Data-Driven POD-Galerkin reduced order model for turbulent flows

Saddam Hijazi[1,*], Giovanni Stabile[1], Andrea Mola[1], Gianluigi Rozza[1]

[a] *SISSA, International School for Advanced Studies, Mathematics Area, mathLab, via Bonomea 265 – 34136, Trieste, Italy*

Abstract

In this work we present a Reduced Order Model which is specifically designed to deal with turbulent flows in a finite volume setting. The method used to build the reduced order model is based on the idea of merging/combining projection-based techniques with data-driven reduction strategies. In particular, the work presents a mixed strategy that exploits a data-driven reduction method to approximate the eddy viscosity solution manifold and a classical POD-Galerkin projection approach for the velocity and the pressure fields, respectively. The newly proposed reduced order model has been validated on benchmark test cases in both steady and unsteady settings with Reynolds up to $Re = O(10^5)$.

Keywords: proper orthogonal decomposition; finite volume approximation; data-driven ROM; eddy viscosity ROM; turbulence modeling; turbulent ROM; supremizer velocity space enrichment; Navier-Stokes equations.
2010 MSC: 00-01, 99-00

1. Introduction

A large part of physical problems (fluid dynamics, mechanics and heat transfer, ...) in relevant engineering and physics applications is governed by conservation laws. Over the years, several different numerical methods have been developed to solve the systems of Partial Differential Equations (PDEs) resulting from these conservation laws. Among these we mention the finite difference (FDM), the finite element (FEM), the finite volume (FVM), and the spectral element method (SEM). In particular, the Finite Volume one [1, 2] is very often used to solve fluid dynamics and more in general hyperbolic problems.

Despite the recent increase of available computational power and new computational methods, the resolution of the governing equations, using one of the classical discretization methods previously mentioned, may become for several reasons not convenient. This is evident in common situations such as real-time control problems, where a small computational time is a major requirement or in a multi-query contest (e.g. optimization, uncertainty quantification, repetitive computational environment), where one needs to compute a certain output of interest for a large number of different input settings. This makes the cost of resorting to standard numerical methods (that will be referred as the Full Order Model (FOM)) prohibitive. These challenges in simulating computational problems has pushed the scientific community to seek techniques which could reduce the computational cost. Reduced Order Methods (ROMs) have been successful in meeting the needs of reducing the computational time offering high speed up rates. For a comprehensive review on ROMs, the reader may refer to [3, 4, 5, 6, 7].

*First Corresponding author
**Second Corresponding author
Email addresses: shijazi@sissa.it (Saddam Hijazi), gstabile@sissa.it (Giovanni Stabile), amola@sissa.it (Andrea Mola), grozza@sissa.it (Gianluigi Rozza)

Projection based ROMs [8, 9], on which this article is focused, have been applied in several scientific contributions dealing with laminar fluid dynamics problems and the methodology is already well established. On the other side, for what concerns turbulent flows, there are still several issues that need to be addressed. For instance, it is well known that projection based ROMs of turbulent flows suffer from energy stability issues [10]. This is due to the fact that the POD retrieves the modes which are biased toward large, high-energy scales, but the turbulent small scales are the responsible scales for the dissipation of the turbulent kinetic energy [11].

Several strategies have been proposed to stabilize ROMs for turbulent flows and here a brief overview of the possible strategies is outlined. A possible approach suggests to include dissipation via a closure model, see [12, 13]. In [14], it has been theoretically and numerically shown that the POD modes have similar energy transfer to the one of the Fourier modes. This suggests that the use of Large Eddy Simulations (LES) at the full order level could be beneficial in the case of POD-Galerkin-based ROMs.

Another possible approach [15] to obtain more dissipative ROMs, and justified by the fact that small scale modes have H^1 norm value that is higher than their L^2 norm value, proposes the usage of the H^1 inner product instead of the L^2 one in order to compute the POD modes.

Efforts to reduce CFD problems for turbulent flows include also employing minimum residual formulation in the reduced order model [16, 17, 18] or the use of the Dynamic Mode Decomposition (DMD) [19, 20, 21, 22].

Recently in [23], the authors proposed a constrained formulation to deal with long time instabilities. In the latter work, a constrained Galerkin formulation is proposed in order to correct the standard Galerkin approach. The reduced order model in [23] was generated using H^1-POD-hGreedy strategy, which is a simplified version of the h-type Greedy [24]. In [25] the authors presented a reduced order model (based on the FEM) for the Smagorinsky turbulence model [26] for steady flows. Their approach consisted into the approximation of the non-linear eddy diffusion term using the Empirical Interpolation Method. The contribution [27] presents a reduced order model which is designed also to deal with Smagorinsky turbulence model. The authors in [27] proposed a model which solves for the degrees of freedom of the velocity components and does not take into account pressure, the matrix coefficients, which come from the projection of the eddy viscosity term onto the velocity POD modes, have been assumed to be time dependent, and thus, these coefficients were dynamically updated during the time integration of the momentum equation at reduced order level. Additional works on Smagorinsky ROMs are presented in [28, 29, 30]. In the context of ROMs for turbulent flows it is also worth mentioning the Variational Multi-Scale (VMS) method [31, 32]. Smagorinsky VMS-ROMs are proposed in [33, 34]. In [35] started from a Discontinuous Galerkin formulation to inherits the stability of the full order discretization.

Most of the works mentioned above make use of Projection-based methods. However, ROMs can also be obtained by data-driven approaches [36, 37, 38, 39, 40, 41, 42, 43]. A recent work on data-driven reduced order modeling for time-dependent problems can be found in [44], where the authors proposed a regression based model to approximate the maps between the time-parameter values and the projection coefficients onto the reduced basis.

Since the final aim is to develop ROMs for flows with high Reynolds number, at the FOM level a Direct Numerical Simulation (DNS) is not affordable and thus we have to introduce turbulence modeling. In the FVM setting the most used techniques to introduce turbulence modeling are based on the Reynolds Averaged Navier–Stokes (RANS) equations and on the Large Eddy Simulation (LES) method. In this work, the RANS approach is considered. In order to solve the RANS equations a turbulence closure model that describes the effect of sub grid scales is required. In order to approximate the Reynolds stress tensor, we analyzed eddy viscosity closure models for both steady parametrized flows and unsteady flows. We considered closure models with both $k-\epsilon$ and SST $k-\omega$ [45, 46] which are two equations models, in which the eddy viscosity ν_t depends algebraically on two variables k and ϵ or ω. These variables stand respectively for the turbulent kinetic energy, turbulent dissipation and the specific turbulent dissipation rate. An additional PDE is solved for each of the turbulence variables.

In this work we present a mixed approach between projection-based ROMs and data-driven-based ROMs, for some references on hybrid projection/data-driven ROM see [47, 48, 49, 50, 51, 52]. In [47] the authors presented a combination of projection based ROM with a Data Driven Filtering technique. In particular the work proposed to modify the standard Galerkin ROM by introducing a correction term which models

the interaction between resolved modes and truncated modes. The authors used data driven modeling only to approximate the correction term, and tested the ROM on a 2D channel flow past a circular cylinder at Reynolds number of 100, 500 and 1000.

In [48, 49], calibration methods have been constructed for the goal of reducing the Navier–Stokes equations, the authors used POD-Galerkin projection strategy and then they utilized data-driven techniques for calibrating the reduced order models. In [48], this is done by assuming that the term which contains the pressure gradient (in the projected momentum equation) is modeled by the product of a calibration matrix and the reduced vector of velocity coefficients. Afterwards, the calibration matrix entries can be found by minimizing a functional that depends on the values of the interpolated velocity vector of L^2 projection coefficients. In [49], the calibration is done by finding the polynomial function that sets up the reduced dynamical system for the velocity coefficients as the solution to an optimization problem, where the functional which has to be minimized has two weighted terms. The first term measures the error between the values of the projection coefficients obtained from the data and the reduced solution of the dynamical system. The second imposes a cost for the difference between the original polynomial of the reduced dynamical system and the new one that determines the calibrated system.

In [51], the hybrid approach is similar to [48], where an empirical pressure model is used to approximate the pressure term in the projected momentum equation. The data-driven approach utilized is a linear regression which fitted a set of coefficients in the empirical model from the data. In the last mentioned works, the hybrid/mixed approaches include modeling projected terms at the reduced order level and modifying the reduced order matrices entries. We mentioned only works which focus on reducing the the Navier–Stokes equations in both laminar and turbulent settings. Since such works were focused on reconstructing the velocity field of Direct Navier–Stokes resolutions, we here stress that the corresponding reduced model did not include the pressure field nor any turbulence associated field. In the present reduced approach we instead aim at reconstructing both the velocity and pressure fields and also consider the turbulent viscosity field ν_t. This is motivated by the fact that the eddy viscosity is used at the reduced level to stabilize the momentum equation, as is the case for any FOM employing one or two equations turbulence model based on the Boussinesq eddy viscosity assumption. In fact, including the eddy viscosity in the ROM formulation introduces consistency with the FOM. Furthermore, the motivation behind the computation of a reduced version of the pressure field is that in several applications, important performance parameters not only depend on the velocity field, but also on the pressure one. Among these performance parameters, we mention for instance the fluid dynamics forces acting on the surface of a certain body. Thus, the ROM approach developed aims at approximating the fluid dynamics variables $\boldsymbol{u}$, p and ν_t. For such reason, separate sets of ROM coefficients are employed for the reduced order expansion of the $\boldsymbol{u}$, p and ν_t fields. Yet, if the pressure and velocity coefficients are determined through a well assessed projection methodology, the correct identification of the ROM coefficient for the turbulent variable ν_t is less obvious. Ideally, a proper projection procedure requires that the specific turbulence model equations used in the FOM solver must be taken into account. Unfortunately, given the wealth of one and two equations turbulence models of common use in the engineering community, their several variants and the even higher number of closure coefficients to be tracked at the ROM level, this approach appears not suitable for versatile ROMs which aim at being applied to FOM results obtained with different solvers. For instance, for solvers included in the OpenFOAM® (OF) [53] library — which are employed in this work — users can access to about 20 RANS one or two equation turbulence models. This would not only require the development of a projection strategy for each turbulence equation encountered, but would also force constant monitoring of the FOM solvers libraries updates. In fact, even minimal changes in the turbulence models closure coefficients would make the results of the ROM solvers inaccurate. Hence, a versatile ROM solver, that can be employed in the every day virtual prototyping work by design engineers, should ideally be sensitive to the turbulence models used at the FOM level, but its implementation should not be dependent on their smallest details and intricacies. For such reason, we decided to use data-driven techniques for the computation of the reduced order coefficients of ν_t, while still resorting to reduced order expansion of the $\boldsymbol{u}$ and p fields. Indeed, such approach is able to reproduce differences due to changes in the particular turbulence model employed in the FOM simulations, while avoiding the increased ROM complexity due to the projection of the specific turbulence equations.

As a result, the approach developed in this work exploits the traditional projection methods in the part that computes the degrees of freedom for the reduced velocity and pressure fields. On the other hand, it uses a data-driven technique for the computation of the reduced coefficients of the eddy viscosity field. This is done by means of an interpolation process with Radial Basis Functions (RBF). The approach in the offline stage involves the construction of a RBF interpolant function (with Gaussian kernel functions) based on the set of samples used to train the ROM. In the general case of parametrized unsteady flows, both the coefficients obtained by the L^2 projection of the velocity snapshots (obtained by different values of the parameters and/or acquired at different time instants) onto the spatial modes of the velocity, as well as their vector derivatives, will be used to compute the weights of the RBF interpolant function. In the online stage, the values of the eddy viscosity coefficients are obtained by interpolation. The dynamical system resulted from the projection step can be solved to obtain POD coefficients of the pressure and velocity expansion. To summarize, this approach is based on two main ideas. The first one is to approximate the solution manifold of the eddy viscosity field by means of an interpolation based approach. The second idea is to still exploit projection based methods to determine the expansion coefficients for velocity and pressure.

The work is organized as follows: section 2 deals with the description of the full order model and of the numerical methods used to solve the incompressible Navier–Stokes equations. Section 3 presents the methodologies used in this work to assemble the reduced order model. A review of projection based ROMs is outlined in 3.1, then the POD-Galerkin projection method is addressed in 3.2. Subsection 3.3 focuses on the mixed projection-based/data-driven reduced order model. Subsection 3.4 addresses how boundary conditions are treated at the reduced order level. The numerical examples are presented in 4 with two benchmark test cases which are the steady case of the backstep and the unsteady case of the flow past a circular cylinder. Conclusions and perspectives follow.

2. The full order model (FOM)

The present section is devoted to a description of the governing equations of the full order fluid dynamic model. Thus, the parametrized incompressible Navier–Stokes equations will be presented, along with details of their finite volumes discretization. Finally the Reynolds Averaged Navier–Stokes equations will be presented, including some relevant aspects of the turbulence modeling considered in this work.

2.1. The mathematical problem: parametrized Navier-Stokes equations

In this subsection, the strong form of the mathematical problem of interest is recalled. Given a parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^q$, where $\mathcal{P}$ is a q-dimensional parameter space. The Navier-Stokes equations parametrized by $\boldsymbol{\mu}$ read as follows :

$$\begin{cases} \frac{\partial \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu})}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}) \otimes \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu})) - \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla} \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}) + (\boldsymbol{\nabla} \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}))^T \right) = -\boldsymbol{\nabla} p(t,\boldsymbol{x};\boldsymbol{\mu}) & \text{in } \Omega \times [0,T], \\ \boldsymbol{\nabla} \cdot \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0} & \text{in } \Omega \times [0,T], \\ \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{\mu}) & \text{on } \Gamma_{In} \times [0,T], \\ \boldsymbol{u}(t,\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0} & \text{on } \Gamma_0 \times [0,T], \\ (\nu \boldsymbol{\nabla} \boldsymbol{u} - p\boldsymbol{I})\boldsymbol{n} = \boldsymbol{0} & \text{on } \Gamma_{Out} \times [0,T], \\ \boldsymbol{u}(0,\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}) & \text{in } (\Omega, 0), \end{cases} \tag{1}$$

where $\Gamma = \Gamma_{In} \cup \Gamma_0 \cup \Gamma_{Out}$ is the boundary of the fluid domain $\Omega \in \mathbb{R}^d$, with $d = 1$, 2 or 3. The boundary is formed by three different parts Γ_{In}, Γ_{Out} and Γ_0 , which correspond respectively to the inlet boundary, the outlet boundary and the physical walls. $\boldsymbol{u}$ is the flow velocity vector field, t is the time, ν is the fluid kinematic viscosity, and p is the normalized pressure field, which is divided by the fluid density ρ_f, $\boldsymbol{f}$ is a generic function that describe the velocity on the inlet Γ_{In} and it is parametrized through $\boldsymbol{\mu}$. $\boldsymbol{R}$ is the initial velocity field and $[0,T]$ is the time window under consideration. We remark that in this work the parameter $\boldsymbol{\mu}$ is always a physical parameter.

2.2. The finite volume discretization

The governing equations of (1) are discretized using the FVM [1]. After choosing an appropriate polygonal tessellation, one can write the system of partial differential equations (1) in integral form over each control volume. In the present work 2-dimensional tessellations are considered. The number of degrees of freedom of the discretized problem represents the dimension of the full order model (FOM) which is denoted by N_h. In the next subsections, the discretization methodology of the momentum and continuity equations is addressed. In particular the momentum and continuity equations are solved using a segregated approach in the spirit of Rhie and Chow interpolation. The discretization starts writing the momentum equation in integral form for each control volume V_i as follows:

$$\int_{V_i} \frac{\partial}{\partial t}\boldsymbol{u}dV + \int_{V_i} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u})dV - \int_{V_i} \boldsymbol{\nabla}\cdot\nu\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right)dV + \int_{V_i} \boldsymbol{\nabla}p dV = 0. \tag{2}$$

We define then a generic cell center P and a set of neighboring points around it N (Figure 1). For each cell P, the discretized form of the momentum equation is then written as:

$$a_P^{\boldsymbol{u}}\boldsymbol{u}_P + \sum_N a_N^{\boldsymbol{u}}\boldsymbol{u}_N = -\boldsymbol{\nabla}p, \tag{3}$$

where $\boldsymbol{u}_N$ and $\boldsymbol{u}_P$ are the velocities at the centers of two neighboring cells, $a_P^{\boldsymbol{u}}$ is the vector of diagonal coefficients of the equations and $a_N^{\boldsymbol{u}}$ is the vector that consists off diagonal coefficients. Equation 3 is rewritten for all the cells in matrix form as:

$$\mathcal{A}\boldsymbol{u} = \mathcal{H} - \boldsymbol{\nabla}p. \tag{4}$$

In the above expression the terms $\mathcal{H} = -\sum_N a_N^{\boldsymbol{u}}\boldsymbol{u}_N$ and $\boldsymbol{\nabla}p$ are evaluated in an explicit manner based on previous tentative values of the velocity and pressure fields or on the values converged at the previous iteration or at the previous time step. The $\mathcal{A}$ matrix is a diagonal matrix and can be easily inverted and therefore Equation 4 can be easily solved:

$$\boldsymbol{u} = \mathcal{A}^{-1}\mathcal{H} - \mathcal{A}^{-1}\boldsymbol{\nabla}p. \tag{5}$$

If we apply the divergence operator and then exploit the continuity equation ($\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$) we obtain a Poisson equation for pressure:

$$\boldsymbol{\nabla}\cdot(\mathcal{A}^{-1}\boldsymbol{\nabla}p) = \boldsymbol{\nabla}\cdot(\mathcal{A}^{-1}\mathcal{H}). \tag{6}$$

The equation for pressure can be solved and used together with Equation 5 and with the discretized version of the continuity equation to update F_f (the mass flux through each face of the control volume):

$$F_f = \boldsymbol{u}_f\cdot\boldsymbol{S}_f = -\mathcal{A}^{-1}\boldsymbol{S}_f\cdot\boldsymbol{\nabla}p + \mathcal{A}^{-1}\boldsymbol{S}_f\cdot\mathcal{H}, \tag{7}$$

where $\boldsymbol{S_f}$ is the area vector of each face of the control volume and $\boldsymbol{u_f}$ is the velocity vector evaluated at the center of each face of the control volume. The procedure used to discretize all the different terms inside the Navier-Stokes equations is explained in what follows. The pressure gradient term is discretized with the use of Gauss's theorem:

$$\int_{V_i} \boldsymbol{\nabla}p dV = \int_{S_i} p d\boldsymbol{S} \approx \sum_f \boldsymbol{S_f}p_f, \tag{8}$$

where p_f is the value of pressure at the center of the faces (Figure 1).

Again using Gauss's theorem, the convective term can be discretized as follows:

$$\int_{V_i} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u})dV = \int_{S_i}(d\boldsymbol{S}\cdot(\boldsymbol{u}\otimes\boldsymbol{u})) \approx \sum_f \boldsymbol{S_f}\cdot\boldsymbol{u_f}\otimes\boldsymbol{u_f} = \sum_f F_f\boldsymbol{u_f}. \tag{9}$$

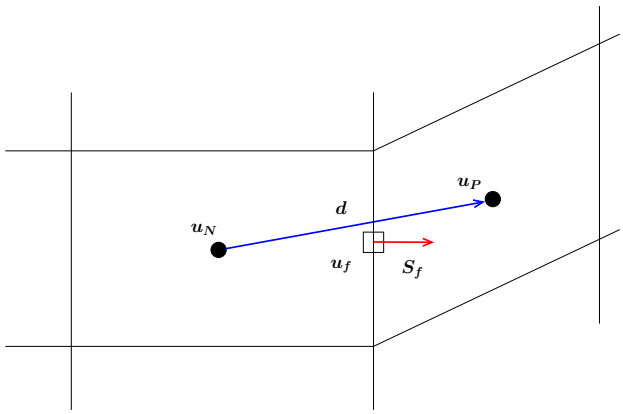


Figure 1: Sketch of a finite volume in 2 dimensions

We remark that the velocity unknowns in the discretized form of the equations are always computed at the center of the faces. Therefore these values must be interpolated using the values at the cell centers. Several interpolation schemes are available such as central, upwind, second order upwind and blended differencing schemes. F_f is the mass flux through each face of the control volume and, in order to remove the non-linearity, it is computed using the previous converged velocity and updated with Equation 7 .

The diffusion term is discretized as follows:

$$\int_{V_i} \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) dV = \int_{S_i} d\boldsymbol{S} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \approx \sum_f \nu \boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f, \tag{10}$$

where $(\boldsymbol{\nabla u})_f$ is the gradient of $\boldsymbol{u}$ at the faces. A procedure similar to the one described for pressure in (8) is used to compute the value of $(\boldsymbol{\nabla u})_f$. As for computing the term $\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f$ in (10), its value depends on whether the mesh is orthogonal or non-orthogonal. The mesh 1 is orthogonal if the line that connects two cell centers is orthogonal to the face that divides these two cells. For orthogonal meshes the term $\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f$ is computed as follows :

$$\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f = |\boldsymbol{S_f}| \frac{\boldsymbol{u}_N - \boldsymbol{u}_P}{|\boldsymbol{d}|}, \tag{11}$$

where $\boldsymbol{u}_N$ and $\boldsymbol{u}_P$ are the velocities at the centers of two neighboring cells and $\boldsymbol{d}$ is the distance vector connecting the two cell centers see Figure 1. If the mesh is not orthogonal, then a correction term has to be added to the above equation. In that case, one has to consider computing a non-orthogonal term to account for the non-orthogonality of the mesh [54] as given by the following equation:

$$\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f = |\boldsymbol{\Delta}| \frac{\boldsymbol{u}_N - \boldsymbol{u}_P}{|\boldsymbol{d}|} + \boldsymbol{J} \cdot (\boldsymbol{\nabla u})_f, \tag{12}$$

where the following relation holds $\boldsymbol{S_f} = \boldsymbol{\Delta} + \boldsymbol{J}$. The first vector $\boldsymbol{\Delta}$ is chosen parallel to $\boldsymbol{S_f}$. The term $(\boldsymbol{\nabla u})_f$ is obtained through interpolation of the the values of the gradient at the cell centers $(\boldsymbol{\nabla u})_N$ and $(\boldsymbol{\nabla u})_P$ in which the subscripts N and P indicate the values at the center of the cells of the two neighboring cells. The coupled system of discretized equations given by Equation 5 and Equation 6 is solved by a segregated approach and specifically using the the SIMPLE [55] algorithm for the steady case and the PIMPLE [1] algorithm for the unsteady case that merges the PISO [56] and the SIMPLE [55] algorithms.

2.3. Turbulence modeling

Since the interest is to solve and and to reduce computational fluid dynamics problems characterized by high Reynolds numbers, the direct numerical resolution of the whole spectrum of temporal and spatial scales is not feasible. In order to model turbulence without resolving all the temporal and spatial scales up to the

Kolmogorov scale two main different approaches are typically used. The first approach — the one considered in this work — is based on Reynolds Averaged Navier-Stokes (RANS) equations and substantially consists into the decomposition of velocity and pressure fields into a mean part and a fluctuating part with zero mean. The decomposition of a generic scalar field $\sigma(x,t)$ will read as follows

$$\sigma = \overline{\sigma} + \sigma', \tag{13}$$

where $\overline{\sigma}$ is the mean part and σ' is the fluctuating one. The RANS equations are obtained after introducing such decomposition for each scalar field (there are four scalar fields consisting of the three velocity components and the pressure field) into Navier-Stokes equations and time averaging them. In RANS, the approach is based on solving the equations for the mean part of each field after making use of the assumption that the fluctuating part has zero mean. A second possible approach — which not considered in this work — consists into Large Eddy Simulations (LES) [57, 58]. LES turbulence modeling is done by filtering and solving the Navier-Stokes equations just for specific scales which are the large scales.

2.3.1. RANS equations

In this subsection, the RANS equations will be presented in further detail. As mentioned earlier, in RANS the turbulence modeling starts by the Reynolds decomposition of the velocity and pressure fields into a mean part and a fluctuating one. These are denoted with $\overline{\boldsymbol{u}}, \overline{p}$ for the mean part and $\boldsymbol{u}', p'$ for the fluctuating part. Inserting the Reynolds decomposition into (1), and time averaging the equations yields the so-called Reynolds Average Navier-Stokes (RANS) equations.

Due to the the non-linearity of Navier-Stokes equations, the velocity fluctuations will not completely vanish in the time averaged equations. In particular the so called Reynolds stress tensor $\mathcal{R} = \overline{\boldsymbol{u}'\boldsymbol{u}'}$ is the single residual term in which the fluctuating components still appear after time averaging. Thus, such tensor must be expressed in terms of the mean part of the flow variables so as to obtain a closed problem for the latter unknowns. In this work we consider eddy viscosity models, that are based on Boussinesq assumption that the Reynolds stress tensor can be expressed by $\mathcal{R} = \frac{\nu_t}{2}[\boldsymbol{\nabla}\overline{\boldsymbol{u}} + (\boldsymbol{\nabla}\overline{\boldsymbol{u}})^T]$. Different possibilities are available for the approximation of the additional coefficient ν_t, which is named eddy viscosity [59]. In the most effective cases the estimation of ν_t is based on the resolution of one or more additional transport-diffusion equations. We mention here the one equation Spalart–Allmaras (S–A) turbulence model [60] and the two equations $k-\epsilon$ [61] and SST $k-\omega$ turbulence models [45].

We here report the RANS equations for the $k-\omega$ turbulence model, which reads:

$$\begin{cases} \frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \boldsymbol{\nabla}\cdot(\overline{\boldsymbol{u}}\otimes\overline{\boldsymbol{u}}) = \boldsymbol{\nabla}\cdot\left[-\overline{p}\mathbf{I} + (\nu+\nu_t)\left(\boldsymbol{\nabla}\overline{\boldsymbol{u}} + (\boldsymbol{\nabla}\overline{\boldsymbol{u}})^T\right)\right] & \text{in } \Omega\times[0,T], \\ \boldsymbol{\nabla}\cdot\overline{\boldsymbol{u}} = 0 & \text{in } \Omega\times[0,T], \\ \overline{\boldsymbol{u}}(t,\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{\mu}) & \text{on } \Gamma_{In}\times[0,T], \\ \overline{\boldsymbol{u}}(t,\boldsymbol{x}) = \mathbf{0} & \text{on } \Gamma_0\times[0,T], \\ (\nu\boldsymbol{\nabla}\overline{\boldsymbol{u}} - \overline{p}\boldsymbol{I})\boldsymbol{n} = \mathbf{0} & \text{on } \Gamma_{Out}\times[0,T], \\ \overline{\boldsymbol{u}}(0,\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}) & \text{in } (\Omega,0), \\ \nu_t = F(k,\omega), & \text{in } \Omega, \\ \text{Transport-Diffusion equation for } k, & \\ \text{Transport-Diffusion equation for } \omega, & \end{cases} \tag{14}$$

where F is the function that describes the algebraic relationship between ν_t and the turbulence variables k and ω.

3. The reduced order model (ROM)

The proposed reduced order model is an extension of the model introduced in [62]. In 3.1 the main notions of projection-based ROMs are recalled. Subsection 3.2 introduces the POD technique and the

general procedure used to construct a POD-Galerkin ROM. Subsection 3.3 addresses in details how data-driven techniques can be exploited to stabilize ROMs for turbulent flows. In particular, the subsection 3.3 explains how the model in the online stage uses data acquired in the offline stage for approximating the Reynolds stress term. Finally subsection 3.4 outlines the treatment of non-homogeneous boundary conditions at the reduced order level.

3.1. Projection based ROMs

In the context of this work, we aim to develop ROMs which are able to approximate the solutions of Parametrized PDEs (PPDEs) in turbulent fluid dynamic problems efficiently and accurately. Reduced order modeling for PPDEs is based on the assumption that the solution field lives in a low dimensional manifold [3]. Based on this assumption any element of the solution manifold can be approximated by the linear combination of a reduced number of global basis functions. The velocity and pressure fields can be approximated as a linear combination of the dominant modes (basis functions) multiplied by scalar coefficients. The modes are assumed to be dependent on space variables only, while the coefficients are allowed to have temporal and/or parameter dependency. The last statement leads to the following approximation of the fields:

$$\boldsymbol{u}(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x}), \quad p(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i(\boldsymbol{x}), \tag{15}$$

where $\boldsymbol{\phi}_i(\boldsymbol{x})$ and $\chi_i(\boldsymbol{x})$ (which do not depend on $\boldsymbol{\mu}$ and t) are the spatial modes for velocity and pressure, respectively, $a_i(t;\boldsymbol{\mu})$ and $b_i(t;\boldsymbol{\mu})$ are temporal coefficients which depend on time t and on the parameter vector $\boldsymbol{\mu}$. The reduced basis spaces $\mathbb{V}_{rb} = \text{span}\,\{\boldsymbol{\phi}_i\}_{i=1}^{N_u}$ and $Q_{rb} = \text{span}\,\{\chi_i\}_{i=1}^{N_p}$ can be obtained either by Reduced Basis (RB) method with a greedy approach [3], using Proper Orthogonal Decomposition (POD) [62], by the Proper Generalized Decomposition (PGD) [63, 64], or by Dynamic Mode Decomposition (DMD) [65]. For unsteady PPDEs, a POD-Greedy approach (POD in time and RB method with greedy algorithm in parameter space) can be used as in [66] or a nested POD can be used where POD is applied on time and later on parameter space. In this work, in order to calculate the reduced basis functions, we rely on a POD approach applied onto the full snapshots matrices formed by the fields obtained for different values of the parameters as well as for different time instants.

3.2. POD-Galerkin projection method for laminar flows

One of the most used approaches to construct reduced order spaces is the proper orthogonal decomposition (POD) [67, 68, 69, 70, 71]. The POD is a method to compress a set of numerical realizations (in the time or parameter space) into a reduced number number of orthogonal basis (modes) that capture the most important information when suitably combined. As mentioned above, in this work the POD is applied on a group of different realizations which are called snapshots. The POD modes are optimal in the sense that, for every number of chosen modes, the difference between the L^2 projection of the snapshots onto the modes and the snapshots themselves is minimized. In this setting, it has to be remarked that the FOM presented in 2 is solved for each value of the vector parameter $\boldsymbol{\mu} \in \mathcal{P}_M = \{\boldsymbol{\mu}_1, ...\boldsymbol{\mu}_M\} \subset \mathcal{P}$ where $\mathcal{P}_M$ is a finite set of samples inside the parameter space $\mathcal{P}$. In case of non-stationary problems, while generating the snapshots for constructing the reduced order space, one has to consider time and parameter dependencies. Consequently, for each parameter value one has the time instants $\{t_1, t_2, ..., t_{N_T}\} \subset [0, T]$ at which snapshots are taken. For this reason, there will be a total number of snapshots $N_s = M * N_T$. The snapshots matrices $\boldsymbol{\mathcal{S}_u}$ and $\boldsymbol{\mathcal{S}_p}$, for velocity and pressure respectively, will be given by:

$$\boldsymbol{\mathcal{S}_u} = \{\boldsymbol{u}(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., \boldsymbol{u}(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N_u^h \times N_s}, \tag{16}$$

$$\boldsymbol{\mathcal{S}_p} = \{p(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., p(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N_p^h \times N_s}, \tag{17}$$

where N_u^h and N_p^h are the degrees of freedom for velocity and pressure fields, respectively. The POD space for velocity is constructed by solving the following optimization problem:

$$\mathbb{V}_{POD} = \arg\min \frac{1}{N_s} \sum_{n=1}^{N_s} ||\boldsymbol{u}_n - \sum_{i=1}^{N_u} (\boldsymbol{u}_n, \boldsymbol{\phi}_i)_{L^2(\Omega)} \boldsymbol{\phi}_i||^2_{L^2(\Omega)}, \tag{18}$$

where $\boldsymbol{u}_n$ is a general snapshot of the velocity field which is obtained for any value of the parameter $\boldsymbol{\mu}$ and acquired at any time instant t_i. It can be shown that solving (18) is equivalent to solve the following eigenvalue problem [72] :

$$\boldsymbol{C^u V^u} = \boldsymbol{V^u \lambda^u}, \tag{19}$$

where $\boldsymbol{C^u} \in \mathbb{R}^{N_s \times N_s}$ is the correlation matrix of the velocity field snapshot matrix $\boldsymbol{\mathcal{S}_u}$, $\boldsymbol{V^u} \in \mathbb{R}^{N_s \times N_s}$ is the matrix whose columns are the eigenvectors, $\boldsymbol{\lambda^u}$ is a diagonal matrix whose diagonal entries are the eigenvalues. The entries of the correlation matrix are defined as follows:

$$(\boldsymbol{C^u})_{ij} = (\boldsymbol{u_i}, \boldsymbol{u_j})_{L^2(\Omega)}. \tag{20}$$

One can compute the velocity POD modes as follows [73],

$$\boldsymbol{\phi}_i = \frac{1}{N_s \lambda_i^u} \sum_{j=1}^{N_s} \boldsymbol{u}_j V_{ij}^u, \tag{21}$$

similar procedure can be followed for the computation of the POD pressure modes $[\chi_i(\boldsymbol{x})]_{i=1}^{N_p}$.

After computing the POD modes of velocity and pressure, one can perform a Galerkin projection of the governing equations onto the POD space. Projecting the momentum equation of (1) onto the POD space spanned by the velocity POD modes yields:

$$\left(\boldsymbol{\phi}_i, \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) + \boldsymbol{\nabla} p\right)_{L^2(\Omega)} = 0. \tag{22}$$

Inserting the approximations (15) into (22) gives the following system:

$$\dot{\boldsymbol{a}} = \nu \boldsymbol{B a} - \boldsymbol{a}^T \boldsymbol{C a} - \boldsymbol{H b}, \tag{23}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the reduced vectors of coefficients $a_i(t; \boldsymbol{\mu})$ and $b_i(t; \boldsymbol{\mu})$, respectively, while the reduced matrices $\boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{H}$ are computed as follows:

$$(\boldsymbol{B})_{ij} = (\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \boldsymbol{\phi}_j)_{L^2(\Omega)}, \tag{24}$$

$$(\boldsymbol{C})_{ijk} = (\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot (\boldsymbol{\phi}_j \otimes \boldsymbol{\phi}_k))_{L^2(\Omega)}, \tag{25}$$

$$(\boldsymbol{H})_{ij} = (\boldsymbol{\phi}_i, \boldsymbol{\nabla} \chi_j)_{L^2(\Omega)}. \tag{26}$$

In [62], one can find more details on the treatment of the non-linear term in Navier-Stokes equations. An important remark is that the system (23) has $N_u + N_p$ unknowns but just N_u equations. Therefore one must seek N_p additional equations in order to close the system. It is not possible to directly exploit the continuity equation at this stage because the velocity snapshots are divergence free and so are the velocity POD modes. The additional equations could be obtained by the usage of a Poisson equation for pressure also at the reduced order level, see [73]. Another possible approach is to employ a supremizer enrichment technique [74, 75] where the velocity POD space is enriched with additional, non divergence-free modes in order to satisfy a reduced version of the inf-sup condition. We refer to [62] for the implementation of this approach in the finite volume setting. There exist also other approaches to obtain pressure-stable ROMs, for example the use of Pressure Stabilized Petrov-Galerkin (PSPG) methods during the online procedure [69, 76] or ROMs based on the assumption that velocity and pressure expansions share the same scalar coefficients [68, 77].

In this work, the supremizer stabilization method has been chosen. This approach will ensure that velocity POD modes are not all divergence free so one can project the continuity equation onto the space spanned by the POD pressure modes. This will give the following reduced system:

$$\begin{cases} \boldsymbol{M}\dot{\boldsymbol{a}} = \nu \boldsymbol{B a} - \boldsymbol{a}^T \boldsymbol{C a} - \boldsymbol{H b}, \\ \boldsymbol{P a} = \boldsymbol{0}, \end{cases} \tag{27}$$

where the new reduced matrices $\boldsymbol{M}$ and $\boldsymbol{P}$ are the mass matrix, that due to the additional supremizer modes is not anymore unitary, and the matrix associated with the continuity equation. The entries of the two additional matrices are given by:

$$(\boldsymbol{M})_{ij} = (\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)_{L^2(\Omega)}, \tag{28}$$

$$(\boldsymbol{P})_{ij} = (\chi_i, \boldsymbol{\nabla} \cdot \boldsymbol{\phi}_j)_{L^2(\Omega)}. \tag{29}$$

3.3. POD-Galerkin Mixed-ROM for turbulent flows

In this subsection, the attention will be shifted to flows characterized by high Reynolds number. As mentioned earlier, turbulence modeling at the full order level is resolved using the RANS equations with a proper closure model (14). This motivated the development of a reduced order model specifically tailored to turbulent flows. This model will be referred to from now on as Mixed-ROM.

A possible approach could consist into a POD procedure applied also onto the additional turbulence variables (k, ω, ϵ) as it was done with the velocity and pressure fields in (15). This phase should be followed by a POD-Galerkin projection of the additional transport diffusion equations that define the specific turbulence model in order to obtain a reduced version of the equations. The last step, in the online phase, would consist into the coupling of all the "reduced" equations and into their simultaneous resolution. The reduced equations come from the momentum equation, the continuity equation and the additional PDEs of the turbulent model. However, this approach has some drawbacks:

- it implies that the reduced order model needs to be customized to the specific turbulence model used during the offline stage;
- since it requires also the projection of the PDEs of the turbulent equations, the effort in the generation of the reduced order model and the number of reduced unknowns is increased.

Since one of the aims of this work is to develop a ROM which is "independent" from the turbulence model used to generate the FOM snapshots the latter approach is ruled out. The chosen approach involves the extension of the assumption of the reduced order expansion only for the eddy viscosity without considering the additional turbulence variables (k, ϵ or ω). In more details, this means introducing the reduced order eddy viscosity as a sum of eddy viscosity POD modes multiplied by temporal or parameter dependent coefficients. The eddy viscosity modes are computed using a POD approach and, during the online stage, the scalar coefficients of the POD expansion are computed with a data-driven approach that uses interpolation with Radial Basis Functions (RBF) [78, 79], thus the reduced order viscosity reads as follows:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) \approx \sum_{i=1}^{N_{\nu_t}} g_i(t, \boldsymbol{\mu}) \eta_i(\boldsymbol{x}), \tag{30}$$

where $\eta_i(\boldsymbol{x})$ are the POD modes for the eddy viscosity field and $g_i(t, \boldsymbol{\mu})$ are the scalar coefficients of the POD expansion. One can see that the temporal coefficients in the above equation are not the same of neither the ones of the velocity $a_i(t, \boldsymbol{\mu})$ nor the ones of the pressure $b_i(t, \boldsymbol{\mu})$. The data-driven approach will be used for the computation of these coefficients. The momentum equation of the RANS (14) is projected onto the spatial modes of velocity, inserting also the POD decomposition of the eddy viscosity field (30). On the other hand, the continuity equation is projected onto the pressure modes with the usage of supremizer enrichment. The POD-Galerkin projection will result in the following reduced system:

$$\begin{cases} \boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a^T C a} + \boldsymbol{g^T}(\boldsymbol{C_{T1}} + \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H b}, \\ \boldsymbol{P a} = \boldsymbol{0}, \end{cases} \tag{31}$$

where $\boldsymbol{g}$ is the vector of the coefficients $[g_i(t, \boldsymbol{\mu})]_{i=1}^{N_{\nu_t}}$, and the new terms with respect to the dynamical

system in (27) are computed as follows:

$$(\boldsymbol{B_T})_{ij} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\boldsymbol{\phi}_j^T)\right)_{L^2(\Omega)}, \tag{32}$$

$$(\boldsymbol{C_{T1}})_{ijk} = (\boldsymbol{\phi}_i, \eta_j \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_k)_{L^2(\Omega)}, \tag{33}$$

$$(\boldsymbol{C_{T2}})_{ijk} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla}\cdot\eta_j(\boldsymbol{\nabla}\boldsymbol{\phi}_k^T)\right)_{L^2(\Omega)}. \tag{34}$$

As one can notice, system (31) has more unknowns $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{g}$ than the available equations. This problem can be resolved by finding a proper way to compute the coefficients of the eddy viscosity POD expansion $\boldsymbol{g}$. This is carried out with the usage of a POD-I approach [80, 81, 82] using radial basis functions.

Before explaining more details about the used methodology we fix a set of notations and conventions. Let $X_{\boldsymbol{\mu},t}$ be the set defined as follows:

$$X_{\boldsymbol{\mu},t} = \mathcal{P}_M \bigtimes \{t_1, t_2, ..., t_{N_T}\}, \tag{35}$$

$X_{\boldsymbol{\mu},t}$ is the Cartesian product of the discretized parameter set and the set of time instants at which snapshots were taken. This set has a cardinality of N_s and its i-th member will be referred as $\boldsymbol{x}^i_{\boldsymbol{\mu},t}$. We remark that for each term $\boldsymbol{x}^i_{\boldsymbol{\mu},t}$ there is a corresponding unique snapshot (for $\boldsymbol{u}$, p and ν_t) that is used to compute the reduced basis for each variable in the offline stage. On the other hand, we define the parameter sample $\boldsymbol{\mu}^*$ as the one introduced to the reduced order model in the online stage. A remark has to be made that $\boldsymbol{\mu}^*$ should be close enough in the parameter space to the parameter samples used in the offline stage that will assure an accurate ROM result. Also we define t^* as the time instant at which the Mixed-ROM solution is sought, where $t_1 \leqslant t^* \leqslant t_{N_T}$. The last statement essentially means that currently it is not possible to extrapolate in time. Also we define $\boldsymbol{z}^* = (t^*, \boldsymbol{\mu}^*)$ as the combination of the online parameter sample and the time instant at which the Mixed-ROM solution is desired.

As done for velocity and pressure in (16) and (17), respectively, we define a matrix of snapshots for the eddy viscosity field as follows:

$$\boldsymbol{\mathcal{S}_{\nu_t}} = \{\nu_t(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., \nu_t(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N^h_{\nu_t}\times N_s}, \tag{36}$$

where the i-th column of the $\boldsymbol{\mathcal{S}_{\nu_t}}$ represents an eddy viscosity snapshot and is denoted by $\boldsymbol{\mathcal{S}^i_{\nu_t}}$. We define $g_{r,l}$ as the coefficient computed from the L^2 projection of the r-th eddy viscosity snapshot $\boldsymbol{\mathcal{S}^r_{\nu_t}}$ onto the l-th eddy viscosity mode η_l.

$$g_{r,l} = (\boldsymbol{\mathcal{S}^r_{\nu_t}}, \eta_l)_{L^2(\Omega)}, \quad \text{for} \quad r = 1, 2, ..., N_s \quad \text{and} \quad l = 1, 2, ..., N_{\nu_t}. \tag{37}$$

The interpolation statement will be the following: given the set $X_{\boldsymbol{\mu},t}$, the corresponding eddy viscosity snapshots $[\boldsymbol{\mathcal{S}^i_{\nu_t}}]_{i=1}^{N_s}$ and the coefficients $[g_{r,l}]_{r=1,l=1}^{N_s,N_{\nu_t}}$, predict the value of the vector $\boldsymbol{g}$ in (31) for the vector $\boldsymbol{z}^*$ defined earlier. The goal can be split to each of the scalar coefficients $[g_i(t^*, \boldsymbol{\mu}^*)]_{i=1}^{N_{\nu_t}}$. Meaning that the interpolation will be done separately N_{ν_t} times for each one of the scalar coefficients. From now on, we will include the dependency as follows $\boldsymbol{g}(\boldsymbol{z}^*)$ or $[g_i(\boldsymbol{z}^*)]_{i=1}^{N_{\nu_t}}$.

The interpolation procedure will be carried out for each mode separately, therefore one could fix the viscosity mode in (37) to be η_L, and then the vector $\boldsymbol{Y}_L = [g_{r,L}]_{r=1}^{N_s} \in \mathbb{R}^{N_s}$ is considered as the set of observations. The next step is to consider the pair of data $(X_{\boldsymbol{\mu},t}, \boldsymbol{Y}_L)$ which is obtained in the offline stage by doing the computations in (37). The objective is to approximate the value of the scalar coefficient $g_L(\boldsymbol{z}^*)$.

The interpolation using RBF functions is based on the following formula :

$$G_L(\boldsymbol{z}) = \sum_{j=1}^{N_s} w_{L,j}\zeta_{L,j}(\|\boldsymbol{z} - \boldsymbol{x}^j_{\boldsymbol{\mu},t}\|_{L^2(\mathbb{R}^{q+1})}), \quad \text{for} \quad L = 1, 2, ..., N_{\nu_t}, \tag{38}$$

where $\boldsymbol{z} = (t, \boldsymbol{\mu})$ with $\boldsymbol{\mu} \in \mathcal{P}$ and $t \in [0, T]$, $w_{L,j}$ are some appropriate weights and $\zeta_{L,j}$ for $j = 1, ..., N_s$ are the RBF functions which are chosen to be Gaussian functions, $\zeta_{L,j}$ is centered in $\boldsymbol{x}^j_{\boldsymbol{\mu},t}$. For the computation of the weights, the following property has to be used, which essentially comes from the data of the FOM:

$$G_L(\boldsymbol{x}^i_{\boldsymbol{\mu},t}) = g_{i,L}, \quad \text{for} \quad i = 1, 2, ..., N_s, \tag{39}$$

and then it follows that,

$$\sum_{j=1}^{N_s} w_{L,j}\zeta_{L,j}(\|\boldsymbol{x}^i_{\boldsymbol{\mu},t} - \boldsymbol{x}^j_{\boldsymbol{\mu},t}\|_{L^2(\mathbb{R}^{q+1})}) = g_{i,L}, \quad \text{for} \quad i = 1, 2, ..., N_s. \tag{40}$$

The last equation can be rewritten as a linear system, namely:

$$\boldsymbol{A}^{\zeta}_L \boldsymbol{w}_L = \boldsymbol{Y}_L, \tag{41}$$

where $(\boldsymbol{A}^{\zeta}_L)_{ij} = \zeta_{L,j}(\|\boldsymbol{x}^i_{\boldsymbol{\mu},t} - \boldsymbol{x}^j_{\boldsymbol{\mu},t}\|_{L^2(\mathbb{R}^{q+1})}$, one can solve the latter linear system to obtain the weights $\boldsymbol{w}_L$, which will be stored to be then used in the online stage.

In the **Online Stage**, as **Input** we have the new time-parameter vector $\boldsymbol{z}^*$ and the goal is to compute $\boldsymbol{g}(\boldsymbol{z}^*) = [g_i(\boldsymbol{z}^*)]_{i=1}^{N_{\nu_t}}$, which is done simply by:

$$g_i(\boldsymbol{z}^*) \approx G_i(\boldsymbol{z}^*) = \sum_{j=1}^{N_s} w_{i,j}\zeta_{i,j}(\|\boldsymbol{z}^* - \boldsymbol{x}^j_{\boldsymbol{\mu},t}\|_{L^2(\mathbb{R}^{q+1})}), \quad \text{for} \quad i = 1, 2, ..., N_{\nu_t}. \tag{42}$$

To summarize the procedure, the interpolation using RBF is done in the online stage. The procedure consists into separated N_{ν_t} times interpolation tasks for the interpolation of the elements of the vector $\boldsymbol{g}$, which appears in (31) for some value of the combined time-parameter vector $\boldsymbol{z}^*$.

The interpolation problem has as input a set of known data called $X_{\boldsymbol{\mu},t}$ with cardinality of N_s. A member in that set is a vector called $\boldsymbol{x}^i_{\boldsymbol{\mu},t}$ and lies in $\mathbb{R}^{q+1}$, where one can see that basically time has been treated as another parameter. The other discrete set of outputs (which has the same cardinality N_s) is the set of the coefficients obtained by the projection mentioned in (37) with the viscosity mode being fixed. At the end, based on the observations given in the offline stage, the value of the coefficient $g_i(\boldsymbol{z}^*)$ (the interpolant) will be approximated.

The approach above is general for unsteady parametrized cases, a description of the same approach but just for steady cases can be found in [83]. A modified version of this approach for unsteady flows may involve the splitting of the eddy viscosity field into two parts. The first part, for each individual parameter sample describes the time averaged viscosity field and the second one contains the time varying contribution. In other words we assume that the eddy viscosity field can be rewritten as follows:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) = \overline{\nu_t}(\boldsymbol{x}; \boldsymbol{\mu}) + \nu'_t(\boldsymbol{x}, t; \boldsymbol{\mu}). \tag{43}$$

Such decomposition is justified by the fact that usually the part which is largely affected by parameter changes is the mean contribution. Small fluctuations are in fact poorly affected by parameter variations and in our numerical example we have noticed that excluding the parameter value from the RBF approximation, $\nu'_t(\boldsymbol{x}, t; \boldsymbol{\mu}) \approx \nu'_t(\boldsymbol{x}, t)$, does not lead to a degradation of the accuracy. Therefore, this approach with such an approximation we have the advantage of splitting the time and parameter contributions into two separate terms which will ease the function approximation. At this point the reduced order approximation of the eddy viscosity field will be modified as follows:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) \approx \sum_{i=1}^{M} \overline{g}_i(\boldsymbol{\mu})\overline{\eta}_i(\boldsymbol{x}) + \sum_{i=1}^{N_{\nu_t}} g_i(t)\eta_i(\boldsymbol{x}), \tag{44}$$

where the first sum approximates the averaged part while the second one approximates the time varying contribution. The fields $[\overline{\eta}_i]_{i=1}^M$ are given by the time averaged eddy viscosity fields of the M different parameter samples (each field is computed as the time averaged field of only the eddy viscosity snapshots corresponding to one parameter sample), while $[\overline{g}_i]_{i=1}^M$ are the parameter dependent coefficients which, for a parameter sample outside of the training set $\boldsymbol{\mu}^*$, can be approximated by an interpolation procedure. The dynamical system 31 is modified as follows:

$$\begin{cases} \boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a^T C a} + \overline{\boldsymbol{g}}^T(\overline{\boldsymbol{C}}_{T1} + \overline{\boldsymbol{C}}_{T2})\boldsymbol{a} + \boldsymbol{g^T}(\boldsymbol{C_{T1}} + \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b}, \\ \boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}, \end{cases} \tag{45}$$

where there are two new tensor terms defined as follows:

$$(\overline{\boldsymbol{C}}_{T1})_{ijk} = \left(\boldsymbol{\phi}_i, \overline{\eta}_j \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \boldsymbol{\phi}_k\right)_{L^2(\Omega)}, \tag{46}$$

$$(\overline{\boldsymbol{C}}_{T2})_{ijk} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot \overline{\eta}_j (\boldsymbol{\nabla} \boldsymbol{\phi}_k^T)\right)_{L^2(\Omega)}. \tag{47}$$

One of the drawbacks of the approach in the current setting is that for unsteady cases one can not extrapolate in time. In order to address this issue the RBF interpolation can be rewritten in a different fashion. The idea is to change the independent variable of the RBF interpolation from being the time value t^* to the combination of the reduced order velocity coefficients vectors $\boldsymbol{a}$ and $\dot{\boldsymbol{a}}$. The motivation comes from the fact that the eddy viscosity field ν_t at time t^n, denoted hereafter by ν_t^n, is a function of the time history of the velocity field $\boldsymbol{u}$, in other words $\nu_t^n = \nu_t(\boldsymbol{u}^1, \boldsymbol{u}^2, ..., \boldsymbol{u}^n)$, in the last formula $\boldsymbol{u}^n$ is the FOM velocity field obtained at time t^n. This allows us to write the eddy viscosity coefficients vector in the expansion 44 as follows[1]:

$$\boldsymbol{g}^n = \boldsymbol{g}^n(t^n) \approx \boldsymbol{g}^n(\boldsymbol{a}^n, \dot{\boldsymbol{a}}^n). \tag{48}$$

The training phase of the RBF in the offline stage is done with the L^2 projection coefficients of the velocity modes (excluding the supremizer modes) onto the snapshots as well as the vector of time derivatives of these coefficients. In order to establish a clear idea of the training methodology, the following notation will be used:

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{u}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{u}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{u}} \end{bmatrix}, \boldsymbol{\mathcal{S}}_{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{p}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{p}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{p}} \end{bmatrix}, \boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{\nu_t}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{\nu_t}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{\nu_t}} \end{bmatrix}, \tag{49}$$

where the snapshots matrices for all the variables have been written as M vertically aligned submatrices with each one of the submatrices containing the time snapshots corresponding to a single sample. The next step is to define the L^2 velocity projection coefficients denoted by $\boldsymbol{a}^r_{\boldsymbol{\mu}_k,L^2} \in \mathbb{R}^{N_u}$:

$$\boldsymbol{a}^r_{\boldsymbol{\mu}_k,L^2} = [(\boldsymbol{\mathcal{S}}^{\boldsymbol{r}}_{\boldsymbol{\mu_k},\boldsymbol{u}}, \phi_1)_{L^2(\Omega)}, ..., (\boldsymbol{\mathcal{S}}^{\boldsymbol{r}}_{\boldsymbol{\mu_k},\boldsymbol{u}}, \phi_{N_u})_{L^2(\Omega)}], \quad \text{for} \quad r = 1, 2, ..., N_T, \quad k = 1, 2, ..., M. \tag{50}$$

Let

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{1},\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{a}^1_{\boldsymbol{\mu}_k,L^2} \\ \boldsymbol{a}^2_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \boldsymbol{a}^{N_T-1}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}, \boldsymbol{\mathcal{A}}_{\boldsymbol{2},\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{a}^2_{\boldsymbol{\mu}_k,L^2} \\ \boldsymbol{a}^3_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \boldsymbol{a}^{N_T}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}, \tag{51}$$

then the needed time derivative vectors for the RBF interpolation are simply computed by the backward differentiation scheme as follows:

$$\dot{\boldsymbol{a}}^r_{\boldsymbol{\mu}_k,L^2} = \frac{\boldsymbol{a}^r_{\boldsymbol{\mu}_k,L^2} - \boldsymbol{a}^{r-1}_{\boldsymbol{\mu}_k,L^2}}{\Delta t_{\boldsymbol{\mu}_k}}, \quad \text{for} \quad r = 2, 3, ..., N_T, \quad k = 1, 2, ..., M. \tag{52}$$

In the formula above $\Delta t_{\boldsymbol{\mu}_k}$ is the time step at which snapshots were acquired for the parameter sample $\boldsymbol{\mu}_k$. As a result, the following matrix of time derivative velocity coefficients is formed

$$\dot{\boldsymbol{\mathcal{A}}}_{\boldsymbol{k}} = \frac{\boldsymbol{\mathcal{A}}_{\boldsymbol{2},\boldsymbol{k}} - \boldsymbol{\mathcal{A}}_{\boldsymbol{1},\boldsymbol{k}}}{\Delta t_{\boldsymbol{\mu}_k}} = \begin{bmatrix} \dot{\boldsymbol{a}}^2_{\boldsymbol{\mu}_k,L^2} \\ \dot{\boldsymbol{a}}^3_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \dot{\boldsymbol{a}}^{N_T}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}. \tag{53}$$

[1]This expression aim to mimic the dependency between the eddy viscosity field and the velocity field. At the FOM level this dependency is described by a PDE. Therefore the expression is an approximation that could be extended with further terms in order to get closer to the map described by the PDE. However, for the numerical examples considered in this work such an approximation turned out to be sufficient.

Finally, merging the L^2 projection coefficients of velocity starting from the second time snapshot with the time derivative coefficients will yield the following matrix

$$\tilde{\boldsymbol{A}}_k = \begin{bmatrix} \boldsymbol{\mathcal{A}_{2,k}} & \dot{\boldsymbol{\mathcal{A}}}_{\boldsymbol{k}} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times 2N_u}. \tag{54}$$

On the other hand, the projection coefficients of the eddy viscosity modes onto the snapshots are given by:

$$g^r_{\boldsymbol{\mu_k},i,L^2} = (\boldsymbol{\mathcal{S}^r_{\mu_k,\nu_t}}, \eta_i)_{L^2(\Omega)}, \quad \text{for} \quad r = 2,3,...,N_T \quad , \quad i = 1,2,...,N_{\nu_t} \quad \text{and} \quad k = 1,2,...,M. \tag{55}$$

One may define the vector $\tilde{\boldsymbol{G}}_{i,k} \in \mathbb{R}^{(N_T-1)}$ as the vector containing the coefficients in (55) for a fixed i and k. The combined matrices and vectors for all parameter samples will be called $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{G}}_i$, respectively, which are defined as follows:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \tilde{\boldsymbol{A}}_1 \\ \tilde{\boldsymbol{A}}_2 \\ \vdots \\ \tilde{\boldsymbol{A}}_M \end{bmatrix} \in \mathbb{R}^{(N_s-M)\times 2N_u}, \tilde{\boldsymbol{G}}_i = \begin{bmatrix} \tilde{\boldsymbol{G}}_{i,1} \\ \tilde{\boldsymbol{G}}_{i,2} \\ \vdots \\ \tilde{\boldsymbol{G}}_{i,M} \end{bmatrix} \in \mathbb{R}^{(N_s-M)}, \tag{56}$$

At this point, the goal of the interpolation will be to approximate the maps $[f_i]_{i=1}^{N_{\nu_t}}$ in $g_i = f_i(\boldsymbol{a}, \dot{\boldsymbol{a}})$, where:

$$f_i : \mathbb{R}^{2N_u} \rightarrow \mathbb{R}. \tag{57}$$

This approximation is based on the interpolation points given in each row of the matrix $\tilde{\boldsymbol{A}}$ and the vector $\tilde{\boldsymbol{G}}_i$.

3.4. Treatment of boundary conditions

In reduced order modeling, it is often the case that the parameterization is in the boundary conditions, and in particular at the inlet boundary. In this subsection, the available methodologies to tackle this aspect are presented. The main two methods to take into account boundary conditions at reduced order level are the penalty method [84, 85, 86, 87, 88] and the lifting function method [89, 90, 91].

Let Γ_D be the Dirichlet boundary that might be composed by separate boundaries, i.e. $\Gamma_D = \Gamma_{D1} \cup \Gamma_{D2} ... \cup \Gamma_{DK}$. Let N_{BC} be the number of velocity boundary conditions we would like to impose on some parts of the Dirichlet boundary. We emphasize that, each non-zero scalar component value of the velocity field that has to be set at one part of the boundary, is counted as one boundary condition. As an example let $\boldsymbol{U_{Dir}} = (U_x, U_y)$ be the velocity vector that must be imposed at the Dirichlet boundary for the problem under interest. It is supposed that U_x and U_y are the values of the velocity components in the x and y directions, respectively, in this case there are two boundary conditions to set and thus $N_{BC} = 2$. Let $U_{BC,i,j}$ be the value of i-th component of the velocity to be imposed at reduced order level at the j-th part of the Dirichlet boundary Γ_{Dj}. We define $\boldsymbol{U_{BC}}$ as the vector of all scalar velocities $U_{BC,i,j}$, this vector has a dimension of N_{BC}, and U_{BCk} is the k-th element of $\boldsymbol{U_{BC}}$.

3.4.1. The penalty method

In the penalty method, an additional term is added in the formulation of the dynamical system of the reduced order model. The added term represents a constraint that has to be satisfied at the reduced order level on certain parts of the boundary. The penalty method has been used for both laminar and turbulent reduced order models as presented in [77]. If we consider employing the method addressed in [77] to the POD-Galerkin Mixed-ROM model, the result will be the following system:

$$\begin{cases} \boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a^T C a} + \boldsymbol{g^T}(\boldsymbol{C_{T1}} + \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H b} + \tau(\sum_{k=1}^{N_{BC}}(U_{BCk}\boldsymbol{D^k} - \boldsymbol{E^k a})), \\ \boldsymbol{P a} = \boldsymbol{0}, \end{cases} \tag{58}$$

where τ is called the penalization factor, and its value is usually determined by sensitivity analysis. In general the higher the value of τ is and the stronger is the enforcement of the boundary conditions. The additional boundary terms with respect to system (31) are defined as follows:

$$(\boldsymbol{D^k})_i = (\boldsymbol{\phi_i})_{L^2(\Gamma_{Dk})}\,, \tag{59}$$

$$(\boldsymbol{E^k})_{ij} = (\boldsymbol{\phi_i}, \boldsymbol{\phi_j})_{L^2(\Gamma_{Dk})}\,. \tag{60}$$

In this method, the POD is applied directly on the snapshots matrices for all variables without the homogenization of the fields. This will result in POD modes which don't have homogeneous Dirichlet boundary conditions.

3.4.2. The lifting function method

The lifting or control function method involves the use of the so-called lifting function which handles the non-homogeneous values on the boundaries. The method involves the creation of a new set of snapshots for the velocity field where the non-homogeneous Dirichlet boundary conditions are removed. After that the POD procedure is applied on the newly formed snapshots and this gives POD modes which have homogeneous Dirichlet conditions at the Dirichlet boundary.

The procedure of modifying the velocity snapshots is done as follows:

$$\tilde{\boldsymbol{u}}_{\boldsymbol{k}} = \boldsymbol{u_k} - \boldsymbol{\phi_L} \cdot \boldsymbol{U_{BC}}, \tag{61}$$

where $\boldsymbol{\phi_L} \in \mathbb{R}^{N_u^h \times N_{BC}}$ is a matrix of the lifting functions $\boldsymbol{\phi_{Li,j}}$. Each lifting function $\boldsymbol{\phi_{Li,j}}$ has homogeneous Dirichlet boundary conditions in all parts of the Dirichlet boundary except in the i-th component at Γ_{Dj} where it has unitary value. We would like to remark that the same lifting method can be used for pressure fields if the formulation of the problem involves non-homogeneous Dirichlet boundary condition for pressure. In that case if the non-homogeneous pressure value is p_{out} then the new pressure snapshots will be computed as follows:

$$\tilde{p}_k = p_k - p_{out}\chi_c, \tag{62}$$

where χ_c is the pressure lifting function. The new snapshots matrices for velocity and pressure are denoted, respectively, by $\tilde{\boldsymbol{\mathcal{U}}} = [\tilde{\boldsymbol{u}}_{\boldsymbol{1}}, \tilde{\boldsymbol{u}}_{\boldsymbol{2}}, \ldots, \tilde{\boldsymbol{u}}_{\boldsymbol{N_s}}]$ and $\tilde{\boldsymbol{\mathcal{P}}} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_{N_s}]$. These snapshots matrices will be used for computing the reduced order bases for the velocity and pressure POD spaces, respectively.

During the online stage, it is required to approximate the velocity and pressure fields for the value of the combined time-parameter vector $\boldsymbol{z}^*$ (which might contain new velocity values to be imposed at some parts of the boundary), this could be done as follows:

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z}^*) \approx \boldsymbol{\phi_L} \cdot \boldsymbol{U^*_{BC}} + \sum_{i=1}^{N_u} a_i(\boldsymbol{z}^*)\boldsymbol{\phi_i}(\boldsymbol{x}), \quad p(\boldsymbol{x}, \boldsymbol{z}^*) \approx p_{out}\chi_c + \sum_{i=1}^{N_p} b_i(\boldsymbol{z}^*)\chi_i(\boldsymbol{x}), \tag{63}$$

where $\boldsymbol{U^*_{BC}}$ is the vector of boundary velocity values that corresponds to $\boldsymbol{z}^*$.

4. Numerical results

In this section, we present the results obtained applying the proposed POD-Galerkin Mixed-ROM on two turbulent flow problems. The first problem is that of the turbulent flow past a backstep. Such classical benchmark in the turbulence modeling community is here considered in a steady state parametrized setup. The second problem analyzed is that of the turbulent flow past a circular cylinder. In the second problem the case is parametrized with Reynolds number being the parameter. In both cases we will present a comparison of the Mixed-ROM results with the ones obtained by the ROM developed in [77]. The authors in their work proposed a ROM which considers a single set of reduced coefficients for the velocity, pressure and eddy viscosity field. In such a way it is possible to resort on a unified approach to deal with turbulent flows. Such

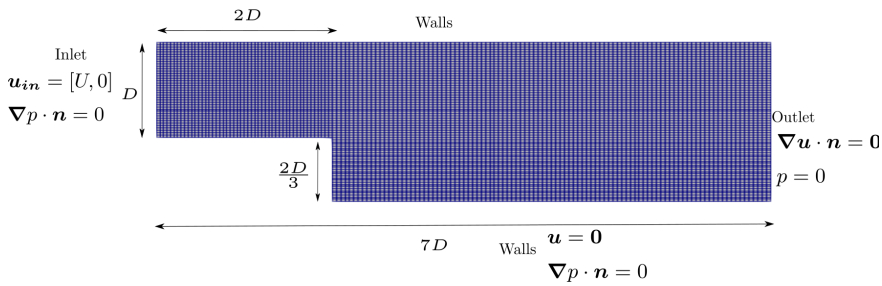


Figure 2: The computational domain used in the numerical simulations, all lengths are described in terms of the characteristic length D that is equal to 1 meter.

an approach, that is used as comparison with respect to methodology developed here, will be referred as P-ROM from now on.

The finite volume C++ library OpenFOAM® (OF) [53] is used as the numerical solver at the full order level. At the reduced order level the reduction and resolution of the reduced order system is carried out using the C++ based library ITHACA-FV [92].

4.1. Steady case

The steady Mixed-ROM solver has been tested on the backward step benchmark case. Figure 2 depicts the layout of the domain with details of the computational mesh. The plot also reports the boundary conditions enforced on every side of the domain. The inflow velocity U has been modified in the simulations to parametrize the problem with respect to the Reynolds number. Thus, the objective of this numerical experiment is to assess the Mixed-ROM solver ability to reproduce flows with high Reynolds number and their dependence on the parameters. To this end, the Mixed-ROM results will be compared both to the full order results and to the results of the P-ROM model. Moreover, to test the Mixed-ROM solver capability to deal with different turbulence models, we tested the model both on full order solutions obtained with $k - \epsilon$ and SST $k - \omega$ models.

The 100 snapshots required for the training during the offline phase were generated by solving the FOM with inlet velocity values ranging from 1 m/s to 25 m/s on an equally spaced distribution. Given the physical viscosity $\nu = 10^{-3}$ m^2/s and the characteristic length is $D = 1$ m, this corresponds to a Reynolds number that varies from 1×10^3 to 2.5×10^4. In the full order simulations, Gauss linear scheme was selected for the approximation of the gradients and Gauss linear scheme with non-orthogonal correction was selected to approximate the Laplacian terms. A 2-nd order bounded Gauss upwind scheme was instead used for the approximation of the convective term. Finally, 1st order bounded Gauss upwind scheme is used to approximate all terms involving the turbulence model parameters k, ϵ and ω.

The modes of velocity, pressure and eddy viscosity fields have been obtained by POD analysis of the snapshots matrices. Figure 3 shows the cumulative eigenvalues decay for velocity, pressure and eddy viscosity. As can appreciated relatively small number of modes is sufficient to recover most of the energetic information in the snapshots.

Once the reduced model training was carried out and the modes were computed, Mixed-ROM and P-ROM simulations have been carried out on new set of sampling points in the parameter space. More specifically, the online sample values for U, denoted by U_i^* where $i = 1, ..., N_{online-samples}$, have been chosen as 80 equally distributed samples in the range of $[3, 20]$. This set of samples includes both samples close to those used in the offline stage and samples which lie almost midway between two offline samples. Clearly, the test is aimed at assessing how accurate the reduced approximation is for parameter values that were not in the training set.

The enforcement of the correct inflow velocity in the reduced simulations is carried out by means of the penalty method as in 3.4.1. In this regard, we must here remark that the simulations results appeared quite

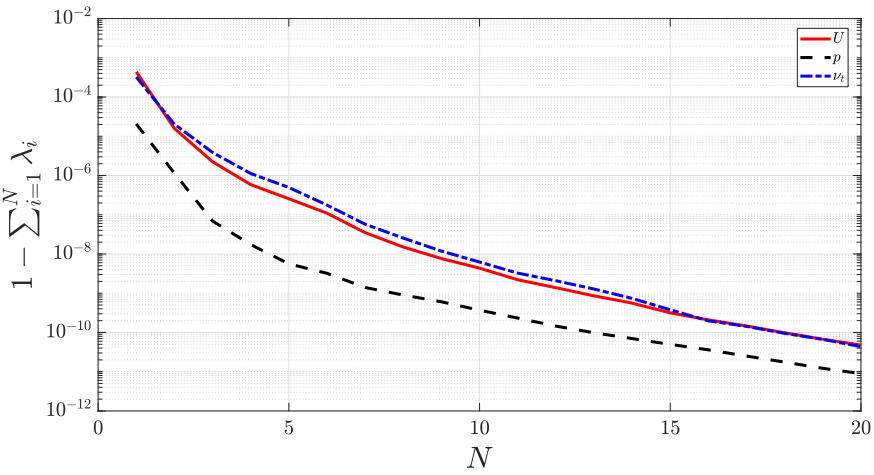


Figure 3: Cumulative ignored eigenvalues decay. In the plot, the solid red line refers to the velocity eigenvalues, the dashed black line indicates the pressure eigenvalues and the dash-dotted blue line finally refers to the eddy viscosity eigenvalues.

sensitive to the penalization factor τ. Thus, a sensitivity analysis had to be performed to set the value of τ for both $k-\epsilon$ and SST $k-\omega$ turbulence models considered.

The first step of the online stage is represented by the interpolation of the eddy viscosity coefficients with respect to the values of the considered parameter (the inflow velocity). More specifically, the result of the interpolation is the vector $\boldsymbol{g}$, which is used to solve the reduced system (31) and finally obtain the vectors of coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$. The interpolation using the RBF in this work has been carried out using the C++ library SPLINTER [93].

Figure 4 depicts the velocity fields corresponding to $U^* = 7.0886$ m/s computed via the FOM, the P-ROM and Mixed-ROM in the case of $k-\epsilon$ turbulence model. A similar comparison is presented in Figure 5 for the pressure fields. We remark that all the solutions were generated using 10 velocity, pressure, supremizer and eddy viscosity modes in the online stage for both the Mixed-ROM and the P-ROM. The images clearly indicate that the hybrid projection/data-driven-based approach allows for qualitatively accurate approximations of the FOM solutions. This is clearly not the case when the P-ROM approach is employed since the pressure field does not correctly reproduce its FOM counterpart. To provide a quantitative measurement of both reduced order models performance, we evaluate the relative L^2 error for velocity and pressure which, respectively, read

$$\epsilon_u = \frac{\|\boldsymbol{u} - \boldsymbol{u}^*\|_{L^2(\Omega)}}{\|\boldsymbol{u}\|_{L^2(\Omega)}} \times 100\%, \epsilon_p = \frac{\|p - p^*\|_{L^2(\Omega)}}{\|p\|_{L^2(\Omega)}} \times 100\%, \tag{64}$$

in which $\boldsymbol{u}^*$ and p^* are general reduced order velocity and pressure fields, respectively. The relative L^2 errors between the FOM and the Mixed-ROM velocity and pressure fields presented in Figure 4 and Figure 5 are respectively $\epsilon_u = 0.4444$ % and $\epsilon_p = 0.3654$ %. As for the P-ROM results, the corresponding errors are $\epsilon_u = 0.6522$ % and $\epsilon_p = 20.9441$ %, respectively.

A further simulation campaign has been carried out with a different, SST $k-\omega$ turbulence model, to evaluate how responsive the hybrid Mixed-ROM and the P-ROM results are with respect to the turbulence model employed for the FOM simulations. Thus, a new set of SST $k-\omega$ FOM simulations has been run using the same inflow velocity values as in $k-\epsilon$ model case. The snapshots generated have been again used to train both reduced models considered. Figure 7, Figure 8 and Figure 9 show the velocity, pressure and eddy viscosity fields obtained by the FOM, the P-ROM and the Mixed-ROM for the inflow velocity value $U^* = 7.0886$, respectively. Again, the Mixed-ROM results appear in good qualitative agreement with their

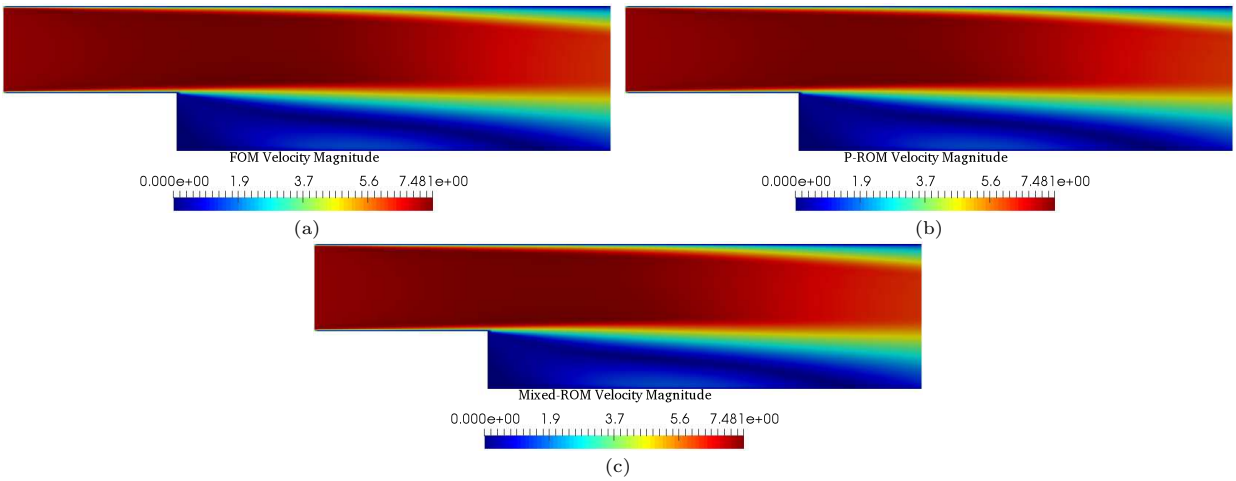


Figure 4: $k-\epsilon$ turbulence model case, velocity fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM velocity, while in (b) one can see the P-ROM velocity, and finally in (c) we have the Mixed-ROM velocity.

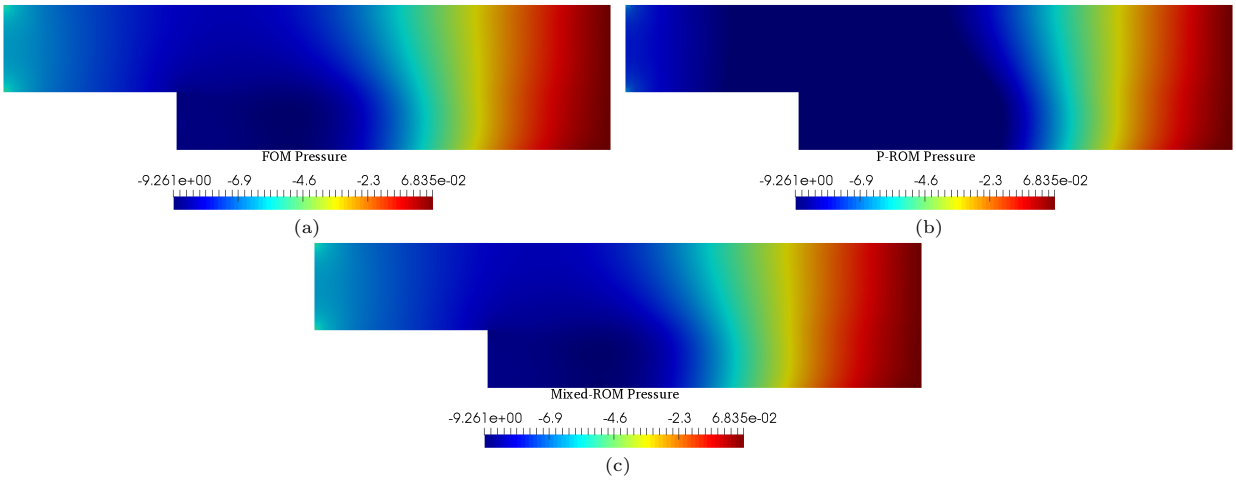


Figure 5: $k-\epsilon$ turbulence model case, pressure fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM pressure, while in (b) one can see the P-ROM pressure, and finally in (c) we have the Mixed-ROM pressure.

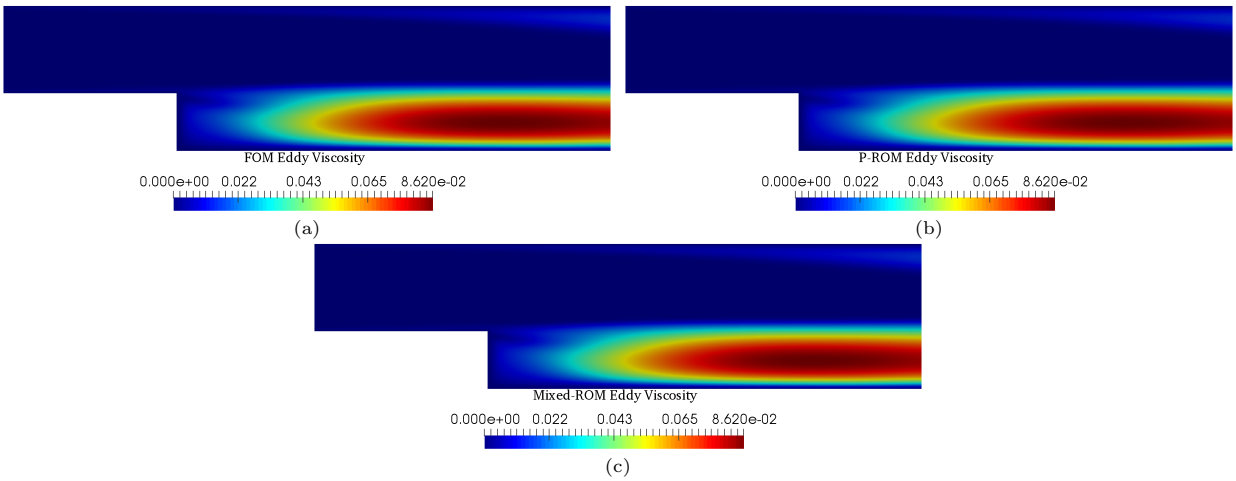


Figure 6: $k-\epsilon$ turbulence model case, eddy viscosity fields: (a) shows the FOM eddy viscosity, while in (b) one can see the P-ROM eddy viscosity, and finally in (c) we have the Mixed-ROM eddy viscosity.

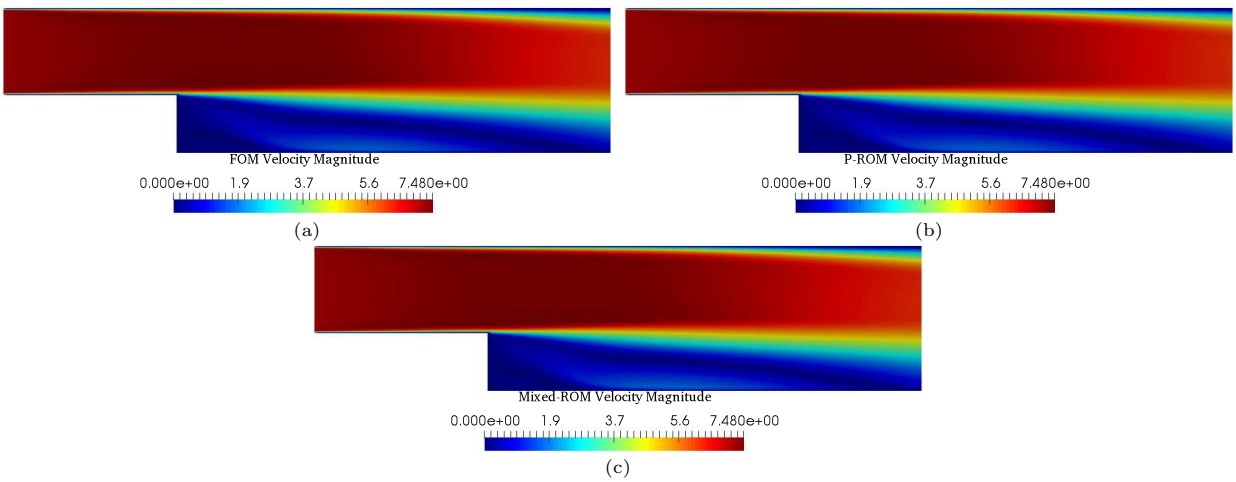


Figure 7: SST $k-\omega$ turbulence model case, velocity fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM velocity, while in (b) one can see the P-ROM velocity, and finally in (c) we have the Mixed-ROM velocity.

SST $k-\omega$ FOM counterparts, while the same cannot be claimed for the P-ROM results. By a quantitative standpoint, the L^2 relative errors between the FOM and the Mixed-ROM velocity and pressure fields are respectively $\epsilon_u = 0.8088$ % and $\epsilon_p = 0.7329$ %. As for the P-ROM results, the corresponding errors are $\epsilon_u = 0.8177$ % and $\epsilon_p = 22.3972$ %, respectively.

The FOM fields obtained solving the RANS equations with the two different turbulence models have quantitatively speaking different values across the domain (except for the velocity). In order to give a clear idea about how accurate was the reduction performed by the Mixed-ROM regardless of the turbulence model employed at full order level, one may plot the FOM and the Mixed-ROM pressure fields (obtained by the two turbulence models) for a fixed value along the x_2 axis (the perpendicular axis) versus the values along the x_1 axis (the horizontal one). The last test is done in Figure 10, where one can see the FOM and the Mixed-ROM pressure fields along the horizontal direction at a fixed height of $x_2 = \frac{5D}{6}$ which is half the height of the domain. The plot is done for both $k-\epsilon$ and SST $k-\omega$. As can be appreciated from the figure, the Mixed-ROM was successful in obtaining pressure field values which are close the FOM ones regardless of the turbulence model utilized at full order level. This accomplishes one of the main goals of the Mixed-ROM developed in this work.

Finally, the convergence analysis for the Mixed-ROM results is shown in Figure 11. The plots show the mean L^2 relative error for all the 80 samples used in the cross validation test in the online stage, as a function of the number of modes used. As previously mentioned, the number of modes used for velocity (N_u), pressure (N_p), supremizer (N_S) and eddy viscosity (N_{ν_t}) was kept uniform in these preliminary tests. The plots indicate that for the problem considered, the Mixed-ROM results exhibit fast convergence to the FOM solution for both $k-\epsilon$ and SST $k-\omega$. Yet, after less then ten modes, the convergence appears to stall, as the error settles on non zero, but fairly acceptable values. This is likely due to the fact that as the number of modes grow, the gain in accuracy becomes only marginal compared to the ν_t field interpolation error.

4.2. Unsteady case

The present subsection presents the application of the Mixed-ROM on a parametrized non-stationary case. The problem considered is that of the turbulent and unsteady flow around a circular cylinder. For more details on such classical benchmark flow, the reader may refer to [94, 95]. In the framework of the unsteady

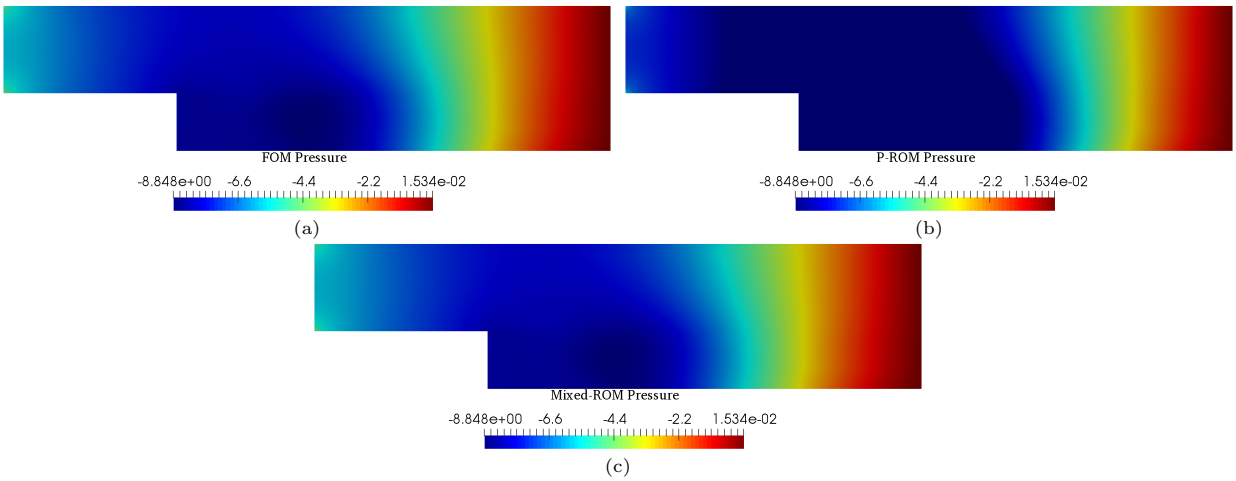


Figure 8: SST $k-\omega$ turbulence model case, pressure fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM pressure, while in (b) one can see the P-ROM pressure, and finally in (c) we have the Mixed-ROM pressure.

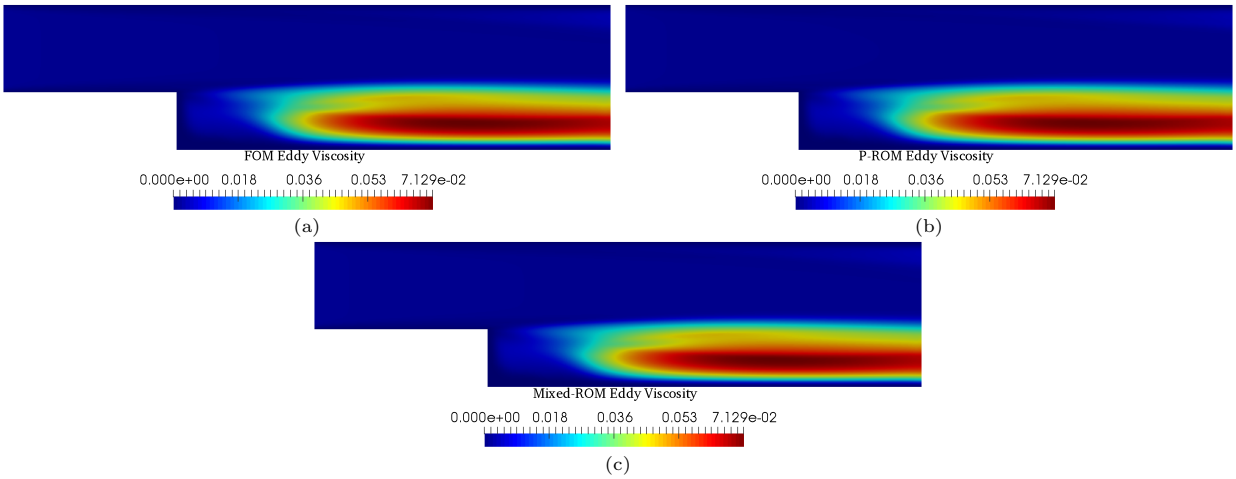


Figure 9: SST $k-\omega$ turbulence model case, eddy viscosity fields: (a) shows the FOM eddy viscosity, while in (b) one can see the P-ROM eddy viscosity, and finally in (c) we have the Mixed-ROM eddy viscosity.

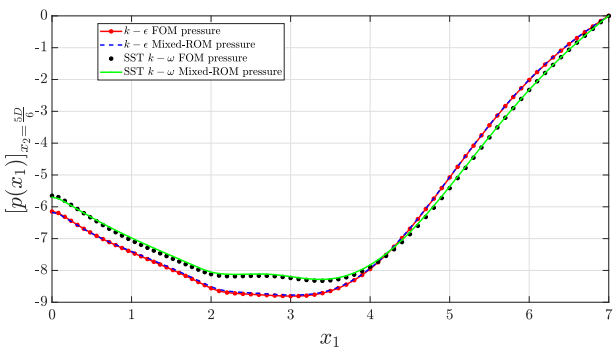


Figure 10: The pressure fields obtained using both $k-\epsilon$ and SST $k-\omega$ turbulence models and the Mixed-ROM ones. The plot is for the pressure value along the x_1 direction keeping the value of x_2 fixed at half the maximum height.

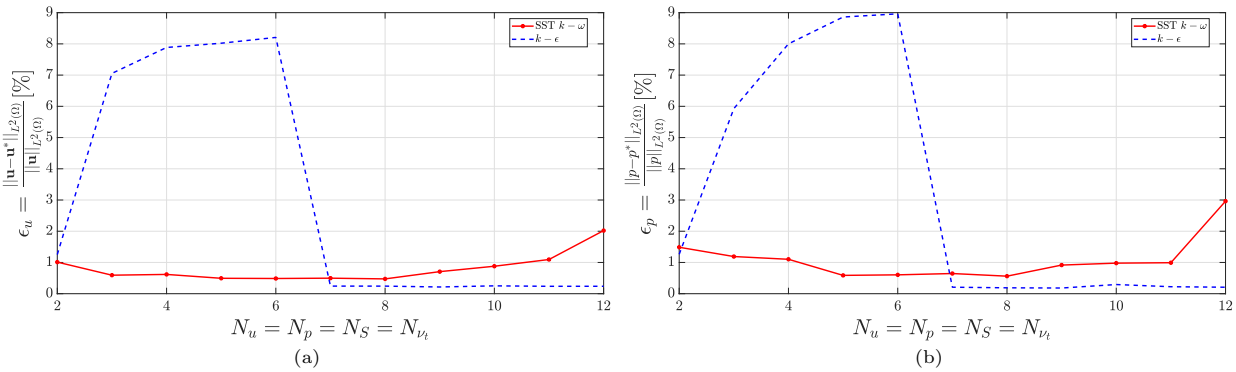


Figure 11: The mean of the L^2 relative errors for all the online samples versus the number of modes used in the online stage. The convergence analysis is done for both Mixed-ROM models obtained with two different turbulence models at the full order level which are $k-\epsilon$ and SST $k-\omega$. The errors are reported in percentages, in (a) we have the velocity fields mean error, while in (b) the pressure fields mean error .

ROM analysis described in Section 3, time is treated as one of the parameters characterizing the PDE problem. A further parameter considered in this numerical investigation is the Reynolds number associated with the inflow velocity. The domain and the 2D computational grid used are depicted in Figure 12, which also reports the boundary conditions imposed in the simulations. In the picture, all the lengths reported are referred to the problem characteristic length which is the diameter of the cylinder $D = 1$ m. The grid features 11644 cells, while the physical viscosity ν is equal to 10^{-4} m^2/s. Uniform and constant horizontal velocities $\mathbf{U}_\infty = (U_{in}, 0)$ with $U_{in} \in [7.5, 12]$ m/s (corresponding to Reynolds number in the range of $[7.5 \times 10^4, 1.2 \times 10^5]$) were imposed at the inlet boundary, and the simulations evolve in time from rest until a final periodic regime solution is reached.

In this test, the turbulence model considered is SST $k-\omega$. As for the numerical schemes used to set up the FOM simulations, time discretization is done using backward Euler scheme, while gradients are approximated using Gauss scheme. The convection term is discretized through a 2nd order bounded upwind divergence scheme which utilized upwind interpolation weights, with an explicit correction based on the local cell gradient. Finally, the diffusive term is discretized by Gauss linear scheme.

The main objective of this numerical test is that of building a reduced order model which can successfully reproduce the flow fields corresponding to the final periodic regime solution. For such reason it is important to properly select the time window from which snapshots will be taken and ensure that it contains enough solution cycles ($1.5 - 2$ cycles at least). The evaluation of the cycles period length has been carried out through Fourier analysis of the FOM time signal of lift and drag fluid dynamic forces acting on the cylinder.

As mentioned, the physical parameter varied in the numerical tests, is the horizontal velocity at the inlet called U_{in}. Ten samples are taken from the velocity range $[7.5, 12]$ m/s, and for each of such samples the FOM simulator was run and snapshots were acquired at time steps covering approximately two cycles after reaching the final regime solution. It must be remarked that the extent of the time windows in which snapshots were taken was adapted for each velocity sample to track the solution period variations due to the change of the frequency of vortex shedding of the system.

As an example of this procedure, if we consider the inlet velocity of 10 m/s, the FOM simulation has been run for 12 seconds using the OpenFOAM solver pimpleFoam which adapts the timestep so as to keep the Courant number CFL [96, 97] under a prescribed value $CFL_{max} = 0.9$. Figure 13 depicts the resulting lift coefficient curve, which is obtained from the lift L as $C_l = \frac{L}{\frac{1}{2}\rho U^2 D}$. The non-uniformly spaced time signal of the lift coefficient has been interpolated on equally distributed time nodes so as to allow the use of Fast Fourier Transform (FFT) for the computation of the time period corresponding to the principal, vortex shedding, frequency. The time period computed is 0.4299 s, corresponding to a Strouhal number [98] of

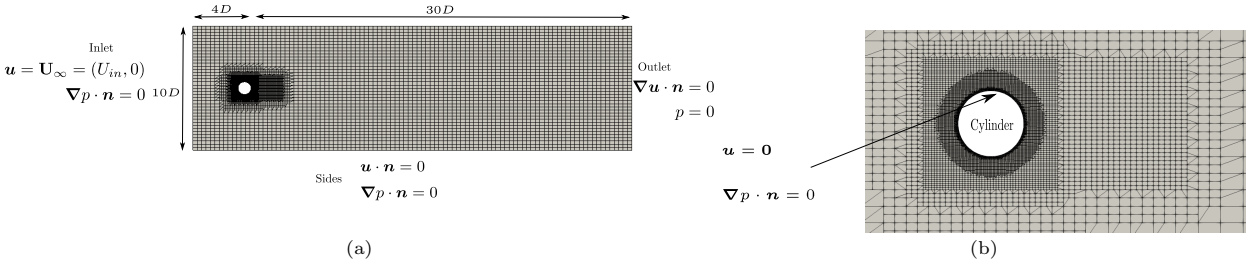


Figure 12: (a) The OpenFOAM mesh used in the simulations for the unsteady case of the flow around a circular cylinder. (b) A picture of the mesh zoomed near the cylinder.

Parameter sample : U_{in} **in** m/s	**FOM time step**	**Snapshot acquiring time**
7.5	0.0004	0.008
8	0.0004	0.008
8.5	0.00035	0.007
9	0.0003	0.006
9.5	0.0003	0.006
10	0.0003	0.006
10.5	0.0003	0.006
11	0.0003	0.006
11.5	0.00025	0.005
12	0.00025	0.005

Table 1: Offline parameter samples and the corresponding snapshots data

St=0.2326, which is in line with well assessed experimental value of approximately 0.20 [99]. After this value was available, the simulations have been extended keeping a fixed time step of 0.0003 s to start acquiring snapshots which cover two periods at least. More specifically, the simulations were run for 1.2 s additional seconds, saving snapshots of the flow field with a 0.006 s time rate so as to finally obtain 200 snapshots. We remark that, to be as consistent as possible, the time step imposed in the resolution of the Mixed-ROM dynamical system (31) at the reduced level, has been the same one used for the FOM simulations.

The offline stage was carried out taking 200 snapshots for each parameter sample. Table 1 shows the values of the parameters and the corresponding values of the simulation time step and the time interval at which snapshots were acquired.

The non-homogeneous Dirichlet boundary condition at the inlet is enforced with the penalty method. POD modes have been obtained applying POD analysis to snapshots matrices of velocity, pressure and eddy viscosity fields. Figure 14 depicts the decay of the cumulative eigenvalues corresponding to the three correlation matrices. The supremizer problem was then solved for each of the pressure modes, to finally obtain the supremizer modes added to the velocity ones.

The online resolution of the Mixed-ROM system requires that an interpolation strategy is used to obtain the eddy viscosity coefficient vector $\boldsymbol{g}$ at each time step t^* of the simulation corresponding to the parameter value U_{in}^*. More specifically, at each time instant $\boldsymbol{g}(t^*, U_{in}^*)$ should be obtained through interpolation — with respect to the combined time-parameter vector — from its values corresponding to the snapshots. Yet, while all snapshots are contained in the aforementioned 1.2 s time window, the online time integration must extend for much longer times. This means that for each instant outside the time window of the original snapshots, $\boldsymbol{g}$ must be in fact extrapolated. To avoid such problem, the eddy viscosity coefficients are obtained through RBF interpolation from the reduced order velocity coefficients vectors of $\boldsymbol{a}$ and $\dot{\boldsymbol{a}}$ (Equation 48). As the values of the reduced velocity solution vector components oscillate between minima and maxima over time integration, using $\boldsymbol{a}(\boldsymbol{t})$ instead of t as the RBF interpolation variable has in fact the

convenient benefit of avoiding extrapolation. Of course, this is true if the values of the a vector components obtained during the ROM time integration fall within the bounds of the FOM snapshots. For such reason, it is clear that the accuracy of such interpolation outside the offline snapshots window highly depends on how close the current solution vector $\boldsymbol{a}$ is to the vectors of the L^2 projection coefficients used in the offline stage for training the RBF.

As for the dependence on the inlet velocity parameter, the results presented have been obtained by splitting ν_t into its time average and its fluctuating part. The inlet velocity parameter dependence has been then only enforced on the time average degrees of freedom $\overline{\boldsymbol{g}}$, while the aforementioned interpolation based on the reduced velocity vector has been only applied to the fluctuating part $\boldsymbol{g}$. This means that in $M = 10$ different average eddy viscosity fields were computed by taking the average of the set of snapshots which correspond to one value of the ten inlet velocity samples. The average reduced vector $\overline{\boldsymbol{g}}$ has been then obtained from U_{in}^* in the online stage using linear interpolation, while the reduced vector $\boldsymbol{g}$ is obtained from RBF interpolation with respect to $\boldsymbol{a}$ and $\dot{\boldsymbol{a}}$. Finally, the initial values for all vectors $\boldsymbol{a}(0, U_{in}^*)$, $\boldsymbol{b}(0, U_{in}^*)$ and $\boldsymbol{g}(0, U_{in}^*)$ are obtained from the inlet velocity parameter using linear interpolation as well (based on the values of the initial L^2 projection vectors of $\boldsymbol{a}(0, U_{in})$, $\boldsymbol{b}(0, U_{in})$ and $\boldsymbol{g}(0, U_{in})$).

The first numerical test is a cross validation test for the parameter value $U_{in} = 7.75$ m/s, not contained in the samples set. Once the offline phase was completed with the computation of the reduced order matrices, system (45) was solved for $\boldsymbol{a}$ and $\boldsymbol{b}$ and the Mixed-ROM solution fields were computed. A comparison is made between the fields obtained by the FOM solver and the ones computed by both the Mixed-ROM and the P-ROM ones. The FOM simulator was run for enough time to reach a periodic regime and then it was launched again with a constant simulation time step of 0.0004 s exporting the solution fields every 0.008 s. The total FOM simulation time for this test was 8 s which contained 13 periods. The starting time of the simulation of the final periodic regime of all tests in this section is set to 0. The first results shown correspond to the flow fields computed by the FOM, the P-ROM and the Mixed-ROM at $t = 2.8$ s. Figure 15 shows the velocity fields while Figure 16 and Figure 17 present the pressure and eddy viscosity fields, respectively. Figure 15 indicates that both the P-ROM and the Mixed-ROM are able to obtain accurate velocity prediction. The L^2 relative norm of the error committed by the two models is in fact 1.218 % and 0.6921 %, respectively. On the other hand the pressure fields shown in Figure 16 suggest that the P-ROM model fails in giving sufficiently accurate results for the reduction of the pressure fields. In fact, the P-ROM pressure field does not match the FOM one. This is particularly true in the region near the cylinder, which is of course crucial for an accurate reproduction of the forces acting on the body. As for the Mixed-ROM, the reduced pressure field appear to be in closer agreement with the FOM one. This is confirmed by more quantitative assessments, as the L^2 relative norm of the Mixed-ROM pressure field error is 4.7894 %, while that of the P-ROM is 29.5958 %. We must remark that to obtain the best results with each model developed in this numerical test, the number of modes used in the online stage for the P-ROM is 9, while in the Mixed-ROM case 20 velocity modes were used and 10 modes were employed for pressure, supremizers and eddy viscosity.

The behavior of the reduced approximation accuracy over time is analyzed considering the time evolution of the relative L^2 error of both the ROM velocity and pressure fields with respect to their FOM counterparts. Figure 18 depicts values of the velocity error ϵ_u plotted as a function of time for both the P-ROM and Mixed-ROM models. A similar graph for the pressure field is presented in Figure 19. Both diagrams suggest that the P-ROM model pressure approximation is not as accurate as that obtained with the Mixed-ROM. Again, we must remark that the modal truncation order used in the P-ROM model to generate Figure 18 and Figure 19 represent the most accurate choices among all values of $N_r \in [1, 30]$, as will be shown in the next results.

One of the main goals for researchers and engineers studying fluid dynamic problems such as the crossflow

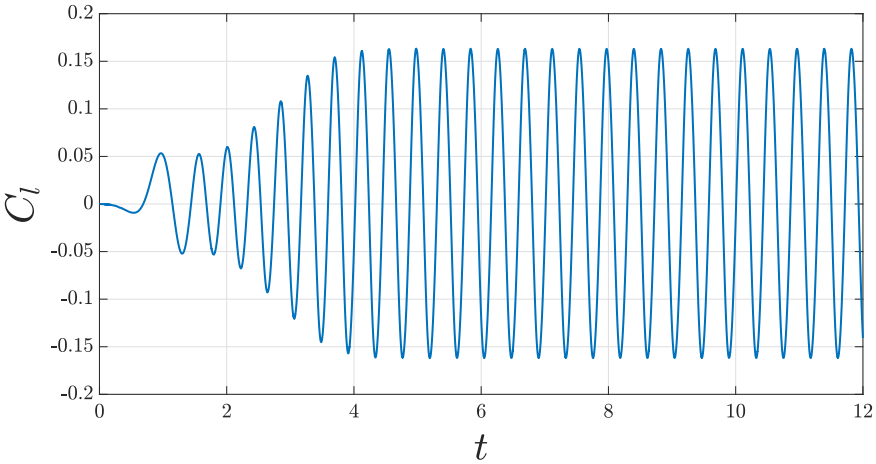


Figure 13: The lift coefficient curve for parameter sample $U_{in} = 10$ m/s.

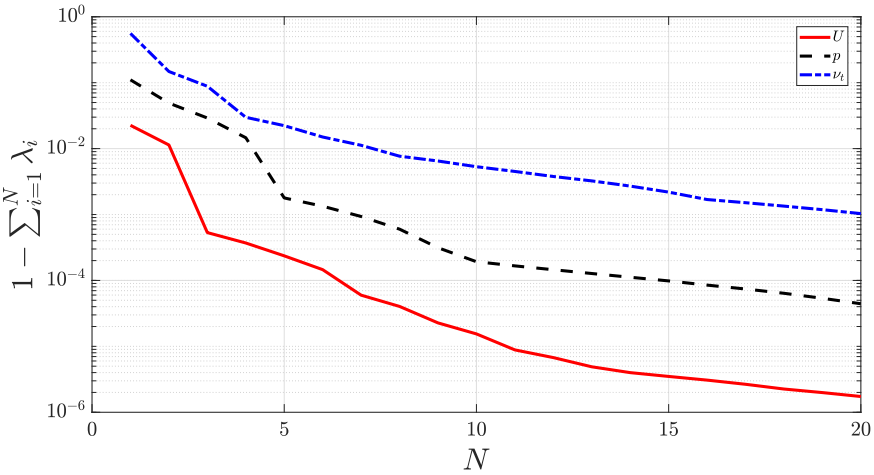
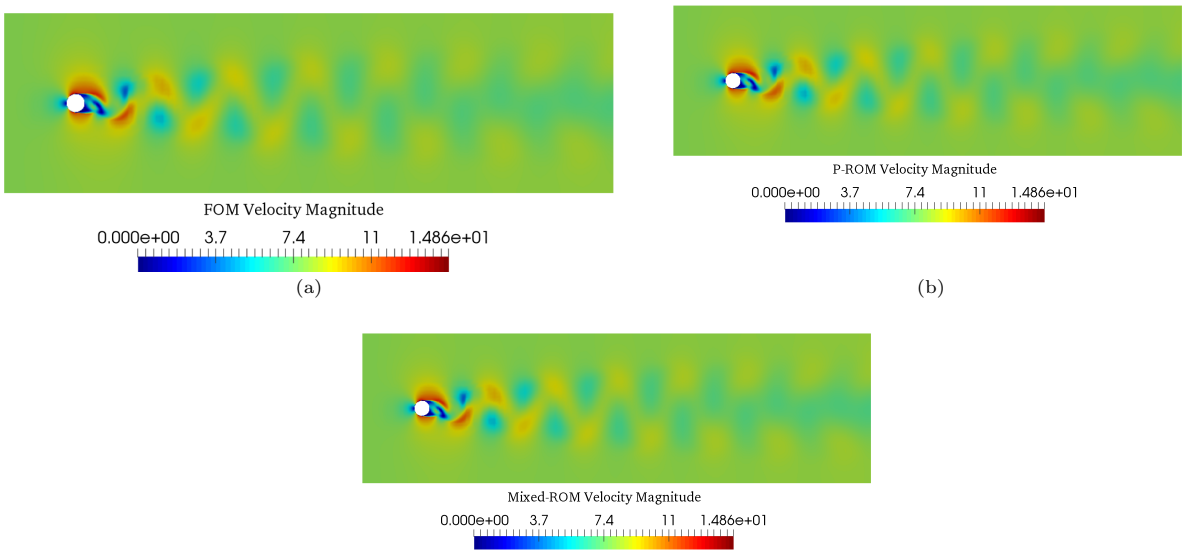


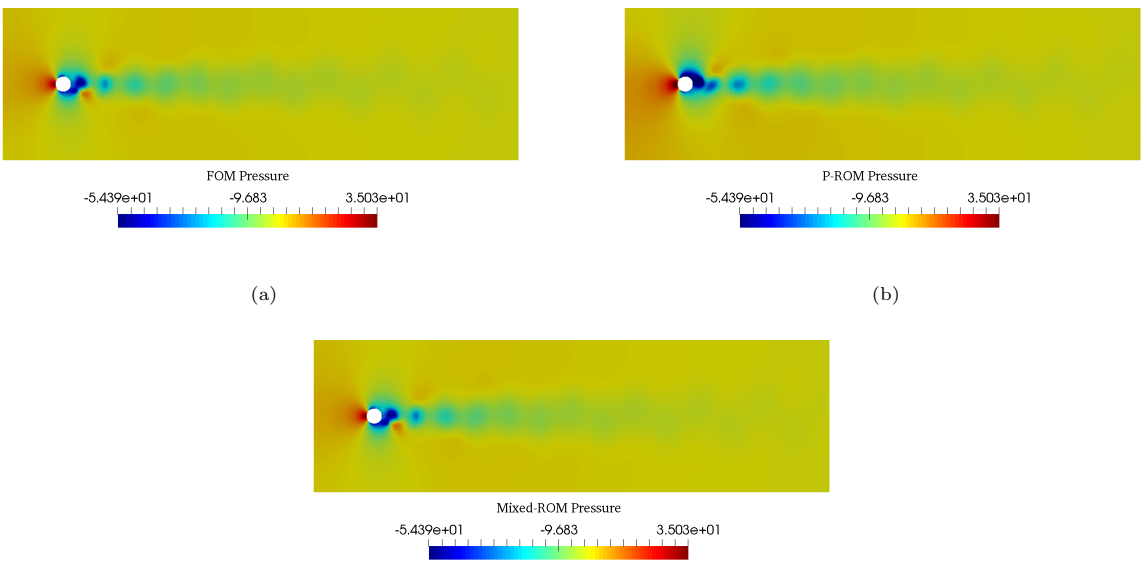
Figure 14: Cumulative ignored eigenvalues decay. In the plot, the solid red line refers to the velocity eigenvalues, the dashed black line indicates the pressure eigenvalues and the dash-dotted blue line finally refers to the eddy viscosity eigenvalues.



(a) (b)

(c)

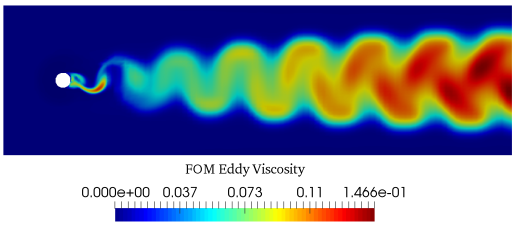
Figure 15: Velocity fields for the parameter value $U_{in} = 7.75$ m/s at $t = 2.8$ s: (a) shows the FOM velocity, while in (b) one can see the P-ROM velocity, and finally in (c) we have the Mixed-ROM velocity.



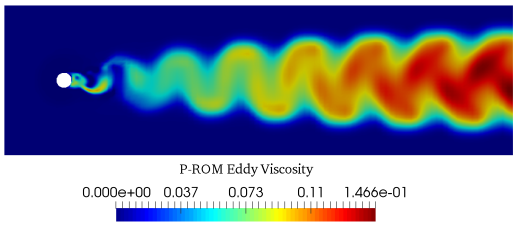
(a) (b)

(c)

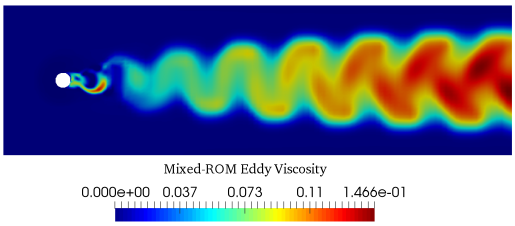
Figure 16: Pressure fields for the parameter value $U_{in} = 7.75$ m/s at $t = 2.8$ s: (a) shows the FOM pressure, while in (b) one can see the P-ROM pressure, and finally in (c) we have the Mixed-ROM pressure.



(a)

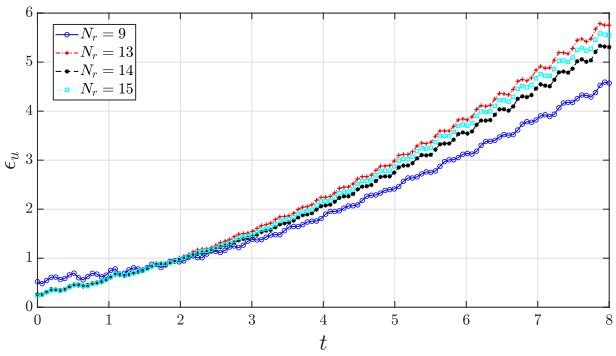


(b)

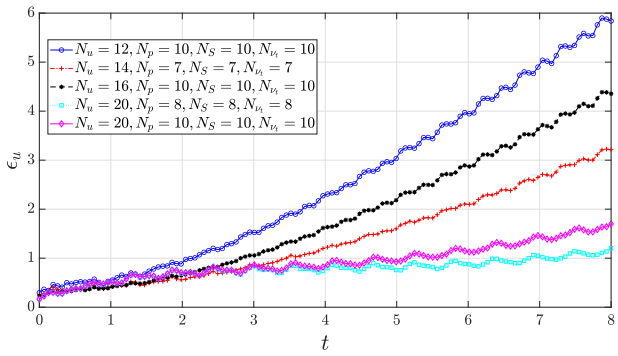


(c)

Figure 17: Eddy viscosity fields for the parameter value $U_{in} = 7.75$ m/s at $t = 2.8$ s: (a) shows the FOM eddy viscosity, while in (b) one can see the P-ROM eddy viscosity, and finally in (c) we have the Mixed-ROM eddy viscosity.



(a)



(b)

Figure 18: The time evolution of the L^2 relative errors of the velocity reduced approximations for both the P-ROM and the Mixed-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the P-ROM model. Figure (b) depicts the case of the Mixed-ROM model. The error values in both graphs are in percentages.

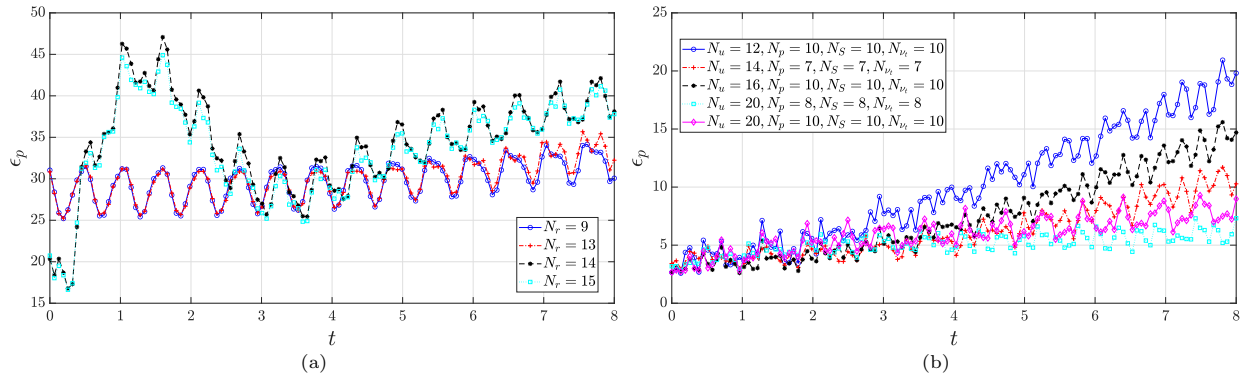


Figure 19: The time evolution of the L^2 relative errors of the pressure reduced approximations for both the P-ROM and the Mixed-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the P-ROM model. Figure (b) depicts the case of the Mixed-ROM model. The error values in both graphs are in percentages.

cylinder one here considered, is often the evaluation of a force acting on a body or a boundary surface in general. As such forces depend on the local values of the pressure and velocity fields around the body of interest, global error evaluators shown so far might not be good indicators if the aim is that of assessing how well the ROM solvers are able to predict the fluid dynamic forces acting on a body. In the case of the present numerical test for instance, a considerable pressure or velocity error localized in the small region around the cylinder might have a substantial impact on the forces values, while having little effect on the global fields errors. For such reason, the following analysis considers the time evolution of the lift coefficient C_l, *i.e.*: the non-dimensionalized vertical component of the fluid dynamic force acting on the cylinder. It is important to point out that the lift and drag forces exerted by the fluid on the cylinder are not a direct result of the Mixed-ROM computations. The reduced system solution consists in fact in the modal coefficients of the velocity and pressure fields at each time instant, which are in turn used to obtain the Mixed-ROM approximation of the full rank flow field. Such approximation can be obviously used to obtain — through integration of pressure and skin friction on the cylinder surface — the reduced order approximation of the fluid dynamic force components and the corresponding force non-dimensional coefficients. Yet, in the reduced order model community this procedure is typically avoided, as it involves a possibly expensive operation such as the evaluation of the full rank flow field. For this reason, the lift and drag coefficients in this work are computed in a fully reduced order fashion, based on the offline computation of suitable matrices which are then used in the online stage. The detailed procedure for online fluid dynamic forces computation is explained in Appendix B. To provide an evaluation of the reduced model C_l approximation throughout the whole time integration, Figure 20 depicts time evolution of the lift coefficient obtained with the FOM, the P-ROM and the Mixed-ROM solvers for the inlet velocity parameter value $U_{in} = 7.75$ m/s. The left plot (a) shows the C_l values for the full time range under consideration which is $[0, 8]$ s. The right diagram (b) represents a detail of the last three cycles of the time span. The plots clearly indicate that the Mixed-ROM model outperforms the P-ROM model in the C_l approximation. The Mixed-ROM C_l curves seem in fact to closely approximate the FOM lift coefficient ones. The P-ROM C_l approximations are instead not completely accurate and it is evident that the P-ROM suffers from instability issues, as through time integration the P-ROM curve diverges from the FOM one. More quantitative assessment of the lift coefficient accuracy during the time integration is obtained through the evaluation of the L^2 relative percentage error, in the integration time interval $[T1, T2]$, between the reduced model approximations of the lift coefficients and their FOM counterparts, namely

$$\epsilon_{C_L} = \frac{\left\| C_l(t) - C_l{}^*(t) \right\|_{L_2(T_1,T_2)}}{\left\| C_l(t) \right\|_{L_2(T_1,T_2)}} \times 100\%. \tag{65}$$

Here, $C_l(t)$ is the time signal of the values of the FOM lift coefficients at all time instants between T_1 and T_2.

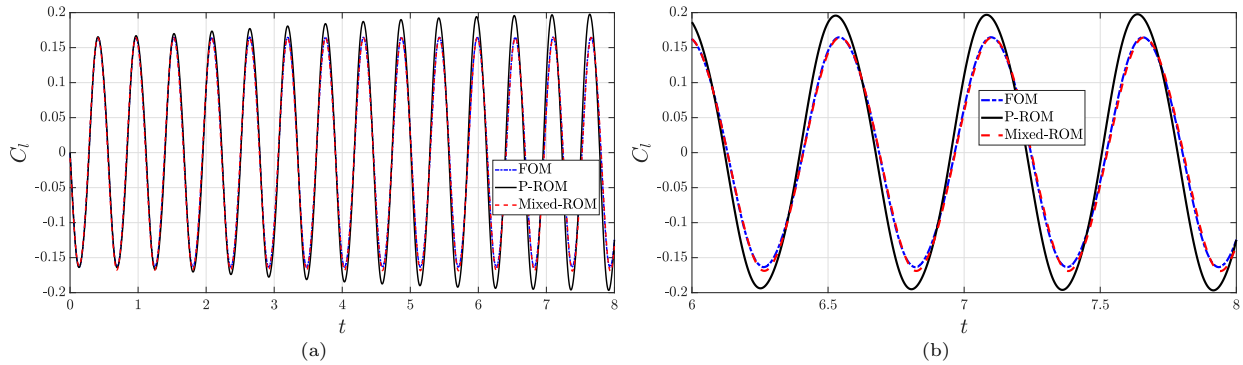


Figure 20: Lift coefficients curves for the cross validation test done for the parameter value $U_{in} = 7.75$ m/s for the time range $[0, 8]$ s, the figure shows the FOM, the P-ROM and the Mixed-ROM lift coefficients histories : (a) the full range is shown (b) the last 2 s C_l is shown.

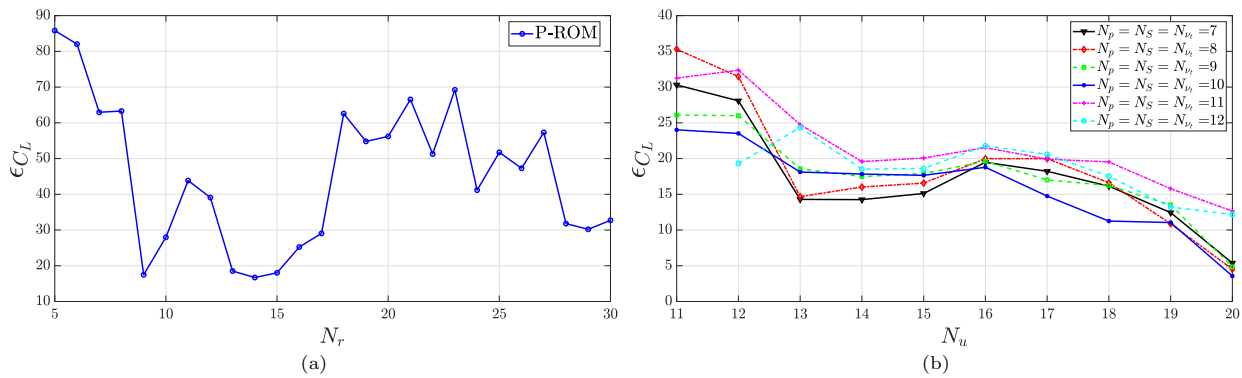


Figure 21: The graph of the L^2 relative errors for the lift coefficients curve versus number of modes used in the online stage in both cases of the P-ROM and the Mixed-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s. The error is computed between the lift coefficients curve obtained by the FOM solver and the one reconstructed from both the P-ROM and the Mixed-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the P-ROM model, where N_r is the number of modes used in the online stage for all variables (by construction of the P-ROM it is not possible to choose different number of online modes for the reduced variables). Figure (b) depicts the case of the Mixed-ROM model, where one can see the error values varying the number of modes used for the pure velocity with different fixed settings for the three other variables (the pressure, the supremizers and the eddy viscosity). The error values in both graphs are in percentages.

On the other hand ${C_l}^*(t)$ is the time evolution of the lift coefficients computed by the reduced order model — whether P-ROM or Mixed-ROM. Figure 21 depicts the L^2 relative errors between the reduced model approximation of the lift coefficients and their FOM counterparts, as a function of the online phase modes employed. As expected from the previous figures the convergence plots highlight that the Mixed-ROM model is able to reproduce the FOM force coefficient with significantly greater accuracy than the ROM. In fact, the Mixed-ROM error reaches values as low as 3 %, while the ROM C_l are consistently above 16 % off the FOM values.

To further analyze the results in Figure 21, we also attempt to assess how much the curve L^2 error is due to incorrect reproduction of the amplitude or frequency of the lift coefficient oscillations. To evaluate, from quantitative perspective, the accuracy of the lift coefficients peak prediction, we define the relative peak error ϵ_{peak} as follows:

$$\epsilon_{n,peak} = \frac{PK_{n,FOM} - PK_{n,*}}{PK_{n,FOM}} \times 100\%, \tag{66}$$

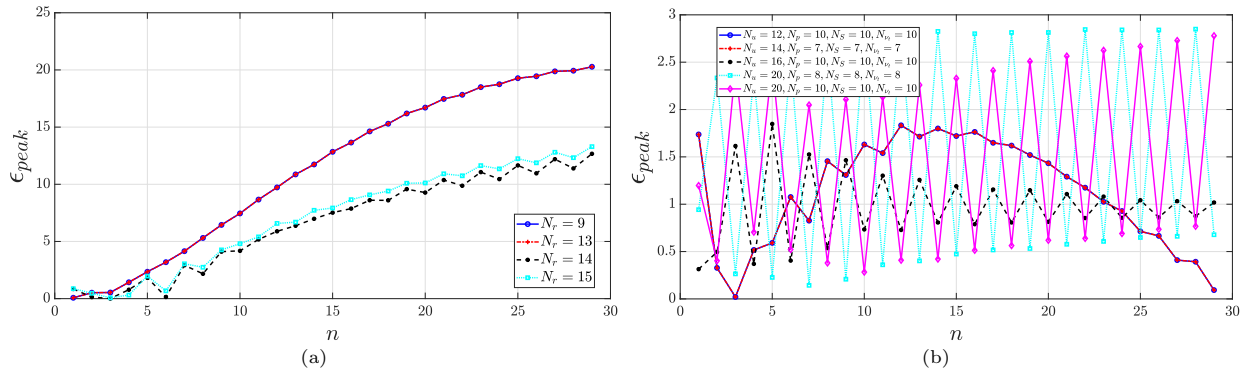


Figure 22: The graph of the peaks relative errors for the lift coefficients curves for varied values of the number of modes used in the online stage in both cases of the P-ROM and the Mixed-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s. The error is computed between the peaks values of the lift coefficients curve obtained by the FOM solver and the ones reconstructed from both the P-ROM and the Mixed-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the P-ROM model, where N_r is the number of modes used in the online stage for all variables (by construction of the P-ROM it is not possible to choose different number of online modes for the reduced variables). Figure (b) depicts the case of the Mixed-ROM model. The error values in both graphs are in percentages.

where $PK_{n,FOM}$ is the value of the $n-$th FOM C_l peak and $PK_{n,*}$ is the value of the $n-$th P-ROM or Mixed-ROM C_l peak. The relative peak error is plotted for both P-ROM and Mixed-ROM models in Figure 22. Figure 22 presents the relative peak error values obtained for each of the 29 peaks the time interval $[0, 8]$ s. We point out that the values of modal truncation order N_r in correspondence of which the peak errors are computed are the resulted most accurate in the relative L^2 lift error analysis presented. Figure 22 suggests that the P-ROM peaks relative error grows in time and settles around values as high as $10 - 20$ %. On the other hand, the corresponding error values for the Mixed-ROM model are less than 3.5 % for several different modal truncation order for velocity, pressure, supremizers and eddy viscosity.

A final numerical test is aimed at assessing the accuracy of the Mixed-ROM model for higher Reynolds value. The inlet velocity parameter sample considered in this case is $U_{in} = 11.75$ m/s. The time interval considered for the reduced order simulations is $t \in [0, 10]$ s which contains around 27 solution cycles. The results reported for this case are relative to the lift coefficient history, the L^2 relative error value for C_l, the C_l peaks error and the approximated time period by the Mixed-ROM.

The Mixed-ROM dynamical system is solved with time step equal to 0.00025 s. The Mixed-ROM fields were reproduced using 12 modes for velocity and 10 for each of pressure, supremizers and eddy viscosity. The C_l curves obtained with the FOM solver and the Mixed-ROM model are presented in Figure 23. The results in Figure 23 prove that the Mixed-ROM was successful in reducing the problem with satisfactory accuracy. The L^2 relative error between FOM and Mixed-ROM solution is in fact 1.9654%. As for the relative peak error, the highest value detected in the $[0, 10]$ s time interval is 2.0672%. Finally the average time period computed in by the FOM solver is about 0.3641 s, while the average time period computed by the Mixed-ROM is roughly 0.3642 s. The corresponding 0.2% relative error suggests that the main source of error in the ROM predictions is due to the amplitude inaccuracies rather then to incorrect frequency reconstructions.

The final test in this section is meant to assess one of the main objectives of this work, that is to test the presented reduction approach for the variance of the FOM turbulence model. We considered the non-parametrized case of $Re = 10^5$, the FOM was run for both $k - \epsilon$ and SST $k - \omega$ models. After having reached the fully periodic regime, snapshots were taken for the first 1.2 s and 1.6 s for $k - \epsilon$ and SST $k - \omega$ models, respectively. The reduction was done extrapolating in time, where the Mixed-ROM simulations were run for 8 s. The lift coefficient curves are shown in Figure 24, where one can see both FOM C_l signals for the two different turbulence models and their reduced counterparts. It is evident from the graph that

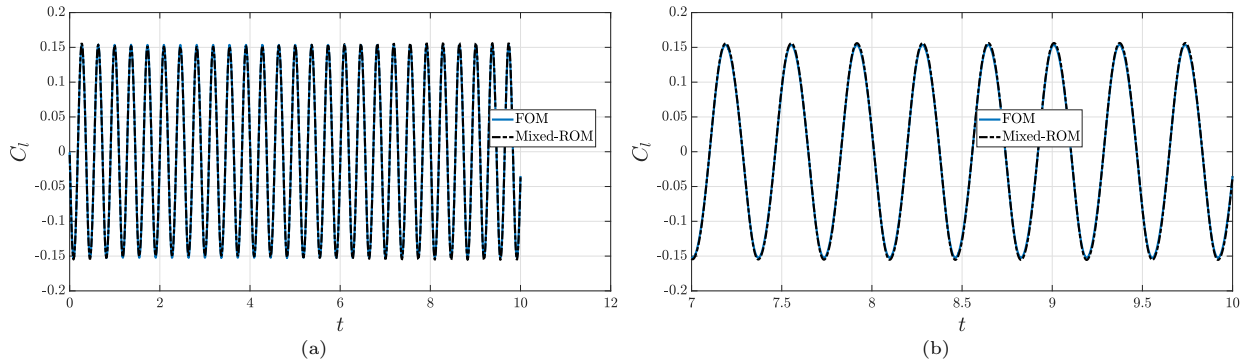


Figure 23: Lift coefficients curves for the cross validation test done for the parameter value $U_{in} = 11.75$ m/s for the time range $[0, 10]$ s, the figure shows the FOM and the Mixed-ROM lift coefficients histories : (a) the full range is shown (b) the last 3 s history of C_l is shown.

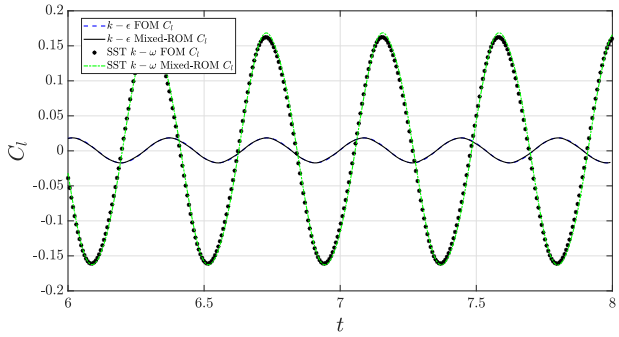


Figure 24: The lift coefficient curves obtained using both $k-\epsilon$ and SST $k-\omega$ turbulence models and the Mixed-ROM ones. The case considered is a non-parametrized one with $U_{in} = 10$ m/s corresponding to $Re = 10^5$. The plot is for the time range $t \in [6, 8]$, the Mixed-ROM achieved relative L^2 errors (over the range $t \in [0, 8]$) which are less than 5 % in both cases.

the Mixed-ROM proves sensitive to the specific turbulence model used in the FOM solver, although no additional PDEs for the turbulent quantities are solved at the reduced level.

5. Conclusions and Outlook

This work presents a hybrid data-driven/projection-based approach to reduce turbulent flows. The approach developed in this work called Mixed-ROM is based on introducing a non-intrusive reduced order version of the eddy viscosity field to the formulation of the reduced order model. The Mixed-ROM employs interpolation using radial basis function in the online stage for the computation of the reduced order eddy viscosity coefficients. This interpolation can be done with the independent variable being the combined time-parameter vector or the combined vector of the velocity L^2 projection coefficients and their vector time derivatives. The Mixed-ROM proved to be accurate in reconstructing the fluid dynamics fields in both cases of steady and unsteady flows with a Reynolds number on the order of 10^5. In the unsteady case considered in this work which is the flow around a circular cylinder, the Mixed-ROM showed that it is capable of reconstructing an important variable of interest that is the lift coefficient time history which mainly depends on local flow features around the cylinder. In that same example, the Mixed-ROM gives satisfactory results when it comes to the extrapolation in time. The Mixed-ROM has been able to obtain accurate predictions with an acceptable computational cost, showing with a speed up of $SU = 10$ in the unsteady case and around $SU = 1000$ in the steady case.

As for potential future work, data-driven techniques can be used in building reduced order models and other methodologies could help in approximating certain maps which are needed for the ultimate goal of

reducing CFD problems. Such methodologies include the Artificial Neural Networks (ANN) with which one could potentially improve the accuracy of the approximation of the eddy viscosity coefficients conducted in this work. Another idea is to use DMD for the extrapolation problem for the unsteady flows. In addition, there is a need to find stabilization techniques for the long time integration problem for unsteady flows [23] and for multi-physics problems [100, 101, 102, 103].

6. Acknowledgements

We acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD and INdAM-GNCS projects. In addition, this work was partially performed in the framework of the project SOPHYA - "Seakeeping Of Planing Hull YAchts" and of the project PRELICA - "Advanced methodologies for hydro-acoustic design of naval propulsion", both supported by Regione FVG, POR-FESR 2014-2020, Piano Operativo Regionale Fondo Europeo per lo Sviluppo Regionale.

Appendix A. List of abbreviations and symbols

Nomenclature

Abbreviations

EVM	Eddy Viscosity Models
FEM	Finite Element Methods
FVM	Finite Volume Methods
Mixed-ROM	The mixed projection/data-driven based reduced order model developed in this work
POD	Proper Orthogonal Decomposition
RANS	Reynolds Average Navier-Stokes
RBF	Radial Basis Functions
RB	Reduced Basis
ROM	Reduced Order Model

Symbols

$(\boldsymbol{\nabla u})_f$	the gradient of $\boldsymbol{u}$ at the faces
$\times$	Cartesian product
$\boldsymbol{\delta}$	a matrix calculated in the offline stage that represents the contribution of viscous forces acting on a surface in the domain
$\boldsymbol{\lambda^u}$	eigenvalues matrix of the correlation matrix of the velocity field snapshot matrix
$\tilde{\boldsymbol{\mathcal{P}}}$	homogenized pressure snapshots matrix
$\tilde{\boldsymbol{\mathcal{U}}}$	homogenized velocity snapshots matrix
$\boldsymbol{\mu}^*$	The sample parameter introduced to the ROM in the online stage
$\boldsymbol{\nabla}$	gradient operator
$\boldsymbol{\nabla}\cdot$	divergence operator
$\boldsymbol{\phi_L}$	The matrix of the lifting functions $\boldsymbol{\phi_{L_{i,j}}}$
$\boldsymbol{\phi}_i$	i-th POD basis function for velocity
$\boldsymbol{\theta}$	a matrix calculated in the offline stage that represents the contribution of pressure forces acting on a surface in the domain
$\boldsymbol{a}$	reduced vector of unknowns for velocity
$\boldsymbol{B_T}$	ROM diffusion turbulent matrix
$\boldsymbol{B}$	ROM diffusion matrix
$\boldsymbol{b}$	reduced vector of unknowns for pressure
$\boldsymbol{C^u}$	correlation matrix of the velocity field snapshot matrix
$\boldsymbol{C_{T1}}$	ROM turbulent tensor
$\boldsymbol{C_{T2}}$	ROM turbulent tensor
$\boldsymbol{C}$	ROM convection tensor
$\boldsymbol{D^k}$	a vector in the penalty method for treatment of the boundary conditions at reduced order level, that contains the values of the L^2 norms of the velocity POD modes at the Γ_{Dk} in the Dirichlet boundary
$\boldsymbol{E^k}$	a matrix in the penalty method for treatment of the boundary conditions at reduced order level, that contains the values of the L^2 scalar products of the velocity POD modes at the Γ_{Dk} in the Dirichlet boundary

$\boldsymbol{g}$	reduced vector of unknowns for eddy viscosity
$\boldsymbol{H}$	ROM pressure gradient matrix
$\boldsymbol{M}$	ROM mass matrix
$\boldsymbol{n}$	outward normal vector
$\boldsymbol{P}$	ROM divergence matrix
$\boldsymbol{U_{BC}}$	the vector of non-zero boundary velocity values which are imposed as non-homogeneous Dirichlet conditions at the Dirichlet boundary
$\boldsymbol{u}$	velocity field
$\boldsymbol{u}'$	fluctuating velocity field in RANS equations
$\boldsymbol{u}_f$	the velocity vector evaluated at the centre of each face of the control volume
$\boldsymbol{u}_N$	the velocity at the centre of one neighboring cell
$\boldsymbol{u}_P$	the velocity at the centre of one neighboring cell
$\boldsymbol{V^u}$	eigenvectors matrix of the correlation matrix of the velocity field snapshot matrix
$\boldsymbol{x}^i_{\boldsymbol{\mu},t}$	i-th member of $X_{\boldsymbol{\mu},t}$
$\boldsymbol{Y}_L$	the vector of observed outputs (the L^2 projection coefficients of the L-th viscosity mode onto the snapshots) for the interpolation procedure
$\boldsymbol{z}$	The generic parameter-time vector which lives in $\mathbb{R}^{q+1}$
$\boldsymbol{z}^*$	The sample parameter combined with the time instant at which Mixed-ROM solution is desired in the online stage
$\boldsymbol{\mathcal{S}_p}$	snapshots matrix for the pressure field
$\boldsymbol{\mathcal{S}_u}$	snapshots matrix for the velocity field
$\boldsymbol{\mathcal{S}_{\nu_t}}$	snapshots matrix for the eddy viscosity field
$\boldsymbol{\phi_{Li,j}}$	The velocity lifting function that has unitary value in its i-th component at Γ_{Dj}
χ_i	i-th POD basis function for pressure
ϵ	turbulent dissipation
ϵ_p	the L^2 relative error between the FOM velocity field and a reduced order pressure field
ϵ_u	the L^2 relative error between the FOM velocity field and a reduced order velocity field
ϵ_{C_L}	the L^2 relative error between the FOM lift coefficient time signal and the reduced order reconstructed one
η_i	i-th POD basis function for eddy viscosity
$\mathbb{V}_{POD}$	POD space for velocity
$\mathcal{P}$	parameter space of dimension q
$\mathcal{P}_M$	training set in the parameter space $\mathcal{P}$ with cardinality of M
ν_t	eddy viscosity field
ω	specific turbulent dissipation rate
$\otimes$	Tensor product
$\overline{\boldsymbol{u}}$	mean velocity field in RANS equations
τ	The penalization factor in the penalty boundary treatment method
$\zeta_{i,j}$	The RBF functions used in interpolating the i-th eddy viscosity coefficient in the expansion and centered at the j-th element of $X_{\boldsymbol{\mu},t}$
C_l	the lift coefficient which corresponds to the force component in the lift direction L
k	turbulence kinetic energy
L	The component that is parallel to the lift direction of the force which acts on a certain surface
N_p	number of modes used in the online phase for pressure
N_S	number of modes used in the online phase for the supremizer
N_s	total number of snapshots
N_u	number of modes used in the online phase for velocity
N_{ν_t}	number of modes used in the online phase for eddy viscosity
N_{BC}	The number of scalar non-zero boundary conditions needed to be set at reduced order level
q	the dimension of the parameter space $\mathcal{P}$
T	final time
V_i	control volume in the mesh

$w_{i,j}$	The weight of the j-th RBF used in interpolating the i-th eddy viscosity coefficient in the expansion.
$X_{\boldsymbol{\mu},t}$	the combined set of samples and time instants at which snapshots are taken
$(\cdot,\cdot)_{L^2(\Omega)}$	inner product in $L^2(\Omega)$
Γ_0	the physical walls boundary part of Γ
Γ_{Dj}	The j-th part of the boundary where Dirichlet conditions are imposed
Γ_{In}	the inlet boundary part of Γ
Γ_{Out}	the outlet boundary part of Γ
Γ	boundary of Ω
ν	dimensionless kinematic viscosity
Ω	bounded domain
M	number of parameter samples in the training set $\mathcal{P}_M$
N_p^h	number of unknowns for pressure at full-order level
N_u^h	number of unknowns for velocity at full-order level
$N_{\nu_t}^h$	number of unknowns for eddy viscosity at full-order level
p	pressure field

Appendix B. Lift and drag forces offline/online computations

This section introduces the computations done in both the offline and the online stages for obtaining the surface forces acting on a part of the domain called $\partial\Omega_f$.

The total viscous and pressure forces $\boldsymbol{F}$ acting on $\partial\Omega_f$ are given by the following integral:

$$\boldsymbol{F} = \int_{\partial\Omega_f} (2\mu\boldsymbol{\nabla}\boldsymbol{u} - p\boldsymbol{I})\boldsymbol{n}ds. \tag{B.1}$$

In many application in fluid dynamics it is very important to efficiently compute the forces acting on certain objects inside the domain. For instance the problem of flow past a circular cylinder considered in this work is one of them. One should avoid resorting to the full order mesh for computing the integral above because this makes the approach not entirely a reduced one.

The first step in developing an offline/online decoupling approach for computing the forces is to insert the approximation (15) into (B.1), this yields the following:

$$\boldsymbol{F} = \int_{\partial\Omega_f} (2\mu\boldsymbol{\nabla}(\sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x})) - \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i\boldsymbol{I})\boldsymbol{n}ds, \tag{B.2}$$

$$\boldsymbol{F} = \int_{\partial\Omega_f} 2\mu\sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})\boldsymbol{n}ds - \int_{\partial\Omega_f}\sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i\boldsymbol{n}ds, \tag{B.3}$$

$$\boldsymbol{F} = \sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\int_{\partial\Omega_f} 2\mu\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})\boldsymbol{n}ds - \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\int_{\partial\Omega_f}\chi_i\boldsymbol{n}ds. \tag{B.4}$$

After reaching this point one can define the following quantities

$$\boldsymbol{\delta}_i = \int_{\partial\Omega_f} 2\mu\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})\boldsymbol{n}ds, \quad \text{for} \quad i = 1, ..., N_u, \tag{B.5}$$

$$\boldsymbol{\theta}_j = \int_{\partial\Omega_f} \chi_j\boldsymbol{n}ds, \quad \text{for} \quad j = 1, ..., N_p, \tag{B.6}$$

where each term of $\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})$ and χ_j can be seen as velocity and pressure field, respectively. This will make the computations of (B.5) and (B.6) possible in the offline stage and they will be stored in order to be later used in the online stage.

In the online stage when a new time-parameter vector $\boldsymbol{z}^*$ is introduced, the forces are computed as follows:

$$\boldsymbol{F}^* = \int_{\partial\Omega_f} (2\mu \boldsymbol{\nabla} \boldsymbol{u}(\boldsymbol{z}^*, \boldsymbol{x}) - p(\boldsymbol{z}^*, \boldsymbol{x})\boldsymbol{I})\boldsymbol{n} ds, \tag{B.7}$$

which simplify to

$$\boldsymbol{F}^* = \sum_{i=1}^{N_u} a_i(\boldsymbol{z}^*)\boldsymbol{\delta}_i - \sum_{j=1}^{N_p} b_j(\boldsymbol{z}^*)\boldsymbol{\theta}_j. \tag{B.8}$$

References

References

[1] F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab, 1st Edition, Springer Publishing Company, Incorporated, 2015. doi:10.1007/978-3-319-16874-6.

[2] H. K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics. The Finite Volume Method, Longman Group Ltd., London, 1995.

[3] J. S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer International Publishing, 2016. doi:10.1007/978-3-319-22470-1.

[4] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, Springer International Publishing, 2016. doi:10.1007/978-3-319-15431-2.

[5] P. Benner, M. Ohlberger, A. Pater, G. Rozza, K. Urban, Model Reduction of Parametrized Systems., Vol. 1st ed. 2017 of MS&A series, Springer, 2017.

[6] P. Benner, S. Gugercin, K. Willcox, A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems, SIAM Review 57 (4) (2015) 483–531. doi:10.1137/130932715.

[7] E. Bader, M. Kärcher, M. A. Grepl, K. Veroy, Certified Reduced Basis Methods for Parametrized Distributed Elliptic Optimal Control Problems with Control Constraints, SIAM Journal on Scientific Computing 38 (6) (2016) A3921–A3946. doi:10.1137/16m1059898.

[8] M. Balajewicz, E. H. Dowell, Stabilization of projection-based reduced order models of the Navier–Stokes, Nonlinear Dynamics 70 (2) (2012) 1619–1632. doi:10.1007/s11071-012-0561-5.

[9] D. Amsallem, C. Farhat, Stabilization of projection-based reduced-order models, International Journal for Numerical Methods in Engineering 91 (4) (2012) 358–377. doi:10.1002/nme.4274.

[10] K. Carlberg, M. Barone, H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, Journal of Computational Physics 330 (2017) 693–734. doi:10.1016/j.jcp.2016.10.033.

[11] P. Moin, K. Mahesh, Direct Numerical Simulation: A tool in turbulence research, Annual Review of Fluid Mechanics 30 (1) (1998) 539–578. doi:10.1146/annurev.fluid.30.1.539.

[12] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison, Computer Methods in Applied Mechanics and Engineering 237–240 (2012) 10 – 26. doi:http://dx.doi.org/10.1016/j.cma.2012.04.015.

[13] N. Aubry, P. Holmes, J. L. Lumley, E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, Journal of Fluid Mechanics 192 (-1) (1988) 115. doi:10.1017/s0022112088001818.

[14] M. Couplet, P. Sagaut, C. Basdevant, Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow, Journal of Fluid Mechanics 491 (2003) 275–284. doi:10.1017/s0022112003005615.

[15] A. Iollo, S. Lanteri, J.-A. Désidéri, Stability Properties of POD-Galerkin Approximations for the Compressible Navier-Stokes Equations, Theoretical and Computational Fluid Dynamics 13 (6) (2000) 377–396. doi:10.1007/s001620050119.

[16] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations, International Journal for Numerical Methods in Engineering 86 (2) (2010) 155–181. doi:10.1002/nme.3050.

[17] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics 242 (2013) 623–647. doi:10.1016/j.jcp.2013.02.028.

[18] A. Tallet, C. Allery, C. Leblond, E. Liberge, A minimum residual projection to build coupled velocity–pressure POD–ROM for incompressible Navier–Stokes equations, Communications in Nonlinear Science and Numerical Simulation 22 (1-3) (2015) 909–932. doi:10.1016/j.cnsns.2014.09.009.

[19] A. Alla, J. N. Kutz, Nonlinear model order reduction via dynamic mode decomposition, SIAM Journal on Scientific Computing 39 (5) (2017) B778–B796. doi:10.1137/16m1059308.

[20] N. Demo, M. Tezzele, G. Gustin, G. Lavini, G. Rozza, Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition, in: Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research, IOS Press, 2018, pp. 212–219. doi:10.3233/978-1-61499-870-9-212.

[21] M. Tezzele, N. Demo, M. Gadalla, A. Mola, G. Rozza, Model order reduction by means of active subspaces and dynamic mode decomposition for parametric hull shape design hydrodynamics, Stand Alone 0 (Technology and Science for the Ships of the Future) (2018) 569–576. doi:10.3233/978-1-61499-870-9-569.

[22] S. Le Clainche, J. M. Vega, Higher order dynamic mode decomposition, SIAM Journal on Applied Dynamical Systems 16 (2) (2017) 882–925.

[23] L. Fick, Y. Maday, A. T. Patera, T. Taddei, A stabilized POD model for turbulent flows over a range of reynolds numbers: Optimal parameter sampling and constrained projection, Journal of Computational Physics 371 (2018) 214–243. doi:10.1016/j.jcp.2018.05.027.

[24] J. L. Eftang, D. J. Knezevic, A. T. Patera, An hp certified reduced basis method for parametrized parabolic partial differential equations, Mathematical and Computer Modelling of Dynamical Systems 17 (4) (2011) 395–422. doi:10.1080/13873954.2011.547670.

[25] T. R. Chacón, E. Á. Delgado, M. M. Gómez, F. Ballarin, G. Rozza, On a certified Smagorinsky reduced basis turbulence model, SIAM Journal on Numerical Analysis 55 (6) (2017) 3047–3067. doi:10.1137/17m1118233.

[26] J. Smagorinsky, General Circulation Experiments with the Primitive Equations , Monthly Weather Review 91 (3) (1963) 99–164. doi:10.1175/1520-0493(1963)091<0099:gcewtp>2.3.co;2.

[27] S. Ullman, J. Lang, A pod-galerkin reduced model with updated coefficients for smagorinsky les, 2010.

[28] B. R. Noack, L. Cordier, xAMC - a Toolkit for Analysis, Modelling and Control of Fluid Flows (Version 3.0), Technical Report 01/2012.

[29] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Two-level discretizations of nonlinear closure models for proper orthogonal decomposition, Journal of Computational Physics 230 (1) (2011) 126–146. doi:10.1016/j.jcp.2010.09.015.

[30] B. R. Noack, L. Cordier, Low-dimensional Galerkin model of a laminar shear-layer, Tech. Rep.2002-01.

[31] M. Bergmann, C.-H. Bruneau, A. Iollo, Improvement of reduced order modeling based on POD, in: Computational Fluid Dynamics 2008, Springer Berlin Heidelberg, 2009, pp. 779–784. doi:10.1007/978-3-642-01273-0_103.

[32] G. Stabile, F. Ballarin, G. Zuccarino, G. Rozza, A reduced order variational multiscale approach for turbulent flows, Advances in Computational Mathematics 45 (5-6) (2019) 2349–2368. doi:10.1007/s10444-019-09712-x.

[33] T. R. Chacón, E. Á. Delgado, M. M. Gómez, S. Rubino, Assessment of self-adapting local projection-based solvers for laminar and turbulent industrial flows, Journal of Mathematics in Industry 8 (1). doi:10.1186/s13362-018-0045-4.

[34] F. Ballarin, T. R. Chacón, E. Á. Delgado, M. M. Gómez, G. Rozza, Certified reduced basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height (2019).

[35] M. Yano, Discontinuous galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws, Advances in Computational Mathematics 45 (5-6) (2019) 2287–2320. doi:10.1007/s10444-019-09710-z.

[36] A. C. Ionita, A. C. Antoulas, Data-driven parametrized model reduction in the loewner framework, SIAM Journal on Scientific Computing 36 (3) (2014) A984–A1007. doi:10.1137/130914619.

[37] B. Peherstorfer, K. Willcox, Dynamic data-driven reduced-order models, Computer Methods in Applied Mechanics and Engineering 291 (2015) 21–41. doi:10.1016/j.cma.2015.03.018.

[38] J.-C. Loiseau, S. L. Brunton, Constrained sparse Galerkin regression, Journal of Fluid Mechanics 838 (2018) 42–67. doi:10.1017/jfm.2017.823.

[39] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. S. Henningson, Spectral analysis of nonlinear flows, Journal of Fluid Mechanics 641 (2009) 115. doi:10.1017/s0022112009992059.

[40] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, R. K. Niven, Cluster-based reduced-order modelling of a mixing layer, Journal of Fluid Mechanics 754 (2014) 365–414. doi:10.1017/jfm.2014.355.

[41] M. Guo, J. S. Hesthaven, Reduced order modeling for nonlinear structural analysis using gaussian process regression, Computer Methods in Applied Mechanics and Engineering 341 (2018) 807–826. doi:10.1016/j.cma.2018.07.017.

[42] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, Journal of Computational Physics 363 (2018) 55–78. doi:10.1016/j.jcp.2018.02.037.

[43] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, Journal of Fluid Mechanics 497 (2003) 335–363. doi:10.1017/s0022112003006694.

[44] M. Guo, J. S. Hesthaven, Data-driven reduced order modeling for time-dependent problems, Computer Methods in Applied Mechanics and Engineering 345 (2019) 75–99. doi:10.1016/j.cma.2018.10.029.

[45] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA Journal 32 (8) (1994) 1598–1605. doi:10.2514/3.12149.

[46] W. Jones, B. Launder, The prediction of laminarization with a two-equation model of turbulence, International Journal of Heat and Mass Transfer 15 (2) (1972) 301–314. doi:10.1016/0017-9310(72)90076-2.

[47] X. Xie, M. Mohebujjaman, L. G. Rebholz, T. Iliescu, Data-driven filtered reduced order modeling of fluid flows, SIAM Journal on Scientific Computing 40 (3) (2018) B834–B857. doi:10.1137/17m1145136.

[48] B. Galletti, C. H. Bruneau, L. Zannetti, A. Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, Journal of Fluid Mechanics 503 (2004) 161–170. doi:10.1017/s0022112004007906.

[49] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, Journal of Computational Physics 207 (1) (2005) 192–220. doi:10.1016/j.jcp.2005.01.008.

[50] F. Lu, K. K. Lin, A. J. Chorin, Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation, Physica D: Nonlinear Phenomena 340 (2017) 46–57. doi:10.1016/j.physd.2016.09.007.

[51] B. R. Noack, P. Papas, P. A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, Journal of Fluid Mechanics 523 (2005) 339–365. doi:10.1017/S0022112004002149.

[52] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering 306 (2016) 196–215. doi:10.1016/j.cma.2016.03.025.
[53] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in physics 12 (6) (1998) 620–631. doi:10.1063/1.168744.
[54] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College, University of London (1996).
[55] S. Patankar, D. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer 15 (10) (1972) 1787 – 1806. doi:http://dx.doi.org/10.1016/0017-9310(72)90054-3.
[56] R. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1) (1986) 40–65. doi:10.1016/0021-9991(86)90099-9.
[57] L. C. Berselli, T. Iliescu, W. J. Layton, Mathematics of large eddy simulation of turbulent flows, Springer Science & Business Media, 2005.
[58] P. Sagaut, Large eddy simulation for incompressible flows: an introduction, Springer Science & Business Media, 2006.
[59] J. Boussinesq, Essa sur latheories des eaux courantes. memoires presentes par divers savants a l'academic des sciences de l'institut national de france, Tome XXIII (1).
[60] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 1992. doi:10.2514/6.1992-439.
[61] K. Hanjalic, B. E. Launder, A Reynolds stress model of turbulence and its application to thin shear flows, Journal of Fluid Mechanics 52 (04) (1972) 609. doi:10.1017/s002211207200268x.
[62] G. Stabile, G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations, Computers & Fluids 173 (2018) 273–284. doi:10.1016/j.compfluid.2018.01.035.
[63] A. Dumon, C. Allery, A. Ammar, Proper General Decomposition (PGD) for the resolution of Navier-Stokes equations, Journal of Computational Physics 230 (4) (2011) 1387–1407. doi:10.1016/j.jcp.2010.11.010.
[64] F. Chinesta, P. Ladeveze, E. Cueto, A Short Review on Model Order Reduction Based on Proper Generalized Decomposition, Archives of Computational Methods in Engineering 18 (4) (2011) 395. doi:10.1007/s11831-011-9064-7.
[65] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics 656 (2010) 5–28. doi:10.1017/s0022112010001217.
[66] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, Mathematical Modelling and Numerical Analysis 42 (2) (2008) 277–302. doi:10.1051/m2an:2008001.
[67] S. Volkwein, Proper orthogonal decomposition: Theory and reduced-order modelling, Lecture Notes, University of Konstanz 4 (4).
[68] M. Bergmann, C.-H. Bruneau, A. Iollo, Enablers for robust POD models, Journal of Computational Physics 228 (2) (2009) 516–538. doi:10.1016/j.jcp.2008.09.024.
[69] J. Baiges, R. Codina, S. Idelsohn, Reduced-order modelling strategies for the finite element approximation of the incompressible Navier-Stokes equations, Computational Methods in Applied Sciences 33 (2014) 189–216. doi:10.1007/978-3-319-06136-8_9.
[70] J. Burkardt, M. Gunzburger, H.-C. Lee, POD and CVT-based reduced-order modeling of navier–stokes flows, Computer Methods in Applied Mechanics and Engineering 196 (1-3) (2006) 337–355. doi:10.1016/j.cma.2006.04.004.
[71] F. Ballarin, G. Rozza, POD-Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems, International Journal for Numerical Methods in Fluids 82 (12) (2016) 1010–1034. doi:10.1002/fld.4252.
[72] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM Journal on Numerical Analysis 40 (2) (2002) 492–515. doi:10.1137/S0036142900382612.
[73] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, G. Rozza, POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder, Communications in Applied and Industrial Mathematics 8 (1) (2017) 210–236. doi:10.1515/caim-2017-0011.
[74] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1136–1161. doi:10.1002/nme.4772.
[75] G. Rozza, K. Veroy, On the stability of the reduced basis method for Stokes equations in parametrized domains, Computer Methods in Applied Mechanics and Engineering 196 (7) (2007) 1244 – 1260. doi:http://dx.doi.org/10.1016/j.cma.2006.09.005.
[76] A. Caiazzo, T. Iliescu, V. John, S. Schyschlowa, A numerical investigation of velocity-pressure reduced order models for incompressible flows, Journal of Computational Physics 259 (2014) 598 – 616. doi:http://dx.doi.org/10.1016/j.jcp.2013.12.004.
[77] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, Computer Methods in Applied Mechanics and Engineering 311 (2016) 151–179. doi:10.1016/j.cma.2016.08.006.
[78] D. Lazzaro, L. B. Montefusco, Radial basis functions for the multivariate interpolation of large scattered data sets, Journal of Computational and Applied Mathematics 140 (1-2) (2002) 521–536. doi:10.1016/s0377-0427(01)00485-x.
[79] C. A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Constructive Approximation 2 (1) (1986) 11–22. doi:10.1007/bf01893414.
[80] Y. Wang, B. Yu, Z. Cao, W. Zou, G. Yu, A comparative study of POD interpolation and POD projection methods for fast and accurate prediction of heat transfer problems, International Journal of Heat and Mass Transfer 55 (17-18) (2012) 4827–4836. doi:10.1016/j.ijheatmasstransfer.2012.04.053.

[81] S. Walton, O. Hassan, K. Morgan, Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions, Applied Mathematical Modelling 37 (20-21) (2013) 8930–8945. doi:10.1016/j.apm.2013.04.025.
[82] F. Salmoiraghi, A. Scardigli, H. Telib, G. Rozza, Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation, International Journal of Computational Fluid Dynamics 32 (4-5) (2018) 233–247. doi:10.1080/10618562.2018.1514115.
[83] S. Hijazi, S. Ali, G. Stabile, F. Ballarin, G. Rozza, The Effort of Increasing Reynolds Number in Projection-Based Reduced Order Methods: from Laminar to Turbulent Flows, Springer LNCSE Vol. 136.
[84] K. Bizon, G. Continillo, Reduced order modelling of chemical reactors with recycle by means of POD-penalty method, Computers & Chemical Engineering 39 (2012) 22–32. doi:10.1016/j.compchemeng.2011.10.001.
[85] I. Babuška, The finite element method with penalty, Mathematics of Computation 27 (122) (1973) 221–221. doi:10.1090/s0025-5718-1973-0351118-5.
[86] J. W. Barrett, C. M. Elliott, Finite element approximation of the dirichlet problem using the boundary penalty method, Numerische Mathematik 49 (4) (1986) 343–366. doi:10.1007/bf01389536.
[87] I. Kalashnikova, M. F. Barone, Efficient non-linear proper orthogonal decomposition/galerkin reduced order models with stable penalty enforcement of boundary conditions, International Journal for Numerical Methods in Engineering 90 (11) (2012) 1337–1362. doi:10.1002/nme.3366.
[88] S. Sirisup, G. Karniadakis, Stability and accuracy of periodic flow solutions obtained by a POD-penalty method, Physica D: Nonlinear Phenomena 202 (3-4) (2005) 218–237. doi:10.1016/j.physd.2005.02.006.
[89] W. R. Graham, J. Peraire, K. Y. Tang, Optimal control of vortex shedding using low-order models. Part I–open-loop model development, International Journal for Numerical Methods in Engineering 44 (7) (1999) 945–972. doi:10.1002/(sici)1097-0207(19990310)44:7<945::aid-nme537>3.0.co;2-f.
[90] M. D. Gunzburger, J. S. Peterson, J. N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Computer Methods in Applied Mechanics and Engineering 196 (4-6) (2007) 1030–1047. doi:10.1016/j.cma.2006.08.004.
[91] S. Hijazi, G. Stabile, A. Mola, G. Rozza, Non-Intrusive Polynomial Chaos Method Applied to Problems in Computational Fluid Dynamics with a Comparison to Proper Orthogonal Decomposition, In press Springer LNCSE .
[92] G. Stabile, G. Rozza, ITHACA-FV - In real Time Highly Advanced Computational Applications for Finite Volumes, http://www.mathlab.sissa.it/ithaca-fv, accessed: 2018-01-30.
[93] B. Grimstad, et al., SPLINTER: a library for multivariate function approximation with splines, http://github.com/bgrimstad/splinter, accessed: 2015-05-16 (2015).
[94] M. M. Zdravkovich, Flow around Circular Cylinders: Volume 1: Fundamentals, Vol. 350, Cambridge University Press, 1997.
[95] M. M. Zdravkovich, Flow around Circular Cylinders: Volume 2: Applications, Vol. 2, Oxford University Press, 2003.
[96] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen differenzengleichungen der mathematischen physik, Mathematische Annalen 100 (1) (1928) 32–74. doi:10.1007/bf01448839.
[97] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, IBM Journal of Research and Development 11 (2) (1967) 215–234. doi:10.1147/rd.112.0215.
[98] V. Strouhal, Ueber eine besondere art der tonerregung, Annalen der Physik und Chemie 241 (10) (1878) 216–251. doi:10.1002/andp.18782411005.
[99] R. D. Blevins, C. S. Coughran, Experimental investigation of vortex-induced vibration in one and two dimensions with variable mass, damping, and reynolds number, Journal of Fluids Engineering 131 (10) (2009) 101202. doi:10.1115/1.3222904.
[100] S. Georgaka, G. Stabile, K. Star, G. Rozza, M. J. Bluck, A Hybrid Reduced Order Method for Modelling Turbulent Heat Transfer Problems (2019). arXiv:1906.08725.
[101] S. Georgaka, G. Stabile, G. Rozza, M. J. Bluck, Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems (2019). arXiv:1808.05175v3.
[102] K. Star, G. Stabile, S. Georgaka, F. Belloni, G. Rozza, J. Degroote, POD-Galerkin reduced order model of the Boussinesq approximation for buoyancy-driven enclosed flows, in: Building theory and applications : proceedings of M&C 2019, American Nuclear Society (ANS), 2019, pp. 2452–2461.
[103] S. Busto, G. Stabile, G. Rozza, M. Vázquez-Cendón, POD–Galerkin reduced order methods for combined Navier–Stokes transport equations based on a hybrid FV-FE solver, Computers & Mathematics with Applications 79 (2) (2020) 256–273. doi:10.1016/j.camwa.2019.06.026.