ORIGINAL PAPER

Anna Ivagnes · Niccolò Tonicello · Paola Cinnella ·
Gianluigi Rozza

Enhancing non-intrusive reduced-order models with space-dependent aggregation methods

Abstract In this manuscript, we combine non-intrusive reduced-order models (ROMs) with space-dependent aggregation techniques to build a *mixed-ROM*, able to accurately capture the flow dynamics in different physical settings. The flow prediction obtained using the *mixed* formulation is derived from a convex combination of the predictions of several previously trained reduced-order models (ROMs), with each model assigned a space-dependent weight. The ROMs incorporated in the *mixed* model utilize different *reduction* methods, such as proper orthogonal decomposition and autoencoders, and various *approximation* techniques, including radial basis function interpolation (RBF), Gaussian process regression, and feed-forward artificial neural networks. Each model's contribution is given higher weights in regions where it performs best and lower weights where its accuracy is lower compared to the other models. Additionally, a random forest regression technique is used to determine the weights for previously unseen conditions. The performance of the aggregated model is assessed through two test cases: the 2D flow past a NACA 4412 airfoil at a 5-degree angle of attack, with the Reynolds number ranging between 1×10^5 and 1×10^6, and a transonic flow over a NACA 0012 airfoil, with the angle of attack as the varying parameter. In both scenarios, the mixed-ROM demonstrated improved accuracy compared to each individual ROM technique, while providing an estimate for the predictive uncertainty.

Abbreviations

AE-ANN	ROM using AE as reduction and ANN as approximation
AE-GPR	ROM using AE as reduction and GPR as approximation
AE-RBF	ROM using AE as reduction and RBF as approximation
AE	AutoEncoder
ANN	Artificial neural network
CFD	Computational fluid dynamics
DNS	Direct numerical simulation
FOM	Full order model

A. Ivagnes · N. Tonicello (✉) · G. Rozza
Mathematics Area, International School of Advanced Studies, SISSA, Via Bonomea, 265, 34136 Trieste, Italy
E-mail: niccolo.tonicello@sissa.it

A. Ivagnes
E-mail: anna.ivagnes@sissa.it

G. Rozza
E-mail: gianluigi.rozza@sissa.it

P. Cinnella
Institut Jean Le Rond D'Alembert, Sorbonne Université, Boîte 162, Tour 55-65, 75252 Paris, France
E-mail: paola.cinnella@sorbonne-universite.fr

GPR	Gaussian process regression
LES	Large eddy simulation
Mixed-ROM	Aggregation of different ROMs
PCA	Principal component analysis
POD-ANN	ROM using POD as reduction and ANN as approximation
POD-GPR	ROM using POD as reduction and GPR as approximation
POD-RBF	ROM using POD as reduction and RBF as approximation
PODAE-ANN	ROM using PODAE as reduction and ANN as approximation
PODAE-GPR	ROM using PODAE as reduction and GPR as approximation
PODAE-RBF	ROM using PODAE as reduction and RBF as approximation
PODAE	Mixed reduction with proper orthogonal decomposition and AutoEncoder
POD	Proper orthogonal decomposition
RANS	Reynolds–averaged Navier–Stokes
RBF	Radial basis function interpolation
RF	Random forest regressor
ROM	Reduced order model
XMA	Space-dependent aggregation model

List of symbols

$(\cdot)_{\text{eval}}$	Variable in the evaluation database
$(\cdot)_{\text{test}}$	Variable in the test database
$(\cdot)_{\text{train}}$	Variable in the train database
α	Angle of attack
$\boldsymbol{\eta}$	Set of independent features on which the aggregation prediction depends
$\boldsymbol{\mu}$	Parameters considered in the specific test case
∇	Gradient operator
$\nabla\cdot$	Divergence operator
$\nabla\times$	Curl operator
$\boldsymbol{\phi}_i$	$i-$th POD spatial mode
$\boldsymbol{a}(\boldsymbol{\mu})$	Parametrized reduced representation of a generic field
$\boldsymbol{E}_{ij}$	$(ij)_{th}$ component of the averaged strain rate tensor
$\boldsymbol{s}(\boldsymbol{\mu})$	Generic parametric field
$\boldsymbol{u}$	Velocity field
$\boldsymbol{x}$	Spatial coordinates
$\triangle$	Laplacian operator
$\hat{\boldsymbol{s}}$	Random variable notation (used for uncertainty analysis)
$\|\cdot\|_{l^2}$	norm in vectorial space l^2
$\mathcal{A}$	Approximation mapping in ROMs
$\mathcal{D}$	Decoder part in the AutoEncoder
$\mathcal{E}$	Encoder part in the AutoEncoder
$\mathcal{I}$	Domain of research of hyperparameter σ_p
$\mathcal{M}$	Set of non-intrusive ROMs considered in aggregation
$\mathcal{R}$	Reduction mapping in ROMs
$\mathcal{R}_{\text{AE}}$	Reduction mapping produced with the AE in PODAE strategy
$\mathcal{R}_{\text{POD}}$	Reduction mapping produced with the POD in PODAE strategy
$\mathcal{R}_{ij}$	$(ij)_{th}$ component of the Reynolds stress tensor
K	Covariance function in the normal distribution of GPR
m	Mean function in the normal distribution of GPR
ω_i	Weights associated with the $i-$th individual ROM in the aggregation model
σ_p	Hyperparameter of aggregation model
$\tilde{\boldsymbol{s}}^{(\text{mix})}$	Prediction of the aggregation model
$\tilde{\boldsymbol{s}}^{(i)}$	Prediction of the $i-$th individual ROM (used in aggregation)
$E[\hat{\boldsymbol{s}}]$	Expected value or variable $\hat{\boldsymbol{s}}$
g_i	Numerator of weights associated with the $i-$th individual ROM in the aggregation model
n_M	Number of ROM models considered in the aggregation and composing set $\mathcal{M}$

Re	Reynolds number
$Var[\hat{s}]$	Variance associated to variable $\hat{s}$
κ	Turbulent kinetic energy
ν_t	Eddy viscosity
ν	Kinematic viscosity
Ω	Bounded domain
$\partial\Omega$	Boundaries of domain Ω
C_f	Skin friction coefficient coefficient
C_p	Pressure coefficient coefficient
P	Number of parameters (dimension of vector $\boldsymbol{\mu}$)
p	Pressure field
r_{med}	Medium dimension (or reduced POD dimension) in mixed reduction PODAE
n_{dof}	Number of spatial degrees of freedom in high-fidelity simulations
N_{eval}	Number of snapshots used to build the aggregation model (*evaluation* database)
N_{train}	Number of snapshots used to train the ROMs (*train* database)
N_{test}	Number of snapshots used to test the aggregation model (*test* database)
r	Reduced (or latent) dimension used in non-intrusive ROMs

1 Introduction

Thanks to the constantly increasing computational resources provided by modern hardware architectures, computational fluid dynamics (CFD) has become a fundamental tool for the design processes of the aeronautical industry [43].

In the last few years, great improvements have been made in using scale-resolving simulations such as large-eddy simulations (LES) for real-world aeronautical applications of CFD [13,14]. However, for full-scale problems such as full wings or fuselage of operating airplanes, their computational cost is still prohibitive in terms of CPU time even with the intensive use of powerful supercomputers. Consequently, with the additional cost of a reduced accuracy in the smallest details of the flow field, the use of Reynolds-averaged Navier–Stokes (RANS) equations still represents the most common technique to deal with turbulent flows in the aerospace industry. Even if RANS are much less expensive from a computational point of view with respect to LES, they may still require a high computational effort for design processes, which require repeated queries of the CFD model for computing the performance of new geometries and configurations. This is even more true if such computations are performed within an optimization algorithm to find optimal designs. In such case, the use of reduced-order modeling (ROM) can significantly improve the computational efficiency of the overall algorithm with a small loss in terms of accuracy. In other words, LES, RANS and ROMs for RANS are in this way ordered for decreasing computational cost. Depending on the different tasks required by the industrial needs, one or the other can be used in order to simulate the aerodynamics of aeronautical vehicles.

The general paradigm of model order reduction is the *offline-online* procedure. The offline stage is typically expensive and time-consuming. It is usually performed on supercomputers and it allows to collect the flow solutions, named *snapshots*, from a large number of high-fidelity simulations, named full-order model (FOM). The online stage is instead considerably faster than the first one and it consists in performing a linear or nonlinear projection of the solutions' manifold into a space of reduced dimensionality.

We focus here on the use of ROMs for RANS, which are particularly suitable for applications where many queries in the parametric space are required, or for scenarios where real-time response is needed. Whereas using ROMs on LES or DNS data is still quite limited due to the large amount of degrees of freedom involved and the strongly multi-scale character of the flow, predicting unseen configurations via ROMs for RANS is much more doable and common in the reduced-order modeling community [17,18,56–58]. More in detail, ROMs can be divided into two distinct classes: *non-intrusive ROMs*, which are only built upon data, and *intrusive* ROMs, which consist in the resolution of a reduced and simplified version of the physical model of interest.

In this project, we restrict our analysis to *non-intrusive* (i.e., *purely data-driven*) ROMs. This approach typically employs a *reduction* step to reduce the snapshots, and an *approximation* technique, to predict the reduced solution for an unseen configuration. These methods have been applied in the past decade in a large variety of different CFD applications showing promising results in approximating the flow dynamics only starting from data. Some examples of applications of ROMs for RANS are: industrial problems with a design optimization goal [7,22,47,48], incompressible [17,18] and compressible turbulent flows [37,53,54,56–58].

Thus, we focus hereafter on the use of ROMs for RANS over two different airfoil configurations, namely, a NACA4412 in subsonic conditions and a transonic NACA0012 test case. Both cases represent well-documented benchmarks often used in the literature to validate RANS models [12,19,33,49]. Specifically, we consider here a set of alternative non-intrusive reduced-order models corresponding to different reduction methodologies and approximations in the latent space. It will be shown that such models perform differently in approximating the reference RANS solution in different regions of the computational domain, and it is often difficult to select *a-priori* a single best-performing ROM for all configurations. Such uncertainty in the ROM choice may be critical in view of their use for further design processes, and should be quantified and, if possible, reduced.

For that purpose, we propose here a novel approach to further improve reduced-order modeling accuracy while quantifying uncertainty, by constructing a spatially dependent convex linear combination of different ROMs depending on their agreement with the reference full-order model. In many problems of interest, it might be difficult to know a-priori which combination of reduction and approximation techniques might be the best in the whole spatial and parametric domain. The typical approach involves testing different models, different architectures, and tuning different hyperparameters until a sufficiently satisfying model is obtained. Such a task is however rather arbitrary and time consuming. The methodology proposed in the following is not only able to identify automatically the optimal model combination, but it can also further improve the overall accuracy by using the information obtained from all the alternative models. For example, nonlinear reduction techniques might be a reasonable choice for advection-dominated problems. However, such techniques might be so finely trained in capturing sharp features such as shock waves and boundary layers that they end up lacking accuracy in smoother regions of the flow where, instead, linear techniques might be more suitable. In practice, complex flows contain various physical processes leading to competing requirements for the ROM to be used.

An efficient framework for combining a set of alternative models based on their merit is the model aggregation framework [8,10,44], which aims at combining multiple predictions stemming from various models—also termed experts or forecasters—to provide a global, enhanced, solution. A solution that provides a spatially constant weighting of competing model predictions being not optimal, is not optimal itself because the accuracy of the models varies according to the local flow physics. Therefore, it becomes interesting to combine the model aggregation approach with so-called mixture-of-local-experts techniques [23], also referred-to as mixture-of-experts [55] or mixture models. In such approaches, the input feature space (covariate space) is softly split into partitions where the locally best-performing models are assigned higher weights. The soft partitioning is accomplished through parametric gate functions, or a network of hierarchical gate functions [25] that rank the model outputs with probabilities. However, mixture-of-experts tend to promote a single best model in every soft partition, thus accounting for the spatial variation of the best model but neglecting the uncertainty in model choice. A methodology combining the model aggregation and the mixture-of-experts ideas has been recently introduced by De Zordo Banliat et al. [59], who applied it to optimally combine a set of competing RANS models depending on some local flow features. The approach, named space-dependent model aggregation (XMA), was specifically designed to improve the prediction over individual component models. In this work, we build on the XMA framework, and we propose a space-dependent aggregation model for combining ROMs.

The new methodology is tested on the two above-mentioned airfoil configurations. A variety of different analyses such as varying the dimensions of the latent space, and the influence of the chosen reduction/approximation techniques are presented in order to assess the robustness and suitability of the proposed approach for the prediction of external turbulent aerodynamics problems.

2 Numerical methods

The use of RANS is extremely widespread in the aeronautical industry. Here we simply considered well-established models and set-ups as full-order models. Consequently, we focus this section more on non-intrusive ROMs and the aggregation algorithm to combine them. Non-intrusive ROMs, described in Sect. 2.1, are employed to predict an approximated version of the full-order fields for unseen operating conditions with a considerably reduced computational cost. The predictions of different ROMs are then combined together with a model mixture technique, deeply explained in Sect. 2.2.

2.1 Non-intrusive reduced-order models

This part of the manuscript is dedicated to the presentation of the theory behind the non-intrusive model order reduction technique. This approach is fully data-based, namely it only exploits the information provided by high-fidelity simulations without any a-priori knowledge on the equations or numerical scheme used for the high-fidelity simulations. Therefore, in this framework, the governing equations are only used at the full-order level to perform the offline simulations.

The original full-order solutions, named *snapshots*, correspond to the fields of interest of our problem and each snapshot is related to a specific set of parameters, i.e., a specific simulation setup. For instance, in both test cases we will consider the pressure and wall shear stress fields acting on the airfoil, and the velocity and pressure fields evaluated in the 2D space around the airfoil.

We consider a generic field s and a number of parameters P. We call $s_i = s(\mu_i)$ the i-th snapshot, having as $\mu_i \in \mathbb{R}^P$ the corresponding set of parameters. We can then assemble the following snapshots' matrix:

$$S = \begin{bmatrix} | & | & & | \\ s_1(x) & s_2(x) & \dots & s_{N_{\text{train}}}(x) \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n_{\text{dof}} \times N_{\text{train}}}$$

where n_{dof} is the number of degrees of freedom (equal for each snapshot), and N_{train} is the number of snapshots. In particular, n_{dof} is the number of cells on the airfoil 1D surface (when considering as snapshots the pressure and wall shear stress acting on it), and the number of cells in the 2D external mesh (when considering the pressure and velocity fields around the airfoil).

The question which is addressed by ROMs is the following.

Question 1: "*How can we find the approximation $\tilde{s}(\mu^*) \simeq s(\mu^*)$ where μ^* represents an unseen configuration?*"

The procedure to address the above-mentioned task is composed of two steps: a *reduction* step, performing a compression onto a space with reduced dimensionality, and an *approximation* step, where interpolation or regression techniques are used to predict the reduced representation for unseen parameters. Finally, the reduced representation is backmapped into the full-order space to find the approximated full-dimensional solution.

2.1.1 Reduction techniques

This first step consists in a linear or nonlinear mapping of the original matrix of snapshots onto a matrix with reduced dimensions. In the following part we will refer to the reduction step with the mapping $\mathcal{R}$. So, the problem here addressed is the following.

Question 2: "*How can we find a reduced representation $a(\mu_i)$ of each $s(\mu_i)$?*"

We consider here two different reduction techniques: a linear one, the proper orthogonal decomposition (POD), and a nonlinear one, the autoencoder (AE).

Reduction through Proper Orthogonal Decomposition The POD [1,5,27] consists in the projection of the snapshots' matrix onto a space spanned by a limited number of the so-called *modes*, which can be computed either via the correlation matrix or through a singular value decomposition (SVD) technique [28,51] in the offline stage. The main hypothesis of the POD is that each snapshot can be approximated as a linear combination of the modes:

$$s_i \simeq \sum_{j=1}^{r} a_j(\mu_i)\phi_j,$$

where $\{\phi_j\}_{j=1}^{r}$ are the modes and $r \ll N_{\text{train}}$. The parameter r is established *a-priori* based on energy criteria and/or the singular values decay. The terms $\{a_j(\mu_i)\}_{j=1}^{r}$ are the reduced coefficients associated with the modes. When the SVD factorization is employed, we will have $S = U\Sigma V^T$, where the columns of U are the POD modes and the operation $U^T s_i = a(\mu_i)$ provides the reduced representation.
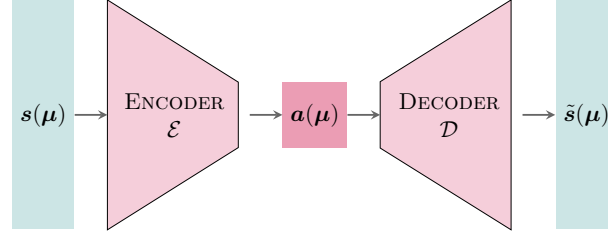
Fig. 1 General structure of an autoencoder

Reduction through Autoencoder Autoencoders are a family of neural networks that have become popular as dimensionality reduction technique [11,15,21,26,29,40] thanks to their peculiar architecture. Indeed, AEs are composed of two networks, as shown in Fig. 1: an *encoder* $\mathcal{E} : \mathbb{R}^{n_{dof}} \to \mathbb{R}^r$, which maps the high-dimensional input into a latent space (the reduced space), and a *decoder* $\mathcal{D} : \mathbb{R}^r \to \mathbb{R}^{n_{dof}}$, which back-maps the latent representation onto the original full dimensionality.

Autoencoders can be composed by either convolutional or dense layers, but in this work we focus on dense feed-forward neural networks for both the encoder and the decoder.

For instance, we consider a single-layer structure for both the decoder and the encoder, we call W and b the weights' matrix and bias for network $\mathcal{E}$, and W$'$ and b$'$ the weights' matrix and bias for network $\mathcal{D}$. Then, considering the activation functions σ and σ' for $\mathcal{E}$ and $\mathcal{D}$, respectively, we obtain the following formulation:

$$\mathcal{E}(s(\boldsymbol{\mu})) = \sigma(\mathrm{W}s(\boldsymbol{\mu}) + \mathrm{b}) = \boldsymbol{a}(\boldsymbol{\mu}),$$
$$\mathcal{D}(\boldsymbol{a}(\boldsymbol{\mu})) = \sigma'(\mathrm{W}'\boldsymbol{a}(\boldsymbol{\mu}) + \mathrm{b}') = \tilde{\boldsymbol{s}}(\boldsymbol{\mu}) \simeq \boldsymbol{s}(\boldsymbol{\mu}).$$

The general structure with more hidden layers can be easily derived from the above formulation.

In the training of the AE, the loss function which is minimized is the following:

$$\frac{1}{\mathrm{N}_{\text{train}}} \sum_{i=1}^{\mathrm{N}_{\text{train}}} \left(\|s(\boldsymbol{\mu}_i) - \tilde{s}(\boldsymbol{\mu}_i)\|_{l^2}^2 \right) = \frac{1}{\mathrm{N}_{\text{train}}} \sum_{i=1}^{\mathrm{N}_{\text{train}}} \left(\|s(\boldsymbol{\mu}_i) - \sigma'(\mathrm{W}'(\sigma(\mathrm{W}s(\boldsymbol{\mu}) + \mathrm{b})) + \mathrm{b}')\|_{l^2}^2 \right), \quad (1)$$

where $\tilde{s}(\cdot)$ is the approximated high-fidelity field and $s(\cdot)$ is the ground truth.

Reduction through PODAE The mixed approach PODAE consists in a two-step reduction, where the total mapping $\mathcal{R} : \mathbb{R}^{n_{dof}} \to \mathbb{R}^r$ is expressed as $\mathcal{R} = \mathcal{R}_{\text{AE}} \circ \mathcal{R}_{\text{POD}}$.

The first reduction $\mathcal{R}_{\text{POD}} : \mathbb{R}^{n_{dof}} \to \mathbb{R}^{r_{med}}$ is, in general, the most expensive from a computational point of view. In our case, it coincides with the POD retaining all the modes, namely $r_{med} \equiv N_{train}$. On the other hand, the second reduction $\mathcal{R}_{\text{AE}} : \mathbb{R}^{r_{med}} \to \mathbb{R}^r$ is operated by an autoencoder.

The reason for employing such a reduction technique is the considerable gain in the computational time with respect to a purely nonlinear AE. The computational effort needed to train an autoencoder is, in fact, much higher than in the case of a mixed nonlinear approach. However, since the PODAE includes a nonlinear mapping, it allows to capture complex behaviors that the POD alone would not describe properly. In Sect. 3, we will employ this technique when n_{dof} is of the order of magnitude of 10^6.

It is important to highlight that in all the numerical tests we considered the same latent dimension for both the POD, the AE and the PODAE techniques.

2.1.2 Approximation techniques

Once we have all the reduced representations of the snapshots, we can address the final task.

Question 3: "*How can we find the reduced representation $\boldsymbol{a}(\boldsymbol{\mu}^*)$, starting from the knowledge of $\{\boldsymbol{a}(\boldsymbol{\mu}_i)\}_{i=1}^{N_{train}}$?*" which can be also written as: "*Which is the mapping $\mathcal{A} : \mathbb{R}^p \to \mathbb{R}^r$ such that $\boldsymbol{a}(\boldsymbol{\mu}_i) = \mathcal{A}(\boldsymbol{\mu}_i)$?*"

We here consider different types of mapping $\mathcal{A}$: a radial basis function (RBF) interpolation technique, a Gaussian process regression (GPR) technique, and a feed-forward artificial neural network (ANN). We provide in the following part a summary of the above-mentioned techniques, without going through all the details.

Radial Basis Function Interpolation   Following the RBF [36] technique, the unknown coefficients are computed as follows:

$$\mathcal{A}(\boldsymbol{\mu^*}) = \boldsymbol{a}(\boldsymbol{\mu^*}) = \sum_{i=1}^{N_{\text{train}}} \omega_i \varphi(\|\boldsymbol{\mu^*} - \boldsymbol{\mu_i}\|), \tag{2}$$

where $\varphi(\|\boldsymbol{\mu^*} - \boldsymbol{\mu_i}\|)$ are the radial basis functions having center $\boldsymbol{\mu_i}$ and weight ω_i. The weights are found considering known conditions, namely Eq. (2) with $\boldsymbol{\mu^*} = \boldsymbol{\mu_j}$, $j = 1, \dots, N_{\text{train}}$. More details about the kernel are provided in the supplementary section.

Gaussian Process Regression   The GPR [52] is a supervised learning technique that exploits a stochastic model to build the regression function $\mathcal{A}$. In particular, the GPR model assumes that the regression function that relates the inputs $\boldsymbol{\mu}$ to the outputs $\boldsymbol{a}$ is drawn from a Gaussian process with mean function m and covariance function K:

$$\mathcal{A}(\boldsymbol{\mu}) \sim \mathcal{N}(\text{m}(\boldsymbol{\mu}), \text{K}(\boldsymbol{\mu})), \tag{3}$$

where $K_{ij}(\boldsymbol{\mu}) = \mathcal{K}(\boldsymbol{\mu_i}, \boldsymbol{\mu_j})$. The shape of the covariance matrix, also called *kernel*, is chosen depending on the particular problem of interest. The details of the kernel chosen for this work are provided in the supplementary results' section.

Artificial Neural Network   The last approximation technique considered in this work is a regression through a feed-forward neural network, which have been subject of a lot of research works in recent years in several fields, and in particular in reduced-order modeling [2,16,21,41,42].

The architecture that we consider is a fully connected neural network, defined as the concatenation of an input layer, multiple hidden layers, and a final output layer. If we isolate just an hidden layer, we can express the i-th output of the h-th layer as follows:

$$o_i^h = \sigma \left(\sum_{k=1}^{n_{h-1}} W_{ik}^h o_k^{h-1} + b_i^h \right), \tag{4}$$

where σ is the activation function representing the nonlinearity of the model, W^h and b^h are the weights' matrix and the bias of the h-th layer. All weights and bias are tuned during the training procedure in order to fit the input data, and the function can be here expressed as: $\mathcal{A}_\theta(\boldsymbol{\mu})$, where θ is the set of all the weights of the ANN. The architecture of the ANN is reported in the supplementary results' part.

2.2 Space-dependent aggregation methods and ROM mixture

The present subsection of the article is dedicated to the explanation of the logic of the space-dependent model aggregation (XMA) [6,59]. The main principle is to mix models into an "hypermodel" (the model mixture) with improved predictive capabilities with respect to the mixture components themselves.

We call the set of all the models taken into account for the mixture $\mathcal{M} = \{M_1, M_2, \dots, M_{n_M}\}$, where n_M is the number of models we are considering.

In our case, each M_i correspond to a non-intrusive reduced-order model, characterized by specific reduction technique $\mathcal{R}$ and approximation technique $\mathcal{A}$, i.e., $M_i = M(\mathcal{R}, \mathcal{A})$.

Moreover, we can write the prediction of each model M_i as follows:

$$\tilde{s}^{(i)}(\boldsymbol{\eta}) = \tilde{s}(\boldsymbol{\eta}, M_i), i = 1, \dots, n_M,$$

where $\boldsymbol{\eta}$ is the set of independent features on which the prediction depends, in our case the space coordinates and the ROM parameters: $\boldsymbol{\eta} = [\boldsymbol{x}, \boldsymbol{\mu}]$.

The model mixture is based on the assumption that the aggregated prediction can be written as a convex combination of the given models, namely:

$$\tilde{s}^{(\text{mix})}(\boldsymbol{\eta}) = \sum_{i=1}^{n_M} \omega_i(\boldsymbol{\eta})\tilde{s}^{(i)}(\boldsymbol{\eta}), \tag{5}$$

where $\{\omega_i(\boldsymbol{\eta})\}_{i=1}^{n_M}$ are the weights' distribution associated with each model.

In particular, the weights have to satisfy the following conditions:

$$\omega_i(\boldsymbol{\eta}) \in [0, 1], \quad \sum_{i=1}^{n_M} \omega_i(\boldsymbol{\eta}) = 1 \quad \forall \boldsymbol{\eta} \,.$$

At this point, it is important to specify that we split the original dataset of parameters and snapshots $\{(\boldsymbol{\mu}_i, \boldsymbol{s}_i)\}_{i=1}^{N_{\text{tot}}}$ into the following three parts:

- a *training* set, used to train all the reduced-order models in $\mathcal{M}$, composed of N_{train} components;
- an *evaluation* set, used to compute the optimal weights $\{\omega_i(\boldsymbol{\eta})\}_{i=1}^{n_M}$, composed of N_{eval} components;
- a *test* set, composed of N_{test} components and used to compare the results among the single models and the mixture. In this set, the weights are computed with a *regression* technique.

For the sake of clarity, in the following part, we will indicate the parameters and snapshots in the evaluation set as $\boldsymbol{\eta}_{\text{eval}}$ and s_{eval}, respectively, omitting the dependency on the parameters in the snapshots' notation. The same is done for elements in the test set, where we indicate parameters and snapshots as $\boldsymbol{\eta}_{\text{test}}$ and s_{test}.

The key question that we have to address now is:

Question 4: "*How can we compute the weights in the evaluation set in an optimal way?*"

We consider the following expression for the weights [59]:

$$\omega_i(\boldsymbol{\eta}_{\text{eval}}) = \frac{g_i(\boldsymbol{\eta}_{\text{eval}})}{\sum_{j=1}^{n_M} g_j(\boldsymbol{\eta}_{\text{eval}})}, \quad \text{where } g_j(\boldsymbol{\eta}_{\text{eval}}) = \exp\left(-\frac{1}{2}\frac{(\tilde{s}_{\text{eval}}^{(j)} - s_{\text{eval}})^2}{\sigma_p^2}\right). \tag{6}$$

In the above formulation, we consider $\boldsymbol{\eta}_{\text{eval}} = [\boldsymbol{x}, \boldsymbol{\mu}_{\text{eval}}]$, where $(\boldsymbol{\mu}_{\text{eval}}, s_{\text{eval}})$ is the evaluation set. Instead, $\tilde{s}_{\text{eval}}^{(j)}$ is the result predicted by the j-th model. The parameter σ_p is evaluated in each regression with an optimization algorithm, as in [9], in the following way:

$$\sigma_{\text{opt}} = \arg\min_{\sigma_p \in \mathcal{I}} \sum_{k=1}^{N_{\text{eval}}} \left((s_{\text{eval}})_k - \sum_{l=1}^{n} \omega_l((\boldsymbol{\eta}_{\text{eval}})_k; \sigma_p)(\tilde{s}_{\text{eval}})_k\right)^2, \tag{7}$$

where $\mathcal{I}$ is the range within which the optimal value of σ_p is sought. As investigated in detail in [9,59], the algorithm (7) can be used to establish the optimal *order of magnitude* of σ_p, since it slightly influences the results if the order of magnitude is the optimal one.

Finally, one can train a regression technique for each model considering $(\boldsymbol{\eta}_{\text{eval}}, \omega_j(\boldsymbol{\eta}_{\text{eval}}))$ for each $j = 1, \ldots, n_M$ as data and then let the regression algorithm predict $\omega_j(\boldsymbol{\eta}_{\text{test}})$ for each $j = 1, \ldots, n_M$. Being inspired from previous works such as [6,59], a random forest regression has been used here. However, any regression technique may be used to address this task, such as a GPR, or a neural network for example.

Random Forest regression The random forest (RF) regression [4,30,39] is a machine learning algorithm used for supervised regression, which relies on different hyperparameters, namely the number of trees, the criterion for node splitting, and the minimum number of samples in a leaf. In our case, the number of trees is set to 100, the criterion for splitting is the mean squared error, and the minimum number of samples in a leaf is set to 2.

In general, at this point there are two operative choices:

- train $n_M - 1$ regressions, having as output $\{\omega_i(\boldsymbol{\eta}^\star)\}_{i=1}^{n_M-1}$ and compute the remaining weights as $\omega_{n_M}(\boldsymbol{\eta}^\star) = 1 - \sum_{i=1}^{n_M-1} \omega_i(\boldsymbol{\eta}^\star)$, for a fixed $\boldsymbol{\eta}^\star$;
- train separately n_M regressions, having as outputs the Gaussians $\{g_j(\boldsymbol{\eta}^*)\}_{j=1}^{n_M}$ and then normalize the Gaussians as in (6).
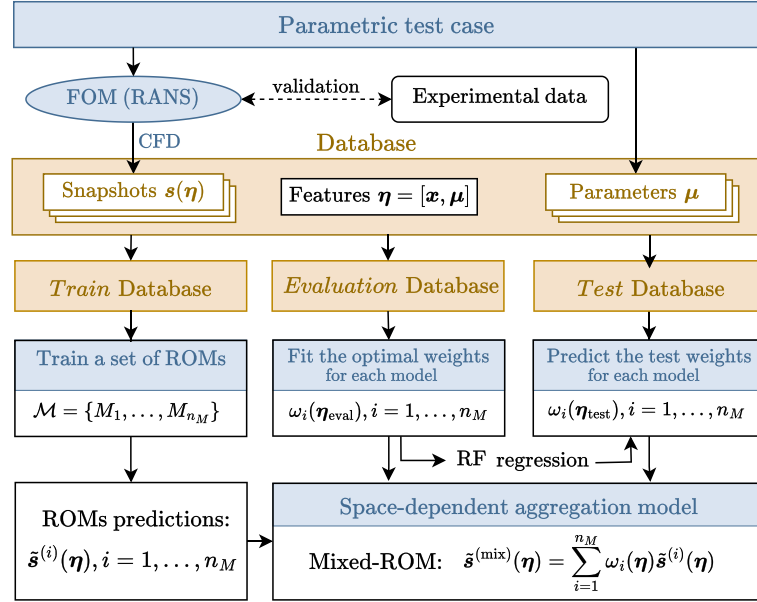
Fig. 2 Flow chart which summarizes the procedure adopted in this article to build a space-dependent aggregation model, as a model-of-mixture of ROMs

In this work, we adopt the second strategy. It is also important to highlight that in some cases it may happen that the sum $\sum_{j=1}^{n_M} g_j(\boldsymbol{\eta}_{\text{eval}})$ is considerably small, leading the division in Eq. (6) to be ill-conditioned. To avoid this numerical issue, we setup a minimum value for the range $\mathcal{I}$ in Eq. (7) to search the optimal value of σ_p. In particular, we set $\mathcal{I} = [10^{-3}, 1]$.

It is also important to remark that we decided to use a random forest algorithm for the regression task, since it has already proven to be efficient in aggregation models [6,59]. Moreover, with respect to other deep learning strategies (artificial neural networks, for instance), it is characterized by shorter training times and better performances in data scarcity conditions.

2.2.1 On uncertainty quantification

The predictions $\tilde{\boldsymbol{s}}^{(i)}$ given by the reduced-order models involved in the *model-mixture* can be seen as n_M possible outcomes of a random variable $\hat{\boldsymbol{s}}$, where the weights $\omega^{(i)}$ are interpreted as the corresponding probability mass functions [59]. We can then consider the expression in (5) as the expected value of the variable taken into account:

$$E[\hat{\boldsymbol{s}}(\boldsymbol{\eta})] = \tilde{\boldsymbol{s}}^{(\text{mix})}(\boldsymbol{\eta}) = \sum_{i=1}^{n_M} \omega_i(\boldsymbol{\eta})\tilde{\boldsymbol{s}}^{(i)}(\boldsymbol{\eta}). \tag{8}$$

The random variable is also characterized by a variance written in the following form:

$$Var[\hat{\boldsymbol{s}}(\boldsymbol{\eta})] = \sum_{i=1}^{n_M} \omega_i(\boldsymbol{\eta}) \left(\tilde{\boldsymbol{s}}^{(i)}(\boldsymbol{\eta}) - E[\hat{\boldsymbol{s}}(\boldsymbol{\eta})]\right)^2. \tag{9}$$

The variance has the meaning of indicator of agreement among the models used in the model-mixture. Indeed, it will be larger in the space regions where the models provide considerably different outcomes and smaller where the models predict similar results.

Now that each step of the proposed strategy has been properly introduced, the whole pipeline followed to build the space-dependent aggregation model can be summarized as shown in the flow chart of Fig. 2.

In the numerical results' Sect. 3, the uncertainty analysis is included both through the *confidence interval* (for one-dimensional fields), and the *graphical variance* (for two-dimensional fields). In particular, the confidence interval is evaluated considering the range $[E[\hat{\boldsymbol{s}}(\boldsymbol{\eta})] - 2\sqrt{Var[\hat{\boldsymbol{s}}(\boldsymbol{\eta})]}, E[\hat{\boldsymbol{s}}(\boldsymbol{\eta})] + 2\sqrt{Var[\hat{\boldsymbol{s}}(\boldsymbol{\eta})]}]$, which corresponds to the 95% of confidence.

3 Numerical results

In the present section, we present the numerical results obtained by aggregating different combinations of ROMs for two different test cases:

(i) steady and 2D flow past a NACA 4412 airfoil, having as parameter the inlet velocity, namely the Reynolds number. In particular, the Reynolds number is in the range $[1 \times 10^5, 1 \times 10^6]$;

(ii) transonic and 2D flow past the symmetric NACA 0012 airfoil, having as parameter the angle of attack, which varies within the range $[0°, 10°]$ at fixed Reynolds number $Re = 1 \times 10^7$.

The results' section is divided into the following parts:

- description of the RANS models herein considered for both test cases, and validation of them with respect to the high-fidelity and/or experimental counterparts (Sect. 3.1);
- analysis of the POD eigenvalues' decay for both cases, and, brief study of the ROMs' results (Sect. 3.2);
- presentation of the results of the aggregated models for test case 1 (Sect. 3.3.1) and 2 (Sect. 3.3.2), and finally a brief discussion of the results, including a comparison between the performances in the two test cases.

3.1 Full-order model description and validation

This subsection is dedicated to the presentation of the full-order models considered to obtain the datasets for the ROMs in the two test cases. The software used to perform the simulations for both cases is OpenFOAM [24], a CFD open-source software, based on a finite-volume space discretization. The following parts, namely 3.1.1 and 3.1.2, are dedicated to the description of the single FOMs for test case 1 and 2, respectively.

3.1.1 Test case 1

The first test case is the 2D steady flow past a NACA 4412 airfoil at fixed angle of attack $\alpha = 5°$. This case is simulated employing the Reynolds-averaged Navier–Stokes (RANS) formulation.

The main hypothesis that characterizes the RANS approach is the *Reynolds decomposition* [38]. This theory is based on the assumption that each flow field can be expressed as the sum of its mean and fluctuating parts. Such mean has different definitions depending on the case of application. Classical choices are, for example, time averaging, averaging along homogeneous directions or ensemble averaging.

We briefly recall here the standard RANS formulation for the incompressible Navier–Stokes equations:

$$
\begin{cases}
\dfrac{\partial \overline{u}_i}{\partial x_i} = 0, \\[2ex]
\overline{u}_j \dfrac{\partial \overline{u}_i}{\partial x_j} = -\dfrac{\partial \overline{p}}{\partial x_i} + \dfrac{\partial (2\nu \overline{\mathbf{E}}_{ij} - \mathcal{R}_{ij})}{\partial x_j}.
\end{cases}
\tag{10}
$$

In system (10), the Einstein notation has been adopted. Moreover, $\mathcal{R}_{ij} = \overline{u_i' u_j'}$ is the Reynolds stress tensor, and $\overline{\mathbf{E}}_{ij} = \dfrac{1}{2}\left(\dfrac{\partial \overline{u}_i}{\partial x_j} + \dfrac{\partial \overline{u}_j}{\partial x_i}\right)$ is the averaged strain rate tensor.

The RANS formulation in (10) needs to be coupled with a turbulence model to close system (10). In particular, we adopt the $\kappa - \omega$ *shear stress transport* (SST) model [32].

This model belongs to the class of *eddy viscosity models*, which are based on the Boussinesq hypothesis, i.e., the turbulent stresses are related to the mean velocity gradients as follows:

$$
-\mathcal{R}_{ij} = 2\nu_t \overline{\mathbf{E}}_{ij} - \frac{2}{3}\kappa \delta_{ij},
$$

where $\kappa = \frac{1}{2}\overline{u_i' u_i'}$ is the turbulent kinetic energy and ν_t is the eddy viscosity. For the complete model, we refer the reader to the original paper [32].
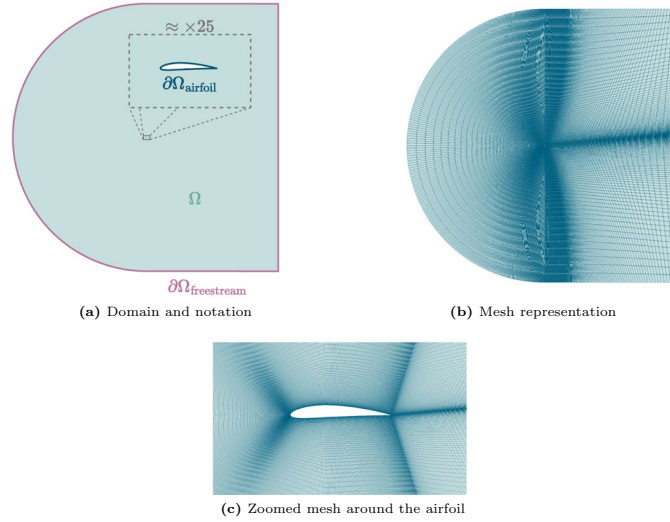
(a) Domain and notation   **(b)** Mesh representation

(c) Zoomed mesh around the airfoil

Fig. 3 Domain (**a**) and mesh (**b**) representation for test case 2, with a zoomed representation for the zone around the airfoil (**c**)

The domain and the mesh considered for this simulation are represented in Fig. 3. The domain boundaries $\partial\Omega_{\text{freestream}}$ are at a distance 100 times the airfoil chord length, and so essentially in a freestream condition. Hence, the boundary conditions are the following:

$$
\begin{cases} \boldsymbol{u} = \boldsymbol{u}_{\text{freestream}}, \\ p = p_{\text{freestream}}, \end{cases} \text{on } \partial\Omega_{\text{freestream}}, \quad \begin{cases} \boldsymbol{u} = \boldsymbol{0}, \\ p = 0, \end{cases} \text{on } \partial\Omega_{\text{airfoil}}, \tag{11}
$$

where the freestream condition corresponds to a mixed condition, where the mode of operation switches between fixed value and Neumann, based on the sign of the flux. The fixed velocity value is evaluated starting from the Reynolds number, the dataset parameter, whereas the fixed pressure value is $p = 0$.

The pressure–velocity coupling is treated considering the semi-implicit method for pressure-linked equations (SIMPLE) algorithm. For more details, we refer the reader to [34,35].

In order to analyze the flow behavior, in Fig. 4 we report a zoomed detail around the airfoil of the pressure and velocity magnitude snapshots, for two values of the Reynolds number. From the snapshots' representation in Fig. 4, we can notice that the solutions represented by the snapshots are all characterized by a similar behavior. Moreover, also the trend of the pressure coefficient in this test case is similar for all the parameters taken into account, as represented in Fig. 5.

For the above-mentioned reasons, non-intrusive ROMs are particularly suitable to approximate this kind of dataset, as we will see in Sect. 3.2.

The FOM dataset in this first test case is inspired by the AirfRANS database presented in [3], where the reader can find more details about the FOM and the turbulence setting. The above-mentioned FOM has also been validated with the NASA DNS and RANS high-fidelity simulations.

Indeed, in Fig. 6 we report the pressure and skin friction coefficients for the NACA 4412 airfoil at Re $= 4e5$ and angle of attack $\alpha = 5°$. In addition to the FOM data, we reported the DNS data [50] and the RANS simulation data provided in [45]. Similar validation studies have been performed in other works such as [49] and [46], where the tripping point was also taken into account in the simulation setup. We can observe a quite good agreement between the performed RANS and the simulated data. The main difficulty for this specific test case is the considerably high Reynolds number. Due to the predominance of advection effects in this specific set-up, we can expect a particularly challenging test case in terms of model order reduction.

3.1.2 Test case 2

The second test case is the compressible flow past a NACA 0012 airfoil, with fixed inlet velocity at $Re = 1 \times 10^7$ having as a parameter the angle of attack.

The domain, with the corresponding notation and mesh, is represented in Fig. 7. It is important to remark here that the mesh is fixed for all snapshots. Within the different FOM simulations, what changes is the orientation of the inlet velocity, depending on the angle of attack.
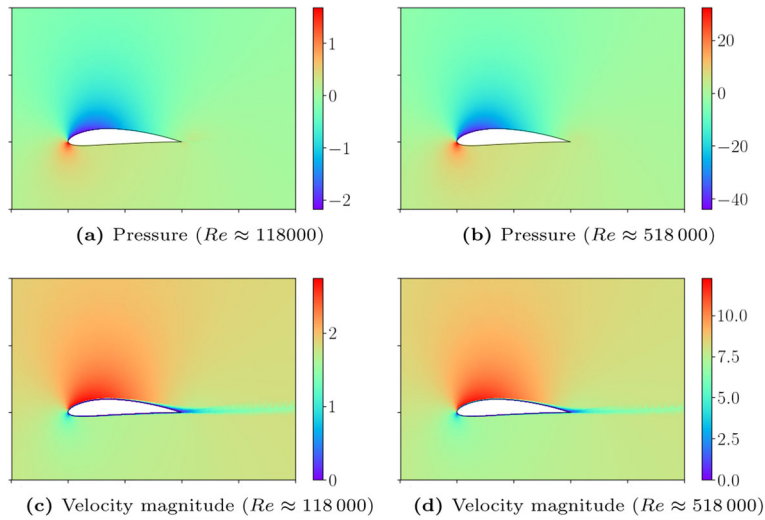
(a) Pressure ($Re \approx 118000$)   **(b)** Pressure ($Re \approx 518\,000$)

(c) Velocity magnitude ($Re \approx 118\,000$)   **(d)** Velocity magnitude ($Re \approx 518\,000$)

Fig. 4 Examples of pressure and velocity magnitude high-fidelity snapshots for the first test case, for $Re \approx 118,000$ and $Re \approx 518,000$
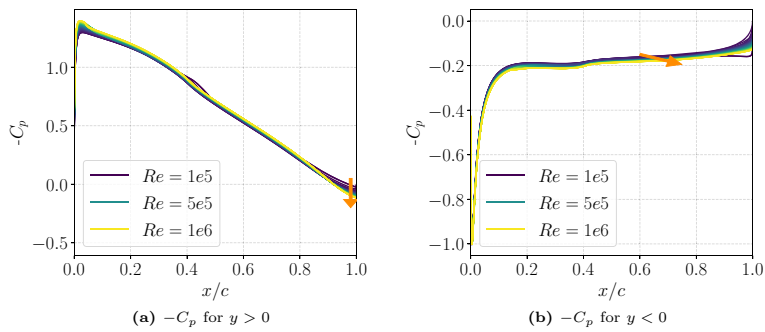

(a) $-C_p$ for $y > 0$ **(b)** $-C_p$ for $y < 0$

Fig. 5 Pressure coefficient for all the elements of the database considered for the first test case, for both suction and pressure sides of the airfoil. The orange arrows represent the direction of growing parameter Re. The legend refers only to the smallest, medium and largest Re numbers, but all the parameters in the database are here considered

(a)   **(b)**
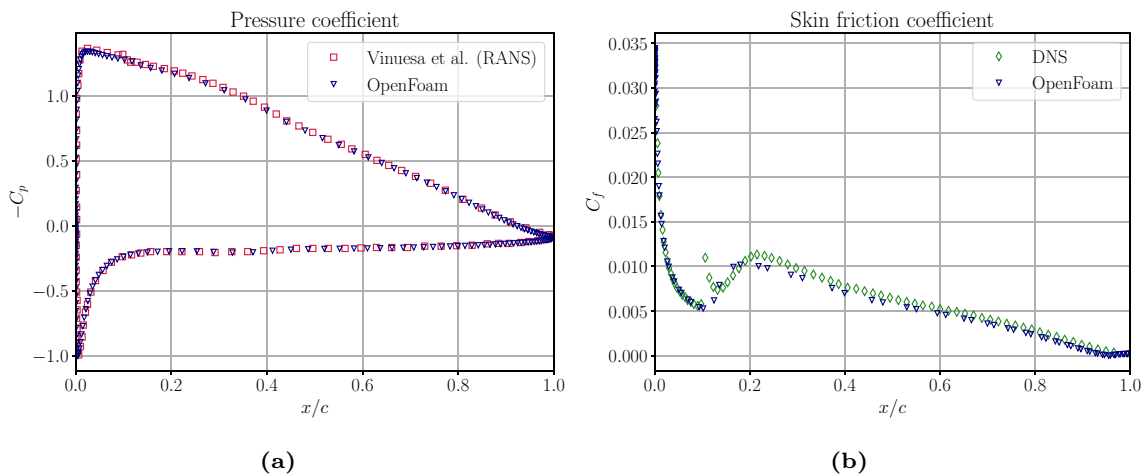
Fig. 6 Validation of pressure (**a**) and upper skin friction (**b**) coefficients for the first type of full-order simulation. The FOM validation is for NACA 4412 at Re $= 4e5$ and angle of attack $\alpha = 5°$. The reference simulations are taken from [50] (DNS) and [45] (RANS)

(a) Domain and notation

(b) Mesh representation

(c) Zoomed mesh around the air-
foil
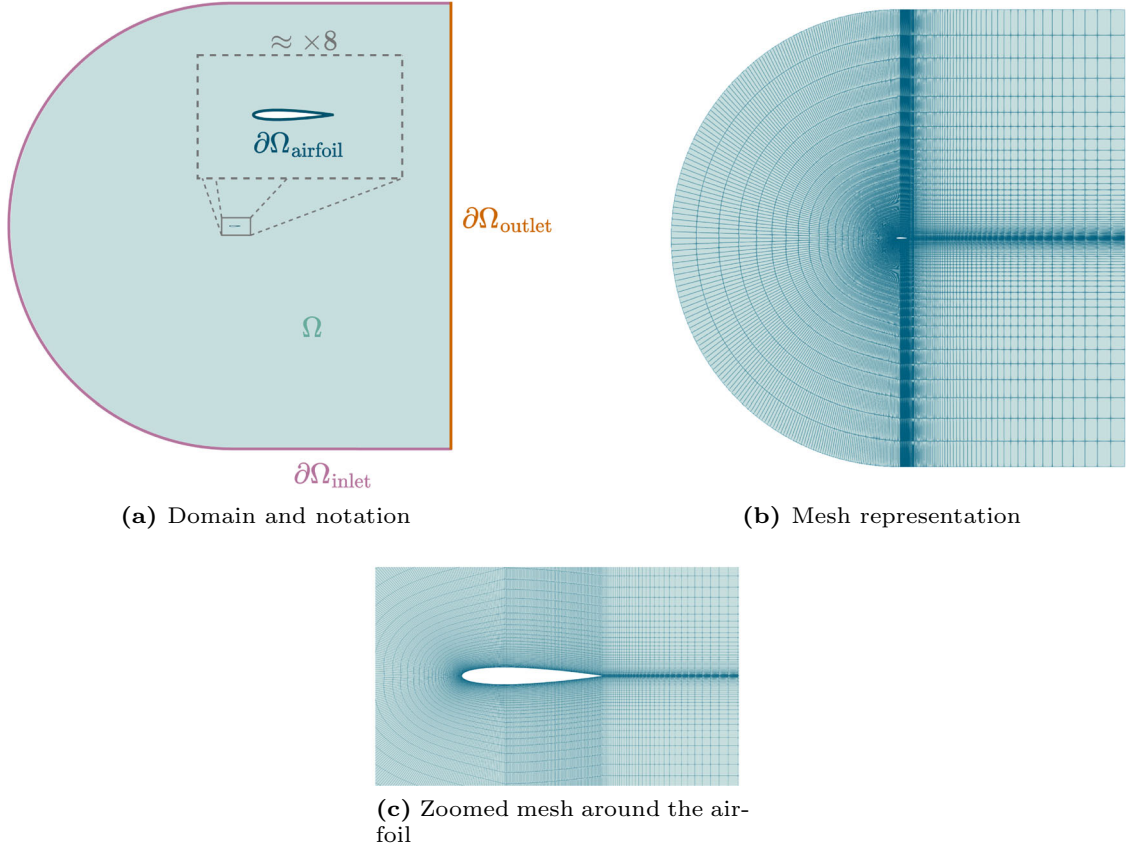
Fig. 7 Domain (**a**) and mesh (**b**) representation for test case 2, with a zoomed representation for the zone around the airfoil (**c**)

In this case, the boundary conditions read as follows:

$$
\begin{cases} \boldsymbol{u} = \boldsymbol{u}_{\text{freestream}}, \\ p = p_{\text{freestream}}, \end{cases} \text{on } \partial\Omega_{\text{inlet}}, \quad \begin{cases} \boldsymbol{u} = \boldsymbol{u}_{\text{freestream}}, \\ \dfrac{\partial\left(\int_{\partial\Omega_{\text{outlet}}} w \cdot p \, dx\right)}{\partial t} = 0, \end{cases} \text{on } \partial\Omega_{\text{outlet}}, \quad \begin{cases} \boldsymbol{u} = \boldsymbol{0}, \\ \nabla p \cdot \boldsymbol{n} = \boldsymbol{0}, \end{cases} \text{on } \partial\Omega_{\text{airfoil}},
$$

(12)

where the pressure condition at the outlet corresponds to the *wave transmission* boundary condition and w is the wave speed. The freestream conditions are mixed conditions, as in the first test case, but here the pressure fixed value is 1×10^5 whereas the velocity inlet value has a fixed magnitude $\|\boldsymbol{u}_{\text{freestream}}\| \approx 277 \, \text{m s}^{-1}$.

The solver for the pressure–velocity coupling used in OpenFOAM is *rhoPimpleFoam*, which coincides with a version of the PIMPLE algorithm for compressible flows. The PIMPLE algorithm is a combination of the SIMPLE and PISO (pressure implicit with splitting of operator) algorithms, and it is used also for steady-state simulations to obtain more robust convergence toward the solution with respect to the SIMPLE counterpart.

Figure 8 displays two different velocity and pressure snapshots, one for $\alpha = 0.3°$ and one for $\alpha = 4.7°$.

Among the solutions retained in the snapshots matrix, considerable differences can be observed. Such differences can be quantified by looking at the shock location, which is clearly visible from the representation of the pressure coefficient with respect to the variation of the angle of attack α (Fig. 9). In the comparison between test case 1 (Fig. 5) and 2 (Fig. 9), the data variability appears significantly wider in test case 2. Indeed, we face here an additional challenge due to the presence of shock waves which are well-known complex features to be captured using reduced-order modeling. For this particular application, furthermore, the shock location varies significantly within the range of angles of attack herein considered. In particular, we considered an angle of attack varying between 0° to 10°. Such a range offers many different behaviors both in terms of
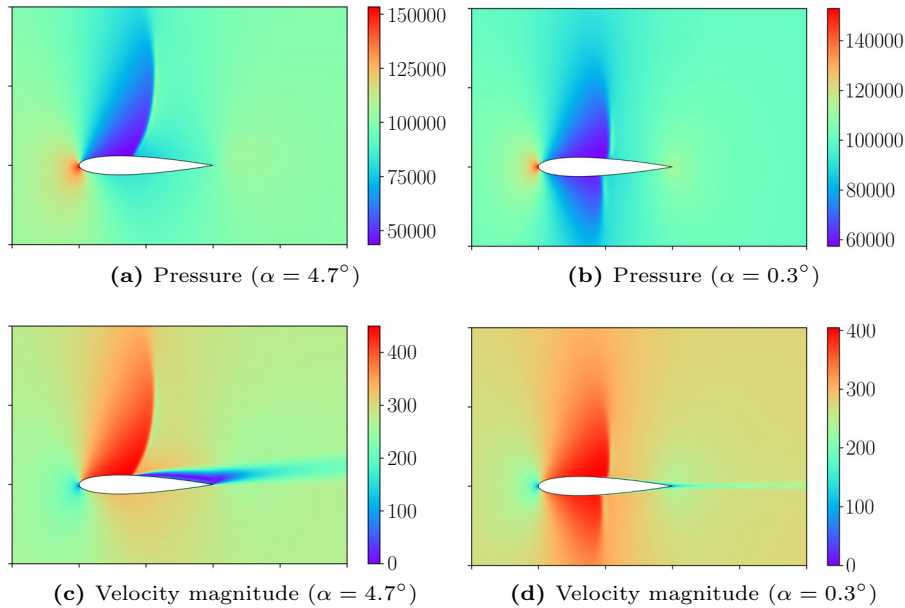
(a) Pressure ($\alpha = 4.7°$)

(b) Pressure ($\alpha = 0.3°$)

(c) Velocity magnitude ($\alpha = 4.7°$)

(d) Velocity magnitude ($\alpha = 0.3°$)

Fig. 8 Examples of FOM snapshots for the second test case, for two values of the angle of attack, $\alpha = 0.3$ and $\alpha = 4.7$
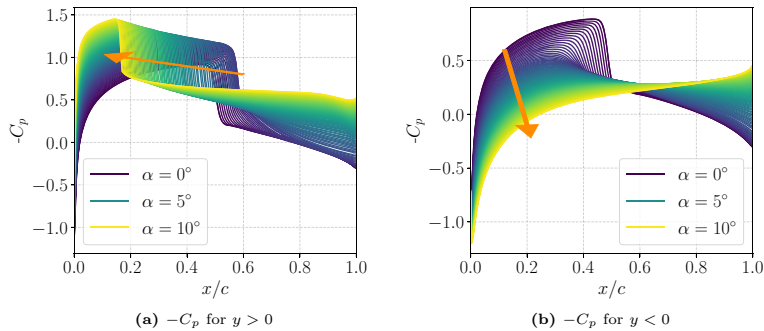

(a) $-C_p$ for $y > 0$

(b) $-C_p$ for $y < 0$

Fig. 9 Pressure coefficient for the elements of the database considered for the second test case, for both suction and pressure sides of the airfoil. The orange arrows represent the direction of growing parameter α. The legend refers only to the smallest, medium and largest α numbers, but all the parameters in the database are here considered

boundary layer separation and shock location. Because of all the above considerations, we expect this test case to be considerably more challenging than the first one in terms of dimensionality reduction.

A validation analysis has also been conducted for the second test case. Figure 10 represents the pressure coefficient in two different operating conditions, namely at $\alpha = 1°$ and $\alpha = 2°$. Here, the results obtained by OpenFOAM are compared with the experimental data presented in [31] and a wall-resolved reference simulation [20], which used the same turbulence model. The agreement is quite good, especially with respect to the simulation taken as reference. The deviations from the experimental data are intrinsically related to the RANS models employed. This set-up has been used precisely because different models can provide quite different outcomes. In this scenario, it would be useful to apply the same type of RANS aggregation proposed by de Zordo-Banliat et al. [59]. However, such improvements in terms of RANS modeling is out of the scope of the present work and future research will be focused on improving the RANS modelization and combine it with the reduced-order modeling strategies herein proposed.

3.1.3 Computational time

In the first test case, the CPU time required for running each full-order simulation is in the range [2 h 30 min , 7 h], depending on the value of the Reynolds number considered. In the second test case, the CPU time is in the interval [1 h 45 min , 3 h], depending on the angle of attack considered. The FOM simulations
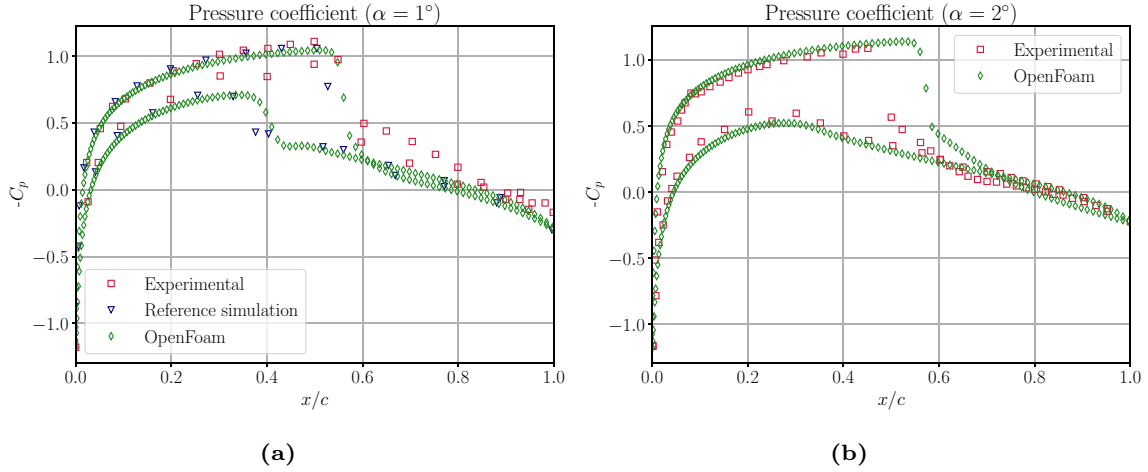
Fig. 10 Validation of pressure coefficient of the full-order model for two angles of attack: $\alpha = 1°$ (**a**) and $\alpha = 2°$ (**b**). The FOM validation is for NACA 0012 at Mach Ma $= 0.8$, and the reference results are presented in [31] (experimental) and [20] (wall-resolved reference simulation)

are all performed using 4 processor cores on SISSA HPC cluster Ulysses (200 TFLOPS, 2TB RAM, 7000 cores).

3.2 Individual ROM analysis

This part of the paper is dedicated to the analysis of the individual ROMs considered for the model mixture.

The fields considered for the model reduction can be divided into two types: (i) one-dimensional (1D) wall variables on the airfoil (pressure and wall shear stress); (ii) two-dimensional (2D) fields in the internal mesh (pressure and velocity magnitude).

In both the test cases taken into account, we collect the predictions of the following non-intrusive models:

- POD-RBF;
- POD-GPR;
- POD-ANN;
- AE-RBF and PODAE-RBF, for the 1D airfoil fields and the 2D mesh fields, respectively;
- AE-GPR and PODAE-GPR, for the 1D airfoil fields and the 2D mesh fields, respectively;
- AE-ANN and PODAE-ANN, for the 1D airfoil fields and the 2D mesh fields, respectively.

As above-mentioned, the nonlinear reduction approach considered for the 2D fields is the mixed technique PODAE, instead of the pure AE. The reason is that the PODAE allows for a valuable reduction in the computational effort in the offline part, and especially in the training of the autoencoder.

Following the notation introduced in Sect. 2.2, the training dataset considered for all the data-based ROMs is composed of $N_{train} = 70$ snapshots, the evaluation set of $N_{eval} = 20$ elements, and the test set of $N_{test} = 10$ elements.

Figures 11 and 12 represent the POD eigenvalues' decays for all the variables considered and for the two test cases, respectively.

It can be seen that the eigenvalues' decays for all variables in the second test case is considerably slower than in the corresponding variables in the first test case. This fact is reflected on the accuracy of the ROMs and of the model mixture, that are presented in Sects. 3.3.1 and 3.3.2 for the first and second test cases, respectively. The problems herein considered are consequently presented for increasing complexity in terms of dimensionality reduction.

At this point, we address the accuracy of the individual ROMs on the *ROM test set*, namely the union of the evaluation and test datasets. The two test cases are separately considered in 3.2.1 and 3.2.2.
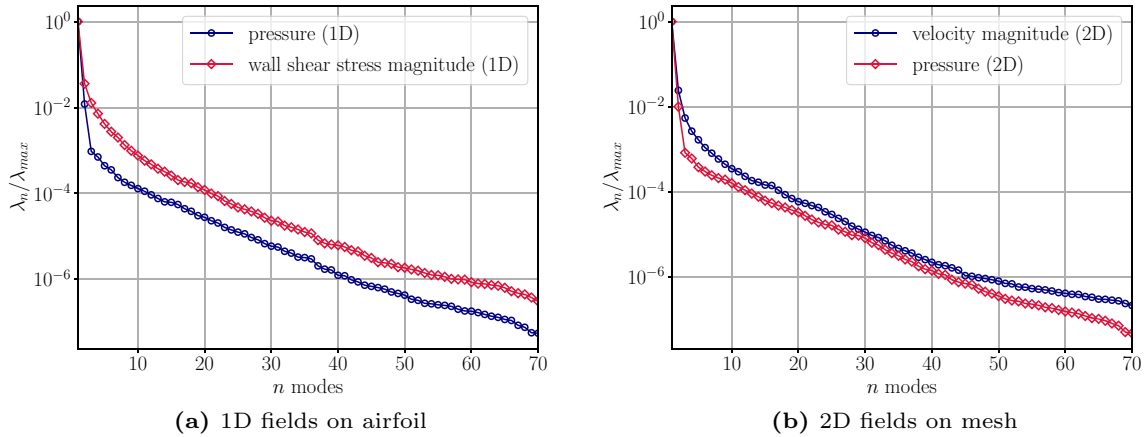
(a) 1D fields on airfoil

(b) 2D fields on mesh

Fig. 11 Decay of POD eigenvalues for the 1D fields on the airfoil and for 2D fields on the computational mesh around the airfoil for test case 1
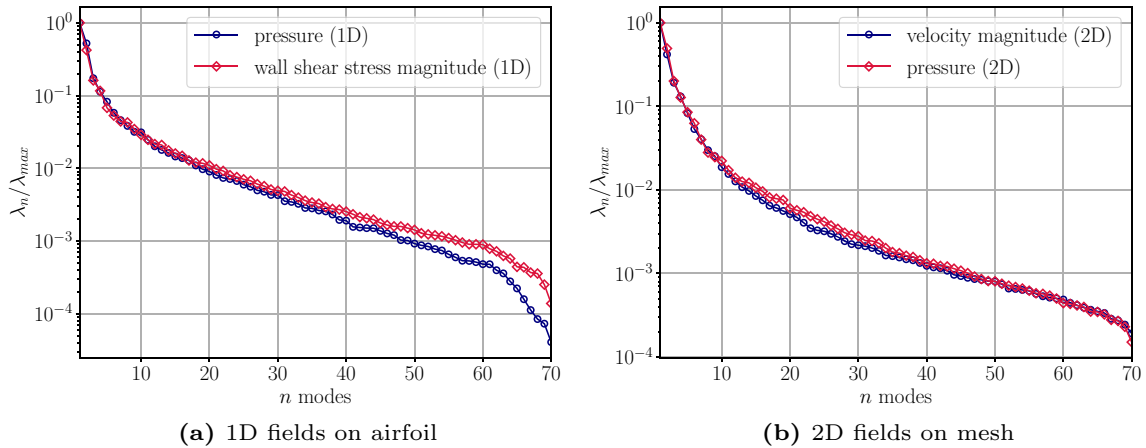

(a) 1D fields on airfoil

(b) 2D fields on mesh

Fig. 12 Decay of POD eigenvalues for the 1D fields on the airfoil and for 2D fields on the computational mesh around the airfoil for test case 2

3.2.1 Test case 1

In Figs. 13 and 14, we represent the relative errors on the ROM test set with respect to the FOM counterpart, on the 1D wall shear stress field and on the 2D velocity magnitude field, respectively.

From both the figures, the following considerations can be drawn:

- all the individual ROMs in both the latent dimensions considered are characterized by better performances for high values of the Reynolds parameter;
- the AE-based ROMs have similar accuracy for the two latent dimensions;
- the AE outperforms the results of the POD as a reduction approach almost everywhere in the parametric domain when the latent dimension is small. Indeed, in this case we are in the so-called *under-resolved* modal regime, namely the POD modes are not able to fully characterized the dynamics of the system.

As a comparison between the two fields (1D wall shear and 2D velocity), the results are less accurate for the wall shear stress field, as can be seen from the order of magnitude of the errors. Indeed, it is usually a complex field to characterize in the flow past an airfoil.

3.2.2 Test case 2

In this second test case, we represent the relative test errors for the 1D wall shear stress and the 2D pressure fields, in Figs. 15 and 16, respectively.
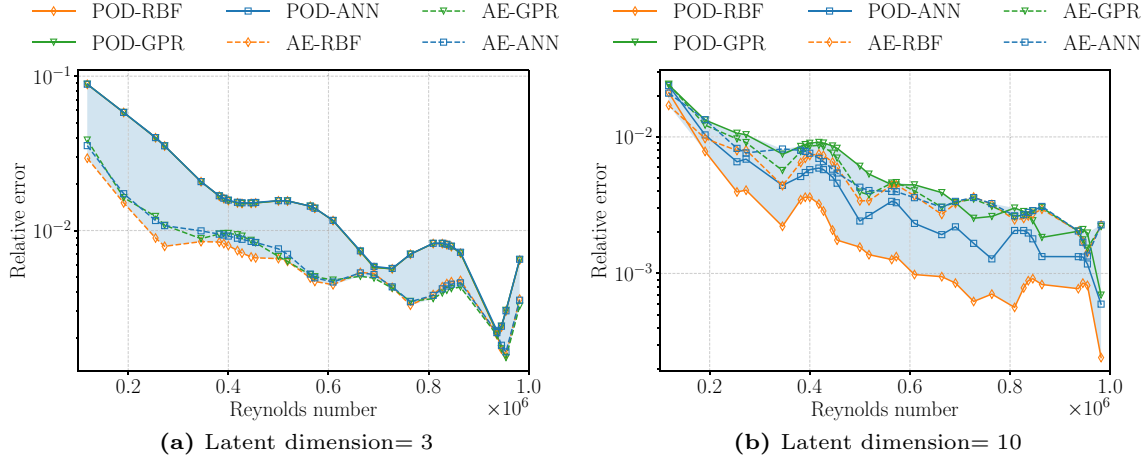
Fig. 13 Relative test errors of individual ROMs for the 1D wall shear stress magnitude field, for two different latent dimensions, namely 3 and 10. The results are for test case 1. The shading represents the accessible area of all the reduced models' predictions
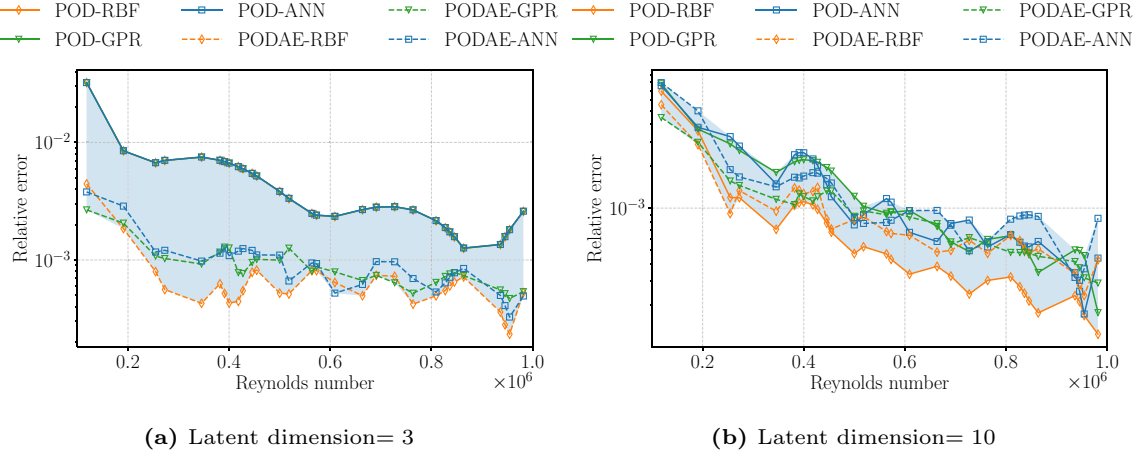
Fig. 14 Relative test errors of individual ROMs for the 2D velocity magnitude field, for two different latent dimensions, namely 3 and 10. The results are for test case 1. The shading represents the accessible area of all the reduced models' predictions

As expected, the performances of the individual ROMs are remarkably different from the first test case. We can, hence, notice that:

- the performances of the single ROMs are more differentiated than in the first test case, and, therefore, the accessible region by all the ROMs is bigger;
- there is not a clear trend of the errors as the parameter (the angle of attack in this case) increases;
- the AE (or PODAE)-RBF model outperforms all the other techniques in most of the parameter's values;
- while in the first test case, the choice of the reduction is the most relevant in the ROM performance, in this test case also the type of approximation plays an important role.

The complexity of this problem can be quantified by observing the decay of the singular values of the snapshots' matrix for the pressure coefficient and the wall shear stress already depicted in Fig. 12. In order to achieve an error of 1%, in fact, at least 20 modes are needed.

In order to have a large variability between the different reduced-order models herein considered, we decided to consider a relatively small number of latent variables (namely 3 and 10). The motivation for this choice is twofold: first, we obviously want the latent space to be small so that we can achieve a significant speed-up by means of ROMs. Secondly, we want to position ourselves in a case where the different ROMs provide significantly different results. Only in this scenario, the aggregation can be beneficial.

Section 3.3 will thoroughly discuss how the different decays of the two cases will affect both the individual ROMs and their aggregation on the overall accuracy of the proposed approach.
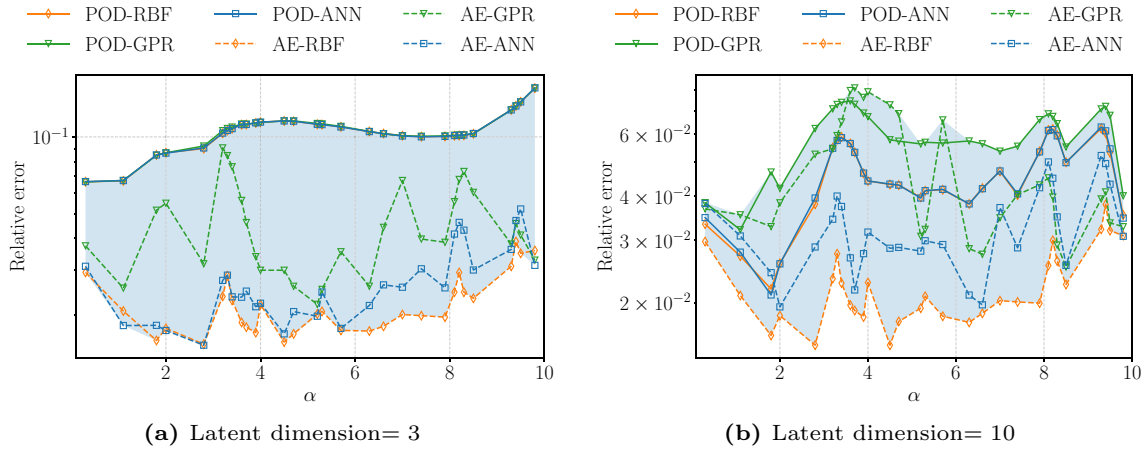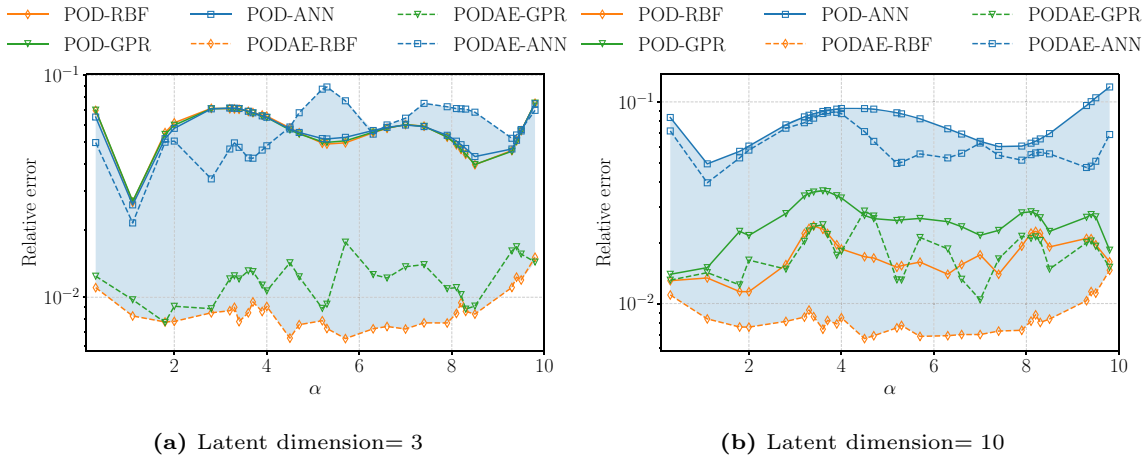
(a) Latent dimension= 3   **(b)** Latent dimension= 10

Fig. 15 Relative test errors of individual ROMs for the 1D wall shear stress field, for two different latent dimensions, namely 3 and 10. The results are for test case 2. The shading represents the accessible area of all the reduced models' predictions

(a) Latent dimension= 3   **(b)** Latent dimension= 10

Fig. 16 Relative test errors of individual ROMs for the 2D velocity magnitude field, for two different latent dimensions, namely 3 and 10. The results are for test case 2. The shading represents the accessible area of all the reduced models' predictions

3.2.3 Computational time

All the ROMs evaluations are performed on an Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz 16GB RAM on only one processor core. Tables 1 and 2 report the execution times of non-intrusive ROMs for offline and online stages, respectively. It is worth remarking that the offline times do not include the FOM times, which are reported in Sects. 3.1.1 and 3.1.2, for the two test cases, respectively. The online execution time is here the time required by the ROM prediction in the test set. The difference in the execution times between 1D airfoil and 2D internal fields for the reduction techniques is caused by the different initial dimension. Indeed, the compression of the 2D fields' snapshots' matrix takes more time than in the 1D fields' case.

We also specify that the initial dimension of the snapshots is $n_{dof} = 965$ for the 1D airfoil fields, $n_{dof} \sim 2.5e5$ for 2D fields, in the first test case. In the second test case we have $n_{dof} = 448$ for 1D fields, and $n_{dof} \sim 2.5e5$ for 2D fields. The results in terms of computational times are similar for the two test cases, so we report in the tables the execution time only for test case 1.

3.3 Aggregation results

This section is dedicated to the results of the aggregated ROM models for the first and second test cases, addressed in 3.3.1 and 3.3.2, respectively. The following results will also be supported by an uncertainty analysis based on the concepts introduced in Sect. 2.2.1.

Table 1 ROM execution times in offline stage, for all the reduction and approximation techniques for r = 3

Model	Field	Offline time—Test case 1 [s]
POD	Airfoil	$\sim 4 \times 10^{-3}$
	Internal	~ 2.18
AE	Airfoil	$\sim 40 - 50$
PODAE	Internal	$\sim 25 - 35$
RBF	(All cases)	$\sim 3.2 \times 10^{-2}$
GPR	(All cases)	$\sim 3.5 \times 10^{-2}$
ANN	Airfoil	$\sim 7 - 10$
	Internal	$\sim 80 - 100$

Table 2 ROM execution times in online stage, for all the reduction and approximation techniques for r = 3

Model	Field	Online time—test case 1 (s)
POD-RBF	Airfoil	$\sim 1 \times 10^{-4}$
	Internal	$\sim 3.3 \times 10^{-2}$
AE-RBF	Airfoil	$\sim 5 \times 10^{-3}$
PODAE-RBF	Internal	$\sim 4.3 \times 10^{-2}$
POD-GPR	Airfoil	$\sim 2 \times 10^{-4}$
	Internal	$\sim 4.5 \times 10^{-2}$
AE-GPR	Airfoil	$\sim 6 \times 10^{-3}$
PODAE-GPR	Internal	$\sim 7.6 \times 10^{-2}$
POD-ANN	Airfoil	$\sim 2 \times 10^{-4}$
	Internal	$\sim 2 \times 10^{-3}$
AE-ANN	Airfoil	$\sim 4 \times 10^{-4}$
PODAE-ANN	Internal	$\sim 4.3 \times 10^{-2}$

3.3.1 Test case 1

As a first quantitative analysis, we include Figs. 17 and 18, showing a comparison in the relative errors among the individual ROM models and the aggregated models (or *mixed-ROMs*). In particular, in both the test cases we built two different aggregation models, considering:

- POD-RBF and AE(or PODAE)-RBF, namely two methods with different reduction techniques but with the same approximation method;
- the two "*best*" ROMs, based on the relative errors of the individual ROMs in the training set.

Based on Figs. 17 and 18, we can make different considerations. First of all, as already noticed in the preliminary analysis in 3.2.1, the performance of POD-based methods improves in general as the number of modes increases, whereas AE-based methods are not affected as much by the latent dimension. Secondly, the aggregated models always outperform the individual ROMs in the evaluation set. That is obvious by mathematical law, since the weights are calculated to minimize the discrepancy with the full-order counterpart in that specific set. However, this may not happen in the test set, where the weights are computed through regression law, namely a random forest regression. In fact, the performance of the aggregation in the test set highly depends on the database used to fit and test the regression. In particular, we have in all cases an improvement of the results in the test set for the airfoil pressure field and for the internal fields, but not for the wall shear field. In that case, we have comparable results with respect to the individual non-intrusive ROMs. Moreover, the errors for the wall shear stress field are about one order of magnitude higher than for the other fields. This may lead to an increased complexity when combining the individual models.

As an example of how the aggregation performs along the chord of the airfoil, we include here Fig. 19, displaying the skin friction coefficient C_f for a test parameter. In particular, the figure shows the predictions of the *mixed-ROM*, the FOM ground truth, and the regions accessible by the ROMs considered for the model mixtures. In addition, the light orange/green region represents the accessible area for the aggregated ROMs in the two types of aggregation here considered. It essentially quantifies the envelope of all the possible linear combinations of the individual ROMs aggregated.

The *mixed-ROM* provides a prediction that is the closest as possible to the FOM snapshot, while remaining inside the accessible ROMs region, as can be seen in the zoomed region in Fig. 19. The figure also includes an
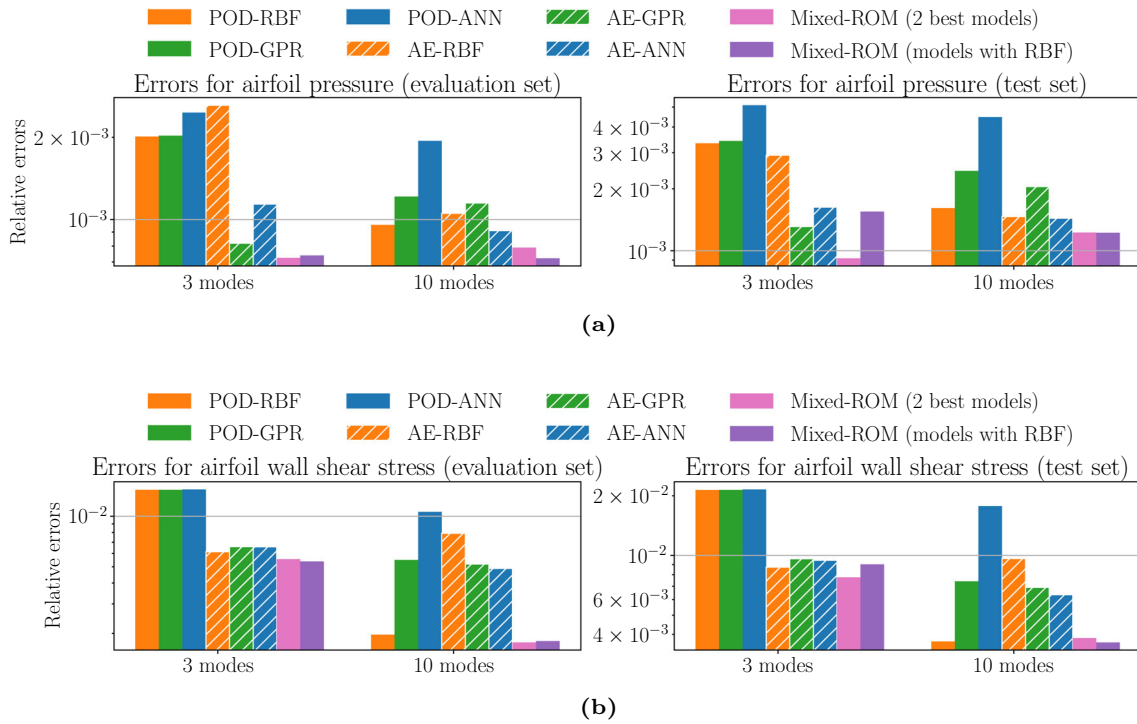
Fig. 17 Relative errors on evaluation and test set for 1D fields on the airfoil for all ROMs and for aggregated models in test case 1
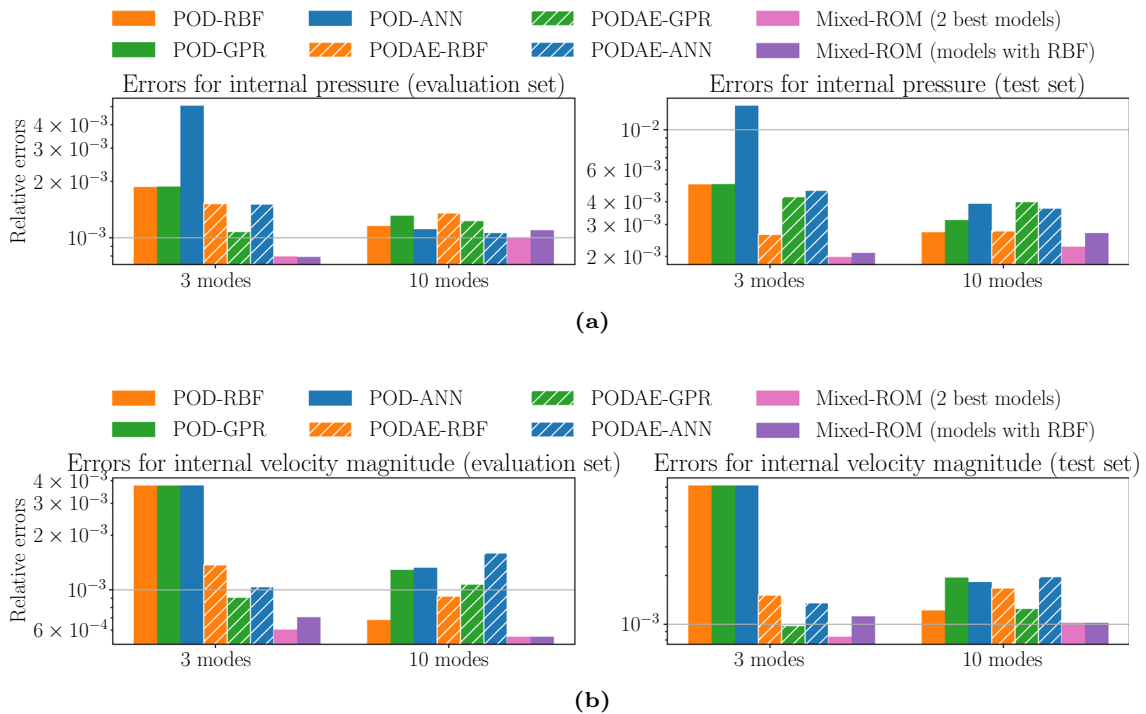

Fig. 18 Relative errors on evaluation and test set for 2D internal fields for all ROMs and for aggregated models in test case 1
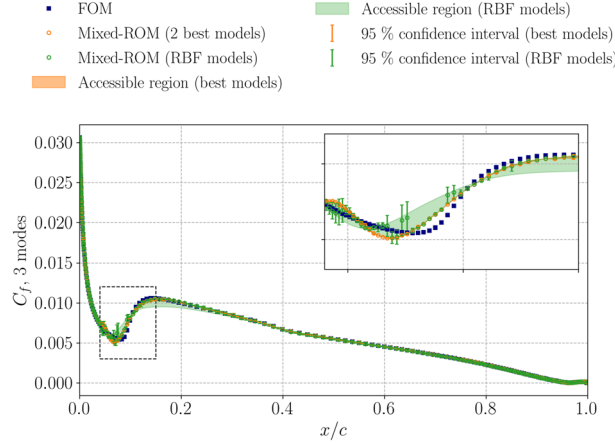
Fig. 19 Representation of the skin friction coefficient C_f on the suction side for mixture models and FOM for a test parameter $Re \simeq 518{,}000$. We consider 3 as reduced dimension. The *accessible region*, namely the region accessed by the ROMs predictions considered in the aggregation, is also here represented. The Mixed-ROMs correspond to the expected values of the aggregations $E[C_f]$, while the error bars correspond to interval $[E[C_f] - 2\sqrt{Var[C_f]}, E[C_f] + 2\sqrt{Var[C_f]}]$
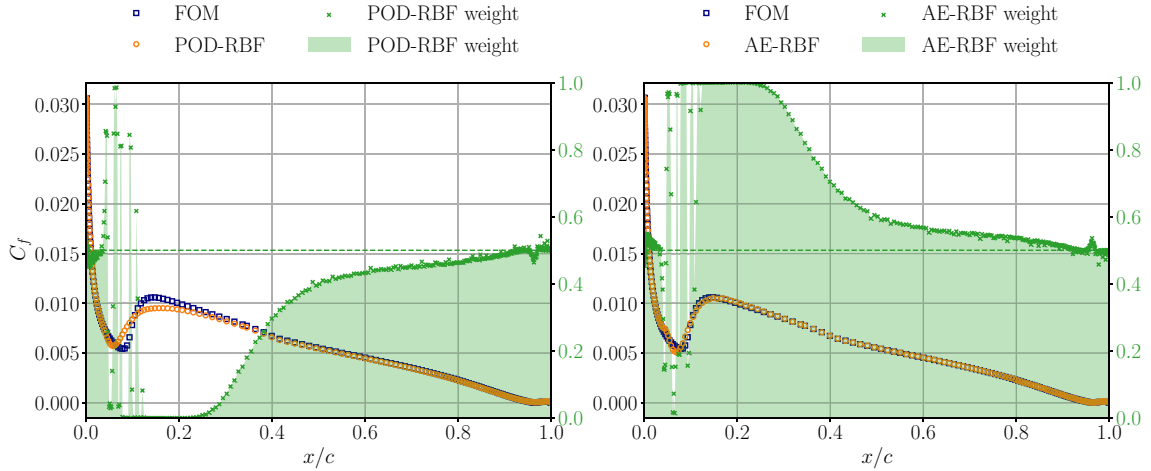
Fig. 20 Representation of the skin friction coefficient C_f on the suction side for two ROMs (POD-RBF and AE-RBF) and of the corresponding weights for the aggregation with only RBF models. We consider as test parameter $Re \simeq 518{,}000$ and as reduced dimension 3

uncertainty quantification analysis via the evaluation of the confidence intervals. Indeed, as stated in Sect. 2.2.1, we considered the aggregation prediction as the expected value of a random variable and computed the corresponding variance using expression (9). The error bars reveal that the variance is significantly small in most of the spatial domain, except for the region with the largest accessible area. The meaning of the variance can be indeed linked to the consensus among the models and to the concept of the accessible area itself. It can be also seen as a measure of the risk of having a completely wrong prediction if only one single model was used.

Moreover, Fig. 20 provides the spatial distributions of the weights for the mixture of RBF-based models. The figure shows that the AE-RBF has weights close to 1.0 for $x/c \in [0.15, 0.3]$, where it provides the most accurate approximation. In the region $x/c \in [0.6, 1]$, since the two ROMs have similar predictions, the weights are close to 0.5, i.e., the aggregated model retains the two individual ROMs in the same percentage. On the other hand, the fact that the weights are oscillating in space in the region $x/c \in [0, 0.15]$ indicates more uncertainty of the aggregation model and it is reflected also in the oscillatory width of the confidence interval in the zoomed region of Fig. 19. Indeed, it may happen that in some spatial regions the weights are not well informed due to an inaccurate prediction of the random forest regressor in the test set. However, as already mentioned above, the error bars are overall very small, indicating that the algorithm is well-performing and the best-performing model is preferred.
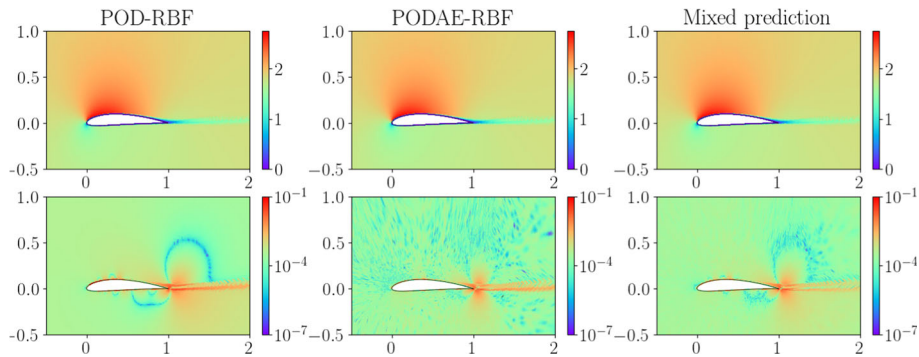
Fig. 21 Velocity magnitude predictions for two ROMs and for the aggregation model (first row), and corresponding relative errors with respect to FOM (second row). The test parameter considered is $Re \approx 118,000$, and the reduced dimension is 10
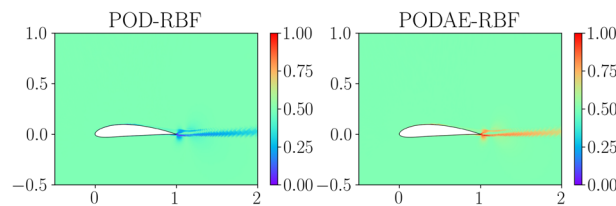

Fig. 22 Weights for RBF models for the velocity magnitude field when considering the test parameter $Re \approx 118,000$. The reduced dimension is 10

If we consider a velocity internal field (Figs. 21 and 22), we can notice that the two ROMs have a similar performance and are really close to the FOM reference.

However, the AE-RBF model is closer to the FOM prediction in the wake, as can be seen from the spatial relative error in Fig. 21, and, hence, the corresponding weights are close to 1 in that specific zone of the domain (Fig. 22).

3.3.2 Test case 2

Based on the results obtained in the previous section, we wanted to consider a test case that could be considered more challenging in terms of reduction. Consequently, we moved toward a transonic airfoil where the angle of attack has been chosen as the parameter of interest. In this scenario, due to the significant movement of the shock wave for different parametric samples, the reduction is expected to be more demanding.

In Figs. 23 and 24, we show the relative errors of each individual ROM (varying the type of reduction and approximation in the parametric space) and with the different types of mixing combinations.

As a general observation, we can notice that the relative error of the aggregated models is always significantly small.

The same type of analysis is carried out also on the variables on the internal mesh with respect to the airfoil. The relative errors are thereby shown in Fig. 24. Also in this case we can observe that the aggregated model can significantly improve the best possible combination of individual ROMs, or, at least, identify the best possible model for each configuration.

In Fig. 25, we show the results for a specific solution of the test set. In particular, we depict the pressure coefficient at the wall given by the full-order model and the two aggregated ROMs. On the left, we show the results obtained with a latent space of dimension 3 and on the right with a latent space of dimension 10. Clearly, the accessible area may be significantly larger when the dimension of the latent space is small: in this scenario, we can expect much different results from each individual ROMs. As the dimension of the latent space increases, all of the different ROMs will tend to converge to the exact solution, limiting the variability among them. We can notice that within this region, the aggregated model correctly predicts a mixed solution that is as close as possible to the reference (Fig. 26).

It is important to stress that having a significantly large accessible region is fundamental in the success of the proposed strategy. Such a feature, in fact, indicates that the large differences between the models can be exploited in order to obtain a better aggregated solution. It can be noticed, in fact, that considering a latent
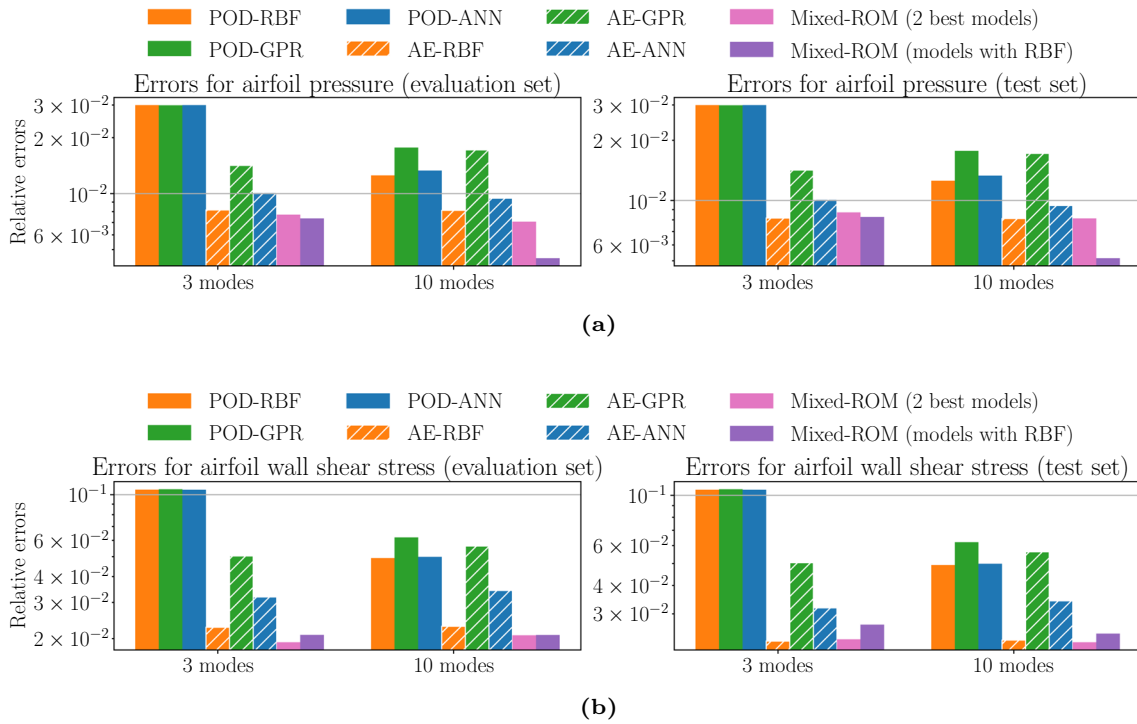
Fig. 23 Relative errors on evaluation and test set for the 1D fields on the airfoil for all ROMs and for aggregated models
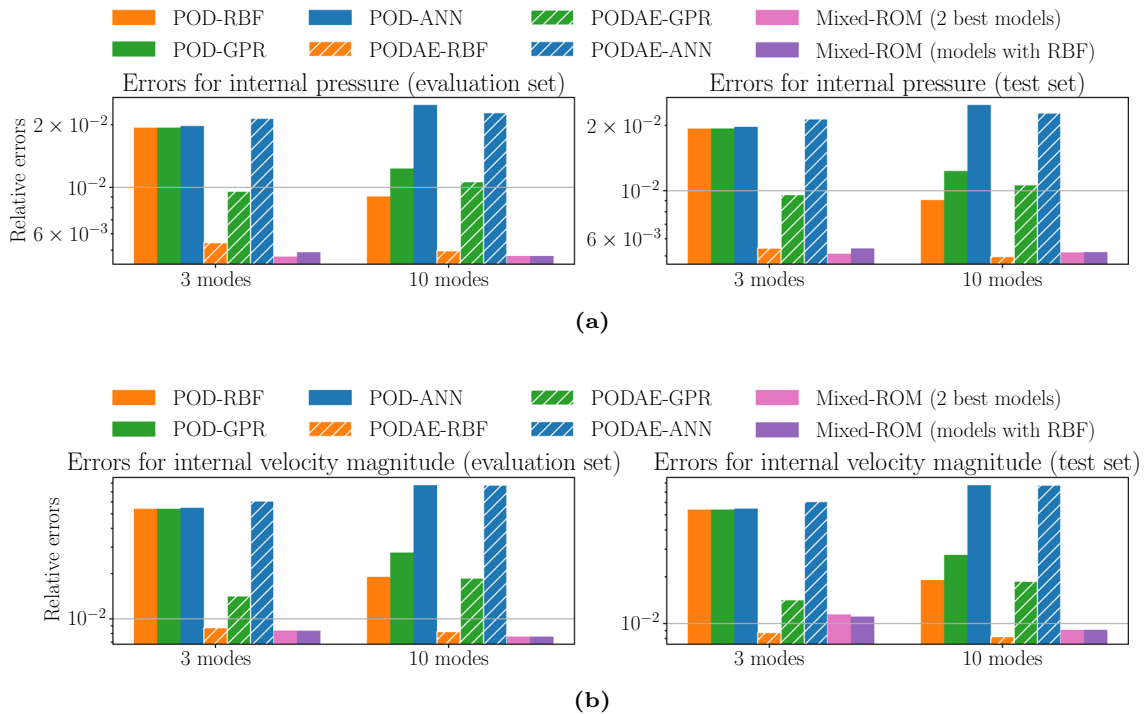

Fig. 24 Relative errors on evaluation and test set for the 2D fields in the internal mesh for all ROMs and for aggregated models
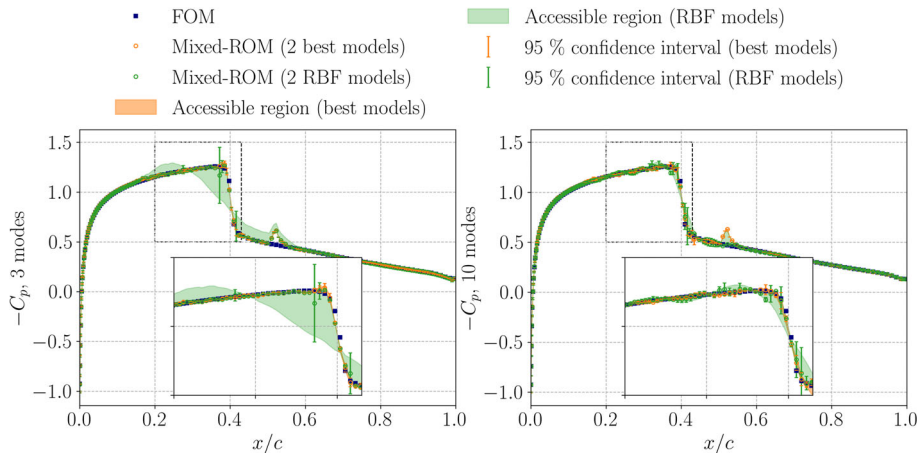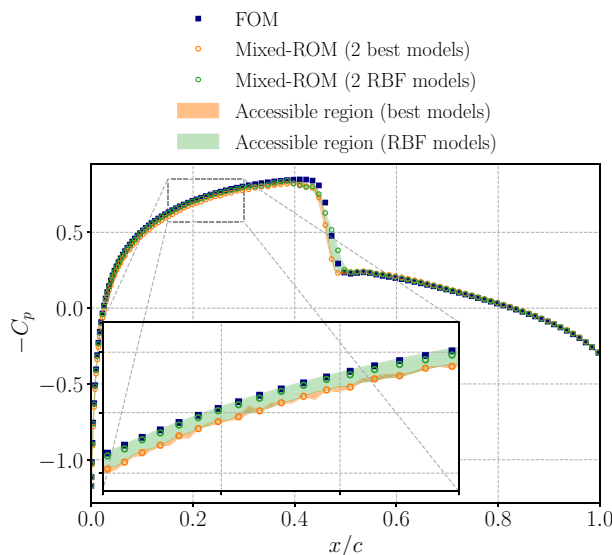
Fig. 25 Representation of the pressure coefficient C_p for mixture models and FOMs for a test parameter $\alpha = 4.7°$. We consider two different reduced dimensions, 3 (on the left) and 10 (on the right). The *accessible region*, namely the region accessed by the ROMs predictions considered in the aggregation, is also here represented. The mixed-ROMs correspond to the expected values of the aggregations $E[C_p]$, while the error bars correspond to interval $[E[C_p] - 2\sqrt{Var[C_p]}, E[C_p] + 2\sqrt{Var[C_p]}]$

Fig. 26 Representation of the pressure coefficient C_p for mixture models and FOMs for a test parameter $\alpha = 0.3°$. We consider as reduced dimension r = 10. The *accessible region*, namely the region accessed by the ROMs predictions considered in the aggregation, is also here represented

space of dimension 10, the prediction of the aggregated model is better when using the two models based on RBF interpolation rather than the two best models. In this latter case, in fact, the differences would be so small between the two best models, that there would be practically no benefit in their aggregation. Moreover, as done in Fig. 19, also in this case we included the error bars, quantifying the model uncertainty along the airfoil's chord for the test parameter considered. In general, the confidence intervals are very small and show that the aggregation algorithm prefers the best-performing model in almost all the space coordinates. The oscillation for $x/c \simeq 0.37$ in Fig. 25 (left, namely 3 modes) may be due to a not accurate weights' prediction, also reflected in the oscillating weights' behavior in Fig. 27a in the region near $x/c = 0.37$. The reason for the slightly oscillatory behavior of the error bars in the case of 10 modes (the right plot in Fig. 25) may be due to the consistently different behavior of the ROMs, as shown also in Fig. 27.

To better understand the interplay between the different models and how they work together, in Fig. 27 we show also the values of the weights of the individual ROMs. We can observe that, as one might expect, the intrinsic nonlinearity of the autoencoder helps in predicting correctly the sharp feature represented by the
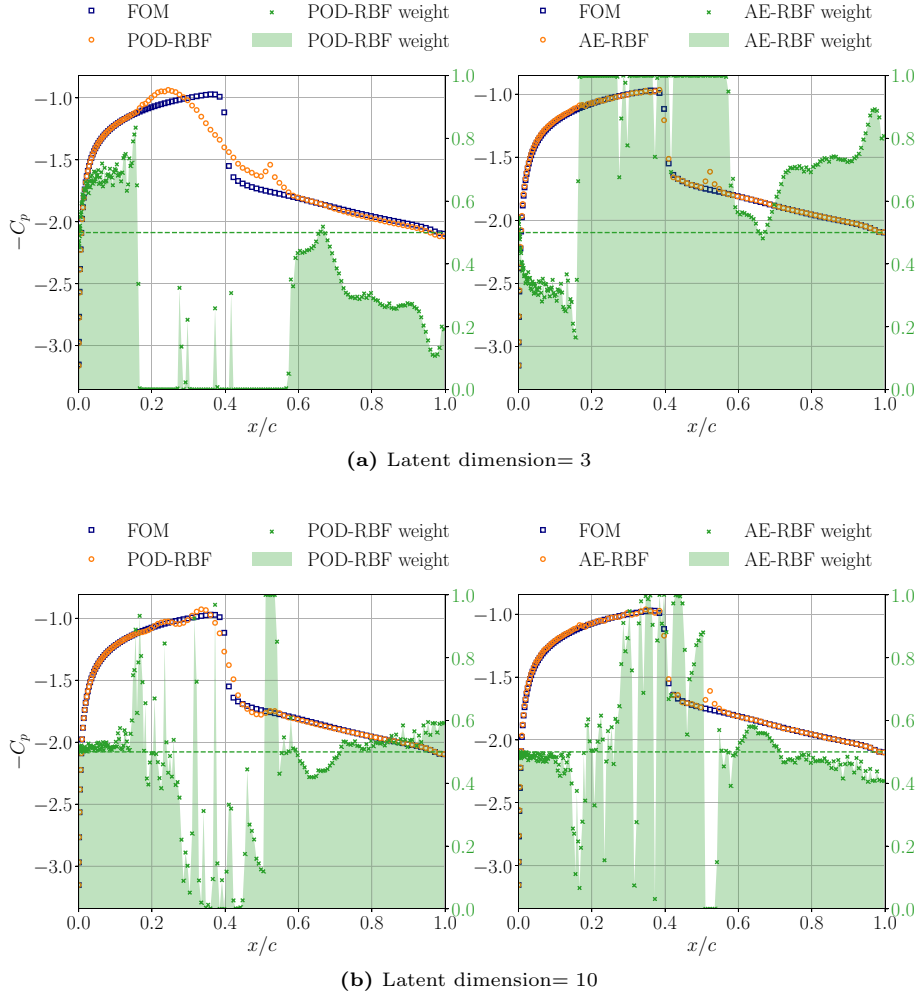
(a) Latent dimension= 3

(b) Latent dimension= 10

Fig. 27 Representation of the pressure coefficient C_p for two ROMs (POD-RBF and AE-RBF) and of the corresponding weights for the aggregation with only RBF models. We consider as test parameter $\alpha = 4.7°$ and two different reduced dimensions, 3 and 10

shock wave. This automatically implies a full activation of the AE at this location, whereas in other regions, such as in proximity of the leading edge, the POD is more active. This is more evident when the number of modes is small. In fact, in this scenario, the POD prediction does not provide a lot of information to properly reconstruct the manifold in the parametric space. By increasing the number of modes (right figure), we can still observe similar trends, although the POD is reasonably good in proximity of the shock and the activation of the AE is less evident.

The velocity magnitude on the surroundings of the airfoil is shown in Fig. 28. In this figure, the POD and POD-AE reductions are compared and aggregated together.

Both models provide a reasonably good approximation of the unseen parametric solution. However, if we observe more closely the weights of the two models (shown in Fig. 29), we can clearly observe that regions which are particularly challenging in terms of reduction are correctly detected in the spatial domain and the nonlinear technique based on PODAE is much more active with respect to the linear POD.

Figure 30 depicts the variance corresponding to the normalized velocity magnitude field for a test parameter. It is interesting to notice that the variance can be directly linked to the weights' space distribution. Indeed, the variance is larger in the regions where there is considerable discrepancy in the weights, namely where there is less consensus among the models. These zones also coincide with the wake and the region nearby the shock position, where the nonlinear phenomena are dominant.

If we increase the dimension of the latent space, it is even more interesting to notice that the wake behind the airfoil is not a challenging feature anymore and both models are correctly predicting its behavior. However,
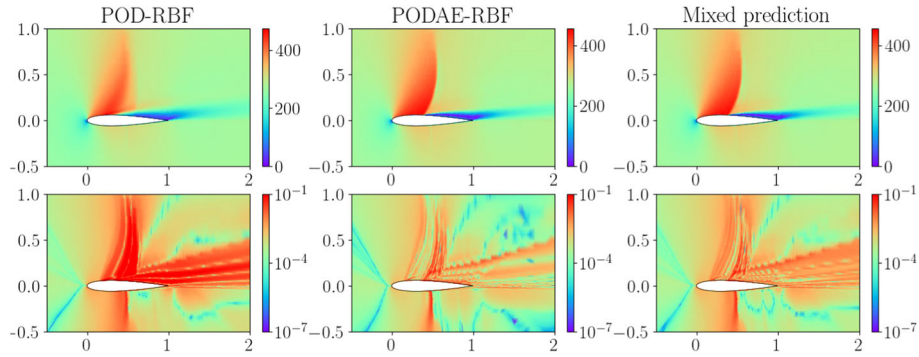
Fig. 28 Velocity magnitude predictions for two ROMs and for the aggregation model (first row), and corresponding absolute errors with respect to FOM (second row). The test parameter considered is $\alpha = 4.7°$, and the reduced dimension is 3
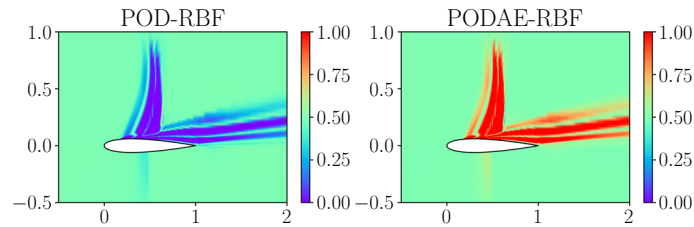

Fig. 29 Weights representation for the aggregation model when considering the POD-RBF and PODAE-RBF models. The test parameter considered is $\alpha = 4.7°$, and the reduced dimension is 3
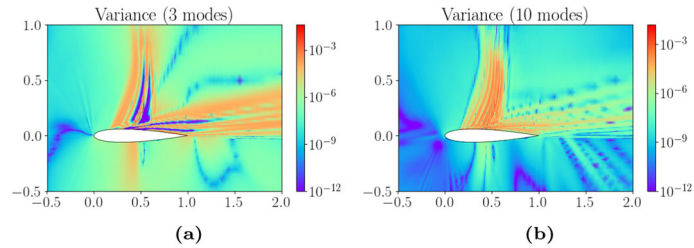

Fig. 30 Variance representation ($Var[v]$) for the normalized velocity magnitude field ($u/u_{\text{freestream}}$) for the test parameter $\alpha = 4.7°$. The figure shows the results for both 3 (left) and 10 modes (right)

at the shock, the PODAE is much more active than the POD. This observation is a further indication that even if turbulent boundary layers can be challenging in terms of reduction, sharp features moving significantly in the parametric space (such as shock-waves) are still the dominant bottleneck in terms of reduced-order modeling. This aspect reiterates how the second test case is more complex to reduce and represents a more challenging scenario for the aggregation of ROMs.

3.3.3 Computational time

For what concerns the computational time needed for the random forest regression, most of the execution time is due to the fitting process to the evaluation data. In particular, the execution time for fitting the regression is ~ 1 s for the 1D airfoil fields, and $\sim 40-45$ s for 2D fields, because of the larger space dimensionality.

The execution times for prediction are much smaller and comparable with the prediction times for the individual ROMs, i.e., $\sim 1 \times 10^{-2}$ s for the airfoil fields and $\sim 11 \times 10^{-1}$ s for the internal fields.

In order to summarize the efficiency of the proposed methodology in terms of computational times, we report in Table 3 the average speed-up of both the individual ROMs used in the aggregation, and of the aggregation itself. The speed-up values reported refer to the first test case, but the results are similar also for the second test case.

For the individual ROMs, the speed-up is evaluated starting from the sum of the training and online times, which is divided by the medium execution time of a single FOM simulation.

Table 3 Average speed-ups in terms of execution time of individual ROMs (POD-RBF and AE-RBF), with respect to one single FOM simulation

Field type	ROM speed-up		Mixed-ROM speed-up
	POD-RBF	AE(or PODAE)-RBF	
1D	$\sim 1 \times 10^7$	$\sim 8.5 \times 10^3$	$\sim 8 \times 10^3$
2D	$\sim 2 \times 10^5$	$\sim 1.4 \times 10^4$	$\sim 1.3 \times 10^4$

Both the offline training and the online evaluation are considered for computing the speed-ups. The data reported refer to test case 1, for both 1D fields acting on the airfoil, and 2D fields around the airfoil

For the aggregation, the speed-up is evaluated considering the sum of the training and online times of both ROMs and of the aggregation.

As can be seen from Table 3, the acceleration is considerably high both for the individual ROMs and for the mixed-ROM. However, it can be immediately noted that the aggregation model has a speed-up very close to the one of the most expensive individual ROMs, namely the AE-based approach. Hence, the aggregation strategy allows to retain the best prediction among the available ones, without increasing the computational effort. Even if one individual ROM outperforms the other in each region of the space, the mixture-of-model automatically detects the best model without any a-priori knowledge of the physical problem.

4 Conclusions and perspectives

In this manuscript, we proposed a strategy to enhance the performance of individual data-driven ROMs, which are previously trained in the so-called *training set*. The ROM methodology is extensively discussed in Sect. 2.

In particular, each individual ROM is mainly characterized by two parts, a *reduction* stage and an *approximation* step, which have been discussed in Sect. 2.1.

The space-dependent aggregation approach, described in Sect. 2.2, builds a convex combination of a set of alternative ROMs, with space-dependent weights evaluated on an *a-priori-selected evaluation* set. A machine learning technique (random forest regression) is then trained with the given weights' data, and used to predict the weights in a *test set*.

The model mixture, referred to as the *mixed-ROM*, is evaluated on two distinct test cases, both involving the flow past an airfoil but differing in the full-order models (FOMs) and parameters used: the Reynolds number in the first test case and the angle of attack in the second.

As detailed in Sect. 3, the proposed methodology significantly enhances the accuracy of individual reduced-order models (ROMs) in both test cases. It effectively identifies the ROM with the optimal performance across various regions of the spatial and parameter domains. Notably, in the more challenging second test case, the aggregation approach successfully and automatically detects the nonlinear reduction techniques in the vicinity of the shock position, demonstrating its robustness and efficacy.

Therefore, we can definitely say that it provides an automatic detection of the need between linear and nonlinear reduction without any a-priori knowledge of the problem of interest.

Moreover, the computational time needed for computing the aggregation is small if compared to the FOM simulation time. In particular, most of the execution time in the aggregation is employed for fitting the RF regression, but it takes only a few seconds, as pointed out in Sect. 3.3.3.

It is important to remark that the performance of the technique in the test set strongly depends on the type of regression used to predict the weights' space distributions and future work will be focused on testing the method with more advanced machine learning techniques for the weights' regression.

A possible extension is to change the features space in order to allow for different space coordinates, and hence, different geometries.

Another extension that the authors will address in the future is related to more complex test cases. This project is the first stage to show that the mixture-of-models is able to improve the individual ROMs' predictions. We included here the results on two 2D test cases, but with substantial differences in the physical setting (incompressible and transonic regimes). We plan to extend this investigation on more challenging and also 3D test cases in future research.

Moreover, a possible extension of this work is to enrich the methodology by building a *mixed-ROM* on the top of FOMs characterized by different turbulence models. In this case, a further layer of variability at the ROM level would be introduced by the different performances of the turbulence models at the FOM level.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

A Supplementary material

In this supplementary Section, we report the specifications of the hyperparameters of the techniques exploited in non-intrusive ROMs.

In particular, in the RBF interpolation, we consider a *thin plate spline* kernel, which reads $\varphi(r) = r^2 \log(r)$, and the degree of the added polynomial is 0.

In the GPR approximation, we consider a squared exponential form:

$$\mathcal{K}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \sigma^2 \exp\left(-\frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2}{2l}\right). \tag{13}$$

Once the hyperparameters σ and l of the covariance kernel are computed in order to fit our data, we can exploit the distribution (3) to evaluate $\mathcal{M}(\boldsymbol{\mu}^*)$ in (3) and predict the new modal coefficients. The hyperparameters are automatically fitted by the algorithm itself. For what concerns the neural networks' setting, both as reduction technique (AE or PODAE) and as approximation technique (ANN), the hyperparameters are reported in Table 4.

Table 4 Neural networks setting in ROMs

Network	Hidden layers	Nonlinearity	Learning rate	Stop criteria		Weight decay
				Epochs	Final loss	
ANN	[20, 20, 20]	Softplus	5×10^{-3}	100,000	1×10^{-4}	1×10^{-7}
AE	[50, 20, r, r, 20, 50]	Softplus	5×10^{-4}	20,000	5×10^{-6}	0
AE (PODAE)	[r_{med}, 50, 20, r, r, 20, 50, r_{med}]	Softplus	5×10^{-4}	20,000	5×10^{-6}	0

References

1. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**(1), 539–575 (1993)
2. Bhattacharya, K., Hosseini, B., Kovachki, N.B., et al.: Model reduction and neural networks for parametric pdes. SMAI J. Comput. Math. **7**, 121–157 (2021)

3. Bonnet, F., Mazari, J., Cinnella, P., et al.: Airfrans: high fidelity computational fluid dynamics dataset for approximating Reynolds-averaged Navier–Stokes solutions. Adv. Neural. Inf. Process. Syst. **35**, 23463–23478 (2022)
4. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
5. Chatterjee A.: An introduction to the proper orthogonal decomposition. Current Science, 808–817 (2000)
6. Cherroud S, Merle X, Cinnella P, et al. (2023) Space-dependent aggregation of data-driven turbulence models. arXiv:2306.16996
7. Demo, N., Ortali, G., Gustin, G., et al.: An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques. Bollettino dell'Unione Matematica Italiana **14**, 211–230 (2021)
8. Deswarte, R., Gervais, V., Stoltz, G., et al.: Sequential model aggregation for production forecasting. Comput. Geosci. **23**(5), 1107–1124 (2019)
9. Deswarte, R., Gervais, V., Stoltz, G., et al.: Sequential model aggregation for production forecasting. Comput. Geosci. **23**, 1107–1124 (2019)
10. Devaine, M., Gaillard, P., Goude, Y., et al.: Forecasting electricity consumption by aggregating specialized experts: a review of the sequential aggregation of specialized experts, with an application to Slovakian and French country-wide one-day-ahead (half-)hourly predictions. Mach. Learn. **90**(2), 231–260 (2013)
11. Eivazi H., Veisi H., Naderi M.H., et al.: Deep neural networks for nonlinear model order reduction of unsteady flows. Physics and Fluids, 32(10) (2020)
12. Garcia-Uceda Juarez, A., Raimo, A., Shapiro, E., et al.: Steady turbulent flow computations using a low mach fully compressible scheme. AIAA J. **52**(11), 2559–2575 (2014)
13. Goc K, Bose S, Moin P (2020) Wall-modeled large eddy simulation of an aircraft in landing configuration. In: AIAA Aviation 2020 Forum, p. 3002
14. Goc, K.A., Lehmkuhl, O., Park, G.I., et al.: Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. Flow **1**, E14 (2021)
15. Halder, R., Fidkowski, K.J., Maki, K.J.: Non-intrusive reduced-order modeling using convolutional autoencoders. Int. J. Numer. Meth. Eng. **123**(21), 5369–5390 (2022)
16. Hesthaven, J.S., Ubbiali, S.: Non-intrusive reduced order modeling of nonlinear problems using neural networks. J. Comput. Phys. **363**, 55–78 (2018)
17. Hijazi S, Ali S, Stabile G, et al (2020a) The effort of increasing Reynolds number in projection-based reduced order methods: from laminar to turbulent flows. Numerical Methods for Flows: FEF 2017 Selected Contributions pp. 245–264
18. Hijazi, S., Stabile, G., Mola, A., et al.: Data-driven POD-Galerkin reduced order model for turbulent flows. J. Comput. Phys. **416**, 109513 (2020)
19. Iorio, M.C., Gonzalez, L.M., Ferrer, E.: Direct and adjoint global stability analysis of turbulent transonic flows over a naca0012 profile. Int. J. Numer. Meth. Fluids **76**(3), 147–168 (2014)
20. Iovnovich, M., Raveh, D.E.: Reynolds-averaged Navier–Stokes study of the shock-buffet instability mechanism. AIAA J. **50**(4), 880–890 (2012)
21. Ivagnes, A., Demo, N., Rozza, G.: Towards a machine learning pipeline in reduced order modelling for inverse problems: neural networks for boundary parametrization, dimensionality reduction and solution manifold approximation. J. Sci. Comput. **95**(1), 23 (2023)
22. Ivagnes, A., Demo, N., Rozza, G.: A shape optimization pipeline for marine propellers by means of reduced order modeling techniques. Int. J. Numer. Methods Eng. **125**, e7426 (2024)
23. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., et al.: Adaptive mixtures of local experts. Neural Comput. **3**, 79–87 (1991)
24. Jasak, H.: OpenFOAM: open source CFD in research and industry. Int. J. Naval Archit. Ocean Eng. **1**(2), 89–94 (2009)
25. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Comput. **6**(2), 181–214 (1994)
26. Kadeethum, T., Ballarin, F., Choi, Y., et al.: Non-intrusive reduced order modeling of natural convection in porous media using convolutional autoencoders: comparison with linear subspace techniques. Adv. Water Resour. **160**, 104098 (2022)
27. Kerschen, G., Jc, Golinval, Vakakis, A.F., et al.: The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. Nonlinear Dyn. **41**, 147–169 (2005)
28. Klema, V., Laub, A.: The singular value decomposition: its computation and some applications. IEEE Trans. Autom. Control **25**(2), 164–176 (1980)
29. Lee, K., Carlberg, K.T.: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J. Comput. Phys. **404**, 108973 (2020)
30. Liaw, A., Wiener, M., et al.: Classification and regression by random Forest. R News **2**(3), 18–22 (2002)
31. McDevitt JB, Okuno AF (1985) Static and dynamic pressure measurements on a NACA 0012 airfoil in the Ames high Reynolds number facility. Tech. rep
32. Menter, F.R.: Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J. **32**(8), 1598–1605 (1994). https://doi.org/10.2514/3.12149
33. Nguyen N, Persson PO, Peraire J (2007) RANS solutions using high order discontinuous Galerkin methods. In: 45th AIAA Aerospace Sciences Meeting and Exhibit, p. 914
34. Patankar, S.: Numerical heat transfer and fluid flow. Taylor & Francis, Abingdon-on-Thames (2018)
35. Patankar SV, Spalding DB (1983) A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In: Numerical prediction of flow, heat transfer, turbulence and combustion. Elsevier, pp. 54–73
36. Powell M.J. Radial basis functions for multivariable interpolation: a review. Algorithms for approximation, 143–167 (1987)
37. Rahman, S.M., Pawar, S., San, O., et al.: Nonintrusive reduced order modeling framework for quasigeostrophic turbulence. Phys. Rev. E **100**(5), 053306 (2019)
38. Reynolds, O.: IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. Philos. Trans. R. Soc. Lond. **186**, 123–164 (1895). https://doi.org/10.1098/rsta.1895.0004

39. Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., et al.: Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines. Ore Geol. Rev. **71**, 804–818 (2015)
40. Romor F, Stabile G, Rozza G (2022) Non-linear manifold ROM with convolutional autoencoders and reduced over-collocation method. arXiv:2203.00360
41. Salvador, M., Dede, L., Manzoni, A.: Non intrusive reduced order modeling of parametrized PDEs by kernel POD and neural networks. Comput. Math. Appl. **104**, 1–13 (2021)
42. San, O., Maulik, R., Ahmed, M.: An artificial neural network framework for reduced order modeling of transient flows. Commun. Nonlinear Sci. Numer. Simul. **77**, 271–287 (2019)
43. Slotnick JP, Khodadoust A, Alonso J, et al. (2014) CFD vision 2030 study: a path to revolutionary computational aerosciences. Tech. rep
44. Stoltz, G.: Agrégation séquentielle de prédicteurs:méthodologie générale et applications à la prévision de la qualité de l'air et à celle de la consommation électrique. J. de la Société Française de Stat. **151**(2), 41 (2010)
45. Tabatabaei, N., Hajipour, M., Mallor, F., et al.: RANS modelling of a NACA4412 wake using wind tunnel measurements. Fluids **7**(5), 153 (2022)
46. Tabatabaei, N., Vinuesa, R., Örlü, R., et al.: Techniques for turbulence tripping of boundary layers in rans simulations. Flow Turbul. Combust. **108**(3), 661–682 (2022)
47. Tezzele, M., Salmoiraghi, F., Mola, A., et al.: Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems. Adv. Model. Simul. Eng. Sci. **5**, 1–19 (2018)
48. Thomas, P.V., ElSayed, M.S., Walch, D.: Review of model order reduction methods and their applications in aeroelasticity loads analysis for design optimization of complex airframes. J. Aerosp. Eng. **32**(2), 04018156 (2019)
49. Tober H., Hänninger E.:Evaluation of OpenFOAM performance for rRANS simulations of flow around a NACA 4412 airfoil, Student Bachelor Thesis (2018)
50. Vinuesa, R., Hosseini, S.M., Hanifi, A., et al.: Pressure-gradient turbulent boundary layers developing around a wing section. Flow Turbul. Combust. **99**, 613–641 (2017)
51. Wall ME, Rechtsteiner A, Rocha LM (2003) Singular value decomposition and principal component analysis. In: A practical approach to microarray data analysis. Springer, pp. 91–109
52. Williams C., Rasmussen C.: Gaussian processes for regression. Advances in neural informatio processing systems, 8 (1995)
53. Xiao, D., Heaney, C., Fang, F., et al.: A domain decomposition non-intrusive reduced order model for turbulent flows. Computers and Fluids 182, 15–27 (2019)
54. Yu, J., Yan, C., Jiang, Z., et al.: Adaptive non-intrusive reduced order modeling for compressible flows. J. Comput. Phys. **397**, 108855 (2019)
55. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE Trans. Neural Netw. Learn. Syst. **23**(8), 1177–1193 (2012)
56. Zancanaro, M., Mrosek, M., Stabile, G., et al.: Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. Fluids **6**(8), 296 (2021)
57. Zancanaro M, Hijazi S, Girfoglio M, et al. (2022a) Finite volume-based reduced order models for turbulent flows. In: Advanced reduced order methods and applications in computational fluid dynamics. SIAM-Society for Industrial and Applied Mathematics, pp. 165–202
58. Zancanaro M, Stabile G, Rozza G (2022b) A segregated reduced order model of a pressure-based solver for turbulent compressible flows. arXiv:2205.09396
59. de Zordo-Banliat, M., Dergham, G., Merle, X., et al.: Space-dependent turbulence model aggregation using machine learning. J. Comput. Phys. **497**, 112628 (2024)