


RESEARCH ARTICLE

Open Access



A segregated reduced-order model of a pressure-based solver for turbulent compressible flows

Matteo Zancanaro^{1†}, Valentin Nkana Ngan^{1†}, Giovanni Stabile^{2†} and Gianluigi Rozza^{1*†} 

[†]Matteo Zancanaro, Valentin Nkana Ngan, Giovanni Stabile and Gianluigi Rozza contributed equally to this work.

*Correspondence: grozza@sissa.it

¹Mathematics Area, mathLab, SISSA, via Bonomea 265, I-34136 Trieste, Italy

²The Biorobotics Institute, Sant'Anna School of Advanced Studies, Viale Rinaldo Piaggio 34, 56025 Pontedera, Pisa, Italy

Abstract

This article provides a reduced-order modelling framework for turbulent compressible flows discretized by the use of finite volume approaches. The basic idea behind this work is the construction of a reduced-order model capable of providing closely accurate solutions with respect to the high fidelity flow fields. Full-order solutions are often obtained through the use of segregated solvers (*solution variables are solved one after another*), employing slightly modified conservation laws so that they can be decoupled and then solved one at a time. Classical reduction architectures, on the contrary, rely on the Galerkin projection of a complete Navier–Stokes system to be projected all at once, causing a mild discrepancy with the high order solutions. This article relies on segregated reduced-order algorithms for the resolution of turbulent and compressible flows in the context of physical and geometrical parameters. At the full-order level turbulence is modeled using an eddy viscosity approach. Since there is a variety of different turbulence models for the approximation of this supplementary viscosity, one of the aims of this work is to provide a reduced-order model which is independent on this selection. This goal is reached by the application of hybrid methods where Navier–Stokes equations are projected in a standard way while the viscosity field is approximated by the use of data-driven interpolation methods or by the evaluation of a properly trained neural network. By exploiting the aforementioned expedients it is possible to predict accurate solutions with respect to the full-order problems characterized by high Reynolds numbers and elevated Mach numbers.

Keywords: Compressible flows, Reduced-order model, SIMPLE algorithm, Turbulence, Neural networks

Introduction

In the last decades, fluid flow simulations have progressively enlarged their applicability and their influence in many different research fields (general overviews can be found in [1–3]). Nowadays, applications of computational fluid dynamics (CFD) have reached widespread application areas such as, shape optimization for naval/automotive/aerospace engineering [4,5], cardiovascular in real time surgery [6], chemistry industrial processes [7,8] or weather forecasts [9]. As the demand for usability and reliability in CFD methods

increases, current hardware designs are becoming insufficient to meet the computational requirements in a timely manner. As a result, a considerable amount of CFD research is focused on developing new, efficient techniques to reduce computation times. This challenge frequently arises in applications such as shape optimization problems, uncertainty quantification studies, and optimal control frameworks.

To address this issue, various strategies have been explored recently. One of such approach is Galerkin projection, which has been widely applied to develop new reduction techniques that offer efficient, accurate, and more cost-effective solutions for varying parameter selections. This method leverages information from only a few full-order solutions across different parameter values. In fact, in a reduced-order model (ROM) framework, parameters and parametric studies are central to understanding the influence of different variables on the system's behavior while maintaining the computational efficiency offered by ROMs. A reduced-order model is a simplified representation of a complex, high-dimensional system, designed to capture the essential behavior and dynamics of the system with significantly reduced computational effort. By focusing on the most important modes or features of the full-order model, ROMs provide fast and efficient simulations while maintaining accuracy within an acceptable range for various analyses. The most commonly used technique in ROMs is Galerkin projection. In Galerkin projection, the original high-dimensional problem is projected onto a lower-dimensional subspace. This significantly reduces computational complexity while preserving the system's dominant behavior [10]. For fluid flow applications, relevant studies can be found in [11–15]. There are numerous options for leveraging the dynamic content included in high-fidelity systems. The most commonly used one is the Proper Orthogonal Decomposition (POD) [16–21], the Proper Generalized Decomposition (PGD) [22,23], the Dynamic Mode Decomposition (DMD) [24,25]. The initial concept of the POD, which was first developed in the domain of fluid dynamics to examine turbulence, is to decompose a vector field into a series of deterministic spatial functions weighted by time/parameter coefficients.

In recent years, while the concept of machine learning (ML) is not new, its applications within the fluid dynamics community have significantly expanded. This growth is driven by advancements in algorithms, increased computational power, more affordable memory, and the availability of vast amounts of data. As a result, ML has emerged as a prominent area of research in the field. Leveraging ML algorithms has made solving complex, non-linear parametric partial differential equations (PDEs) more efficient and accessible than ever before. Numerous approaches for applying ML to CFD problems have been investigated in the literature, as demonstrated in several studies, for instance in [26–37] to name few. For example, the combination of the POD and neural networks (NNs) has been applied across a wide range of cases, including the non-linear Poisson equation in one and two spatial dimensions, as well as two-dimensional cavity flows governed by the steady incompressible Navier–Stokes equations. Both POD-based projection techniques and ML methods, offer valuable insights but also have their limitations. Projection techniques are closely aligned with the physical laws of the problem, using modal basis functions derived from real solutions to capture the primary dynamics. These modes are then used to project and reconstruct the solutions of conservation equations on reduced manifolds. However, challenges arise in dealing with non-linearity and the non-affine nature of parameterized formulations, which can complicate their application. Additionally, projection methods

may not be feasible if the governing equations are inaccessible, such as in commercial software where the underlying laws are not fully disclosed. In contrast, ML techniques offer greater versatility. They require only a set of pre-trained solutions and are independent of the complexity of the problem's mathematical formulation. These methods are designed to produce accurate approximations rapidly. However, a significant drawback is their weaker connection to the underlying physics, making it difficult to interpret the individual components of the ML architecture in terms of physical phenomena. For this reason, they may give inaccurate results thanks to impossibility in having a deeper check on networks responses.

Taking all the aforementioned examinations under consideration, this work applies the new mixed technique [26] on compressible Navier–Stokes problems, which is capable of merging the advantages of projection techniques together with data-driven architectures. In particular, in our approach, classical projection methods are used for the Favre Averaged Navier Stokes (FANS) equations while a neural network gets trained to provide the eddy viscosity solutions in a turbulence modelling approach. These new contributions result to a reduced-order models that are independent of the selection of turbulence models for any segregated solvers for compressible flows capable to reduce the computational cost associated with fluid flow problems characterized by high Reynolds numbers and elevated Mach numbers. The main goal is to propose an architecture proficient in dealing with different types of parametrizations for compressible flows. Moreover, one of the most relevant focuses concerning this work is constituted by a coherent approach between full-order and reduced-order solutions, by developing a new reduced compressible SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.

This manuscript is structured as follows: The section “[The compressible Navier–Stokes equations](#)” presents the equations used in this work; subsection “[Proper Orthogonal Decomposition procedure](#)” explains the POD procedure employed to obtain the modal basis functions. In subsection “[Reduced-SIMPLE algorithm for compressible flows](#)” the core algorithm used for our technique is introduced together with subsection “[Turbulence treatment](#)” where the AI architecture for turbulence treatment is shown. Two different test cases, a physically parameterized and a geometrically parameterized ones, are exposed in subsection “[Physical parametrization test case](#)” and subsection “[Geometrical parametrization test case](#)” respectively. Finally, in subsection “[Conclusions and future perspectives](#)”, few considerations on the results and some possible developments for this work are presented.

The compressible Navier–Stokes equations

In this work, we want to deal with parameterized compressible Navier-Stokes equations problems. To manage the compressibility of the fluid, we selected a common strategy for this kind of applications: the Favre averaging. The equations describing the physics are

the following ones:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \text{ in } \Omega(\pi), \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot [\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I} - \boldsymbol{\tau}] = 0 \text{ in } \Omega(\pi), \quad (2)$$

$$\frac{\partial \rho e_0}{\partial t} + \nabla \cdot [\rho \mathbf{u} e_0 + p \mathbf{u} - \mathbf{u} \cdot \boldsymbol{\tau} + \mathbf{q}] = 0 \text{ in } \Omega(\pi), \quad (3)$$

$$\mathbf{u} = \mathbf{g}_D \text{ in } \Gamma_D, \quad (4)$$

$$\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g}_N \text{ in } \Gamma_N, \quad (5)$$

where ρ represent the density, \mathbf{u} the flow velocity, p the pressure, $\boldsymbol{\tau}$ the viscous stress tensor, e_0 the total energy, and \mathbf{I} the identity tensor. The boundary conditions include Γ_D , where Dirichlet conditions \mathbf{g}_D are applied, and Γ_N , where Neumann conditions \mathbf{g}_N are imposed. Here, ν refers to the kinematic viscosity, and \mathbf{n} is the unit normal vector. The computational domain is denoted by $\Omega(\pi)$, which, in cases of geometric parametrization, can depend explicitly on the parameter π . The heat-flux \mathbf{q} is given by Fourier's law:

$$\mathbf{q} = \lambda \nabla T \equiv C_p \frac{\mu}{Pr} \nabla T, \quad (6)$$

with the laminar Prandtl number Pr is given by $Pr = \frac{C_p \mu}{\lambda}$. To close these equations, it is also necessary to specify an equation of state. Assuming air to be an ideal gas, the following relations are valid:

$$\gamma \equiv C_p / C_v, \quad p = \rho R T, \quad e = C_v T, \quad C_p - C_v = R.$$

Being R the gas constant, C_v is the constant volume, and C_p means specific heat at constant pressure, γ is the adiabatic index, e the internal energy, and T the temperature. In the Favre Averaged Navier–Stokes (FANS) equations, all the variables (density ρ , pressure p , velocity \mathbf{u} , total energy e_0 , temperature T and internal energy e) are decomposed in an averaged part and a fluctuating one as follows:

$$\rho = \bar{\rho} + \rho', \quad p = \bar{p} + p', \quad T = \bar{T} + T'' \quad (7)$$

$$e_0 = \bar{e}_0 + e_0'', \quad \mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'', \quad e = \bar{e} + e''. \quad (8)$$

Superscript $\bar{\square}$ indicates the Favre averaging which correspond to a density weighted Reynolds averaging $\bar{\square}$. Given a certain variable $\Phi(t)$, we have:

$$\bar{\Phi} = \frac{1}{T} \int_T \Phi(t) dt \Rightarrow \Phi' = \Phi - \bar{\Phi} \quad (9)$$

$$\bar{\Phi} = \frac{\bar{\rho} \Phi}{\bar{\rho}} \Rightarrow \Phi'' = \Phi - \bar{\Phi}. \quad (10)$$

Plugging Eq. 7, Eq. 8, Eq. 9 and Eq. 10 in Eq. 1, Eq. 2, Eq. 3 lead to:

$$\frac{\partial \bar{\rho}}{\partial t} + \nabla \cdot (\bar{\rho} \bar{\mathbf{u}}) = 0 \text{ in } \Omega(\pi), \quad (11)$$

$$\frac{\partial \bar{\rho} \bar{\mathbf{u}}}{\partial t} + \nabla \cdot [\bar{\rho} \bar{\mathbf{u}} \otimes \bar{\mathbf{u}} - \bar{\boldsymbol{\tau}}_{turb} - \bar{\boldsymbol{\tau}} + \bar{p} \mathbf{I}] = 0 \text{ in } \Omega(\pi), \quad (12)$$

$$\frac{\partial \bar{\rho} \bar{e}_0}{\partial t} + \nabla \cdot \left[\bar{\rho} \bar{\mathbf{u}} \bar{e}_0 - C_p \frac{\alpha_{eff}}{\gamma} \nabla \bar{T} \right] + \nabla \cdot [\bar{p} \bar{\mathbf{u}} - \bar{\mathbf{u}} \cdot \bar{\boldsymbol{\tau}} - \bar{\mathbf{u}} \cdot \bar{\boldsymbol{\tau}}_{turb}] = 0 \text{ in } \Omega(\pi), \quad (13)$$

$$\bar{\mathbf{u}} = \mathbf{g}_D \text{ in } \Gamma_D, \quad (14)$$

$$\nu \frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{n}} - \bar{p} \mathbf{n} = \mathbf{g}_N \text{ in } \Gamma_N, \quad (15)$$

with $\alpha_{eff} = \gamma \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right)$, and where \bar{p} , $\tilde{\mathbf{u}}$ and \tilde{e} become the unknowns of the problem. μ is the dynamic viscosity, μ_t is the eddy viscosity owing to turbulence, Pr indicates the Prandtl number and Pr_t its turbulent counterpart which is a constant value. The molecular $\tilde{\boldsymbol{\tau}}$ and Reynolds-Stress $\tilde{\boldsymbol{\tau}}_{turb}$ tensors are given by:

$$\tilde{\boldsymbol{\tau}} = 2\mu\tilde{\boldsymbol{S}}, \quad \tilde{\boldsymbol{\tau}}_{turb} = 2\mu_t\tilde{\boldsymbol{S}} - \frac{2}{3}\bar{\rho}k\mathbf{I}, \quad (16)$$

where $\tilde{\boldsymbol{S}} = \frac{\nabla\tilde{\mathbf{u}} + \nabla\tilde{\mathbf{u}}^T}{2} - \frac{1}{3}\nabla \cdot \tilde{\mathbf{u}}\mathbf{I}$, and $k = \frac{\tilde{\mathbf{u}}' \cdot \tilde{\mathbf{u}}''}{2}$. Moreover, the density averaged total energy \tilde{e}_0 is rewritten in the internal energy form:

$$\tilde{e}_0 = \tilde{e} + \frac{\tilde{\mathbf{u}} \cdot \tilde{\mathbf{u}}}{2} + k, \quad (17)$$

Eq. 12, and Eq. 13 are obtained after some approximations and assumptions from an eddy viscosity point of view. The reader interested in the averaging procedure and modelling should refer to [38].

From now on, Eq. 11, Eq. 12, and Eq. 13 will be considered only in its steady-state formulation. All the averaged variables are dependent on the parameter π but, for the sake of simplicity, the following notation will be used:

$$\bar{\rho} = \bar{\rho}(\pi), \quad \bar{p} = \bar{p}(\pi), \quad \tilde{\mathbf{u}} = \tilde{\mathbf{u}}(\pi), \quad \tilde{T} = \tilde{T}(\pi), \quad \tilde{e} = \tilde{e}(\pi).$$

In the energy equation, the viscous terms are neglected in many solvers. This, can be reasonably true if compared with the other terms present into the energy equation. Moreover, the turbulent kinetic energy is neglected in the total energy. This results in the following system:

$$\nabla \cdot (\bar{\rho}\tilde{\mathbf{u}}) = 0 \quad \text{in } \Omega(\pi), \quad (18)$$

$$\nabla \cdot \left[\bar{\rho}\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}} - \mu_{eff} \left(\nabla\tilde{\mathbf{u}} + \nabla\tilde{\mathbf{u}}^T - \frac{2}{3}\nabla \cdot \tilde{\mathbf{u}}\mathbf{I} \right) + \bar{p}\mathbf{I} \right] = 0 \quad \text{in } \Omega(\pi), \quad (19)$$

$$\nabla \cdot \left[\bar{\rho}\tilde{\mathbf{u}} \left(\tilde{e} + \frac{\tilde{\mathbf{u}} \cdot \tilde{\mathbf{u}}}{2} \right) - \alpha_{eff} \nabla\tilde{e} + \bar{p}\tilde{\mathbf{u}} \right] = 0 \quad \text{in } \Omega(\pi), \quad (20)$$

$$\tilde{\mathbf{u}} = \mathbf{g}_D \quad \text{in } \Gamma_D, \quad (21)$$

$$v \frac{\partial \tilde{\mathbf{u}}}{\partial \mathbf{n}} - \bar{p}\mathbf{n} = \mathbf{g}_N \quad \text{in } \Gamma_N. \quad (22)$$

With $\mu_{eff} = \mu + \mu_t$, and $\alpha_{eff} = \gamma \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right)$. It is now clear that all the turbulence-related terms of the equations rely on μ_t to be calculated. For this reason, since only the eddy viscosity is required, a common 2-equations turbulent model as, e.g., $k - \epsilon$ or $k - \omega$ [38], is sufficient as a closure for the problem.

Reduced-order modelling architecture

This part of the manuscript presents the POD in a nutshell, after that, addresses the reduced-SIMPLE algorithm and finally the treatment of the eddy viscosity when dealing with turbulence flows.

Proper orthogonal decomposition procedure

The scope of this work is to find an efficient and reliable reduced-order model to be able to solve the full-order model for many different values of the parameter π without solving the Finite Volume discretized equations every time from scratch. For this reason, we developed a new procedure based on a POD-Galerkin scheme.

The whole machinery is divided in two main steps: an offline phase which consists on the resolution of a certain number N_π of full-order solutions, trying to extract as much information as possible from this set, and an online phase consisting on the resolution of a dimensionally reduced problem for all the different needed parametric configurations. What is new in this method is to be capable of resulting as general as possible with respect to the selected full-order turbulence model and, at the same time, as coherent as possible with respect to high fidelity solutions.

Let $\mathbb{P} = \{\pi_1, \dots, \pi_{N_\pi}\}$ be the training parameters set. For every parameter $\pi_i \in \mathbb{P}$, the full-order problem can be solved to obtain the corresponding solution s_i . All these offline solutions are then stored in the snapshots matrix:

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{21} & \dots & s_{N_\pi 1} \\ \vdots & \vdots & \vdots & \vdots \\ s_{1N_h} & s_{2N_h} & \dots & s_{N_\pi N_h} \end{bmatrix}.$$

In our case we want to construct an online solver able to mimic the offline convergence dynamics. For this reason the use of a monolithic (*non-segregated*) approach for the reduced problem is not a good choice as the *offline* solutions are obtained relying on a segregated solver; also at the *online* level a segregated strategy has to be applied to obtain solutions which are as consistent as possible. For a discussion on a similar consistent approach in the context of explicit time integration schemes, the reader is referred to [39]. To obtain an algorithm able to properly follow the behaviour of the high fidelity algorithm, the set of snapshots is enriched by adding a certain amount of intermediate solutions s_i^j obtained during the offline iterations. The distance between exported intermediate solutions is set to Δ as shown in Fig. 1. Since the solution fields during these iterations vary a lot, from the first attempt for the variables to the last resolution, the information contained into the converged snapshots is not sufficient to ensure the correct reduced reconstruction of the path to the global minimum for Eq. 1, Eq. 2, Eq. 3. By adding some non-physical solutions to the snapshots' matrix, which is what is happening by inserting non-converged fields, we are somehow polluting the physical content, but the convergence properties of the algorithm are *quite acceptable* in any case. To reach a balance between convergence and reliability, Δ can be varied and the total amount N_{int} of selected intermediate solutions can be modified. The new snapshots matrix then reads:

$$\mathbf{S} = \left[s_1^1, s_1^2, \dots, s_1^{N_{int}}, s_1, \dots, s_{N_\pi}^1, s_{N_\pi}^2, \dots, s_{N_\pi}^{N_{int}}, s_{N_\pi} \right],$$

where s_i^j is the solution obtained at the $(j \Delta)$ -th iteration for the i -th offline parameter.

In a POD-Galerkin approach, the reduced order solution s^r is obtained as a linear combination of some pre-calculated basis functions ξ :

$$s^r(\mathbf{x}, \pi) = \sum_{i=1}^{N_r} \beta_i(\pi) \xi_i(\mathbf{x}), \quad (23)$$

where $N_r < N_\pi$ is the number of basis functions to be used for the reconstruction and the β_i are the coefficients depending only on the parameter representing the reduced solution.

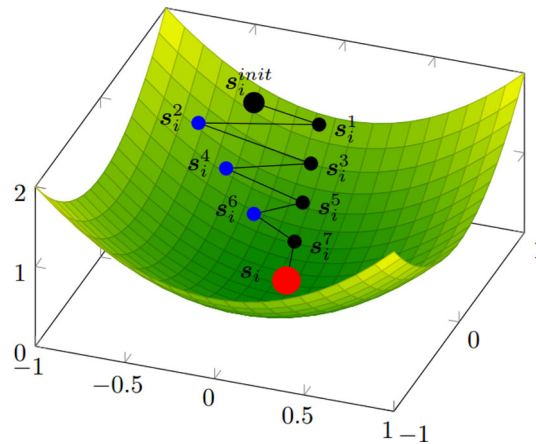


Fig. 1 Scheme of the snapshots selection for $\Delta = 2$: black dots are discarded intermediate solutions, blue dots are saved intermediate solutions while the red dot represents the final solution

Once provided a certain amount N_t of high fidelity solutions, with $N_t > N_\pi$ because of the intermediate snapshots, the best reduced order model we can get is the one able to fully reproduce the training offline solutions with no error with respect to it. Of course this is not achievable but we would like the L^2 norm of the error E_{ROM} between all the offline solutions and the respective online ones to be as low as possible:

$$E_{ROM} = \sum_{i=1}^{N_t} \|s_i^{ROM} - s_i\|_{L^2} = \sum_{i=1}^{N_t} \left\| \sum_{j=1}^{N_r} \beta_j(\pi) \xi_j(\mathbf{x}) - s_i \right\|_{L^2}.$$

It is well known (see, e.g., [19]) that the basis functions best performing in this sense are the ones obtained through a Proper Orthogonal Decomposition (POD) applied to the snapshots matrix \mathcal{S} . The eigen problem

$$\mathbf{C}\mathbf{V} = \mathbf{V}\boldsymbol{\lambda},$$

has to be resolved, where $\mathbf{C} \in \mathbb{R}^{N_t \times N_t}$ is the correlation matrix containing all the inner products in the form $(s_i, s_j)_{L^2(\Omega)}$. $\mathbf{V} \in \mathbb{R}^{N_t \times N_t}$ is the matrix containing its eigenvectors while $\boldsymbol{\lambda}$ is the diagonal matrix containing the eigenvalues.

The basis functions are then constructed as just a linear combination of the snapshots contained in \mathcal{S} :

$$\xi_i(\mathbf{x}) = \frac{1}{N_t \sqrt{\lambda_i}} \sum_{j=1}^{N_t} V_{ji} s_j(\mathbf{x}).$$

The basis functions matrix is then defined as:

$$\boldsymbol{\Xi} = [\xi_1, \dots, \xi_{N_r}] \in \mathcal{R}^{N_h \times N_r}.$$

The interested reader may refer to [21,40,41] for a detailed explanation regarding POD approaches.

Reduced-SIMPLE algorithm for compressible flows

In this paragraph, we will discuss the reduced algorithm developed for the resolution of compressible flows where no discontinuities are present. This means it is only suited for subsonic cases where the Mach number is lower than 1 in all the points of the domain. In

this paragraph, the reduced algorithm is based on the SIMPLE algorithm. In the following, it is assumed that all the equations: continuity, momentum and energy equations are discretized and written in the following form:

$$\mathbf{A}_u \mathbf{u}_h = \mathbf{b}_u, \quad \mathbf{B}_p \mathbf{p}_h = \mathbf{b}_p, \quad \mathbf{E}_e \mathbf{e}_h = \mathbf{b}_e \quad \text{where;} \quad (24)$$

$$\mathbf{A}_u \in \mathbb{R}^{dN_h \times dN_h}, \quad \mathbf{B}_p \in \mathbb{R}^{N_h \times N_h}, \quad \text{and} \quad \mathbf{E}_e \in \mathbb{R}^{N_h \times N_h}, \quad (25)$$

indicate the matrices containing the terms related to velocity, pressure, and energy for the discretized continuity, momentum, and energy equations respectively.

$$\mathbf{b}_u \in \mathbb{R}^{dN_h}, \quad \mathbf{b}_p \in \mathbb{R}^{N_h}, \quad \text{and} \quad \mathbf{b}_e \in \mathbb{R}^{N_h}; \quad (26)$$

are the related source terms. In addition, \mathbf{u}_h , \mathbf{p}_h , and \mathbf{e}_h are the vectors where all the \tilde{u}_i , \tilde{p}_i , and \tilde{e}_i variables are collected with $d = 2$ the dimension of the computational domain and N_h being the number of control volumes (cells) in the mesh. In the following, Galerkin projection (on the fully discrete equations) is used for the construction of the reduced-order method. We assume the following decompositions introduced in section “[Proper Orthogonal Decomposition procedure](#)”

$$\mathbf{u}_h = \sum_{i=1}^{N_u} a_i(\cdot) \phi_i(\mathbf{x}) = \Phi \mathbf{a}^\top, \quad \mathbf{p}_h = \sum_{i=1}^{N_p} b_i(\cdot) \xi_i(\mathbf{x}) = \Xi \mathbf{b}^\top, \quad \mathbf{e}_h = \sum_{i=1}^{N_e} c_i(\cdot) \theta_i(\mathbf{x}) = \Theta \mathbf{c}^\top \quad (27)$$

Where $\tilde{\mathbf{u}}_r \simeq \mathbf{u}_h$, $\tilde{\mathbf{p}}_r \simeq \mathbf{p}_h$, and $\tilde{\mathbf{e}}_r \simeq \mathbf{e}_h$. Similarly, $a_i(\cdot)$, $b_i(\cdot)$, and $c_i(\cdot)$ are modal coefficients which can time, parameters dependent or both. ϕ_i , ξ_i , and θ_i are the basis functions corresponding to the POD modes of the velocity, pressure, and energy fields stored respectively in $\Phi \in \mathbb{R}^{dN_h \times N_u}$, $\Xi \in \mathbb{R}^{N_h \times N_p}$, and $\Theta \in \mathbb{R}^{N_h \times N_e}$. In addition, N_u , N_p , and N_e being the numbers of basis functions selected for the predicted velocity, pressure, and energy solutions respectively. $\mathbf{a} \in \mathbb{R}^{N_u}$, $\mathbf{b} \in \mathbb{R}^{N_p}$, and $\mathbf{c} \in \mathbb{R}^{N_e}$ are the vectors containing the coefficients for the velocity expansion while the same reads for pressure and energy. The linear systems in eq. (24) are projected using the respective basis functions defined in eq. (27) respectively leading to:

$$\mathbf{A}_u^r \mathbf{a} = \mathbf{b}_u^r, \quad \mathbf{A}_p^r \mathbf{b} = \mathbf{b}_p^r, \quad \mathbf{A}_e^r \mathbf{c} = \mathbf{b}_e^r. \quad (28)$$

Where $\mathbf{A}_u^r = \Phi^\top \mathbf{A}_u \Phi \in \mathbb{R}^{N_u \times N_u}$, $\mathbf{A}_p^r = \Xi^\top \mathbf{A}_p \Xi \in \mathbb{R}^{N_p \times N_p}$, and $\mathbf{A}_e^r = \Theta^\top \mathbf{A}_e \Theta \in \mathbb{R}^{N_e \times N_e}$. The resulting reduced linear systems in eq. (28) can be solved using any method for dense matrices. For example, the Householder rank-revealing QR decomposition of a matrix with full pivoting is used, and it is available in the Eigen library [42].

As the main idea here is to rely on a method capable of being as coherent as possible concerning the high-fidelity problem (Algorithm 1), in the following, the main steps for the reduced SIMPLE algorithm related to compressible turbulent flows are reported in Algorithm 1.

In Algorithm 1, $\mathbf{H}(\mathbf{u}_h^n)$ is the extra-diagonal parts of the momentum matrix \mathbf{A}_u so that the following holds:

$$\mathbf{A}_u \mathbf{u}_h^n = \mathbf{D} \mathbf{u}_h^n - \mathbf{H}(\mathbf{u}_h^n), \quad (29)$$

Algorithm 1: The reduced-SIMPLE algorithm

Input : $\mathbf{u}_h^n, \mathbf{p}_h^{n-1}$, and $\mathbf{e}_h^{n-1}, \mu_t^{n-1}$; spatial basis Φ, Ξ and Θ ;

1 **while** $n \leq N_{ols}$ **do**

2 Assemble the discretized momentum equation, **relax** it employing α_u ;

3 **Project** it over the velocity basis functions Φ and solve it to obtain \mathbf{a} ;

4 Reconstruct $\mathbf{u}_h^n = \Phi \mathbf{a}^\top$ and calculate $\mathbf{H}(\mathbf{u}_h^n)$;

5 Evaluate the eddy viscosity field μ_t^n with the **neural network**;

6 Assemble the discretized energy equation, **relax** it employing α_e ;

7 **Project** it over the energy basis functions Θ and solve it to obtain \mathbf{c} ;

8 Reconstruct $\mathbf{e}_h^n = \Theta \mathbf{c}^\top$ and calculate ρ_h^n and \mathbf{T}_h^n ;

9 Assemble the discretized pressure equation;

10 **Project** it over the pressure basis functions Ξ solve to get the new \mathbf{b} ;

11 Reconstruct $\mathbf{p}_h^n = \Xi \mathbf{b}^\top$;

12 Relax the \mathbf{p}_h^n field with α_p ;

13 $\mathbf{u}_h^n \leftarrow \mathbf{A}^{-1} \mathbf{H}(\mathbf{u}_h^n) - \mathbf{A}^{-1} \mathbf{B}_p \mathbf{p}_h^n$; ▷ velocity corrector ;

14 $n \leftarrow n + 1$;

Output: $\mathbf{u}_h^n, \mathbf{p}_h^n, \mathbf{e}_h^n$, and μ_t^n ;

with \mathbf{D} being the diagonal part of \mathbf{A}_u , and \mathbf{u}_h^n the velocities at iteration n . In addition, the *relaxation* of a quantity Q is given by:

$$Q^n = Q^{n-1} + \alpha(Q^{n*} - Q^{n-1}). \quad (30)$$

Where α is the factor that defines the relaxation such that:

- $\alpha < 1$ means under-relaxation. This will slow down the convergence rate but increase the stability.
- $\alpha = 0$ means no relaxation at all. The predicted value of Q is simply used.
- $\alpha > 1$ means over-relaxation. It can sometimes be used to accelerate the convergence rate but will decrease stability.

Q^n refers to the new used value, Q^{n-1} the previous value, and Q^{n*} the new predicted value. For more details, the interested reader can refer to OpenFOAM[®] [43].

Turbulence treatment

In this work some assumptions were taken in section “The compressible Navier–Stokes equations” leading to a simplified FANS system, Eq. 18. Turbulence effects in Eq. 18 are all due to the presence of the eddy viscosity field μ_t . A technique has to be selected to model the eddy viscosity. Within this scope, many different approaches are possible [44–47].

To make our architecture as independent as possible on the turbulence model used during the offline phase to evaluate the μ_t field, this study combines a classical POD–Galerkin approach for what concerns the physical variables $\bar{\mathbf{p}}, \bar{\mathbf{u}}$ and $\bar{\mathbf{e}}$ together with a data driven scheme for what concerns the eddy viscosity evaluation in the Boussinesq hypothesis [48].

Let us imagine to approximate the eddy viscosity field similarly to what has been done for all the other variables:

$$\mu_{t,r} = \sum_{i=1}^{N_{\mu_t}} m_i(\pi) \eta_i(\mathbf{x}),$$

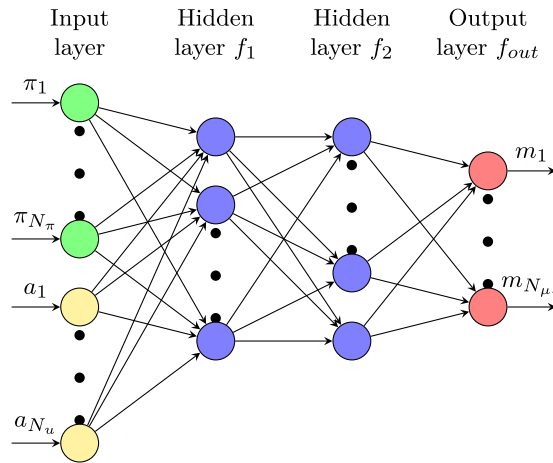


Fig. 2 Schematic perspective of a fully connected neural network composed by an input layer, two hidden layers and an output layer, linking parameters π_i and reduced velocity coefficients a_i to predict the eddy viscosity coefficients m_j . N_π being the number of parameters possibly existing in the problem

where N_{μ_t} is the number of basis functions selected to reconstruct/ predict the eddy viscosity field, m_i are the coefficients depending only on the parameter π while η_i are the μ_t basis functions depend on the position \mathbf{x} . During the offline phase, together with all the other saved solutions, also the eddy viscosity fields are exported and stored. Those snapshots are then collected into the \mathcal{S}_{μ_t} matrix and used, as explained in , to obtain the requested basis functions. For what concerns the parameter coefficients m_i , they are evaluated through a Neural Network (NN) scheme linking the parameters of the problem π_i and the reduced velocity coefficients a_i . In fact, it is well known that, no matter what turbulence model is employed, the eddy viscosity μ_t depends on the velocity field but, especially for geometrically parametrized problems, it also depends on the parameter itself. The reduced problem is thus completely independent of the choice of the turbulence model, and *step 2* into algorithm 1 can be performed efficiently. This would not have been the case if turbulence equations were projected. In case there was the necessity of changing the adopted turbulence model, all the architecture had to be modified.

In this work, we selected a fully connected Neural Network composed by an input layer, two hidden layers and an output layer. The input vector \mathbf{z} and output vector \mathbf{m} are defined as mentioned before:

$$\mathbf{z}^\top = [\pi_1, \dots, \pi_{N_\pi}, a_1, \dots, a_{N_u}], \quad \mathbf{m}^\top = [m_1, \dots, m_{N_{\mu_t}}].$$

It is clear that the Neural Network has to be trained in some way. To this scope, the snapshots contained into \mathcal{S}_{μ_t} are projected over their own basis functions η_i to obtain the set of real coefficients $\{\mathbf{m}_i\}_{i=1}^{N_t}$. They can be compared with the NN estimated coefficients $\{\tilde{\mathbf{m}}_i\}_{i=1}^{N_t}$ into a loss function to target the training procedure. The loss function ℓ we adopted is a widely used quadratic one:

$$\ell = \|\mathbf{m} - \tilde{\mathbf{m}}\|_{L^2}.$$

The quantity \mathcal{L} to be minimized during the training of the network is the sum of the loss function evaluated for all the different snapshots:

$$\mathcal{L} = \sum_{i=1}^{N_t} \|\mathbf{m}_i - \tilde{\mathbf{m}}_i\|_{L^2}.$$

Table 1 A summary of the boundary conditions OpenFOAM® [43]

	Inlet	Outlet	Airfoil
\mathbf{u}	freestreamVelocity	freestreamVelocity	no Slip
p	freestreamPressure	freestreamPressure	zero Gradient
T	inletOutlet	inletOutlet	zero Gradient
k	inletOutlet	inletOutlet	kqRWallFunction
ω	inletOutlet	inletOutlet	omegaWallFunction
ν_t	calculated	calculated	nutkWallFunction

The coefficients estimated by the network can be written as:

$$\hat{\mathbf{m}} = f_{out} (W_{out} f_2 (W_2 f_1 (W_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{out}),$$

where f_1, f_2 and f_{out} are the activation functions, W_1, W_2 and W_{out} are the weights while $\mathbf{b}_1, \mathbf{b}_2$ and \mathbf{b}_{out} are the biases, related to the first and the second hidden layers and to the output layer respectively. For the hidden layers, the best performing activation function appears to be the hyperbolic tangent, while the output layer has been simply implemented as a linear combination of the received data. The previous formula can then be simplified as follows:

$$\hat{\mathbf{m}} = W_{out} \tanh (W_2 \tanh (W_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{out},$$

where $\tanh(\mathbf{y})^T = [\tanh(y_1), \dots, \tanh(y_{dim_y})]$, being $\mathbf{y} = [y_1, \dots, y_{dim_y}]$ a generic vector quantity.

Numerical results

In this section, we first discuss the mesh characteristics with boundary conditions. After that, we discuss the two types of parametrization here considered. Firstly a physical parametrization on the viscosity, and secondly a geometrical parametrization on the airfoil's shape.

Definition of the test case

For a wall boundary condition, \mathbf{u} is assigned a noSlip condition, p and T are assigned a zero gradient. ω is assigned a omegaWallFunction, eddy viscosity is also assigned a nutkWallFunction, and k is assigned a kqRWallFunction. The values at the inlet of the turbulence quantities (ν_t, k , and ω) are set in a way to have already at the inlet a very small turbulent viscosity in order to resample the low turbulence intensity of the domain [49]. The boundary conditions applied are summarized in Table 1.

For the pressure at the inlet, the freestreamPressure boundary conditions are used. It is an outlet-inlet condition that uses the velocity orientation to continuously blend between zero gradient for normal inlet and fixed value for normal outlet flow. The freestreamVelocity velocity boundary condition switches between fixed Value and zero Gradient depending if the mass flux points inside (fixed value) or outside (zero Gradient) the domain.

Physical parametrization test case

The first test case we present in this work is a physical parametrization of an external flow. A NACA0012 airfoil is immersed into a fluid with variable viscosity μ . The unperturbed velocity is fixed and is equal to $\hat{\mathbf{u}}_{inlet} = [250, 0, 0]^T$ m/s while the chord of the airfoil is equal to one. As already said, the viscosity can vary so that $\mu \in [10^{-5}, 10^{-2}]$. The speed of

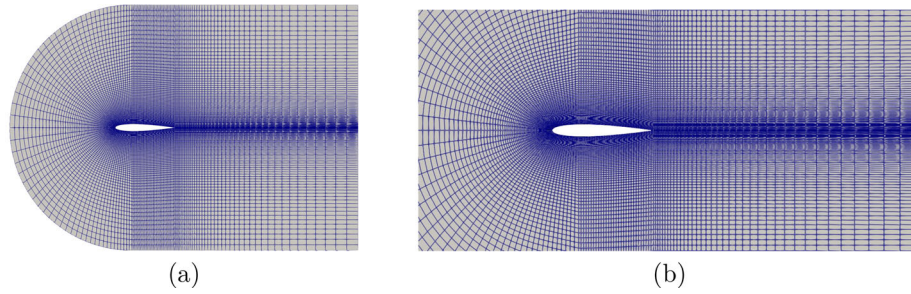


Fig. 3 **a** Picture showing the computational domain and the mesh around the foil with 16,000 cells (control volumes) and 32 480 node points with a chord length 1.0 m, **b** a near view mesh around the airfoil

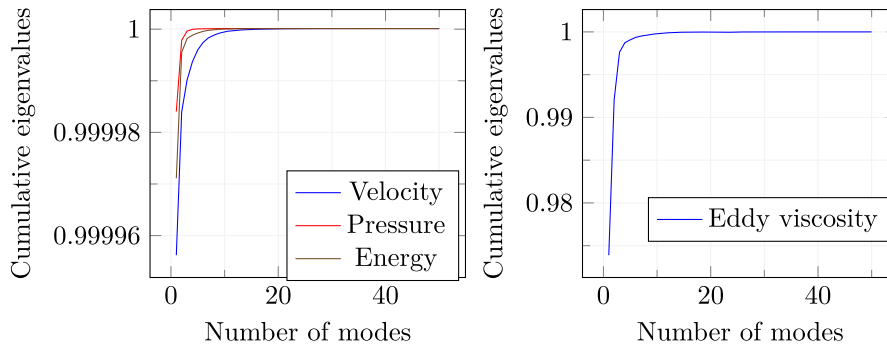


Fig. 4 Cumulative eigenvalues trends

sound at the inlet can easily be evaluated by taking into consideration the thermophysical properties of the gas we are working with. We consider perfect gases. Thus, the specific heat transfer at constant pressure is sufficient to evaluate $\gamma = \frac{C_p}{C_v} = \frac{C_p}{C_p - R}$ where $C_p = 1005 \text{ J kg}^{-1} \text{ K}^{-1}$ while $R = 8,314 \text{ mol K}^{-1}$ is the constant for perfect gases. The airfoil is supposed to move into air so that $M = 28,9 \text{ g mol}^{-1}$ where M stands for the molar weight. Temperature is fixed at $T = 298 \text{ K}$. Collecting all these data together, we end up with $C = \sqrt{\frac{\gamma RT}{M}} = 341.17 \text{ m s}^{-1}$. This means that at the inlet the Mach number can be calculated as

$$\text{Mach} = \frac{\tilde{u}_{inlet}}{C} \simeq 0.73.$$

For this test case, consequently, a compressible treatment for the flow is needed since we are approaching the *Transonic regime* and compressible effects are pretty significant. At the inlet, pressure is fixed to 10^5 Pa . Then the Reynolds number can be evaluated as

$$\text{Re} = \frac{\rho L \tilde{u}_{inlet}}{\mu} = \frac{\rho L \tilde{u}_{inlet} M}{\mu R T}.$$

The resulting Reynolds number is then $\text{Re} \in 2.92 \times [10^4, 10^7]$, which clearly requires treatment for turbulence since the system is operating in a fully turbulent regime.

For the offline phase, 50 random values have been selected: $\pi_i \in [10^{-5}, 10^{-2}]$ for $i = 1, \dots, 50$ where $[\pi_1, \dots, \pi_{50}] = \mathbb{P}$. Full-order eddy viscosity is calculated by the resolution of a $k - \omega$ turbulence model [38].

Figure 4 shows the trends of the *cumulative* eigenvalues for velocity, pressure, energy, and eddy viscosity. As we may notice, by just considering a few modes for every variable, the amount of discarded information is pretty low. For this reason, just the first 20 modal basis functions have been selected for velocity, pressure, and energy while 30 modal basis

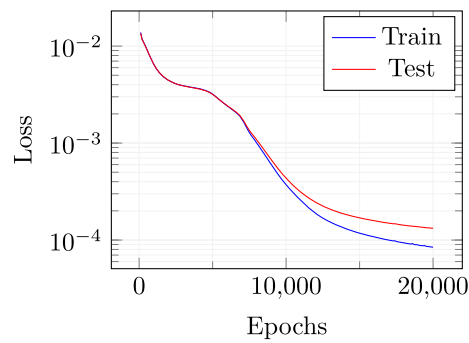


Fig. 5 Loss function decay for both train and test sets

functions are used to predict the eddy viscosity field. This is due to the fact that by analyzing Fig. 4, it can be noticed that a higher number of basis functions are needed in order to approach the unity in the cumulative eigenvalues plot. With these numbers of modes, the cumulative variance reach approximately 99.999%.

As explained in subsection “[Turbulence treatment](#)”, the neural network for the eddy viscosity coefficients employed here is composed by: an input layer, whose dimension is equal to the dimension of the reduced velocity, i.e. 20, plus one for the parameter, two hidden layers of dimension 256 and 64 respectively, an output layer of dimension 30 for the reduced eddy viscosity coefficients. The net is a fully connected one. Moreover, the neurons of the hidden layers are characterized by the employment of tanh activation functions. For the training procedure, the Adam optimizer has been selected with a learning rate set to 10^{-4} . The training procedure is carried out in 2×10^4 epochs. The training set is composed by both the intermediate and final solutions obtained during the offline phase, randomly selected. To control the training procedure, a test set has also been selected. 20 totally random new parameter values have been chosen, and their related full solutions have been calculated, saving both final and intermediate steps, coherently with the offline snapshots used for training. The conditions on which this model performs well depends with high-quality, representative, and sufficient amount of data. On the contrary, with poor data quality and small datasets, the model could have some limitations.

A mean squared error loss function is used to evaluate the prediction capability of the network for both training and testing sets. The decay behaviour of both losses is depicted in Fig. 5. The training stage was stopped after 2×10^4 epochs to avoid over-fitting as the distance between test and train losses was starting to increase significantly.

Figure 6, left, shows the L^2 norm relative errors for all the different parameters in the online set concerning velocity, pressure, and internal energy. Figure 6, right, shows the L^2 norm relative error for the eddy viscosity between full-order and reduced-order for the whole online parameter set. As we may notice, even if the order of magnitude of the v_t error is equal to 10^{-2} , it is sufficient to ensure a lower error for the quantities of interest, i.e. velocity, pressure, and energy. By this observation we are allowed to employ such a small neural network which is not compromising the computational cost, still ensuring good performances.

In Fig. 7 and Fig. 8 a comparison between full-order and reduced-order solutions is depicted, for a random value of the parameter, included in the online set. By analysing the depicted fields, full-order and reduced-order solutions appear to be very similar, and the

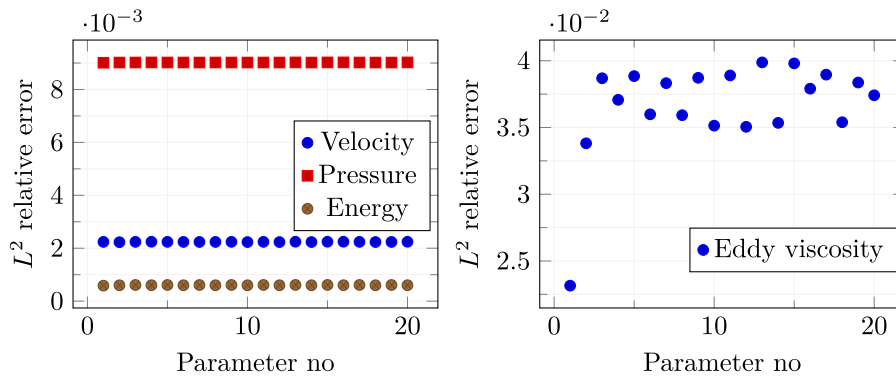


Fig. 6 L^2 norm relative errors

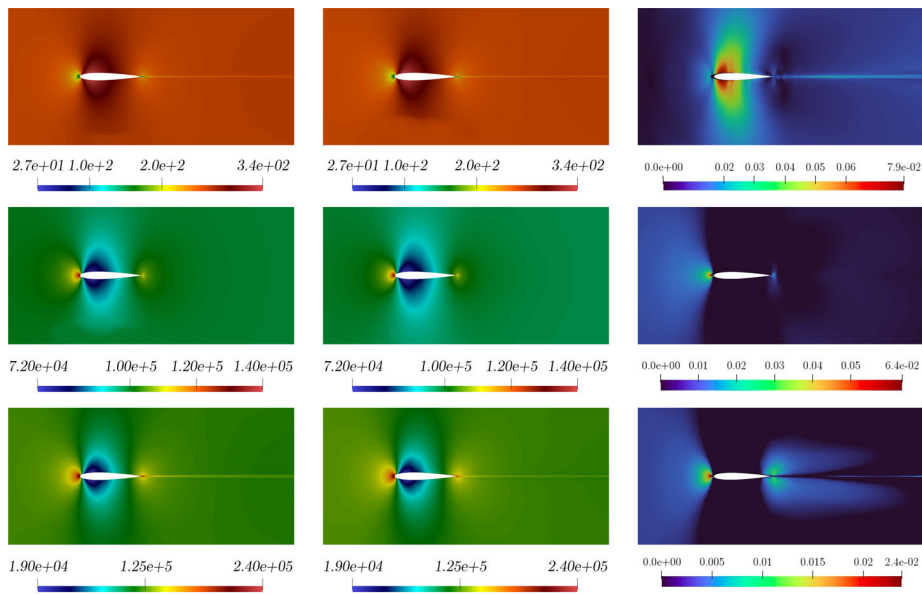


Fig. 7 Comparison between full-order (first column), reduced-order solutions (second column), and relative error associated (third column). *Velocity* (first row), *pressure* (second row), and *energy* (third row) magnitude. These fields refer to the resolution of the problem for $\pi = \mu = 0.21 \times 10^{-3}$ which has been selected as a random value in the online parameter set

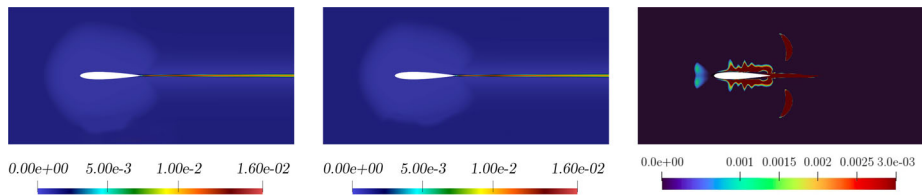


Fig. 8 Comparison between full-order (left picture), reduced-order (middle picture) for the *eddy viscosity* solutions, and relative error associated (third picture). These fields refer to the resolution of the problem for $\pi = \mu = 0.21 \times 10^{-3}$ which has been selected as a random value in the online parameter set

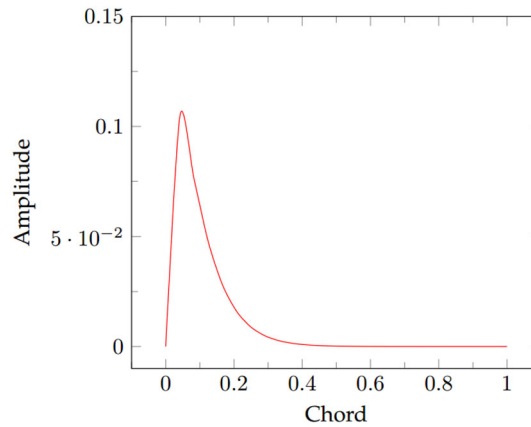


Fig. 9 Shape of the employed bump function

most important areas in the domain, i.e. the zone surrounding the airfoil together with the wake created by the body, are well predicted. Additionally, the related error confirms the above observation.

Geometrical parametrization test case

This section presents the second test case, focused on a geometrically parameterized problem. The shape of the airfoil used into subsection “[Multiscale modeling](#)” is modified by the use of a bump function. In particular, the foil is divided in a top and a bottom part by the chord. The bump function depicted in Fig. 9 is added to the top and subtracted to the bottom surface, premultiplied by two different amplitude scalar factors: every solution is parameterised uniquely by two different scalar values. We used the same thermophysical properties used for subsection “[Multiscale modeling](#)” but the dynamic viscosity is fixed and equal to $1.74 \times 10^{-5} Pa s$. Moreover, the inlet velocity has been slightly decreased since the random modification of the geometry may lead to high curvature areas where the flow could eventually become supersonic: $\vec{u}_{inlet} = [170, 0, 0]^T$ m s⁻¹. This means that the Mach number at the inlet is now around 0.5. For the offline phase, 50 random values have been selected: $\pi_{top_i}, \pi_{bottom_i} \in [0, 0.1]$ for $i = 1, \dots, 50$ where $\mathbb{P} = \{(\pi_{top_i}, \pi_{bottom_i})\}_{i=1}^{50}$. The full-order eddy viscosity is calculated by the resolution of a $k - \omega$ turbulence model [38].

The general POD approach described in subsection “[Proper Orthogonal Decomposition procedure](#)” is not directly applicable to a geometrical parametrization problem, since the L^2 -norm used for the inner products is not well-defined in case of multiple different domains. The mesh in our case is moved thanks to a Radial Basis Functions (RBF) algorithm where the points on the moving boundaries are displaced by the application of the desired law and their displacements are used as boundary conditions for an interpolation procedure, performed in order to move all the remaining points of the grid. The interested reader may find a deeper explanation of this technique in [50] or some applications in [51] and [52]. By exploiting the aforementioned method, the mesh is modified for each offline solution. To take into account the fact that all the snapshots are defined over a different mesh, the grid is taken back to its undeformed state before starting the POD procedure: the mass matrix we consider evaluating the norms is then the reference unperturbed one.

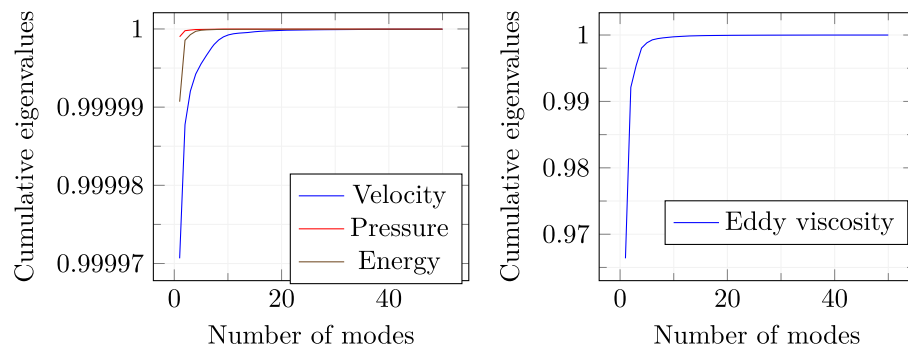


Fig. 10 Cumulative eigenvalues trends

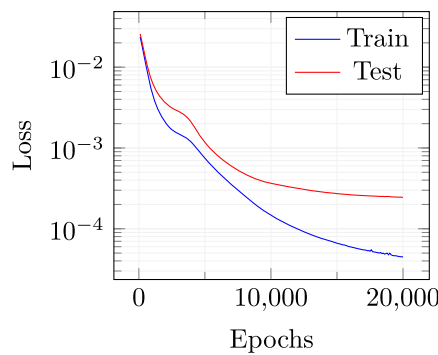


Fig. 11 Loss function decay for both train and test sets

To test the online performances, 20 new scalar amplitude couples have been randomly selected. 30 modal basis functions have been picked for the prediction of velocity, pressure and internal energy fields, while 15 modal basis functions have been employed for v_t . This choice is supported by what is shown in Fig. 10 i.e. the increasing trend of the cumulative eigenvalues is pretty fast and this fact allows the discarding of the modes higher than the fixed quantity. For every new parameter couple, the mesh motion has to be performed, but the procedure is very efficient since the coefficients for the RBF have to be evaluated and stored just once [52].

The same neural network used for Fig. 14 is employed here for what concerns the eddy viscosity. Again, looking at Fig. 11, it can be noticed that the learning of the net seems to stabilize after 2×10^4 epochs which is the threshold we fixed for the training procedure.

The resulting L^2 norm errors for all the parameter couples in the online set are shown in Fig. 12. Once again, a discrepancy of about one order of magnitude can be noticed between the relative errors for the quantities of interest and the one calculated for the eddy viscosity. This is because we are using a very simple and small network, but it reveals to be reliable enough to make the online algorithm work fine.

In Fig. 13, Fig. 14, Fig. 15 and Fig. 16 a comparison between offline and online solutions and the error field associated are presented for two different parameter couples selected from the online set. Even if the two solutions are obtained for airfoil geometries that are perturbed in opposite directions, in both cases the method exhibits good reliability properties even though the intermediate solutions introduced in the snapshots matrix could be highly inaccurate and trigger instabilities in the ROMs. Additionally, Fig. 2 reports

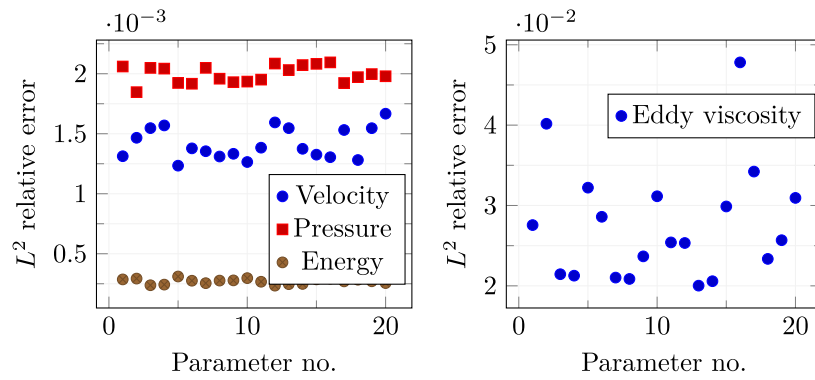


Fig. 12 L^2 norm relative errors

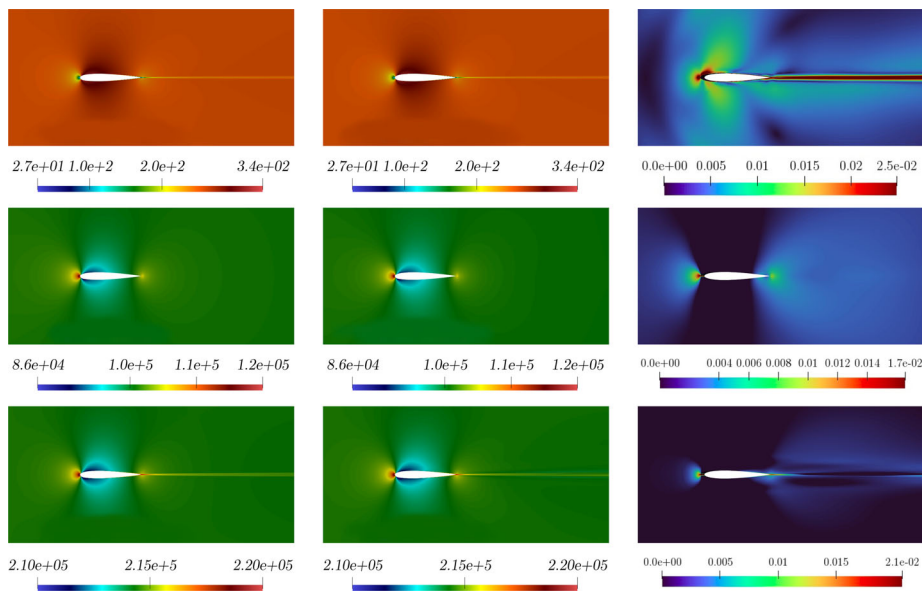


Fig. 13 Comparison between the full-order (first column) and reduced-order (second column) solutions with the relative error associated (third column): *velocity* (first row), *pressure* (second row), and *energy* (third row) magnitude. These fields refer to the resolution of the problem for $(\pi_{top}, \pi_{bottom}) \simeq (0.004, 0.086)$ which has been selected as a random value in the online parameter set

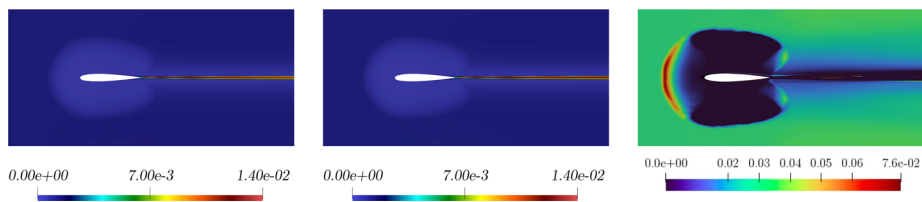


Fig. 14 Comparison between the full-order (left picture) and reduced-order (middle picture) with the relative error associated (right picture) for the *eddy viscosity* solutions. These fields refer to the resolution of the problem for $(\pi_{top}, \pi_{bottom}) \simeq (0.004, 0.086)$ which has been selected as a random value in the online parameter set

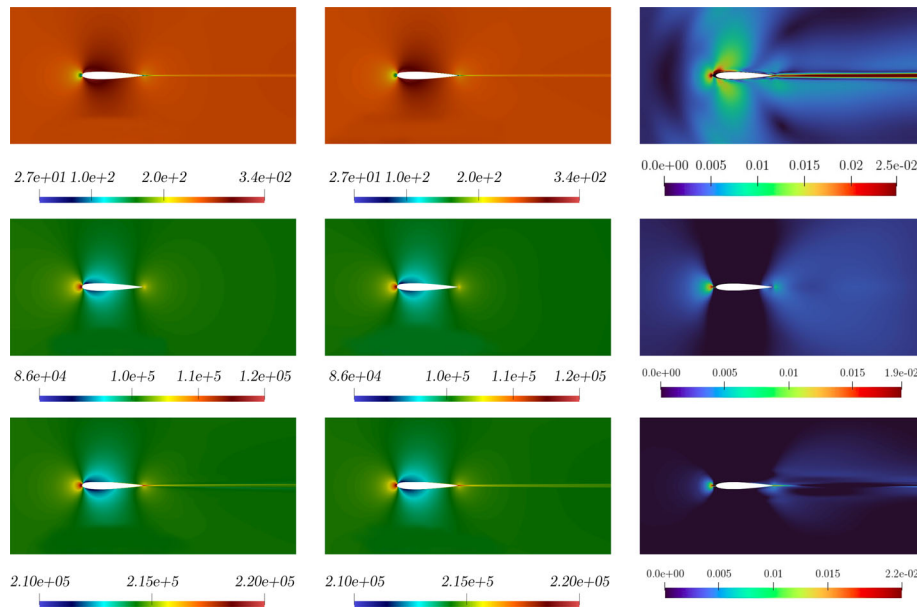


Fig. 15 Comparison between the full-order (first column) and reduced-order (second column) solutions, with the relative error associated (third column): *velocity* (first row), *pressure* (second row), and *energy* (third row) magnitude. These fields refer to the resolution of the problem for $\pi_{top} \simeq 0.095$ and $\pi_{bottom} \simeq 0.003$ which has been selected as a random value in the online parameter set

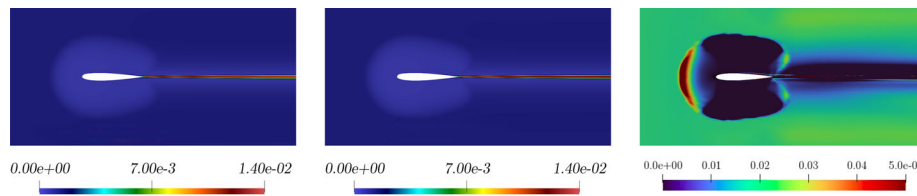


Fig. 16 Comparison between the full-order (left picture) and reduced-order (middle picture) with the relative error associated (right picture) for the *eddy viscosity* solutions. These fields refer to the resolution of the problem for $(\pi_{top}, \pi_{bottom}) \simeq (0.095, 0.003)$ which has been selected as a random value in the online parameter set

Table 2 Error comparison versus the # modes for $(\pi_{top}, \pi_{bottom}) \simeq (0.055, 0.0051)$.

# of modes	L2 relative error			
	u	p	e	v_t
$N_U = N_{v_t} = 10$ and $N_p = N_e = 2$	0.018	0.00311	0.0173	0.652
$N_U = N_{v_t} = 10$ and $N_p = N_e = 5$	0.0163	0.0044	0.0034	0.612
$N_U = N_p = 20$ and $N_e = N_{v_t} = 30$	0.0118	0.0033	0.0019	0.284
$N_U = N_p = N_e = 20$ and $N_{v_t} = 30$	0.0114	0.00320	0.00206	0.2784

an error analysis versus different number of modes. This allows for evaluating the effect of the number of modes variation on the error between the full and reduced order solutions of all the field of interests.

Regarding the computational cost of the reduced-order model, it should be noted that the solution of the reduced system is not proportional to the reduction of the unknowns obtained at the reduced-order level. This is due to the fact that in the presence of a deforming domain such as the one considered in the second case, the entries of the matrices of the ROM system must be computed at each iteration through integrals on the updated full-order grid. Clearly, this is at the moment represents a major bottleneck

towards a ROM which grants significant computational cost reduction with respect to its FOM counterpart, and work is being carried out towards lowering the computational cost associated with the reduced model assembling using hyper-reduction techniques [53,54]. Nonetheless, the main goal of the present work is that of assessing the accuracy of the ROM approach taken. In particular, it is important establishing whether the interaction between the physics based reduction of the fluid dynamic balance equations, and the data driven reduction of the turbulence and shape deformation results in an accurate solver.

Conclusions and future perspectives

This study focused on compressible flows by proposing a new mixed technique, capable of merging the reliability of Galerkin-projection methods together with the versatility of data-driven strategies in turbulence and compressible flows. The good results obtained for both a physical and geometrical parameterized benchmarks make this approach quite promising. From one hand, the possibility to freely select the turbulence model avoiding the necessity of changing the whole architecture is attractive, while on the other hand, the guarantee of a strong connection with physical aspects given by the projection of conservation laws is reassuring.

The segregated compressible algorithm, proposed in subsection “[Multiscale modeling](#)”, also introduces a way to provide accurate reduced solutions without any kind of stabilization: the employment of a decoupled approach for the compressible turbulent Navier–Stokes equations relies on the chipping of the saddle point formulation. For this reason, no stabilization for pressure is required: as shown in both subsection “[Multiscale modeling](#)” and subsection “[Geometrical parametrization test case](#)”, pressure field solutions do not exhibit significant instability or inaccuracy issues. This aspect helps the procedure on being more consistent without pollution of the resulting solution due to stabilization. A natural extension of this work will be a deep analysis with others existing approaches both in the methodology and application. Another extension will be the application of neural networks to approximate the functional evaluations required by the online phase to overtake the necessity of reconstructing the full fields at each iteration. This aspect would increase the performances but it has to be carefully calibrated to avoid possible drifting of the algorithm resulting on the loss of the convergence.

A final aspect that can be improved is the neural network itself: a weighted strategy where eigenvalues play a relevant role in the loss function would, in principle, enhance the training stage since the first modal basis functions, represented by the highest eigenvalues, are the most significant ones on the reconstruction procedure.

Acknowledgements

The main computations in this work were carried out by the usage of ITHACA-FV [55,56], an implementation in OpenFOAM[®] [43] for reduced-order modelling techniques. We acknowledge developers and contributors of each of the aforementioned libraries.

Funding

This research has been supported by the European Union Funding for Research and Innovation -Horizon 2020 Program-in the framework of European Research Council Executive Agency: Consolidator Grant H2020 ERC CoG 2015 AROMA-CFD project 681447 "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" (PI Prof. Gianluigi Rozza) and by the H2020 MSCA RISE ARIA (grant 872442) project. We also acknowledge the INDAM-GNCS project "Tecniche Numeriche Avanzate per Applicazioni Industriali", the support by MIUR (Italian Ministry for University and Research) FARE-X-AROMA-CFD project and PRIN "Numerical Analysis for Full and Reduced-Order Methods for Partial Differential Equations" (NA-FROM-PDEs).

Data availability

The data sets generated during and/or analyzed during the current study are available from the corresponding authors on reasonable request.

Declarations**Competing interests**

The authors report no potential Conflict of interest.

Received: 26 July 2024 Accepted: 18 December 2024

Published online: 19 February 2025

References

1. Anderson JD. Computational fluid dynamics. Computational fluid dynamics: the basics with applications 1995. <https://books.google.it/books?id=dJceAQAIAAJ>
2. Wendt JF. Computational fluid dynamics: an introduction 2008.
3. Blazek J. Computational fluid dynamics: principles and applications 2015.
4. Jameson A. Aerodynamic shape optimization using the adjoint method. Brussels: Lectures at the Von Karman Institute; 2003.
5. Samareh J. Aerodynamic shape optimization based on free-form deformation. In: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics 2004. <https://doi.org/10.2514/6.2004-4630>.
6. Formaggia L, Quarteroni A, Veneziani A. Cardiovascular mathematics: modeling and simulation of the circulatory systems 2010
7. Murthy B, Ghadge R, Joshi J. CFD simulations of gas-liquid-solid stirred reactor: prediction of critical impeller speed for solid suspension. *Chem Eng Sci*. 2007;62(24):7184–95. <https://doi.org/10.1016/j.ces.2007.07.005>.
8. Van Baten J, Krishna R. CFD simulations of mass transfer from Taylor bubbles rising in circular capillaries. *Chem Eng Sci*. 2004;59(12):2535–45. <https://doi.org/10.1016/j.ces.2004.03.010>.
9. Castellani F, Burlando M, Taghizadeh S, Astolfi D, Piccioni E. Wind energy forecast in complex sites with a hybrid neural network and CFD based method. *Energy Proc*. 2014;45:188–97. <https://doi.org/10.1016/j.egypro.2014.01.021>.
10. Holmes PJ, Lumley JL, Berkooz G, Mattingly JC, Wittenberg RW. Low-dimensional models of coherent structures in turbulence. *Phys Rep*. 1997;287(4):337–84. [https://doi.org/10.1016/s0370-1573\(97\)00017-3](https://doi.org/10.1016/s0370-1573(97)00017-3).
11. Amsallem D, Farhat C. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA J*. 2008;46(7):1803–13. <https://doi.org/10.2514/1.35374>.
12. Tezaur I, Fike J, Carlberg K, Barone M, Maddix D, Mussoni E, Balajewicz M. Advanced fluid reduced order models for compressible flow. In: Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 2017
13. Yano M. Model reduction in computational aerodynamics. In: De Gruyter, model order reduction, volume 3 applications, pp. 201–236, 2020. <https://doi.org/10.1515/9783110499001-006>
14. Weller J, Lombardi E, Bergmann M, Iollo A. Numerical methods for low-order modeling of fluid flows based on POD. *Int J Numer Meth Fluids*. 2010;63(2):249–68. <https://doi.org/10.1002/fld.2025>.
15. Stabile G, Rozza G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations. *Comp Fluids*. 2018;173:273–84. <https://doi.org/10.1016/j.compfluid.2018.01.035>.
16. Akhtar I, Nayfeh AH, Ribbens CJ. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theoret Comput Fluid Dyn*. 2009;23(3):213–37. <https://doi.org/10.1007/s00162-009-0112-y>.
17. Baiges J, Codina R, Idelsohn S.R Reduced-order modelling strategies for the finite element approximation of the incompressible navier-stokes equations. In: Springer, pp. 189–216, 2014
18. Burckardt J, Gunzburger M, Lee H-C. POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Comput Methods Appl Mech Eng*. 2006;196(1–3):337–55. <https://doi.org/10.1016/j.cma.2006.04.004>.
19. Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J Numer Anal*. 2002;40(2):492–515. <https://doi.org/10.1137/S0036142900382612>.
20. San O, Iliescu T. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. *Int J Num Anal Model* 2014
21. Hesthaven JS, Rozza G, Stamm B. Certified reduced basis methods for parametrized partial differential equations 2016:590
22. Dumon A, Allery C, Ammar A. Proper Generalized Decomposition method for incompressible Navier-Stokes equations with a spectral discretization. *Appl Math Comput*. 2013;219(15):8145–62.
23. Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng*. 2011;18(4):395–404.
24. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. *J Fluid Mech*. 2010;656:5–28. <https://doi.org/10.1017/s0022112010001217>.
25. Kutz NJ, Brunton SL, Brunton BW, Proctor JL. Dynamic mode decomposition: data-driven modeling of complex systems. SIAM, 2016
26. Zancanaro M, Mrosek M, Stabile G, Othmer C, Rozza G. Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. *Fluids*. 2021;6(8):296. <https://doi.org/10.3390/fluids6080296>.
27. Ivagnes A, Stabile G, Mola A, Iliescu T, Rozza G Hybrid data-driven closure strategies for reduced order modeling. *Appl Math Comput*. 2023;448: 127920. <https://doi.org/10.1016/j.amc.2023.127920>.
28. Ahmed SE, Pawar S, San O, Rasheed A, Iliescu T, Noack BR. On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. *Phys Fluids*. 2021. <https://doi.org/10.1063/5.0061577>.
29. Snyder W, Mou C, Liu H, San O, DeVita R, Iliescu T. Reduced order model closures: a brief tutorial. In: Recent Advances in Mechanics and Fluid-Structure Interaction with Applications: The Bong Jae Chung Memorial Volume, Springer, pp. 167–193, 2022.

30. Maulik R, Lusch B, Balaprakash P. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys Fluids* 2021
31. Ling J, Kurzwski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J Fluid Mech.* 2016;807:155–66.
32. Hesthaven JS, Ubbiali S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J Comput Phys.* 2018;363:55–78.
33. Wang Q, Hesthaven JS, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *J Comput Phys.* 2019;384:289–307.
34. Mohan AT, Gaitonde DV. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. 2018. arXiv preprint [arXiv:1804.09269](https://arxiv.org/abs/1804.09269)
35. Mannarino A, Mantegazza P. Nonlinear aeroelastic reduced order modeling by recurrent neural networks. *J Fluids Struct.* 2014;48:103–21.
36. Papapicco D, Demo N, Girfoglio M, Stabile G, Rozza G. The neural network shifted-proper orthogonal decomposition: a machine learning approach for non-linear reduction of hyperbolic equations. *Comput Methods Appl Mech Eng.* 2022;392: 114687. <https://doi.org/10.1016/j.cma.2022.114687>.
37. Romor F, Stabile G, Rozza G. Non-linear manifold ROM with convolutional autoencoders and reduced over-collocation method. *J Sci Comput.* 2023. <https://doi.org/10.1007/s10915-023-02128-2>.
38. Wilcox DC, et al. Turbulence modeling for CFD. In: DCW Industries La Canada, CA 1998
39. Star SK, Sande B, Stabile G, Rozza G, Degroote J. Reduced order models for the incompressible navier-stokes equations on collocated grids using a ‘discretize-then-project’ approach. *Int J Numer Meth Fluids.* 2021;93(8):2694–722. <https://doi.org/10.1002/flid.4994>.
40. Quarteroni A, Manzoni A, Negri F. reduced basis methods for partial differential equations: an introduction 2015:92
41. Benner P, Ohlberger M, Patera A, Rozza G, Urban K. Model reduction of parametrized systems, 2017
42. Guennebaud G, Jacob B, et al.: Eigen v3. 2010 <http://eigen.tuxfamily.org>
43. *OpenFOAM* website. <https://openfoam.com/>
44. Siddiqui MS, Rasheed A, Kvamsdal T. Numerical assessment of RANS turbulence models for the development of data driven reduced order models. *Ocean Eng.* 2020;196: 106799. <https://doi.org/10.1016/j.oceaneng.2019.106799>.
45. WILCOX D, TRACI R. A complete model of turbulence. In: 9th Fluid and Plasma Dynamics Conference 1976
46. Alfonsi G. Reynolds-averaged Navier-stokes equations for turbulence modeling. *Appl Mech Rev.* 2009. <https://doi.org/10.1115/1.3124648>.
47. Sirovich L. Turbulence and the dynamics of coherent structures, Parts I-III. *Q Appl Math* 561–590, 1987
48. Schmitt FG. About Boussinesq’s turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. *Comptes Rendus Mécanique.* 2007;335(9–10):617–27. <https://doi.org/10.1016/j.crme.2007.08.004>.
49. Bredberg, J.: On the wall boundary condition for turbulence models. Chalmers University of Technology, Department of Thermo and Fluid Dynamics. Internal Report 00/4. Göteborg, 8–16 2000
50. De Boer A, Schoot M, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct.* 2007;85(11–14):784–95.
51. Stabile G, Zancanaro M, Rozza G. Efficient geometrical parametrization for finite-volume-based reduced order methods. *Int J Numer Meth Eng.* 2020;121(12):2655–82. <https://doi.org/10.1002/nme.6324>.
52. Zancanaro M, Mrosek M, Stabile G, Othmer C, Rozza G. Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. *Fluids.* 2021. <https://doi.org/10.3390/fluids6080296>.
53. Reyes R, Codina R. Projection-based reduced order models for flow problems: a variational multiscale approach. *Comput Methods Appl Mech Eng.* 2020;363: 112844.
54. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput.* 2010;32(5):2737–64.
55. *ITHACA-FV* website. <https://github.com/ITHACA-FV/ITHACA-FV>
56. Stabile G, Hijazi S, Mola A, Lorenzi S, Rozza G. POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder. *Commun Appl Indus Math.* 2017;8(1):210–36. <https://doi.org/10.1515/caim-2017-0011>.

Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.