNS of turbulent channel flow at $Re_\tau \in \{180, 550, 1000, 2000, 5200\}$ based on Lee and Moser [20]
- DNS at $Re_H \in \{2800, 5600\}$ and LES at $Re_H = 10595$ of periodic hill flow based on Breuer et al. [21]
- DNS of wavy wall flow at $Re_H = 6850$ based on Rossi [22]
- DNS of converging-diverging channel flow at $Re_\tau = 617$ based on Marquillie et al. [23]

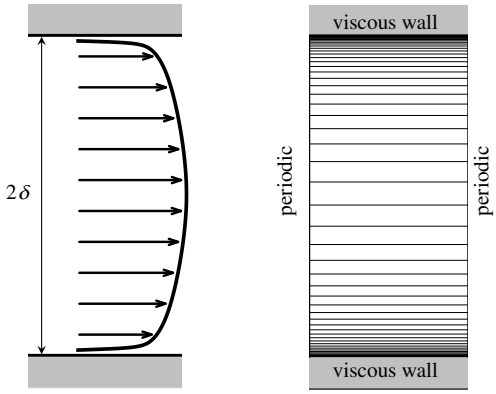
All the DNS and LES data of the described test cases are generated using incompressible solvers. In order to simulate these incompressible flows using the compressible solver TRACE without a low-Mach preconditioning, the simulations are scaled (adapting dimensions of the geometry and/or molecular viscosity) to an incompressible Mach number of about 0.1, while preserving the intended Reynolds numbers. The two-equation, linear eddy viscosity Menter SST $k-\omega$ turbulence model is selected as the baseline model for all conducted RANS simulations [24]. In order to obtain steady state solutions, an implicit time marching algorithm is applied. A flux difference splitting approach is employed to discretize the convection terms making use of a second order accurate Roe scheme in combination with MUSCL extrapolation.
To evaluate proper features as input parameters based on the RANS simulations of the listed flow cases, we conducted a mesh convergence study for each of these. Although the mesh convergence studies are not presented here due to the scope of the paper, we affirm, that we only used RANS simulation data, which shows sufficient grid convergence using a low-Reynolds resolution ($y^+ \leq 1$) at solid walls. The perturbation magnitude p can be determined accordingly to Equation (8) by comparing scale-resolving and RANS solution. In order to generate that intended target quantity of the random forest, the scale-resolving data has to be interpolated onto the RANS data points for every test case. Due to numerical issues some RANS data samples may be located outside the barycentric triangle in terms of anisotropy tensor (see Figure 1). Therefore, we included the opportunity to remove these samples from the training or testing sets. The final application of the UQ perturbation approach is presented for the airfoil test case NACA 4412 at $Re_c = 1.52 \cdot 10^6$.

1. Turbulent channel

Initially, the turbulent channel flow serves as proof of concept in our earlier publications (see [6]). Although, we are more interested in more complex cases, the channel flow data serves as one of the key properties, what the model should be able to recognize and predict: turbulent boundary layer with inaccurate anisotropy represented by the LEVM close to the wall. The configuration for simulating the turbulent boundary layer is sketched in Figure 3. The characteristic Reynolds number is defined by

$$Re_\tau = \frac{\rho u_\tau \delta}{\mu} \,, \tag{12}$$

where δ is the channel half-height and the friction velocity is known as $u_\tau = \sqrt{(\tau_w/\rho)}$ with $\tau_w = \mu \frac{\partial U}{\partial y}|_{y=\text{wall}}$. The turbulent channel flow is homogeneous in the streamwise direction x and the spanwise direction z. A constant pressure gradient $\partial P/\partial x$ is applied to balance the skin friction at the wall. We use the available RANS grid cells in one half of the channel at the five different Reynolds numbers as subsequent data points for training the random forest.



(a) Schematic sketch of a fully developed turbulent boundary layer

(b) $Re_\tau = 1000$ (every fourth line shown) mesh and boundary conditions; symmetry is enforced in spanwise direction

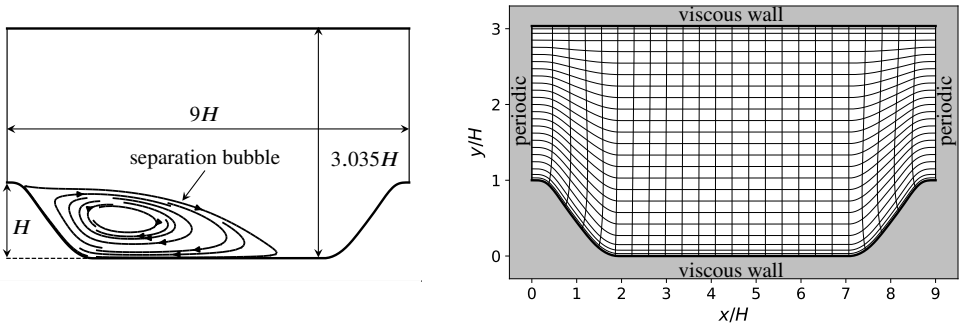
Fig. 3 Turbulent channel flow simulation

2. Periodic hill

The flow over periodic hills features flow separation from curved surfaces, recirculation and a subsequent reattachment on the flat bottom of the channel. Since the Reynolds number has a strong impact on the actual size of the separation bubble, it is worthwhile to add three different Reynolds number flows to our training set. The Reynolds number based on the bulk velocity U_B, evaluated at the crest of the hill, and the hill height H is defined as

$$Re = \frac{\rho U_B H}{\mu} \,. \tag{13}$$

For simulating the periodic hill configuration, periodic boundary conditions are applied as illustrated in Figure 4. A constant pressure gradient $\partial P/\partial x$ is applied to move the fluid through the configuration. The available scale-resolving data sets of the periodic hill only contain data at certain slices ($x/H \in \{0.05, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0\}$). Consequently, the RANS solution is sliced at these locations accordingly. The scale-resolving data is interpolated onto the wall-normal RANS data positions, in order to generate the desired target quantity for the machine learning model.



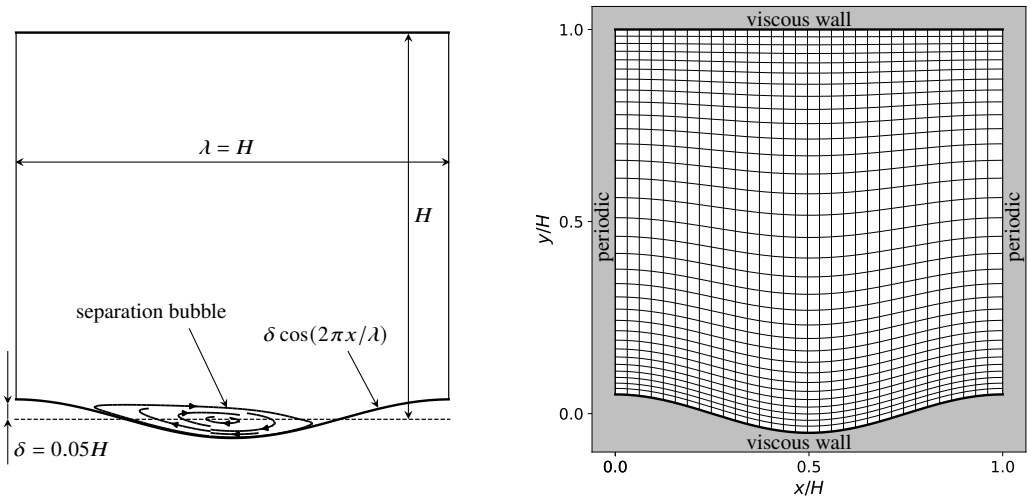
(a) Relative dimensions and sketch of the flow

(b) Mesh ($Re_H = 10595$, every tenth line shown) and boundary conditions; slip conditions/inviscid walls are applied in spanwise direction

Fig. 4 Schematic periodic hill setup

3. Wavy wall

The wavy wall test case is confined by a plane wall and a wavy surface, which is sketched in Figure 5. In former experimental settings, the desired flow situation was generated by stringing together multiple hills and valleys, described by a cosine function. For the CFD simulations (DNS and RANS) periodic boundary conditions in streamwise direction can be applied. In order to adjust the intended Reynolds number of $Re_H = 6850$, based on the bulk velocity and the mean channel height evaluated on the hill crest (accordingly to Equation (13)), a constant pressure gradient $\partial P/\partial x$ is used. Since the available DNS data set is two-dimensional covering the entire domain size, we use all available RANS grid cells as subsequent data points for training the random forest.



(a) Relative dimensions and sketch of the flow

(b) Mesh (every fourth line shown) and boundary conditions; slip conditions/inviscid walls are applied in spanwise direction

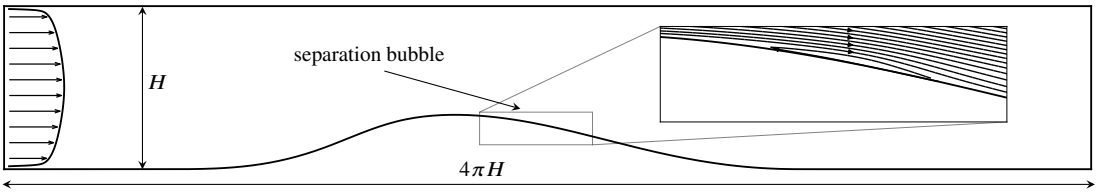
Fig. 5 Schematic wavy wall setup

4. Converging-diverging channel

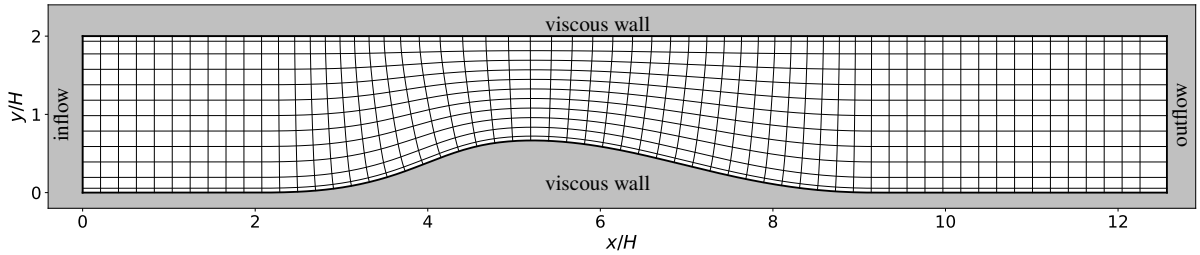
The configuration of a converging followed by a diverging section is an ideal test case to investigate the effect of an adverse pressure gradient with and without curvature. The flow separates slightly at the diverging part at the lower wall but not on the flat top wall, as shown in Figure 6. Similar to the DNS, the inflow boundary conditions are derived from a fully developed turbulent boundary layer at $Re_\tau = 617$. A constant mass flow rate is prescribed at the outflow of the domain, which was derived based on the domain size and the bulk quantities of the streamwise velocity and density of the inflow profile. Since the available DNS data set is two-dimensional as well, we are able to provide all available RANS grid solution points as subsequent data points for training the machine learning model.

5. NACA 4412 airfoil

In order to demonstrate the application of the UQ framework with and without a machine learning model, the near-stall NACA 4412 airfoil is chosen in the presented work. This test case is a NASA benchmark case for testing turbulence models, featuring boundary layer separation close to the trailing edge. The operation condition of this airfoil is a Reynolds number of $Re_c = 1.52 \cdot 10^6$ based on the freestream velocity U_{inf} and the chord length c and a Mach number based on U_{inf} of $Ma = 0.09$ at an angle of attack of 13.87°. The CFD results in this paper are compared against the experimental measurements of Coles and Wadcock [25]. In order to minimize the effect of boundary conditions on the CFD simulation results, far-field conditions are applied to prescribe the specified flow conditions (see Figure 7). A turbulence intensity of 0.086% and an eddy viscosity ratio μ_t/μ of 0.009 is prescribed in accordance with the description of NASA [26]. The used mesh topology is the so-called C-grid featuring a grid cell resolution of $n_x, n_y, n_z = 896, 256, 1$, which can be downloaded NASA's turbulence modeling resource database [26].

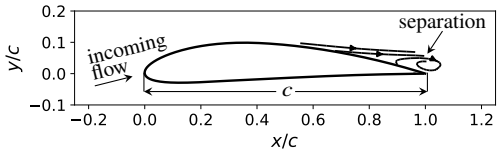


(a) Relative dimensions and sketch of the flow

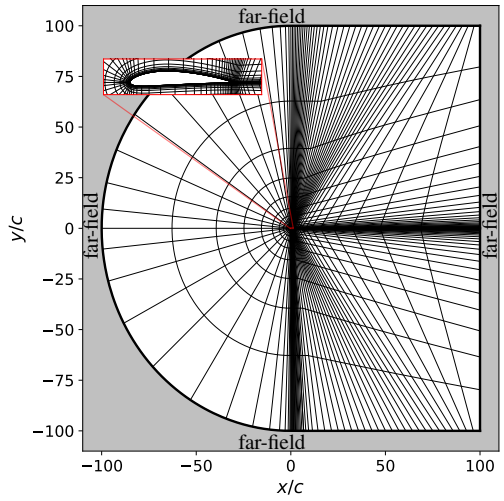


(b) Mesh (every fourth line in streamwise direction and every twentieth line in wall normal direction shown) and boundary conditions; slip conditions/inviscid walls are applied in spanwise direction

Fig. 6 Schematic converging-diverging setup



(a) Relative dimensions and sketch of the flow



(b) Mesh (every eighth line shown) and boundary conditions; slip conditions/inviscid walls are applied in spanwise direction

Fig. 7 Schematic NACA 4412 setup

III. Hyperparameter selection based on generalization study

Before the training of the final random forest regression model was conducted, the impact of four different hyperparameters

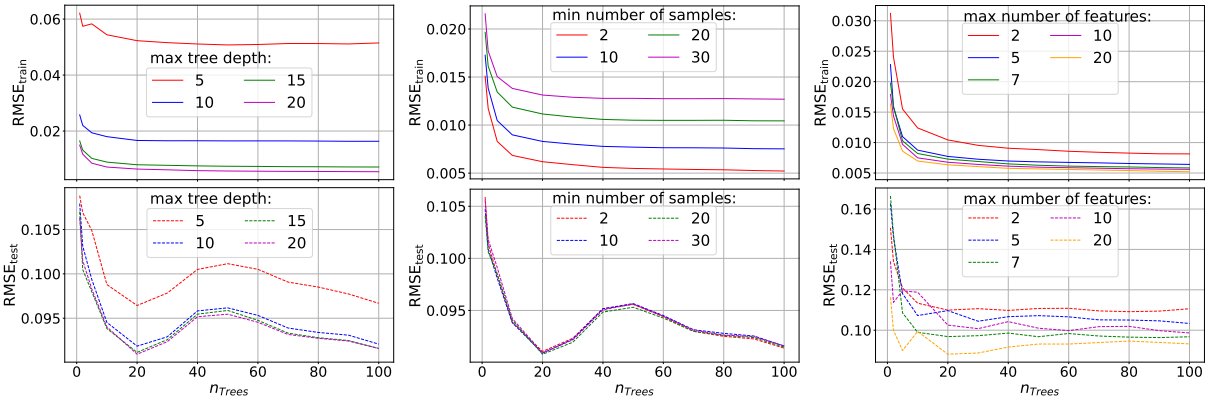
- maximum tree depth: maximum number of decision nodes from the root down to the furthest node allowed
- minimum sample count: minimum number of data samples required at a decision node allowed
- maximum number of features: maximum number of features randomly chosen at each decision node allowed
- number of trees: total number of individual decision trees used

on the accuracy is evaluated. Since the final trained model should be able to generalize for different geometries and flow conditions, it seems to be reasonable to evaluate these hyperparameters with focus on the generalization capabilities of the random forest. Therefore, three out of the four available training data sets (see II.E) are used for training, while the remaining flow case is used to verify the model (see Table 3). Data samples featuring non-physical Reynolds stress tensors (barycentric coordinates are located outside the barycentric triangle) are removed from each data set.

Table 3 Scenarios for hyperparameter study: x means part of training data, $\circ$ means verification data

	Scenario			
Flow cases	I	II	III	IIII
turbulent channel	x	x	x	$\circ$
periodic hill	x	x	$\circ$	x
wavy wall	x	$\circ$	x	x
converging-diverging channel	$\circ$	x	x	x

For each of the first three hyperparameters several different values were studied over a range of the total amount of individual regression trees, while the other two hyperparameters were set to default values (see *scikit-learn* documentation for further information). As an example, Figure 8 presents the effect of the considered hyperparameters on the accuracy of the model prediction in scenario I, where the accuracy is expressed in terms of the root mean square error (RMSE).



(a) Maximum depth of trees over total number of individual trees

(b) Minimum number of required samples over total number of individual trees

(c) Maximum number of active features over total number of individual trees

Fig. 8 Training accuracy (solid) and verification accuracy (dashed) based on RMSE **for selection of hyperparameters in scenario I**

The subsequent statements made coincide with all evaluated scenarios as well. With increasing number of maximum depth of each individual tree, the complexity increases and the rate of overfitting is prone to extend. Because a maximum tree depth of 15 showed excellent performance for the training and test data, it is preferred compared to a higher value of 20, which increases computational costs (see Figure 8(a)).
The smaller the number of data samples for each decision node, the more accurate the performance on the training data (see Figure 8(b)). Since the RMSE based on evaluated testing data is not significantly affected on this hyperparameter, a minimum sample count of 10 is chosen. This enables the model to generalize more, than selecting a smaller value.
A larger number of selected features for each decision node lowers the training error and increases the risk of overfitting. Since a maximum number of 7 features produces accurate prediction performance for the test data as well (in Figure 8(c) it is actually the best choice), it is selected as inferred hyperparameter.
In terms of total number of individual trees, one can observe a steep drop in RMSE for small numbers followed by a constant level of accuracy. The computational costs scale linearly with the number of individual trees. Although

computational costs do not really play a relevant role for our application, as we only evaluate the model once before each simulation run, we sought for the minimum number of trees for maximized performance of the model. Therefore, we concluded to use a total number of 30 individual trees for the random forest by evaluating all described scenarios in Table 3.

IV. Verification of trained machine learning model

Based on the chosen hyperparameters, which was discussed in the previous section, the prediction accuracy of the random forest should be evaluated on the available data (see II.E). Ten different scenarios based on combinations of the flow data cases, listed in Table 4, serve as verification of the right functionality and present the accomplishment of the intended generalization purposes of every model. In order to do so, the RMSE serves as a quantity to compare the prediction accuracy between different scenarios, while data samples featuring non-physical Reynolds stress tensors (barycentric coordinates are located outside the barycentric triangle) are removed from the data sets. Although the prediction accuracy is high for all scenarios, if the flow case, which is chosen for evaluating the RMSE is also part of the training data set, the accuracy of the trained random forest increases, as expected.

Table 4 Prediction accuracy of random forest: x means part of training data, $\circ$ means not part of training data, red means data set for evaluation of RMSE

	Scenario									
Flow cases	I	Ia	II	IIa	III	IIIa	IIIb	IIII	IIIIa	V
turbulent channel	x	x	x	x	x	x	x	$\circ$	x	x
periodic hill										
· $Re_H \in \{2800, 5600, 10595\}$	x	x	x	x	$\circ$	x		x	x	x
· $Re_H \in \{2800, 10595\}$							x			
· $Re_H = 5600$							$\circ$			
wavy wall	x	x	$\circ$	x	x	x	x	x	x	x
converging-diverging channel	$\circ$	x	x	x	x	x	x	x	x	x
RMSE $(p_{\text{pred}}, p_{\text{true}})$	0.098	0.010	0.133	0.029	0.095	0.028	0.041	0.051	0.014	0.013

As soon as a trained machine learning model should make predictions on flow cases, for which accurate data does not exist, judging the model's prediction in terms of accuracy is hardly possible. In order to build the confidence of a machine learning model, the coverage of the training data with reference to a newly used, previously unseen case needs to be evaluated.

One way to do this, is to determine extrapolation metrics, measuring the distance between a test point $\tilde{m}$ and the training data feature set m_i for $i = 1, \ldots, n$ with n as the number of training data points. In this paper, we use the Kernel Density Estimation (KDE) to compute the distance by estimating the probability density

$$f_{\text{KDE}} = \frac{1}{n\sigma^d} \sum_{i=1}^{n} \prod_{j=1}^{d} K\left(\frac{\tilde{m}_j - m_{i,j}}{\sigma}\right), \tag{14}$$

with being d the number of features and σ the bandwidth, determined by Scott's rule [27]. In accordance with the work of Wu et al. [28], we use a Gaussian kernel $K(t) = 1/\sqrt{(2\pi)} \exp\left(-t^2/2\right)$ and the distance is computed as follows:

$$d_{\text{KDE}} = 1 - \frac{f_{\text{KDE}}}{f_{\text{KDE}} + 1/A} \quad \text{with} \quad A = \prod_{i}^{n} \left(\max_j\left(m_{i,j}\right) - \min_j\left(m_{i,j}\right)\right) \quad \text{for} \quad j = 1, \ldots, d \tag{15}$$

Due to the normalization of the distance, it is able to measure the distance of $\tilde{m}$ to the training data with respect to a uniform distribution. This enables a user to interpret the rate of extrapolation needed based on the training data set. Two extreme scenarios are represented according to

- $d_{\text{KDE}} = 0$: no extrapolation is required; the features of the training data set cover the feature of the test point $\tilde{m}$
- $d_{\text{KDE}} = 1$: high extrapolation is required; the features of the test point $\tilde{m}$ are far off the features of the training data.

Equation (14) increases, if $\tilde{m}$ becomes close to a concentrated feature space of the training data points and vice versa. Thus, on the one hand, if $\tilde{m}$ is close to a concentrated feature space, $f_{KDE} \gg 1/A$ implies $d_{\mathrm{KDE}} \to 0$. On the other hand, $d_{\mathrm{KDE}} \to 1$ follows from $f_{KDE} \ll 1/A$. Put simply, the smaller the KDE distance, the more similarity of the training and the test set is expected.
Since the extrapolation metric only assesses the closeness of the features between training and test data sets, Wu et al. [28] demonstrate, that the KDE extrapolation metric can be used to estimate the prediction confidence by quantifying the correlation between the degree of extrapolation and the prediction accuracy.
In our work, the flow case of the converging-diverging channel serves to present the application of the extrapolation metric. The rest of the available data (turbulent channel flow, flow over periodic hills and wavy walls) is used for training individual random forests, while each random forest is evaluated on the converging-diverging channel.
For the computation of the KDE distance d_{KDE}, only selected input features, which are attributed with a higher feature importance compared to the others, are considered. The individual feature importance for each of the 56 input features, is determined after training the final random forest (scenario V in Table 4) using the chosen hyperparameters (see section III). Here, the permutation feature importance, accounting for the reduction in the model accuracy, when a single feature value is randomly shuffled, is applied to determine the helpfulness of each feature by *scikit-learn*'s toolbox [13]. The selected five most important features, to be considered for determining d_{KDE}, are the eddy viscosity q_8, the normalized wall-distance q_3, the Mach number q_7, the turbulent kinetic energy q_2 and the Q-criterion q_1.
Contrary to the work of Wu et al. [28], we cannot confirm a strong correlation between the accuracy of the model, evaluated by predicting the perturbation magnitude p for the converging-diverging channel, and the mean of the KDE distance d_{KDE} for different training data sets, as illustrated in Figure 9. However, Figures 10 and 11 present the explanation for smaller prediction error, when training on the periodic hill compared to the wavy wall. The DNS data based barycentric coordinates of the converging-diverging channel and the periodic hill cover a similar area in the barycentric triangle, while the true values of barycentric coordinates for the wavy wall test case are only placed in the lower range of the triangle. Thus, the derived target quantity, which is the distance in barycentric coordinates, becomes more frequent in a similar range of absolute value. Even Wu et al. [28] mention, that the correlation between accuracy and extrapolation metric is less correlated, if the training set is very similar or very different from the test set, what we might be facing here as well.
Nevertheless, we would like to point out, that the result of the extrapolation metric is highly dependent on the set of considered features. Thus, it seems reasonable to limit the evaluation of the metric to certain important features for the random forest. Lastly, we would like to remark, that the prediction of p (leading to the RMSE in Figure 9) is done by the random forest model, which was trained on all 56 input features and the chosen hyperparameters.

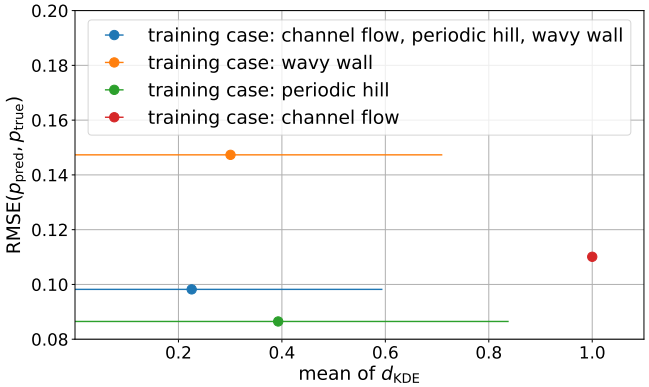
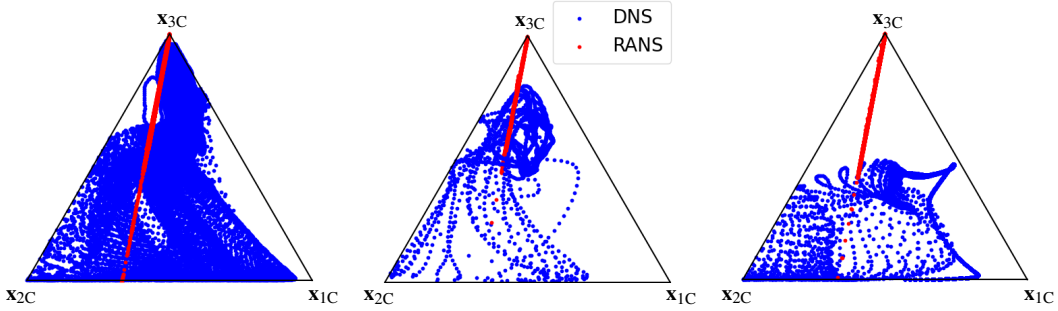


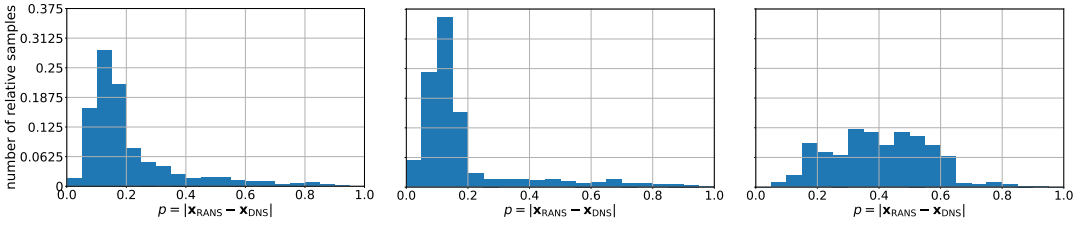
Fig. 9 Relationship between the RMSE of the prediction for the converging-diverging channel and the mean value of KDE extrapolation metric (standard deviation the extrapolation metric is shown as the horizontal bars)

Figure 12(a) presents the two-dimensional distribution of the KDE metric, evaluated based on the five most important features q_8, q_3, q_7, q_2 and q_1, corresponding to the blue data point in Figure 9 and scenario I in Table 4. This can be compared with the deviation of the prediction and the accurate values of the perturbation magnitude p for the converging-diverging channel in Figure 12(b) (which was not part of the training samples for this scenario). When comparing spatial distributions of d_{KDE} and $|p_{\mathrm{pred}} - p_{\mathrm{true}}|$, correlated regions between high KDE distances and



(a) Converging-diverging channel (b) Periodic hill at $Re_H = 5600$ (c) Wavy wall

Fig. 10 Barycentric coordinates for selected flow cases; legend of (b) corresponds to (a) and (c) as well

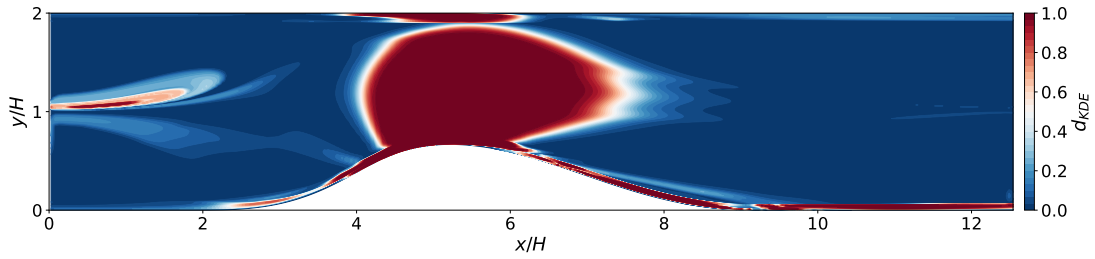


(a) Converging-diverging channel (b) Periodic hill at $Re_H = 5600$ (c) Wavy wall

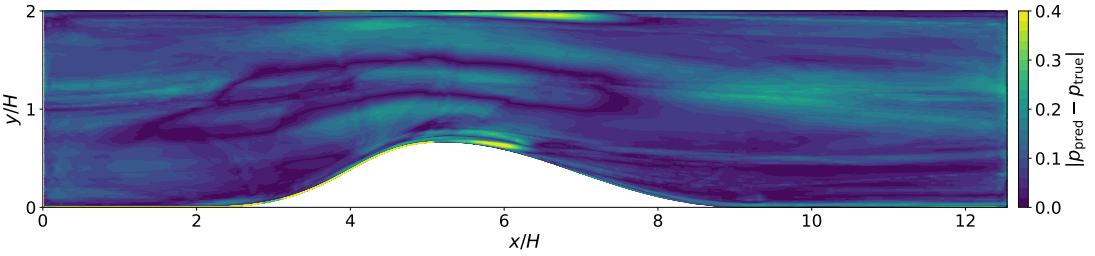
Fig. 11 Frequency of target quantity p for selected flow cases; vertical axis correspond to (a), (b) and (c)

inaccurate model predictions can be recognized, which was also shown by Wu et al. [28]. There are just less accurate model predictions in the area $y/H \approx 1$ and $x/H > 8$, which are not represented by the KDE metric. However, our observation reinforces the trust in the presented KDE distance, when applied in an adequate manner.

Before the actual application of the data-driven UQ perturbation framework on the flow around NACA 4412 can be conducted, the prediction of the random forest model for this case should be discussed. Figure 13(a) shows the evaluated metric distance based on the five most important flow features q_8, q_3, q_7, q_2 and q_1, which are extracted based on the baseline RANS solution. Solely regions close around the airfoil contain $d_{\mathrm{KDE}} \leq 1$. We believe, that this is due to the very different geometry, Reynolds number and flow situation, compared to the training data. This fact also manifests, when taking a look at the level of turbulence intensity for the baseline solution. Because far-field boundary conditions are used, laminar flow is present almost everywhere in the CFD domain. An area with $Tu \geq 0.01\%$ can only be identified in the boundary layer around the profile, in the wake flow and around the stagnation point on the pressure side at $x/c \approx 0.035$. Since the entire mapping of the Reynolds stress tensor onto barycentric coordinates makes only sense for relevant turbulent stresses, which are significantly larger than machine precision, the random forest is restricted to predict values only in the area featuring $Tu \geq 0.01\%$. Figure 13(b) shows the restricted random forest prediction for p trained on all training cases (scenario V in Table 4) and considering all input features. The decision of restricting the actual prediction to certain areas in the flow field is purely based on our observations and physical intuition. The verification of this procedure was numerically justified, by comparing forward data-driven UQ computations based on model predicted p. Some simulations contained p, determined by the random forest, everywhere in the domain, others only in the area $Tu \geq 0.01\%$. Although the random forest is able to predict certain values greater than zero in the region featuring $Tu \leq 0.01\%$, the evaluated flow quantities around the NACA 4412 airfoil did not show any significant difference (not shown here). Even the KDE distance in Figure 13(a) confirms this procedure, by stating, that a machine learning practitioner can only trust the prediction in the area with higher values of turbulence intensity. Thus, these preinvestigations reveal, that restricting the model prediction to certain area closely around the airfoil makes sense from a physical and a machine learning perspective. However, the presented two-dimensional distributions of extrapolation metric and model performance reveal, that predicting the influence of a separated region in terms of anisotropy discrepancy is a very challenging task. This issue might be only overcome with an increasing number of training data sets featuring varying flow conditions and geometry.

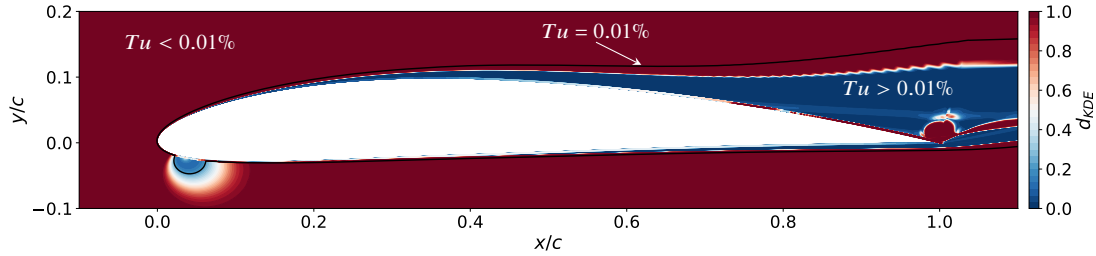


(a) Evaluated metric distance

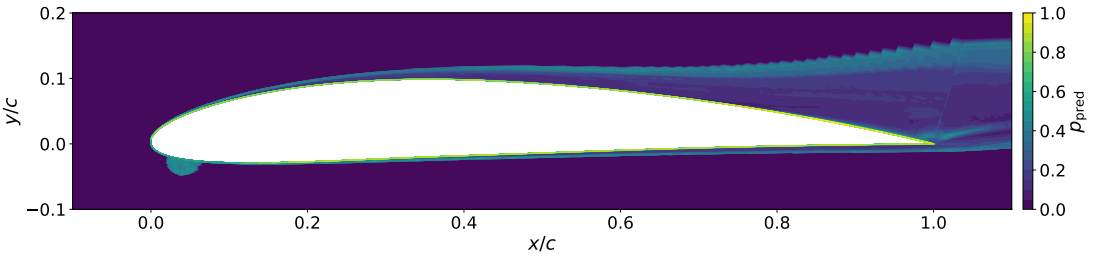


(b) Model prediction p_{pred} presented as absolute deviation based on true values

Fig. 12 Verification of metric for converging-diverging channel



(a) Evaluated metric distance, black line divides areas with higher and lower turbulence intensity with a threshold of 0.01%



(b) Model prediction in the area of $Tu \geq 0.01\%$ (p_{pred} is set to zero for area with $Tu < 0.01\%$)

Fig. 13 Evaluated metric and perturbation magnitude for NACA 4412 profile

V. Application of UQ perturbation framework

The flow around NACA 4412 offers a possibility to demonstrate the general framework of the UQ perturbation approach, presented in II. Before the actual perturbed solutions and the UQ estimates are discussed, we want to make a few points on the general RANS performance of the Menter SST $k-\omega$ on this configuration at a Reynolds number of $Re_c = 1.52 \cdot 10^6$ and a Mach number of $Ma = 0.09$ at an angle of attack of 13.87°.

The baseline simulation conducted with TRACE is in accordance with the presented RANS solutions using the identical

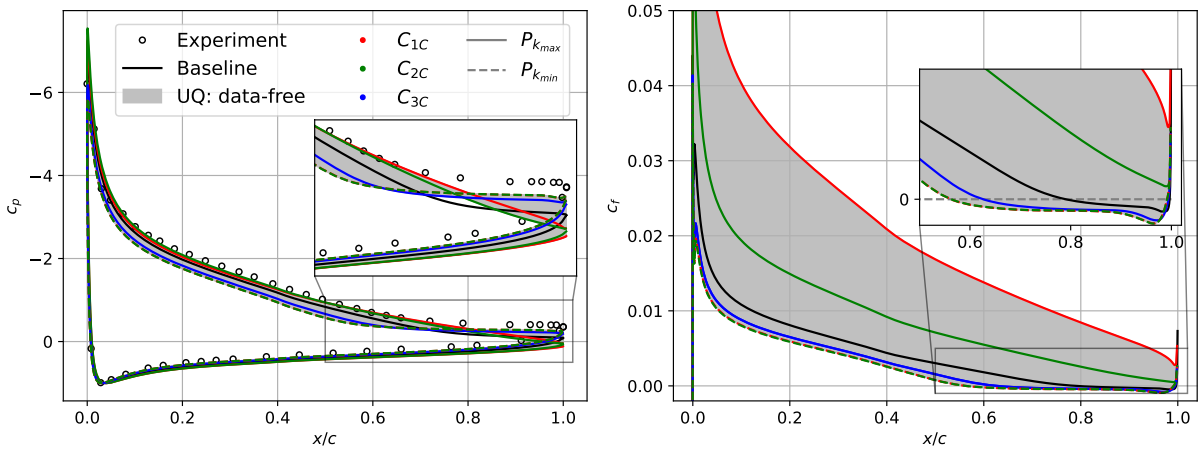
turbulence model on the NASA's turbulence modeling resource site. Although, this test case serves as a turbulence model validation case, there may be still small discretization errors [26]. Similar to NASA's observations, when conducting the steady simulation, CFL = 1 has to be used in order to reach a fully converged steady-state solution with TRACE, which may be also a consequence of the discretization. The main difference in comparison with the experimental surface pressure measurements conducted by Coles and Wadcock [25] can be observed at the suction side trailing edge. The pressure coefficient, presented for example in Figure 14(a), is computed accordingly as

$$c_p = \frac{p - p_{\text{inf}}}{\frac{1}{2}\rho_{\text{inf}} U_{\text{inf}}^2}\,, \tag{16}$$

while the reference quantities are the ones in infinity, which are prescribed at the far-field boundary condition. It is important to note, that the reference velocity, used to nondimensionalize the pressure coefficient, is evaluated at a different location in the experiment and in the CFD setup. Based on the best practice guidelines and in order to retain similar results compared to the NASA findings [26], we apply the far-field freestream velocity here. To sum up, the used quantities are $U_{\text{inf}} = 31.2$ m/s, $p_{\text{inf}} = 76914.1$ Pa and $\rho_{\text{inf}} = 0.9$ kg/m^3, which can be derived by the specified Mach and Reynolds number, when choosing appropriate gas properties.

Table 5 Moderation factor f for each perturbed simulation of NACA 4412 UQ estimation

	Data-free					
Perturbed simulation:	$(1C, P_{k_{\max}})$	$1C, P_{k_{\min}})$	$(2C, P_{k_{\max}})$	$(2C, P_{k_{\min}})$	3C	
moderation factor	1.0	0.03	1.0	0.05	0.1	
	Data-driven					
Perturbed simulation:	$(1C, P_{k_{\max}})$	$(1C, P_{k_{\min}})$	$(2C, P_{k_{\max}})$	$(2C, P_{k_{\min}})$	$(3C, P_{k_{\max}})$	$(3C, P_{k_{\min}})$
moderation factor	1.0	0.2	1.0	0.2	1.0	0.2



(a) Pressure coefficient c_p

(b) Friction coefficient c_f

Fig. 14 Surface quantities for flow around NACA 4412 including data-free evaluation of uncertainty estimates for Menter SST $k-\omega$ turbulence model; legend in (a) applies also for (b)

The data-free uncertainty estimates, presented in Figure 14 are the result of perturbed turbulence model simulations using a relative perturbation magnitude $\Delta_B = 1.0$ everywhere in the domain, since there is no justifiable physical reason to reduce the amount of targeting the extremal states of turbulence [3]. Unfortunately, as already discussed

in II.D.2, the perturbed simulations, trying to minimize the turbulent production, come along with stability issues in terms of convergence or even completely diverge from the steady solution. That is why, the moderation factor f has to be adjusted in order to retain acceptable, converged simulations. Besides examining the overall residuals of each simulation, we evaluate the mean blade force in y-direction in order to distinguish between an unacceptable unstable and an acceptable steady solution. Based on physical experience and consequent intuition, we apply a threshold of a relative standard deviation of 2% based on the mean of the overall blade force in y-direction. Put simply, as one is increasing the moderation factor f for simulations minimizing the turbulent production term ($P_{k_{\text{min}}}$), the standard deviation of the mean blade force rises. In our investigations, we are able to use our high-level python script (see Figure 2) to march the moderation factor as high as possible (by steps of 0.1 for $f \in [0.1, 1.0)$ and steps of 0.01 for $f \in [0, 0.1)$). As the designated solutions still contain small variations, we instrument probes on the airfoil surface and average the solution in order to get the mean.

Figure 14 presents the uncertainty estimates based on the intended five perturbed target states, using the moderation factors presented in Table 5. As discussed above, the baseline Menter SST $k-\omega$ simulation shows significant deviation for the prediction of the flow around the trailing edge, attributed to the flow separation. The results of (1C, $P_{k_{\text{max}}}$) and (2C, $P_{k_{\text{max}}}$) minimize this gap on the suction side for $x/c < 0.7$, whereas perturbed simulations minimizing the turbulent production term by modifying the eigenvectors of the Reynolds stress tensor predict an increased static pressure on the suction side for $x/c > 0.7$. The latter is due to an earlier separation zone because of reduced turbulent kinetic energy. This can be also identified, when having a look at the friction coefficient

$$c_f = \frac{\tau_w}{\frac{1}{2}\rho_{\text{inf}} U_{\text{inf}}^2} \,, \tag{17}$$

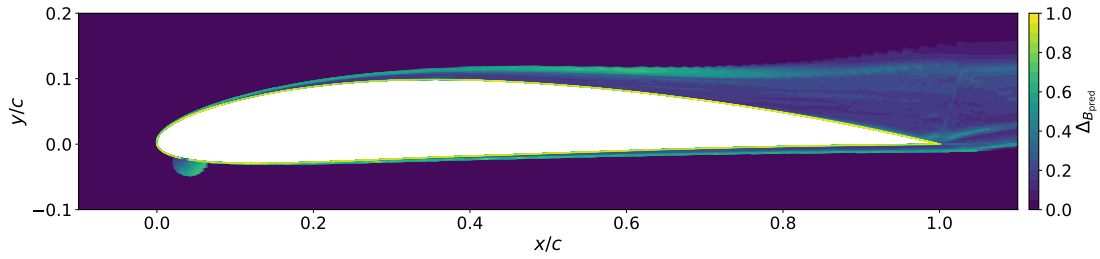
shown in Figure 14(b).

In should be noted, that targeting the 3C turbulent state with $\Delta_B = 1$ results in minimizing the turbulent production term [29], which can be seen on the presented surface quantities as well. In contrast, perturbed solutions (1C, $P_{k_{\text{max}}}$) and (2C, $P_{k_{\text{max}}}$), which do not modify the eigenvectors of the Reynolds stress tensor, completely suppress the separation bubble. The boundary layers of these simulations reveal significantly increased momentum transfer into their viscous sublayers.

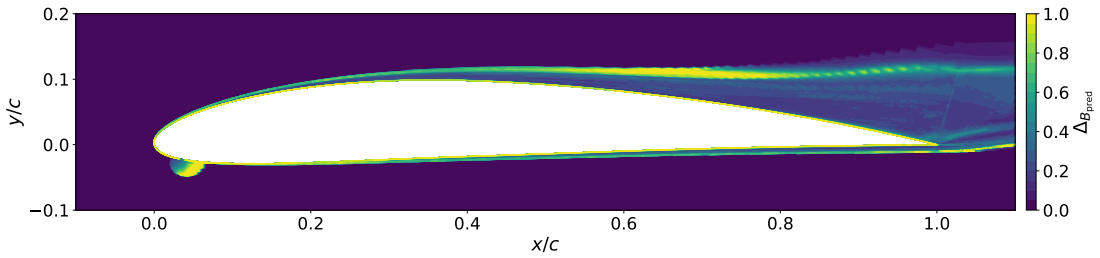
As already mentioned in II.D.2 the UQ estimated bounds by the perturbation framework is only aiming for the extremal state of turbulence in terms of Reynolds stress tensor's shape and orientation. The extremal state of some QoI, in this case the pressure distribution, need not necessarily coincide with this in all areas. In the range of $0.72 < x/c < 0.9$ the baseline solution lies outside of the determined, grey shaded, UQ estimate. On the one hand, this follows from the non-linear relationship of turbulence extreme states and the subsequent impact on some QoI. But on the other hand, it follows from the uniform perturbation in the entire domain, whereas the Reynolds stress tensor based on the unmodified RANS model usually features a distribution along the plane strain line in terms of barycentric triangle. This means, that each solution point of the unmodified RANS model reveals slightly varying barycentric coordinates. Thus, applying random spatial sampling inside the barycentric triangle may result in an overarching UQ estimate (which is out of the scope of this paper).

As already mentioned in section II.C), the random forest predicted perturbation magnitude p around the airfoil (see Figure 13(b)), is forward propagated towards the same three limiting states as in the data-free approach. This means, that the two-dimensional distribution of p is used to determine the respective Δ_B for each target state (1C, 2C and 3C). This is results in the distributions of Δ_B, shown Figure 15. Due to the fact, that the unperturbed RANS solution data points, which are used as input for the model evaluation, are distributed along the plane strain line, the spatially averaged relative distance Δ_B is highest for the the simulations targeting the isotropic corner (3C), followed by the two-component corner (2C) and the one-component corner (1C). In order to reach an acceptable steady solution for each perturbed simulation the moderation factor f is adjusted in the same manner as discussed above for the data-free procedure. Since the overall perturbation is weaker than using $\Delta_B = 1$, the moderation factor could be increased. Table 5 summarizes the actual applied values for the data-driven cases as well. Please note, that due to the fact, that $\Delta_B < 1$, there has to be a distinction between $P_{k_{\text{min}}}$ and $P_{k_{\text{max}}}$ for 3C.

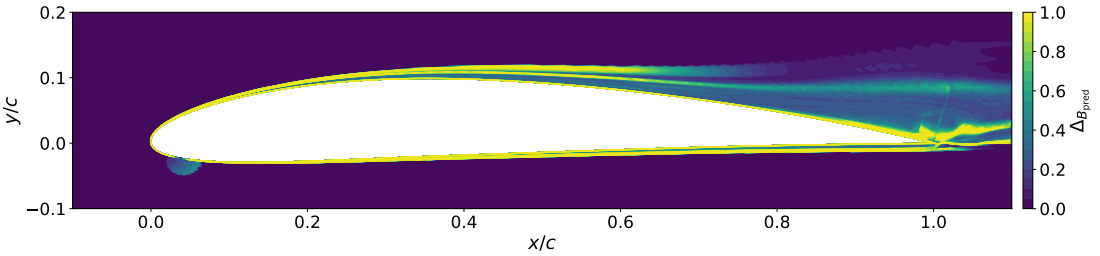
The UQ estimated bands for the surface quantities, shown in Figure 16, become narrower. As the overproduction of turbulent kinetic energy for the data-free 1C and 2C cases disappears, all data-driven perturbed simulations feature a separation zone on the suction surface. Moreover, the uncertainty estimates based on (1C, $P_{k_{\text{max}}}$) and (2C, $P_{k_{\text{max}}}$) are very close to the underlying baseline solution, especially for the pressure coefficient in Figure 16(a).Thus, the actual width of the data-driven UQ envelope is shaped by (3C, $P_{k_{\text{max}}}$) and the simulations featuring minimal turbulent production on the opposite side.



(a) Data-driven relative perturbation magnitude evaluated for 1C target state



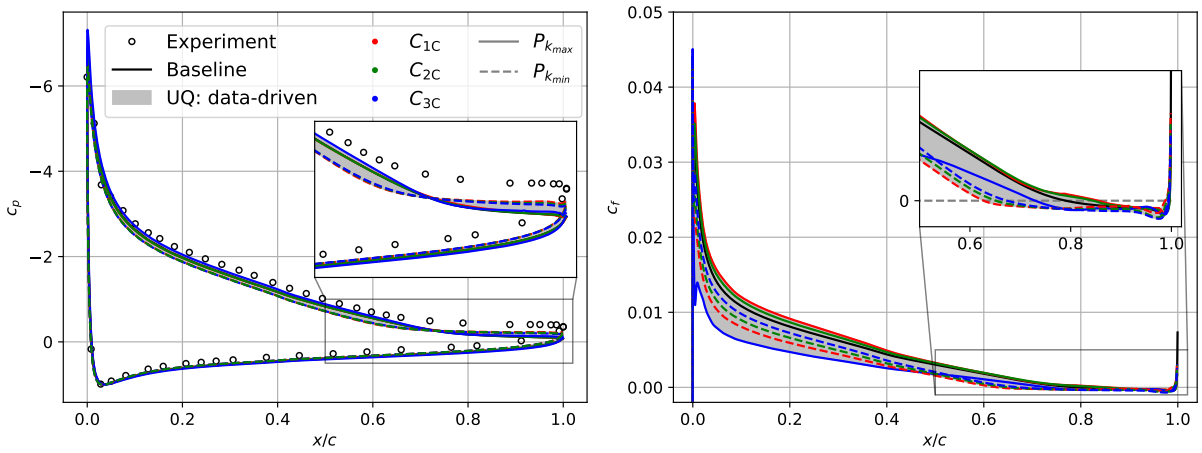
(b) Data-driven relative perturbation magnitude evaluated for 2C target state



(c) Data-driven relative perturbation magnitude evaluated for 3C target state

Fig. 15 Comparison of the effect of identical model predicted perturbation magnitude p on the relative perturbation magnitude Δ_B

Last but not least, the fact, that none of the presented UQ estimates envelopes the experimental surface pressure measurements, needs to be discussed. To start with, as discussed already in the beginning of this section, the CFD setup seems to come along with certain weaknesses. Additionally, as also mentioned on NASA's turbulence modelling resource site [26], there is some inconsistency regarding the determination of the reference velocity in the experiment. A proposed change of multiplying the reference velocity with 0.93, would increase the accuracy with the experiments on the suction side, while the CFD prediction on the pressure side would move away from the experiments. Moreover, the underlying intention of applying the UQ perturbation framework is not to include certain high-fidelity data, whether it originates from experiments or scale-resolving simulation, into its envelope. The methodology seeks to produce limiting states of the Reynolds stress tensor, propagates these states through the CFD solution and results in modified QoI. Consequently, the obtained bands on certain QoI, such as pressure coefficient, can be attributed to the uncertainty of the turbulence model itself. Anyway, the uncertainty envelopes do not represent confidence or strict intervals at all [30]. This statement holds especially for the data-driven approach. The described machine learning procedure only accounts for the spatial varying deviation in Reynolds stress anisotropy, since the turbulence model's uncertainties are not uniform across the computational domain. The impact of the discrepancy in terms of anisotropy between RANS and scale-resolving data on certain QoI, was not part of the machine learning process at all. Thus, it cannot be expected, that the perturbed turbulence model simulations account for accurate envelopes for selected QoI.



(a) Pressure coefficient c_p

(b) Friction coefficient c_f

Fig. 16 Surface quantities for flow around NACA 4412 including data-driven evaluation of uncertainty estimates for Menter SST $k-\omega$ turbulence model; legend in (a) applies also for (b)

VI. Conclusion and Outlook

The present work aims to consolidate an arisen method in the field of turbulence model uncertainty quantification with machine learning practices. We demonstrate the possibility to estimate uncertainty bounds for turbulence models with state-of-the-art methods in DLR's CFD solver suite TRACE.

The UQ perturbation framework is described in detail, presenting its underlying idea while mentioning its limitations and what it is not able to do for industrial applications. Additionally, since TRACE is eminent in turbomachinery industry, designing and implementing a framework to easily conduct uncertainty estimation for turbulence models was a major goal of this work. Moreover, coupling the data-driven extension with the much-cited UQ methodology including Reynolds stress tensor's eigenvalue and eigenvector perturbations and applying it for flow physics featuring separated flows, adverse pressure gradient and reattachment, was the plausible next step in our research. In our investigations, we outline tools and methodologies for assessing and analyzing data-driven models, especially in the context of turbulent flows. We address key points in the field of machine learning such as selection of input features, tuning of hyperparameters, judging the model's accuracy and building trust in a trained model, when there is no accurate data available anymore. In order to predict the desired target quantity for the selected flow cases by the random forest, we admit, that me might not have to use this abundant number of input features. This is due to the fact, that the considered cases show certain similarities in terms of input and target quantities. However, if the amount of training data sets increases even more, covering a wider range of flow phenomena, the machine learning model will likely take advantage of a larger number of input features.

We confirm, that the perturbation approach to account for turbulence model uncertainties, is a purely physics-based, comprehensible framework. Nevertheless, it suffers from reduced convergence or even divergence of steady state solutions. The necessity to moderate certain perturbed simulations by an arbitrary factor, seems unsatisfactory for such an advanced approach. But we do not see any other remedy for this as well, as even the machine learning does not help to overcome this particular issue. Moreover, we also agree on the underlying idea to account for spatially varying of turbulence model uncertainties by using data-driven methodologies to determine certain areas of high deviations from accurate Reynolds stress anisotropy states. Training a machine learning model to predict this deviation and using its model predictions for unseen flow configurations, will always help CFD users to get an indication, in which regions the LEVM assumptions might be violated. However, we believe, based on our experiences with the data-driven UQ framework (which are not only limited to the NACA 4412 flow case), that the UQ estimates would only help designing next turbomachinery generations, if the stability issues in terms of convergence for steady state simulation were solved. As the underlying work of this paper guided us to set up a database for machine learning targeting on turbulence quantities, we would like to utilize it for future investigations. Currently, we are striving for a data-driven a posteriori correction of RANS solutions based on corrected Reynolds stresses. This research will be affiliated in the area of augmented turbulence modelling corrections.

Appendix

Mathematical effect of the moderation factor f in case of pure eigenvalue perturbation

By applying the moderation factor $f \in [0, 1]$, introduced in II.D.1, the perturbed Reynolds stress tensor can be expressed as

$$\begin{aligned} \tau^*_{ij_f} &= \tau_{ij} + f\left[k\left(a^* + \frac{2}{3}\delta_{ij}\right) - \tau_{ij}\right] \\ &= (1-f)\,\tau_{ij} + f\tau^*_{ij}\,, \end{aligned} \tag{18}$$

where τ_{ij} is the Reynolds stress tensor, which was calculated based on Boussinesq assumption in step 1) (see II.D.1). Based on the perturbed anisotropy tensor, the reconstructed Reynolds stress tensor is indicated by τ^*_{ij}. The anisotropy tensor related to this perturbed Reynolds stress tensor can be written as

$$\begin{aligned} a^*_{ij_f} &= \frac{\tau^*_{ij_f}}{k} - \frac{2}{3}\delta_{ij} \\ &= \frac{(1-f)\,\tau_{ij} + f\tau^*_{ij}}{k} - \frac{2}{3}\delta_{ij} \\ &= \frac{(1-f)\left[k\left(a_{ij} + \frac{2}{3}\delta_{ij}\right)\right] + f\left[k\left(a^*_{ij} + \frac{2}{3}\delta_{ij}\right)\right]}{k} - \frac{2}{3}\delta_{ij} \\ &= (1-f)\left(a_{ij} + \frac{2}{3}\delta_{ij}\right) + f\left(a^*_{ij} + \frac{2}{3}\delta_{ij}\right) - \frac{2}{3}\delta_{ij} \\ &= (1-f)\,a_{ij} + f a^*_{ij}\,. \end{aligned} \tag{19}$$

When just applying eigenvalue perturbation of the anisotropy tensor, a_{ij} and a^*_{ij} share identical eigenvectors. Thus, the eigenvalues of $a^*_{ij_f}$ are

$$\lambda^*_{i_f} = (1-f)\,\lambda_i + f\lambda^*_i\,. \tag{20}$$

The barycentric weights C_{iC}, which are used to calculate the barycentric coordinates in Equation (4), can be expressed in terms of moderation factor:

$$\begin{aligned} C^*_{1C_f} &= \frac{1}{2}\left[\lambda^*_{1_f} - \lambda^*_{2_f}\right] \\ &= \frac{1}{2}\left[(1-f)\,\lambda_1 + f\lambda^*_1 - (1-f)\,\lambda_2 + f\lambda^*_2\right] \\ &= (1-f)\,\frac{1}{2}\,(\lambda_1 - \lambda_2) + f\frac{1}{2}\left(\lambda^*_1 - \lambda^*_2\right) \\ &= (1-f)\,C_{1C} + f C^*_{1C} \end{aligned} \tag{21}$$

$$\begin{aligned} C^*_{2C_f} &= \lambda^*_{2_f} - \lambda^*_{3_f} \\ &= (1-f)\,\lambda_2 + f\lambda^*_2 - (1-f)\,\lambda_3 + f\lambda^*_3 \\ &= (1-f)\,(\lambda_2 - \lambda_3) + f\left(\lambda^*_2 - \lambda^*_3\right) \\ &= (1-f)\,C_{2C} + f C^*_{2C} \end{aligned} \tag{22}$$

$$\begin{aligned} C^*_{3C_f} &= \frac{3}{2}\left[\lambda^*_{3_f} + 1\right] \\ &= \frac{3}{2}\left[(1-f)\,\lambda_3 + f\lambda^*_3 + (1-f) + f\right] \\ &= (1-f)\,\frac{3}{2}\,(\lambda_3 + 1) + f\frac{3}{2}\left(\lambda^*_3 + 1\right) \\ &= (1-f)\,C_{3C} + f C^*_{3C} \end{aligned} \tag{23}$$

The perturbed barycentric coordinates $\mathbf{x}_f^*$ (modified by the moderation factor f) can be written using Equations 21, 22 and 23 as

$$\begin{aligned}\mathbf{x}_f^* &= \mathbf{x}_{1\mathrm{C}} C_{1C_f}^* + \mathbf{x}_{2C_f}^* + \mathbf{x}_{3C} C_{3C_f}^* \\ &= \mathbf{x}_{1\mathrm{c}} \left[(1-f)\, C_{1\mathrm{C}} + f C_{1\mathrm{C}}^*\right] + \mathbf{x}_{2\mathrm{c}} \left[(1-f)\, C_{2\mathrm{C}} + f C_{2\mathrm{C}}^*\right] + \mathbf{x}_{3c} \left[(1-f)\, C_{3\mathrm{C}} + f C_{3\mathrm{C}}^*\right] \\ &= (1-f)\left(\mathbf{x}_{1c} C_{1\mathrm{C}} + \mathbf{x}_{2c} C_{2\mathrm{C}} + \mathbf{x}_{3c} C_{3\mathrm{C}}\right) + f\left(\mathbf{x}_{1c} C_{1\mathrm{C}}^* + \mathbf{x}_{2c} C_{2\mathrm{C}}^* + \mathbf{x}_{3c} C_{3\mathrm{C}}^*\right) \\ &= (1-f)\,\mathbf{x} + f\mathbf{x}^* \,.\end{aligned} \tag{24}$$

Remembering Equation (5) and rearranging leads to

$$\mathbf{x}^* = (1-\Delta_B)\,\mathbf{x} + \Delta_B \mathbf{x}_{(t)} \,. \tag{25}$$

The analogy of Equation (24) and (25) reveals the similar effect of adjusting Δ_B or f in case of only perturbing the eigenvalues of the anisotropy tensor. Thus, one can rewrite the actual intended location inside the barycentric triangle as a relative distance towards the corners

$$\mathbf{x}_f^* = (1-\Delta_B f)\,\mathbf{x} + \Delta_B f \mathbf{x}_{(t)} \,. \tag{26}$$

Acknowledgments

The project on which this paper is based was funded by the German Federal Ministry for Economic Affairs and Climate Action under the funding code 03EE5041A. The authors are responsible for the content of this publication. Additionally, the authors thank the financial funding of the prescribed work by the internal DLR project SuperCOOL. Furthermore, we thank Dr. Rossi for providing the DNS data of the wavy wall test case for machine learning purposes [22]. Last but not least, the authors highly appreciate all the discussions with Dr. Mishra about the perturbation framework.

References

[1] Duraisamy, K., Iaccarino, G., and Xiao, H., "Turbulence Modeling in the Age of Data," *Annual Review of Fluid Mechanics*, Vol. 51, No. 1, 2019, p. 357377. https://doi.org/10.1146/annurev-fluid-010518-040547.

[2] Xiao, H., and Cinnella, P., "Quantification of model uncertainty in RANS simulations: A review," *Progress in Aerospace Sciences*, Vol. 108, 2019, pp. 1–31. https://doi.org/https://doi.org/10.1016/j.paerosci.2018.10.001, URL https://www.sciencedirect.com/science/article/pii/S0376042118300952.

[3] Emory, M., Larsson, J., and Iaccarino, G., "Modeling of structural uncertainties in Reynolds-averaged Navier-Stokes closures," *Physics of Fluids*, Vol. 25, No. 11, 2013, p. 110822. https://doi.org/10.1063/1.4824659.

[4] Iaccarino, G., Mishra, A., and Ghili, S., "Eigenspace perturbations for uncertainty estimation of single-point turbulence closures," *Physical Review Fluids*, Vol. 2, 2017. https://doi.org/10.1103/PhysRevFluids.2.024605.

[5] Heyse, J., Mishra, A., and Iaccarino, G., "Estimating RANS model uncertainty using machine learning," *Journal of the Global Power and Propulsion Society*, 2021, pp. 1–14. https://doi.org/10.33737/jgpps/134643.

[6] Matha, M., and Morsbach, C., "Extending turbulence model uncertainty quantification using machine learning," *NeurIPS - Thirty-fifth Conference on Neural Information Processing Systems | Workshop on Machine Learning and the Physical Sciences*, 2021.

[7] Morsbach, C., "Reynolds Stress Modelling for Turbomachinery Flow Applications," Ph.D. thesis, Teschnische Universität Darmstadt, Darmstadt, Germany, 2017.

[8] Banerjee, S., Krahl, R., Durst, F., and Zenger, C., "Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches," *Journal of Turbulence*, Vol. 8, 2007, p. N32. https://doi.org/10.1080/14685240701506896.

[9] Breiman, L., "Random Forests," *Machine Learning*, Vol. 45, 2001, pp. 5–32. https://doi.org/10.1023/A:1010950718922.

[10] Geman, S., Bienenstock, E., and Doursat, R., "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, Vol. 4, 1992, pp. 1–58. https://doi.org/10.1162/neco.1992.4.1.1.

[11] Louppe, G., "Understanding Random Forests: From Theory to Practice," Ph.D. thesis, 10 2014. https://doi.org/10.13140/2.1.1570.5928.

[12] Breiman, L., "Bagging predictors," *Machine Learning*, Vol. 24, 2004, pp. 123–140. https://doi.org/10.1007/BF00058655.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[14] Wang, J.-X., Wu, J.-L., Xiao, H., and Paterson, E., "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, Vol. 3, No. 7, 2018. https://doi.org/10.1103/physrevfluids.3.074602.

[15] Ling, J., Jones, R. E., and Templeton, J., "Machine learning strategies for systems with invariance properties," *J. Comput. Phys.*, Vol. 318, 2016, pp. 22–35.

[16] Spencer, A., and Rivlin, R., "Isotropic integrity bases for vectors and second-order tensors," *Arch. Rational Mech. Anal.*, Vol. 9, 1962, p. 45–63.

[17] Mishra, A., Mukhopadhaya, J., Iaccarino, G., and Alonso, J., "An uncertainty estimation module for turbulence model predictions in SU2," *AIAA Journal*, Vol. 57, 2018. https://doi.org/10.2514/1.J057187.

[18] DLR, German Aerospace Center, Institute of Propulsion Technology, "TRACE User Guide," http://www.trace-portal.de/userguide/trace/index.html, accessed April 2022.

[19] Lumley, J. L., "Computational Modeling of Turbulent Flows," Elsevier, 1979, pp. 123–176. https://doi.org/https://doi.org/10.1016/S0065-2156(08)70266-7.

[20] Lee, M., and Moser, R. D., "Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$," *Journal of Fluid Mechanics*, Vol. 774, 2015, p. 395415. https://doi.org/10.1017/jfm.2015.268.

[21] Breuer, M., Peller, N., Rapp, C., and Manhart, M., "Flow over periodic hills – Numerical and experimental study in a wide range of Reynolds numbers," *Computers Fluids*, Vol. 38, No. 2, 2009, pp. 433–457. https://doi.org/https://doi.org/10.1016/j.compfluid.2008.05.002.

[22] Rossi, R., "Passive scalar transport in turbulent flows over a wavy wall," Ph.D. thesis, Università degli Studi di Bologna, Bologna, Italy, 2006.

[23] Laval, J., and Marquillie, M., "Direct Numerical Simulations of Converging–Diverging Channel Flow," 2011, pp. 203–209. https://doi.org/10.1007/978-90-481-9603-6_21.

[24] Menter, F., Kuntz, M., and Langtry, R., "Ten years of industrial experience with the SST turbulence model," *Heat and Mass Transfer*, Vol. 4, 2003.

[25] Coles, D., and Wadcock, A. J., "Flying-Hot-wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift," *AIAA Journal*, Vol. 17, No. 4, 1979, pp. 321–329. https://doi.org/10.2514/3.61127.

[26] NASA, National Aeronautics and Space Administration, "Turbulence modeling resource database," https://turbmodels.larc.nasa.gov/index.html, accessed April 2022.

[27] Scott, D., "Multivariate density estimation: Theory, practice, and visualization: Second edition," *Multivariate Density Estimation: Theory, Practice, and Visualization: Second Edition*, 2015, pp. 1–360. https://doi.org/10.1002/9781118575574.

[28] Wu, J.-L., Wang, J.-X., Xiao, H., and Ling, J., "A Priori Assessment of Prediction Confidence for Data-Driven Turbulence Modeling," *Flow, Turbulence and Combustion*, Vol. 99, No. 1, 2017, pp. 25–46. https://doi.org/10.1007/s10494-017-9807-0.

[29] Cook, L., Mishra, A., Jarrett, J., Willcox, K., and Iaccarino, G., "Optimization under turbulence model uncertainty for aerospace design," *Physics of Fluids*, Vol. 31, 2019, p. 105111. https://doi.org/10.1063/1.5118785.

[30] Mishra, A. A., Mukhopadhaya, J., Alonso, J., and Iaccarino, G., "Design exploration and optimization under uncertainty," *Physics of Fluids*, Vol. 32, No. 8, 2020, p. 085106. https://doi.org/10.1063/5.0020858, URL https://doi.org/10.1063/5.0020858.