SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area - PhD course in
Mathematical Analysis, Modelling, and Applications

Reduced order methods for laminar and turbulent flows in a finite volume setting: projection-based methods and data-driven techniques

Candidate:
Saddam N Y Hijazi

Advisor:
Prof. Gianluigi Rozza
Co-advisors:
Dr. Giovanni Stabile
Dr. Andrea Mola

Academic Year 2019-20

ABSTRACT

This dissertation presents a family of Reduced Order Models (ROMs) which is specifically designed to deal with both laminar and turbulent flows in a finite volume full order setting. Several aspects associated with the reduction of the incompressible Navier–Stokes equations have been investigated. The first of them is related to the need of an accurate reduced pressure reconstruction. This issue has been studied with the help of two main approaches which consist in the use of the Pressure Poisson Equation (PPE) at the reduced order level and also the employment of the supremizer stabilization method. A second aspect is connected with the enforcement of non-homogeneous Dirichlet boundary conditions at the inlet boundary at the reduced order level. The solutions to address this aspect include two methods, namely, the lifting function method and the penalty method.

Different solutions for the treatment of turbulence at the reduced order level have been proposed. We have developed a unified reduction approach which is capable of dealing with turbulent flows based on the Reynolds Averaged Navier–Stokes (RANS) equations complemented by any Eddy Viscosity Model (EVM). The turbulent ROM developed is versatile in the sense that it may be applied on the FOM solutions obtained by different turbulent closure models or EVMs. This is made possible thanks to the formulation of the ROM which merges 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. The newly proposed turbulent ROM has been validated on benchmark test cases in both steady and unsteady settings with Reynolds up to $Re = O(10^5)$.

To My Family

Acknowledgements

First and foremost praise is to ALLAH, the almighty, the merciful and the most gracious. I thank Allah for having given me the patience, the determination and the perseverance to pursue my PhD degree and reach this point in my career.

My PhD life started four years ago and these years passed in the blink of an eye. Today, I would like to express my deepest gratitude and thanks to all the people who helped me during these years and to those who were very close to me.

I would like to start by thanking my advisor Prof. Gianluigi Rozza for having given me the unique opportunity to be his student during both the master and the PhD studies. I would like to thank Prof. Rozza for his great patience, constant help, enormous support and affection. He has always shown that the well-being of his students comes at the top of his priorities, and he has been a great friend with whom I could always share my opinions, thoughts and suggestions. I appreciate very much his encouraging and reassuring words whenever things were not going the right way. The progress I have made during these years would not have been possible had I not been part of Prof. Rozza's research group, being one of his students was a great fortune for me.

I am also truly thankful to my co-advisors Dr. Giovanni Stabile and Dr. Andrea Mola for their continuous support. I would like to thank them for the numerous hours we spent together in their office writing an article or discussing the results of my work. Their suggestions and advice have contributed immensely to this thesis. The insights they offered were very crucial in improving the quality of our work. I would like to thank Giovanni for his patience throughout these four years, and I also I thank Andrea for being very inspiring in his relentless pursuit of perfection.

In my PhD studies and research, I have collaborated and learnt from many researchers and professors both inside and outside SISSA. On this occasion, I would like to thank all the students, researchers and professors in SISSA mathLab for the insightful discussions and for the enlightening comments they shared.

I am deeply grateful for the close friendship I was able to build with my colleagues in SISSA. Special thanks go to my dear friends Matteo, Federico P., Daniele, Luca, Maria, Monica, Veronica and Federico B. with whom I spent many pleasant moments at SISSA and in Trieste. I will always remember how I improved my Italian by talking to them when we shared the same house for several years. The awesome memories of the many birthdays we celebrated together and the several dinners we organized will be unforgettable for me.

To my parents, Abu Saddam and Um Saddam, I say that this moment is a special one in my life but also in yours, this is the fruit of your endless patience, constant care, deep compassion and generous support. Since the moment I have taken the first step in pursuing my graduate studies in September 2014, I have pledged to do my best to reward your trust in me. It is hard to overstate your role in making this possible, words can not describe how thankful I am to you for your love and support.

To my beloved wife Azza, I am very appreciative for the tender loving care and the positive energy you gave me during our first year of marriage. This graduation is a success for which you take credit for your patience and help which I value highly. I hope that we will be able to spend together many beautiful moments in the future and to have a prosperous life filled with joy and happiness.

To my very dear brothers, Qassam and Dorgam, I would like to thank you from the bottom of my heart for being so close to me during my life. With you I played as a child and with you we celebrated our best family moments. I remember vividly each time one of us graduated from high school or from the university, these moments which were always full of pleasure and satisfaction will be present in my memory forever. The life choices that I have made in the last years have distanced us from each other but I always think about you and pray that you succeed in your life.

Last August 24 in Palestine Azza and I got married after being engaged for one year. I would like to thank every friend who sent me nice wishes and whoever participated in our wedding ceremony and festivities. I would like to thank specifically my international friends who came to my wedding and those who tried their best to be there but could not make it at the end. I am very indebted to my Italian friends Matteo and Chiara for coming on that special occasion. I have always been bound to Matteo by a strong friendship and we have spent a lot of time together. I have met his family in his home near Treviso and he met mine in our home near Nablus. I am delighted that you came to Palestine and that we were able to spend some enjoyable time there. I also thank Chiara for her second visit to Palestine last year to attend my wedding. I will always remember the hospitality of her family in their home near Bergamo when I came to Milan for the first time in order to meet Prof. Rozza. They were very kind to me and I felt at home.. It was such a pleasure for my family to get to know and to welcome you in my house. I would also like to thank my close friend Philip from Hong Kong for having tried his best to be with me on my wedding day. Unfortunately he was not able to make it at the end, but I really appreciate his nice gesture on the day of the wedding. Philip and I lived for more than one year in the same city/town, we followed the same master program which ended in September/October 2016. Yet, we kept in touch and I was happy to welcome him in Trieste and then travel together to Sofia and Berlin. These nice times will always be present in my memory. I thank all my Italian colleagues in SISSA for having tried to attend the wedding. I am very grateful for the nice way they welcomed my wife Azza to Italy and for the nice gestures.

I would like to thank all those in SISSA who made it possible for me to organize my wedding in Palestine as planned. First of all I thank Prof. Gianluigi Rozza for his help and kind approval of all what I needed to make everything possible. I thank all the people in SISSA student secretary for helping me with all the papers required for Azza's visa and with all other issues I faced through the PhD years.

On the wedding day, I was very overwhelmed by the presence of my relatives and also by the huge amount of messages which came from those relatives who were not able to come. I feel extremely grateful to my uncles: Fahmi "Abu Younis", Hijazi "Abu Mohammed", Saqer "Abu Mohammed", Hasan "Abu Shadi", Hosni "Abu Mohammed" and Hussam "Abu Jehad", aunts: Um Emad, Um Rasheed and Um Mohammed and finally my grandma Um Saqer. I also thank all my dear cousins for being there and for their nice wishes. To all my relatives, I always feel very fortunate for having you close to me and for the support which you have never ceased giving during the last years of my study. I would like to thank you for the encouragement and love you give me.

I would like to thank all the library staff in SISSA for being always at hand and helpful during the time in which I collaborated with them. Special thanks to my boss Marina for her patience and for the nice times and talks we had together during my working hours in the library. I was also pleased to work with Valentina during her short period in the library and I thank her for the nice discussions we had and for the numerous things I learnt from her.

During my residence in Trieste, I met many people outside the academic life, with whom I created and maintained a nice friendship and brotherhood. I would like to thank my friends Nabeel from "Bethlehem", Akram from "Tulkarem", Saleh from "Um El-Fahem" and Malik "Pierpaolo" from Trieste. It was a great fortune to meet these nice and generous people.

I was also very fortunate to get to know very nice people filled with positive spirit and energy, people who always choose the right side of every issue. They were very inspiring for me in their strong values and kind attitude. Here I thank the members of the organization "Salaam Ragazzi dell'Olivo": Lorella, Nada, Giorgio, Marta, Alessandra, Letizia, Bruno, Nicoletta, Ada and Mauro. Thank you very much my friends for the nice times, the awesome dinners and the insightful discussions we had together. My memories of Trieste will always include you and I hope that you will continue to tell the true story to all the "Triestini" people.

Saddam N Y, Hijazi

Trieste, Italy, July 30th 2020

Contents

CHAPTER 1

Introduction and Motivation

In this chapter the notions of reduced order modeling are introduced. This chapter also outlines the motivation behind developing Reduced Order Methods (ROMs) in a finite volume setting for both laminar and turbulent flows. The first section gives motivations for the work conducted in this thesis. In addition it provides details about the thesis contents. Then the second section offers a broad review of the works done in the literature on projection-based ROM, turbulent ROMs and data-driven ROMs. The next section explains the added contributions of this thesis. Finally the thesis structure is laid out in the last section.

Contents

1.1 Motivation and Contents of the Thesis

In Computational Fluid Dynamics (CFD), the scientific community has been recently trying to cope with the growing demand for simulating industrial problems. These problems which come from various engineering fields are characterized in being governed by conservation laws resulting in a system of Partial Differential Equations (PDEs). Several numerical methods for solving systems of PDEs have been developed in the last decades. These methods include the finite difference (FDM), the finite element (FEM), the finite volume (FVM), and the spectral element method (SEM).

In the recent years we have seen considerable breakthroughs and progresses in computational sciences and scientific computing. However, solving CFD problems numerically in an efficient way remains a huge challenge and a task which demands greater efforts. In particular, such challenging situations are manifested in the context of Parametric PDEs (PPDEs), where one seeks to solve systems of PDEs for a large set of different input parameters. This may happen in applications like optimization, real-time control and uncertainty quantification. In the aforementioned problems, the computational cost of resorting to a classical numerical method for solving the PDEs system could be prohibitive. Reduced Order Methods (ROMs) [71, 120, 24, 23, 11] have been proposed for the goal of reducing the cost imposed by standard numerical methods and to achieve significant computational speed up. This chapter gives a brief overview on the state of the art of ROMs in the context of PPDEs, and on the relevance of developing ROMs for FVM-based discretization and for turbulent flows.

The main objective of this thesis is to carry out reduced order simulations for turbulent flows and fluid problems of interest in the industrial community. The first part of this thesis is focused on ROMs for fluid problems discretized with the finite volume method [110, 148]. The objectives of this thesis originate from the following points:

- Currently, the usage of the FVM for solving fluid dynamics problems in the industrial community is prevailing and predominant. In fact, most of the CFD solvers used for tackling real-life applications are based on the FVM, we mention Fluent [8] and STAR CCM+ [2] (commercial codes) and OpenFOAM [1] (open-source code). These CFD solvers are well equipped with the computational tools needed for tackling problems coming from various engineering fields.

- Most of the CFD problems in which the industry is interested are turbulent problems. For these problems, the cost of running the full order simulations could be significant.

- CFD researchers and engineers demand novel methods which could eventually reduce the computational burden caused by the full order model simulations. This can be seen for example in shape optimization problems, in which the fluid dynamics problem is required to be run for different configurations.

These observations and demands motivate investing efforts in constructing ROMs for FVM-based discretization, and thereafter introducing turbulence treatment at the reduced order level. In this thesis, the full order CFD solver is chosen to be OpenFOAM, this choice is justified by the fact that it is an open-source code. In addition, OpenFOAM offers well-established documentation and tutorials for various benchmark CFD problems. As a consequence, the reduced order models proposed in this thesis have taken into consideration the full order modeling techniques utilized by OpenFOAM, especially when it comes to turbulence modeling.

The work carried out in this thesis can be divided into two main parts. At first, ROMs for FVM-based discretization are presented, by adapting ROMs techniques used in the finite element environment to work also in a finite volume setting. The second part involves the construction of reduced order models tailored to work with turbulent flows.

The next section reports the relevant contributions on reduced order modeling for the finite volume method, as well as the efforts made for the reduction of turbulent flows. The final part of the next section addresses the ROMs which involve the usage of data-driven methods.

1.2 Literature Review

In this section, I will review works in the literature which are focused on ROMs developed in the finite volume setting. In addition, I will address different approaches for the treatment of turbulent flows at the reduced order level. We shall start by introducing the concept of projection-based ROMs in the next subsection. Then ROMs designed to computationally reduce turbulent flows are presented followed by data-driven reduced order methods. Finally, the notion of hybrid ROMs is introduced.

1.2.1 Projection-based ROMs

The paradigm of projection-based ROMs [23, 13, 19, 7, 22] is a popular reduction approach which has been used in numerous CFD applications. Projection-based ROMs are based on the notion of casting the high-order numerical solution manifold onto a low-dimensional manifold. The construction of such a ROM involves then the projection of the governing equations onto this low-dimensional space. Such a low-dimensional manifold or reduced order space can be generated by the use of different methods. Examples of such methods are the Reduced Basis (RB) with the greedy approach [71], the Proper Orthogonal Decomposition (POD) [132, 42], the Proper Generalized Decomposition (PGD) [50], or by the Dynamic Mode Decomposition (DMD) [130]. As for the projection techniques which can be employed, Galerkin projection and Petrov-Galerkin projection are considered the most used ones. In this thesis, the POD technique is chosen for the generation of the reduced order space and a Galerkin projection is applied for the construction of the projection-based ROM. In this case the ROM is often termed as POD-Galerkin ROM, for references on POD-Galerkin ROMs see [114, 4, 26, 93, 33, 12].

In literature, earlier works on ROMs based on the finite volume discretization include [49, 67, 69, 66, 68]. In these works, the Reduced Basis (RB) approach has been extended to work in general linear evolution schemes such as finite volume schemes. In [103], the authors presented a POD-Galerkin ROM for the finite volume method. The ROM has been constructed to reduce the Navier–Stokes equations for both laminar and turbulent flows. Another POD-Galerkin ROM is presented in [137], where this ROM has been dedicated to deal with the reduction of the problem of vortex shedding around a circular cylinder in a finite volume environment. Stabilization techniques have been extended to work also in finite volume ROMs [139]. In particular, the last work presents a POD-Galerkin ROM which employs the supremizer stabilization method in order to obtain a stable pressure ROM field. The supremizer stabilization method ensures that a reduced order version of the inf-sup condition is met, initially it was constructed for finite element POD-Galerkin ROM in [16]. The POD-Galerkin ROMs in [103, 137, 139] differ in the methodology adapted for the reduction of the Navier–Stokes equations. Further details on the approaches employed in these works are recalled in section 2.4, section 2.5 and section 2.6. We mention works in which numerical analysis has been used for the study of ROMs for turbulent flows and ROMs for pressure approximations [124, 90, 34]. The last works present different ROMs frameworks for the computation of the reduced pressure field.

Recently, the work [140] tackles the issue of geometrical parametrization for FVM-based POD-Galerkin ROM. The authors in [140] propose a reduced order model which is fully consistent with the SIMPLE algorithm [117]. The SIMPLE algorithm is the full order solver approach for steady flows in OpenFOAM and it is a segregated approach. Other works which deal with geometrical parametrization in FVM-based ROM include [98, 164] which focus on inviscid Euler equations. Also addressing the issue of geometrical parametrization, the contributions [154] and [161] deal with turbulent compressible Navier–Stokes equations and PDE-constrained optimization problems, respectively. Extension of the FVM-based POD-Galerkin ROM for thermal mixing problem is presented in [57]. Another POD-Galerkin ROM for the problem of buoyancy-driven enclosed flows is developed in [141].

1.2.2 ROMs for turbulent flows

Turbulent flows are ubiquitous in nature and in real world applications. Turbulence as a physical phenomenon is characterized by being chaotic, disorder, non-stationary and multi-scale [156]. Turbulent flows can be seen in civil, aerospace and naval engineering, for example we mention the flow around circular cylinders [91], the flow around aircraft wing tips [3] and the dynamics of rowing and sailing boats [54, 102]. These kinds of applications are of significant interest in the industrial community. Understanding the underlying dynamics of such flows and capturing the quantities of interest from engineering perspective is of paramount importance. Therefore, numerical simulations of the Navier–Stokes equations are carried out to address these issues. However, the task of simulating the Navier–Stokes equations for turbulent flows is not straightforward. In more details, simulating the Navier–Stokes equations for all the scales in the case of turbulent flows, which is known as the Direct Numerical

Simulations (DNS) approach, is not feasible. The argument for this claim is based on the Kolmogorov length microscales. If one simulates the NSE using the DNS approach (or without any form of turbulence modeling), then the computational grid has to be fine enough to capture all the length scales up to the Kolmogorov length microscales η [94]:

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}}, \quad \varepsilon = \frac{u'^3}{L}. \tag{1.1}$$

In the above relations ε is the turbulent dissipation rate, L is the integral length scale which is the largest scale in the energy spectrum, ν is the kinematic viscosity and u' is the root mean square of the turbulent velocity fluctuation field which is present in the Reynolds decomposition assumption (see 3.1). Consequently, the number of points needed for the discretization of the domain is:

$$N = \frac{L}{\eta} = \left(\frac{u'L}{\nu}\right)^{\frac{3}{4}}, \tag{1.2}$$

where one can see that the term inside the brackets is basically a form of the Reynolds number. Hence, the number of cells N^3 inside a box which has the dimension of the integral length scale scales up with the Reynolds number as follows:

$$N^3 \propto Re^{\frac{9}{4}}. \tag{1.3}$$

The relation above indicates that spatial refinement is needed for flows with high Reynolds number. It has to be remarked that the temporal resolution of the numerical scheme has to be also refined in order to make sure that fluid particles do not cross more than one cell within a time step. The last considerations render the task of carrying out DNS simulations for turbulent flows prohibitive in terms of computational cost.

To overcome the problems caused by the DNS approach, turbulence is treated usually with the help of modeling strategies. These modeling strategies include mainly the idea of averaging and obtaining under-resolved solutions which reflect, to the best extent, the physics of the problem under interest. The first common approach in the CFD community is called Reynolds Averaged Navier–Stokes (RANS) equations. This approach solves for the time-averaged part of the fluid dynamics variables. On the other hand, there is another approach named Large Eddy Simulations (LES) [28, 126] which is based on the concept of filtering. In LES, the Navier–Stokes equations are filtered to some scale, and the large scales are simulated while the small scales are modeled.

After giving this short introduction on turbulence, we will proceed to address the reduction approaches dedicated to deal with turbulent flows. POD-Galerkin ROMs have several issues when it comes to the reduction of turbulent flows. Most of the issues are related to energy stability problems [35]. In more details, the POD method retrieves the modes which are representative of the high-energy scales. This results in ROMs which are not dissipative since the turbulent small scales are the ones responsible for the dissipation of the turbulent kinetic energy [107].

A brief overview of the reduction strategies proposed to deal with turbulent flows will be outlined. The works [25, 116, 153, 9] propose to include dissipation via a closure model. The work [44] shows numerically that the POD modes have similar energy transfer to the one of the Fourier modes. This may suggest that LES ideas could be beneficial in POD-Galerkin ROMs given the analogy with LES which is based on the energy cascade concept. In [77], it is suggested that the usage of the H^1 inner product instead of the L^2 one in the generation of the POD space could result in more dissipative ROMs. This is justified by the fact that small scale modes have H^1 norm value that is higher than their L^2 norm value. In [15, 14, 5], the reduced order space has been enriched in order to better account for the truncated modes. In [5] for example, the POD modes have been enriched with dissipative modes associated with the gradient of the velocity fields. A priori analysis is performed on the POD modes, and a rearrangement of them is done in a way that leads to the enforcement of the energetic dissipative modes within the first orders of the reduced order basis. The ROM in [5] has been tested on an aeronautical injector with *Reynolds* number of 45000.

Minimum residual formulation has been constructed in the reduced order model in order to accurately reduce turbulent flows [36, 37, 145, 60]. Other works include the use of the Dynamic Mode Decomposition (DMD) [6, 48, 146, 97]. The work [52] presents a constrained formulation to deal with long time instabilities in the context of turbulent flows.

The efforts to reduce turbulent flows in the ROM community include the development of turbulent ROMs which are designed to work with specific turbulence model at the full order level. Such ROMs are developed in [38, 147, 113, 152, 112]. In [38], the ROM is dedicated to reduce flows modeled with the Smagorinsky turbulence model [134] for steady flows in a finite element setting. In the last work, the non-linear eddy diffusion term is approximated using the Empirical Interpolation Method (EIM). Also the work in [147] is designed to deal with the Smagorinsky turbulence model. In this Smagorinsky-ROM, it is assumed that the projection matrix coefficients (which come from the projection of the eddy viscosity term onto the velocity POD modes) are time dependent. Later, these coefficients are updated during the time integration of the momentum equation at the reduced order level. The Variational Multi-Scale (VMS) method has been employed in several VMS-ROM works see [27, 136, 76]. In addition, Smagorinsky VMS-ROMs are constructed for turbulent flows as in [39, 18].

It is important to highlight one of the main differences between the works mentioned in this subsection which is the presence or the absence of a turbulence modeling strategy at the full order level. Therefore, one may divide these ROMs into two categories, the first is called *ROMs with FOM turbulence modeling* and the second is referred to as *ROMs without FOM turbulence modeling*. For example, the works [15, 14] are based on DNS full order model snapshots without turbulence modeling and so they belong to the second category. On the other hand, in [5] LES full order turbulence modeling is employed. Likewise, the turbulent ROMs which are based on the Smagorinsky turbulence model such as [147, 38] are considered in first category.

1.2.3 Data Driven and Hybrid ROMs

The majority of the works reported in the previous subsections are for projection-based ROMs. In this thesis, we also address ROMs which employ data-driven techniques, for references on data-driven ROMs see [78, 118, 101, 123, 86, 64, 70, 111]. For instance, we mention the work [65] in which the reduced order model uses regression for the approximation of the maps between time-parameter values and the projection coefficients onto the reduced modes. We remark that the data-driven ROMs mentioned are not necessary designed to deal with turbulent flows. Data-driven techniques can be implemented to approximate certain variables or certain quantities in the reduced order formulation, while other variables are still approximated using classical projection-based techniques. In the latter case, we refer to the reduced order model as hybrid/combined/mixed ROM. Hybrid ROMs have been constructed in previous works, in this subsection we will report hybrid ROMs developed for the reduction of the Navier–Stokes equations whether for laminar or turbulent flows.

In [160], a hybrid ROM is presented, where data-driven filtering techniques have been utilized. The last ROM employ data-driven methods for the approximation of a correction term which has been added in order to model the interaction between truncated and resolved modes. This hybrid ROM has been tested on a $2D$ channel flow past a circular cylinder at various *Reynolds* number values.

Calibration methods in reduced order modeling are considered as data-driven approaches. These methods have been implemented in the formulation of hybrid ROMs such as in [56, 43]. In the last two mentioned contributions, the calibration methods have been implemented in the context of POD-Galerkin ROMs for the purpose of reducing the Navier–Stokes equations. The approach in [56] assumes that pressure gradient term which appears in the projected momentum equation is approximated as the product of a calibration matrix and the reduced velocity vector degrees of freedom. The entries of the calibration matrix are then computed by solving a minimization problem of a functional which has dependence on the interpolated L^2 projection coefficients of the velocity. On the other hand, the calibration procedure in [43] is done differently. In more details, the hybrid ROM [43] involves the tuning of the polynomial that defines the reduced order dynamical system. The reduced order polynomial is therefore different from the original one, where it minimizes a certain meaningful functional. This functional has basically two properly weighted terms, one accounts for the error committed between the projection coefficients (which come from the data) and the reduced order solution, the second term penalizes the difference between the original polynomial and the one which solves the optimization problem.

The hybrid ROM in [115] presents an empirical pressure model for the approximation of the pressure term in the projected momentum equation. Linear regression technique has been used to fit the coefficients of the empirical pressure model from the data.

We mentioned earlier works on ROMs which are focused on the variational multi-scale (VMS) method which focus on turbulent flows. The aforementioned works are based on projection techniques. Recently, a hybrid ROM has been presented in [109] which makes use of the VMS method. In this hybrid ROM (referred to in the article as DD-VMS-ROM), projection-based techniques are used to separate the scales into three categories which are resolved large scales, resolved small scales and unresolved scales. Then the terms which represent the interactions between the three categories of scales (which are called the VMS-ROM closure terms) are identified. Later, the VMS-ROM closure terms are modeled using data-driven approach. The main difference between turbulent VMS-ROMs and the DD-VMS-ROM is that in the former the VMS closure models are utilized to help the dissipation of energy from the ROM, unlike in DD-VMS-ROM where the VMS-ROM closure terms constructed by the usage of data are not required to be dissipative.

Modeling projected terms at the reduced order level and modifying the reduced order matrices entries in the reduced dynamical system is the main pattern observed in the last mentioned works. Moreover, one can notice that these hybrid ROMs concentrate on reproducing the velocity field. The pressure field and the turbulence variables have not been considered in the reduced order model formulation.

1.3 Thesis Contributions

The main objective of this thesis is to construct ROMs for industrial fluid dynamics problems. The prerequisites for achieving this goal are the following:

- To extend ROMs techniques and methodologies to work in the finite volume environment. This is important because of the fact that the FVM is widespread in the industrial community.
- To have ROMs developed specifically to deal with turbulent flows. Since these flows are common in real applications and in industrial problems.

The first part of this thesis deals with the issue of FVM-based ROMs. It offers a survey of projection-based ROMs developed for FVM-based discretization. In addition, the ROMs have been equipped with needed techniques for the reduced treatment of non-homogeneous Dirichlet boundary conditions. Furthermore, a fully offline/online decoupled procedure for computing the fluid dynamics forces is developed for the aforementioned ROMs.

In the second of part of this thesis, the attention is shifted to the treatment of turbulent flows. In chapter 3, a hybrid ROM is proposed for the treatment of turbulent flows. The novelty of this ROM is in its generality and versatility, where it is designed to work with full order models in OpenFOAM based on different turbulence closure models. In contrast to the hybrid ROMs mentioned in subsection 1.2.3, this ROM has introduced a reduced order approximation of the velocity $\boldsymbol{u}$, the pressure p and the eddy viscosity ν_t. The motivation behind having a reduced order version of the pressure and the eddy viscosity fields can be summarized by the following points:

- Having an accurate reconstruction of the pressure field at the reduced order level is vital. This is because of the need to recover certain performance indicators which highly depend on the pressure field. An example of such performance indicators is the fluid dynamics forces acting on the surface of bodies immersed in the flow.

- The reduced eddy viscosity is essential for having a stable reduced order model since the full order model itself has been stabilized by presenting the eddy viscosity into its formulation.

The hybrid ROM proposed in this thesis assumes that each of the fluid dynamics variables of velocity $\boldsymbol{u}$, pressure p and the eddy viscosity ν_t has a different set of reduced degrees of freedom. The model utilizes classical projection techniques for the approximation of velocity and pressure at the reduced order level. In ideal situation, similar projection procedure has to be employed for the specific turbulence model equations. However, this would mean that a turbulent ROM has to be developed for each different turbulence model. Given the abundant number of turbulence models available, the last option would be problematic. This is because of the implications in terms of the number of different ROMs needed to be developed and monitored at the same time for potential updates. Consequently, unlike the case of velocity and pressure, the reduced eddy viscosity in the hybrid ROM proposed here is approximated by the data-driven techniques.

1.4 Thesis Structure

This thesis comprises of 5 chapters which are organized as it follows:

- The first chapter Introduction and Motivation. introduces the general notions of reduced order modeling. It also addresses the motivations and contents of this thesis. It gives an idea of why it is relevant to develop reduced order models for problems in CFD in a finite volume setting. The first chapter provides a broad review of the literature contributions on projection-based ROM, turbulent ROMs and data-driven ROMs.

- The second chapter Projection-based ROMs for the NSE in FV deals with the problem of interest at the full order level which is the incompressible Navier–Stokes equations. It explains how the finite volume discretization of this problem is done and then it lays out the algorithms employed for solving the governing equations. In the same chapter, the POD method for the generation of reduced order spaces is presented. After that, several POD-Galerkin ROMs are proposed, these ROMs are used for the reduction of laminar flows. The last sections of this chapter deal with the issues of non-homogeneous boundary condition treatment and the offline/online decoupling procedure for the computation of the forces.

- The third chapter Hybrid ROMs for the Turbulence Treatment in the NSE is dedicated to the treatment of turbulence at both the full and the reduced order levels. At first, turbulence modeling with the Reynolds Averaged Navier–Stokes (RANS) equations is introduced. Then a hybrid POD-Galerkin ROM is proposed for the goal of reducing turbulent flows. This hybrid ROM utilizes interpolation using radial basis function for the approximation of the turbulent viscosity. The interpolation procedure is done in two different ways which are addressed in detail in this chapter.

- The fourth chapter Applications and Numerical Results presents the results of the ROMs developed in both the second and the third chapters. The results are for benchmark cases in computational fluid dynamics. These cases include the angle of attack for the airfoil (studied in uncertainty quantification context), the backstep problem in turbulent setting and the flow around a circular cylinder in turbulent setting as well. The reduction methodologies are compared to each other on several fronts.

- The final chapter Conclusions and Outlook outlines the conclusions which are drawn from the results of the work carried out in this thesis. It also gives several future paths for the extension of the work presented here.

Chapter 2

Projection-based ROMs for the Incompressible Navier–Stokes Equations in a Finite Volume Setting

This chapter addresses the parametrized incompressible Navier–Stokes equations when discretized using the finite volume method. Then the algorithms employed for solving the equations are presented. Later, the POD method is introduced and then projection-based ROMs for laminar flows are discussed. These ROMs varied according to the equations used at the reduced order level. The first ROM introduces a uniform approach with the usage of solely the momentum equation. Another ROM which is based on the Poisson equation for pressure is introduced. Then we present a ROM which employs the supremizer stabilization method with the use of both the momentum and the continuity equations. In this chapter, the treatment methods for the non-homogeneity at the Dirichlet boundary are addressed. Finally, the offline/online decoupling procedure for the computation of the forces is explained.

Contents

2.1 Full Order Model (FOM)

This section presents the full order model starting with the governing equations of interest which are the incompressible Navier–Stokes Equations (NSE). The NSE are studied here in a parametrized form, we start with recalling their strong form. Given a parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^q$, where $\mathcal{P}$ is a q-dimensional parameter space. The incompressible NSE parametrized by $\boldsymbol{\mu}$ read as follows:

$$\begin{cases} \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T \right) = -\boldsymbol{\nabla} p & \text{in } \mathit{\Omega} \times [0, T], \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = \boldsymbol{0} & \text{in } \mathit{\Omega} \times [0, T], \\ + \text{ Boundary conditions} & \text{on } \mathit{\Gamma} \times [0, T], \\ + \text{ Initial conditions} & \text{in } (\mathit{\Omega}, 0), \end{cases} \tag{2.1}$$

where $\mathit{\Gamma}$ is the boundary of the fluid domain $\mathit{\Omega} \in \mathbb{R}^d$, with $d = 1$, 2 or 3. $\boldsymbol{u}$ is the flow velocity vector field, t is the time, ν is the fluid kinematic viscosity, p is the normalized pressure field, which is divided by the fluid density ρ_f, and the time window under consideration is $[0, T]$. We remark that in this work the parameter $\boldsymbol{\mu}$ could be a physical parameter such as the fluid kinematic viscosity or a geometrical one such as the dimension of a certain part of the domain. We would like to emphasize that the velocity and pressure fields are functions of time, space and the parameter, that is $\boldsymbol{u} = \boldsymbol{u}(t, \boldsymbol{x}; \boldsymbol{\mu})$, $p = p(t, \boldsymbol{x}; \boldsymbol{\mu})$. These dependencies have been dropped in the equations above for the sake of keeping the notation concise.

2.1.1 Finite Volume Discretization

The incompressible Navier–Stokes equations in 2.1 are solved by the finite volume method [110]. The first step in this method is to choose a suitable polygonal tessellation, then the PDEs system is written in integral form over each control volume. Denote by N_h the dimension of the full order model (FOM) which is basically the number of degrees of freedom of the discretized problem. The momentum and continuity equations are solved with the help of a segregated approach which adapts the Rhie and Chow interpolation. The discretization process starts with the momentum equation by writing it in integral form over each control volume V_i as follows:

$$\int_{V_i} \frac{\partial}{\partial t} \boldsymbol{u} dV + \int_{V_i} \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) dV - \int_{V_i} \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T \right) dV + \int_{V_i} \boldsymbol{\nabla} p dV = 0. \tag{2.2}$$

The discretization procedure of all terms in the momentum equation is explained in what follows. The pressure gradient term is discretized using the Gauss's theorem:

$$\int_{V_i} \boldsymbol{\nabla} p dV = \int_{S_i} p d\boldsymbol{S} \approx \sum_f \boldsymbol{S_f} p_f, \tag{2.3}$$

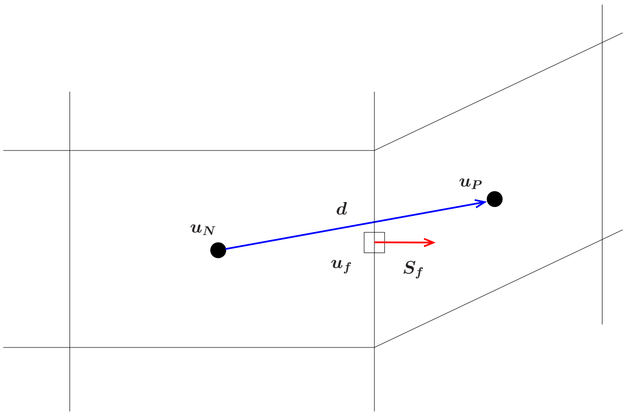


Figure 2.1: Sketch of a finite volume in 2 dimensions.

where p_f is the value of pressure at the center of the faces and $\boldsymbol{S_f}$ is the area vector of each face of the control volume (Figure 2.1).

The convective term is discretized also by exploiting the Gauss's theorem as follows:

$$\int_{V_i} \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) dV = \int_{S_i} (d\boldsymbol{S} \cdot (\boldsymbol{u} \otimes \boldsymbol{u})) \approx \sum_f \boldsymbol{S_f} \cdot \boldsymbol{u_f} \otimes \boldsymbol{u_f} = \sum_f F_f \boldsymbol{u_f}. \tag{2.4}$$

In the above equation $\boldsymbol{u_f}$ is the velocity vector evaluated at the center of each face of the control volume. It has to be remarked that velocity values are initially computed at the cell centres and therefore the values at the center of the faces have to be deduced from the ones calculated at the cell centres. Consequently, these $\boldsymbol{u_f}$ values are interpolated using the values computed at the cell centers. There are plenty of interpolation schemes which can be used such as central, upwind, second order upwind and blended differencing schemes. The mass flux F_f is computed using the previous converged velocity in the first iteration and then updated by $F_f = \boldsymbol{u}_f \cdot \boldsymbol{S}_f$ for removing the non-linearity.

As for the diffusion term, it is discretized as it follows:

$$\int_{V_i} \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) dV = \int_{S_i} d\boldsymbol{S} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \approx \sum_f \nu_f \boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f, \tag{2.5}$$

where $(\boldsymbol{\nabla u})_f$ is the gradient of $\boldsymbol{u}$ at the faces. In similar fashion to the computation of the pressure gradient in (2.3), one may compute $(\boldsymbol{\nabla u})_f$. As for the computation of the term $\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f$ in (2.5), this depends on one particular feature of the mesh which is the orthogonality. The mesh 2.1 is orthogonal if the line that connects two cell centers is orthogonal to the face that divides these two cells. If the mesh is orthogonal, and assuming that the two cell centres are equally distanced to the face, then the term $\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f$ is computed as it follows:

$$\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f = |\boldsymbol{S_f}| \frac{\boldsymbol{u}_N - \boldsymbol{u}_P}{|\boldsymbol{d}|}, \tag{2.6}$$

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 2.1. On the other hand, non-orthogonal correction term is needed for the case of non-orthogonal meshes. The computation of the correction term [83] is given by the following equation:

$$\boldsymbol{S_f} \cdot (\boldsymbol{\nabla u})_f = |\boldsymbol{\Delta}| \frac{\boldsymbol{u}_N - \boldsymbol{u}_P}{|\boldsymbol{d}|} + \boldsymbol{J} \cdot (\boldsymbol{\nabla u})_f, \tag{2.7}$$

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. In the next section, the discretized equations (written in matrix form) for velocity and pressure are addressed. Additionally, the segregated pressure-based solver approaches are presented.

2.2 Segregated Pressure-Based Solvers for The Incompressible NSE

After having introduced the discretization of the different terms in the NSE, one may proceed to address the algorithm used in solving the discretized system. The algorithm employed in OpenFOAM is a segregated pressure-based approach. Its version for the case of steady flows is called the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) [117]. On the other hand the Pressure Implicit with Splitting of Operators (PISO) [79] algorithm is utilized for unsteady flows, also one may use the PIMPLE [110] algorithm which is a combined version of PISO and SIMPLE.

The starting point is the discretized system of the NSE which can be written in matrix form as follows:

$$\begin{bmatrix} [\boldsymbol{A_u}] & [\boldsymbol{\nabla}(\cdot)] \\ [\boldsymbol{\nabla} \cdot (\cdot)] & [0] \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 0 \end{bmatrix}, \tag{2.8}$$

where $\boldsymbol{A_u}$ is the matrix containing the velocity coefficients which comes from the momentum equation, $\boldsymbol{A_u u} = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right)$. The last system is called a saddle-point system.

The momentum equation in Equation 2.8 can be written as:

$$\boldsymbol{u} + [\boldsymbol{A_u^{-1}}][\boldsymbol{\nabla}(\cdot)][p] = 0, \tag{2.9}$$

where $\boldsymbol{A_u^{-1}}$ denotes the inverse of the momentum matrix in the discretized form. By applying the divergence operator on the last equation, one obtains

$$[\boldsymbol{\nabla} \cdot (\cdot)]\boldsymbol{u} + [\boldsymbol{\nabla} \cdot (\cdot)][\boldsymbol{A_u^{-1}}][\boldsymbol{\nabla}(\cdot)][p] = 0, \tag{2.10}$$

then by exploiting the free divergence constraint on velocity $[\boldsymbol{\nabla} \cdot (\cdot)]\boldsymbol{u} = 0$, one may derive the following pressure equation

$$[\boldsymbol{\nabla} \cdot (\cdot)][\boldsymbol{A}_{\boldsymbol{u}}^{\boldsymbol{-1}}][\boldsymbol{\nabla}(\cdot)][p] = 0. \tag{2.11}$$

Thus the matrix $\boldsymbol{A}_{\boldsymbol{u}}^{\boldsymbol{-1}}$ acts as the diffusivity in the Laplace equation for the pressure. For computational reasons, solving Equation 2.11 in this form is inconvenient. This is related to the fact that $\boldsymbol{A}_{\boldsymbol{u}}^{\boldsymbol{-1}}$ is likely to be dense. In addition the product of three matrices could result in a dense matrix which rises the difficulty of solving the linear system. As a result, a different approach is used in which the momentum matrix is decomposed into diagonal and off-diagonal matrices:

$$[\boldsymbol{A}_{\boldsymbol{u}}] = [\boldsymbol{D}_{\boldsymbol{u}}] + [\boldsymbol{LU}_{\boldsymbol{u}}], \tag{2.12}$$

where $[\boldsymbol{D}_{\boldsymbol{u}}]$ is a diagonal matrix and therefore can be easily inverted, and $[\boldsymbol{LU}_{\boldsymbol{u}}]$ is the matrix containing the off-diagonal part of $[\boldsymbol{A}_{\boldsymbol{u}}]$. Inserting Equation 2.12 into Equation 2.8, yields the following modified system

$$\begin{bmatrix} [\boldsymbol{D}_{\boldsymbol{u}}] & [\boldsymbol{\nabla}(\cdot)] \\ [\boldsymbol{\nabla} \cdot (\cdot)] & [0] \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} -[\boldsymbol{LU}_{\boldsymbol{u}}][\boldsymbol{u}] \\ 0 \end{bmatrix}, \tag{2.13}$$

This gives the following equation for pressure:

$$[\boldsymbol{\nabla} \cdot (\cdot)][\boldsymbol{D}_{\boldsymbol{u}}^{\boldsymbol{-1}}][\boldsymbol{\nabla}(\cdot)][p] = -[\boldsymbol{\nabla} \cdot (\cdot)][\boldsymbol{D}_{\boldsymbol{u}}^{\boldsymbol{-1}}][\boldsymbol{LU}_{\boldsymbol{u}}][\boldsymbol{u}]. \tag{2.14}$$

The last equation is a Poisson equation for pressure with the diagonal part of the momentum equation acting as a diffusivity, while divergence of the velocity is on the right hand side. The last derivation of the pressure equation is referred to as the pressure equation as a Schur Complement.

In the next part of this section, the same derivation will be done without having to use the assembly of Schur's complement. The first step is to write the discretized momentum equation, for each control volume, namely:

$$a_P^{\boldsymbol{u}} \boldsymbol{u}_P + \sum_N a_N^{\boldsymbol{u}} \boldsymbol{u}_N = \boldsymbol{r}_{\boldsymbol{p}} - \boldsymbol{\nabla} p, \tag{2.15}$$

where P represents a generic cell center and N is the set of neighboring points around it (Figure 2.1), $\boldsymbol{u}_N$ 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, $a_N^{\boldsymbol{u}}$ is the vector that consists off-diagonal coefficients and $\boldsymbol{r}_{\boldsymbol{p}}$ is a term that contains any r.h.s contributions. For sake of simplicity, one may introduce the operator $\boldsymbol{H}(\boldsymbol{u}) = \boldsymbol{r}_{\boldsymbol{p}} - \sum_N a_N^{\boldsymbol{u}} \boldsymbol{u}_N$ which contains the off-diagonal part of the momentum matrix and any r.h.s contributions. It follows:

$$a_P^{\boldsymbol{u}} \boldsymbol{u}_P = \boldsymbol{H}(\boldsymbol{u}) - \boldsymbol{\nabla} p, \tag{2.16}$$

and

$$\boldsymbol{u}_P = (a_P^{\boldsymbol{u}})^{-1}(\boldsymbol{H}(\boldsymbol{u}) - \boldsymbol{\nabla} p). \tag{2.17}$$

Inserting the last expression for $\boldsymbol{u}_P$ inside the continuity equation $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$, gives:

$$\boldsymbol{\nabla} \cdot [(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{\nabla} p] = \boldsymbol{\nabla} \cdot [(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{H}(\boldsymbol{u})]. \tag{2.18}$$

The last pressure equation is equivalent to the one in Equation 2.14, where it can be noticed that $a_P^{\boldsymbol{u}}$ is a coefficient in the diagonal matrix $[\boldsymbol{D}_{\boldsymbol{u}}]$ and that $\boldsymbol{H}(\boldsymbol{u})$ is the product $[\boldsymbol{LU}_{\boldsymbol{u}}][\boldsymbol{u}]$.

The mass flux through each face of the control volume is denoted by F_f. Equation 2.17 and Equation 2.18 are used together with the discretized version of the continuity equation to update the mass fluxes F_f.

$$F_f = \boldsymbol{u}_f \cdot \boldsymbol{S}_f = -(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{S}_f \cdot \boldsymbol{\nabla} p + (a_P^{\boldsymbol{u}})^{-1} \boldsymbol{S}_f \cdot \boldsymbol{H}(\boldsymbol{u}). \tag{2.19}$$

The term $(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{S}_f \cdot \boldsymbol{\nabla} p$ is computed in a similar manner to Equation 2.6:

$$(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{S}_f \cdot \boldsymbol{\nabla} p = (a_P^{\boldsymbol{u}})^{-1} \frac{|\boldsymbol{S_f}|}{|\boldsymbol{d}|}(p_N - p_P) = a_N^p (p_N - p_P), \tag{2.20}$$

where p_P and p_N are pressure values at the centers of two neighboring cells and $a_N^p = (a_P^{\boldsymbol{u}})^{-1} \frac{|\boldsymbol{S_f}|}{|\boldsymbol{d}|}$ which represents the off-diagonal matrix coefficient in the pressure equation.

At this point, the SIMPLE algorithm may be introduced. The first step in this algorithm is called the momentum predictor step, where one solves the momentum equation 2.16 for an initial guessed pressure field $p^\star$. Then using the obtained value of the velocity $\boldsymbol{u}^\star$ from the last step one may assemble the new off-diagonal vector $\boldsymbol{H}(\boldsymbol{u}^\star)$. This is followed by solving the Poisson equation for pressure 2.18 to correct the pressure field. The next step is to assemble the conservative flux fields F_f before going through the cycle again and carrying out the momentum predictor. This procedure is repeated till convergence is achieved. However, the pressure correction equation in its current form is prone to divergence [148]. In order to assure convergence, under-relaxation is employed.

The pressure field which results from solving the Poisson equation in 2.18 is $p^\star + p'$, where the resulted pressure field is seen as the sum of the guessed initial one $p^\star$ and a correction term denoted by p'. The under-relaxed pressure field is modified as follows:

$$p^{\star\star} = p^\star + \alpha_p p', \tag{2.21}$$

where α_p is the pressure under-relaxation factor. If $\alpha_p = 1$ this means that no under-relaxation is introduced and this choice of α_p is too large for stable computations, particularly when $p^\star$ is far away from the final solution. A zero value of α_p means that no correction is introduced at all, which is obviously unwanted. Therefore, the value of α_p should be taken in the range of $(0, 1)$ so as to allow to add a correction term which is a fraction of p'. This term should be large enough to move forward the algorithm towards convergence and at the same time small enough to secure stable computations.

Unlike pressure, the under-relaxation for the velocity is not explicit, instead momentum under-relaxation is implicit, where the discretized momentum equation in 2.16 is modified by adding artificial terms as follows:

$$\frac{1-\alpha_u}{\alpha_u} a_P^{\boldsymbol{u}} \boldsymbol{u}_P^{\star\star} + a_P^{\boldsymbol{u}} \boldsymbol{u}_P^{\star\star} = \boldsymbol{H}(\boldsymbol{u}^\star) - \boldsymbol{\nabla} p^\star + \frac{1-\alpha_u}{\alpha_u} a_P^{\boldsymbol{u}} \boldsymbol{u}_P^\star. \tag{2.22}$$

The new added terms consists of the new under-relaxed velocity values denoted by $\boldsymbol{u}_P^{\star\star}$ and the old ones denoted by $\boldsymbol{u}_P^\star$. These terms cancel each other in case of convergence. Here α_u is the velocity under-relaxation factor, the above equation simplifies to

$$\frac{1}{\alpha_u} a_P^{\boldsymbol{u}} \boldsymbol{u}_P^{\star\star} = \boldsymbol{H}(\boldsymbol{u}^\star) - \boldsymbol{\nabla} p^\star + \frac{1-\alpha_u}{\alpha_u} a_P^{\boldsymbol{u}} \boldsymbol{u}_P^\star. \tag{2.23}$$

As for the guidelines for choosing under-relaxation factors, there is no straightforward way for setting their optimum values, where in fact their values depend on the type of problem and the flow. However, in the literature it is recommended that:

$$\begin{aligned} &0 < \alpha_u < 1, \\ &0 < \alpha_p < 1, \\ &\alpha_u + \alpha_p \approx 1. \end{aligned} \tag{2.24}$$

Let us summarize the important aspects of the SIMPLE algorithm. In the SIMPLE algorithm, the role of pressure in the momentum equation is to ensure that the velocity field is divergence free. After carrying out the momentum predictor step, the velocity field does not satisfy the divergence constraint since the pressure field is a guessed one. Consequently, the pressure field is corrected and this gives a pressure field which consist into two parts. The first one is physical and thus consistent with the global flow field, while the other one represents the correction term which guarantees the continuity. Only the first component of the pressure field is desired and used in building the physical pressure field. To build this physical pressure field, the SIMPLE algorithm resorts to the under-relaxing approach explained in this section. The full SIMPLE algorithm is outlined in Algorithm 1.

The SIMPLE algorithm is used in OpenFOAM for solving the velocity-pressure coupled system for the case of steady flows. This algorithm may be extended to transient simulations, however, the most used transient solver in CFD codes is PISO. In the remaining part of this section, the PISO algorithm is addressed. The PISO algorithm is based on the idea that for low values of the Courant number (small time steps) the pressure-velocity coupling is much stronger than the non-linear $\boldsymbol{u} - \boldsymbol{u}$ coupling in the convective term. Thus it suggests that a repeated number of pressure corrections can be carried out without having to rediscretize the momentum equation. The PISO algorithm consist into two correction steps for the pressure field. The first one will yield a conservative velocity field, while the second will give back a more physical pressure field. The usage of more than one pressure corrector implies that it is not necessary anymore to under-relax the pressure field. However, one thing that could be seen as a drawback of the PISO algorithm is that it is assumed that the momentum discretization is not needed when the pressure correctors are applied. This last assumption is only true for small time steps, which forces that the maximum Courant number [45, 46] has to be under the value of $Co_{max} = 0.9$.

Algorithm 1 The SIMPLE algorithm

```
Start with an initial guess of the pressure field $p^\star$ and the velocity field $\boldsymbol{u}^\star$.
Momentum predictor step : solve the discretized momentum equation for the guessed pressure field $p^\star$:
    $a_P^{\boldsymbol{u}} \boldsymbol{u}_P^\star = \boldsymbol{H}(\boldsymbol{u}^\star) - \nabla p^\star.$
Calculate the off-diagonal vector $\boldsymbol{H}(\boldsymbol{u}^\star)$ after obtaining the velocity field $\boldsymbol{u}^\star$.
Correct pressure and velocity: the new pressure field is computed based on the obtained velocity field from the last step:
    $\nabla \cdot [(a_P^{\boldsymbol{u}})^{-1} \nabla \tilde{p}] = \nabla \cdot [(a_P^{\boldsymbol{u}})^{-1} \boldsymbol{H}(\boldsymbol{u}^\star)].$
    Then correct the velocity explicitly by:
    $\tilde{\boldsymbol{u}}_P = (a_P^{\boldsymbol{u}})^{-1} (\boldsymbol{H}(\boldsymbol{u}^\star) - \nabla \tilde{p}).$
Relax the pressure field and the velocity equation with the prescribed under-relaxation factors $\alpha_p$ and $\alpha_u$, respectively.
    The under-relaxed fields are called $p^{\star\star}$ and $\boldsymbol{u}^{\star\star}$
Assemble the conservative face flux $F_f$:
    $F_f = \boldsymbol{u}_f \cdot \boldsymbol{S}_f = (a_P^{\boldsymbol{u}})^{-1} \boldsymbol{S}_f \cdot \boldsymbol{H}(\boldsymbol{u}^{\star\star}) - a_N^p (p_N^{\star\star} - p_P^{\star\star}).$
Set $\boldsymbol{u}^\star = \boldsymbol{u}^{\star\star}$ and $p^\star = p^{\star\star}$
Repeat until achieving convergence.
```

The PISO algorithm starts in a similar way to the SIMPLE, where it uses a guessed pressure field to carry out the momentum predictor step. Then it assembles the $\boldsymbol{H}(\boldsymbol{u}^\star)$ vector and corrects pressure and velocities. The next step represents the key difference between the two algorithms, where at this point the velocity field has been corrected after solving the pressure equation. This means that the $\boldsymbol{H}$ vector has been changed and so has the source term in the pressure equation, which makes the pressure field no longer correct and thus one has to update the $\boldsymbol{H}$ vector. To address this issue, the SIMPLE the algorithm goes all the way back to the momentum equation and performs another momentum predictor, and thus it obtains a new velocity value which will be used to update the $\boldsymbol{H}$ vector for resolving the pressure equation. On the other hand, in the PISO loop the momentum predictor is just done once. In more details, rather than solving the momentum equation for the second time, it uses the corrected velocity field to update directly the $\boldsymbol{H}$ vector and then performs the second pressure correction step. The PISO complete procedure is structured in Algorithm 2.

There are other transient solvers which could be used for solving the couple velocity-pressure system. We mention here the PIMPLE algorithm which merges PISO and SIMPLE. The PIMPLE algorithm contains an outer loop which iterates over the PISO procedure resulting in a more robust algorithm in terms of stability. The number of times the outer loop is entered is known as the number of outer correctors of the PIMPLE, if it is set to one then the PIMPLE replicates the PISO.

Algorithm 2 The PISO algorithm

```
1: Perform steps $1-4$ in the SIMPLE algorithm:
       • Solve the discretized momentum equation.
       • Update the off-diagonal vector $\boldsymbol{H}$.
       • Solve the pressure correction equation.
       • Correct pressure and velocities.
2: Solve the second pressure correction, and obtain the second pressure correction component $p''$.
3: Correct pressure and velocities, this step gives $p^{\star\star\star}$ and $\boldsymbol{u}^{\star\star\star}$.
       $p^{\star\star\star} = p^{\star} + p' + p''$
4: Assemble the conservative face flux $F_f$:
       $F_f = \boldsymbol{u}_f \cdot \boldsymbol{S}_f = (a_P^{\boldsymbol{u}})^{-1}\boldsymbol{S}_f \cdot \boldsymbol{H}(\boldsymbol{u}^{\star\star\star}) - a_N^p(p_N^{\star\star\star} - p_P^{\star\star\star}).$
5: Set $\boldsymbol{u}^{\star} = \boldsymbol{u}^{\star\star\star}$ and $p^{\star} = p^{\star\star\star}$
6: Go to step 1 if convergence is not reached.
7: Proceed to the next time step if the final time is not reached.
```

2.3 Proper Orthogonal Decomposition (POD)

The Proper Orthogonal Decomposition (POD) is a technique which was first introduced for the identification of coherent structures in turbulent flows [133, 9]. The POD has been used then extensively in the generation of the reduced order spaces [149, 26, 12, 33, 19]. The method can be described as a process of exploring and compressing information embedded in high-dimensional spaces, and at the end this procedure retains the important information. One of the possible ways of carrying out the POD is based on the method of snapshots. In this thesis, we rely on this method for the generation of the POD reduced order space. The method starts by sampling the parameter space and computing the FOM solutions for each value of the parameter samples. Each of the FOM solutions computed is called a FOM snapshot. A finite discrete parameter set $\mathcal{P}_M = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_M\}$ is sampled from the parameter space $\mathcal{P}$, the cardinality of $\mathcal{P}_M$ is equal to M. In the general case of unsteady flows, the FOM solutions for the parameter samples are computed for different time instants $\{t_1, t_2, ..., t_{N_T}\} \subset [0, T]$, this means that the total number of snapshots is $N_s = M * N_T$. The snapshots matrices $\boldsymbol{\mathcal{S}}_{\boldsymbol{u}}$ and $\boldsymbol{\mathcal{S}}_{\boldsymbol{p}}$, for velocity and pressure, respectively, will be given by:

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{u}} = \{\boldsymbol{u}(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., \boldsymbol{u}(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N_u^h \times N_s}, \tag{2.25}$$

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{p}} = \{p(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., p(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N_p^h \times N_s}, \tag{2.26}$$

where N_u^h and N_p^h are the degrees of freedom for velocity and pressure fields, respectively.

The velocity and pressure POD spaces are the ones that solve the following minimization problems:

$$\mathbb{V}^u_{POD} = \text{arg min}\frac{1}{N_s}\sum_{n=1}^{N_s}||\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{u}} - \sum_{i=1}^{N_u}(\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{u}}, \boldsymbol{\phi}_i)_{L^2(\Omega)}\boldsymbol{\phi}_i||^2_{L^2(\Omega)} \quad \text{for all} \quad N_u = 1, ..., N_s, \tag{2.27}$$

$$\mathbb{V}^p_{POD} = \text{arg min}\frac{1}{N_s}\sum_{n=1}^{N_s}||\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{p}} - \sum_{i=1}^{N_p}(\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{p}}, \chi_i)_{L^2(\Omega)}\chi_i||^2_{L^2(\Omega)} \quad \text{for all} \quad N_p = 1, ..., N_s, \tag{2.28}$$

where $\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{u}}$ and $\boldsymbol{\mathcal{S}}^{\boldsymbol{n}}_{\boldsymbol{p}}$ are the $n-$th velocity and pressure snapshots, respectively, which have been obtained for a value of the parameter which belongs to the finite parameter set $\mathcal{P}_M$ and acquired at any time instant inside the time window [0,T]. The POD spaces are spanned by the POD basis/modes, namely $\mathbb{V}^u_{POD} = span\{[\boldsymbol{\phi}_i]_{i=1}^{N_u}\}$, and $\mathbb{V}^p_{POD} = span\{[\chi_i]_{i=1}^{N_p}\}$, with $N_u << N_u^h$, $N_p << N_p^h$. Therefore, the construction of the POD spaces requires the computation of the velocity POD basis $[\boldsymbol{\phi}_i]_{i=1}^{N_u}$ and the pressure POD basis $[\chi_i]_{i=1}^{N_p}$. Solving the minimization problems reported above is equivalent to solve the eigenvalue problem of the correlation matrix of the velocity and pressure fields, respectively [93]. The case of the velocity is addressed as follows:

$$(\boldsymbol{C}^{\boldsymbol{u}})_{ij} = \left(\boldsymbol{\mathcal{S}}^{\boldsymbol{i}}_{\boldsymbol{u}}, \boldsymbol{\mathcal{S}}^{\boldsymbol{j}}_{\boldsymbol{u}}\right)_{L^2(\Omega)}, \tag{2.29}$$

where $\boldsymbol{C}^{\boldsymbol{u}} \in \mathbb{R}^{N_s \times N_s}$ is the velocity correlation matrix. One can show that the velocity POD modes are computed as follows [137],

$$\boldsymbol{\phi}_i = \frac{1}{N_s\boldsymbol{\lambda}^{\boldsymbol{u}}_i}\sum_{j=1}^{N_s}\boldsymbol{\mathcal{S}}^{\boldsymbol{j}}_{\boldsymbol{u}}\boldsymbol{V}^{\boldsymbol{u}}_{ij}, \tag{2.30}$$

where $\boldsymbol{V}^{\boldsymbol{u}} \in \mathbb{R}^{N_s \times N_s}$ is the matrix whose columns are the eigenvectors of $\boldsymbol{C}^{\boldsymbol{u}}$ and $\boldsymbol{\lambda}^{\boldsymbol{u}}$ is a diagonal matrix whose entries are the eigenvalues of $\boldsymbol{C}^{\boldsymbol{u}}$. Similar procedure can be followed for the computation of the POD pressure modes $[\chi_i]_{i=1}^{N_p}$.

The POD space is optimal in the sense that, for any choice of the number of modes, it minimizes the difference between the snapshots and their projection onto the modes in the L^2 norm. In this thesis, the POD is applied on the snapshots matrices which have snapshots from mixed parameter values and acquired at different time instants. Other procedures for the generation of the reduced order space can be followed. Examples include the nested POD, where the POD is applied on the set of snapshots obtained by each parameter value and then another POD procedure is carried out on the resulting POD modes from the first step. Another approach is the one utilized in [67], where a POD-Greedy method (POD in time and RB method with the greedy algorithm in the parameter space) is used for the generation of the reduced order space.

2.4 Momentum Equation Solely ROM

After having introduced the POD method in the previous section, one may perform the projection procedure in order to construct the reduced order system. As mentioned earlier, in this thesis Galerkin projection is employed resulting in a POD-Galerkin ROM. At this point, different procedures could be followed for what concerns the construction of POD-Galerkin ROMs designed to reduce the NSE. In this section, the most simple option is addressed. This approach consists of the usage of just the momentum equation of the incompressible Navier–Stokes equations 2.1 at the reduced order level [103].

The assumption on which projection-based ROMs sit is that the dynamics of the PPDEs system can be described by few dominant modes. These modes when suitably combined, they give an accurate reproduction of the full order solution. The last assumption translates to the following decomposition approximation of velocity and pressure fields:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \boldsymbol{u_r}(\boldsymbol{x},t;\boldsymbol{\mu}) &= \textstyle\sum_{i=1}^{N_r} a_i(t;\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x}), \\ p(\boldsymbol{x},t;\boldsymbol{\mu}) \approx p_r(\boldsymbol{x},t;\boldsymbol{\mu}) &= \textstyle\sum_{i=1}^{N_r} a_i(t;\boldsymbol{\mu})\chi_i(\boldsymbol{x}), \end{aligned} \tag{2.31}$$

where $\boldsymbol{u_r}$ and p_r are the reduced velocity and pressure fields, respectively, while $\boldsymbol{\phi}_i(\boldsymbol{x})$ and $\chi_i(\boldsymbol{x})$ are the reduced modes for velocity and pressure, respectively. These modes do not depend on $\boldsymbol{\mu}$ and t. The reduced order degrees of freedom are denoted by $a_i(t;\boldsymbol{\mu})$ for both of velocity and pressure, the number of these degrees of freedom is N_r (notice that N_u and N_p in Equation 2.27 and Equation 2.28 are equal to N_r in this formulation). The temporal coefficients $a_i(t;\boldsymbol{\mu})$ depend on time t and on the parameter vector $\boldsymbol{\mu}$. One can notice that in Equation 2.31 the velocity and pressure fields share the same temporal coefficients. This last assumption makes the projection-based ROM simpler as mentioned earlier. However, this assumption has several limitations and drawbacks as it may be foreseen, in the next sections these drawbacks will be addressed in greater details.

The velocity POD modes are computed using Equation 2.30. On the other hand, the pressure modes in this formulation are computed using the eigenvectors and the eigenvalues obtained in the SVD procedure for the velocity POD modes. Thus, the pressure modes are expressed as:

$$\chi_i = \frac{1}{N_s \lambda_i^u} \sum_{j=1}^{N_s} \boldsymbol{\mathcal{S}}_{\boldsymbol{p}}^{j} V_{ij}^{u}. \tag{2.32}$$

The next step in building the POD-Galerkin ROM is to project the momentum equation of 2.1 onto the velocity POD basis $[\boldsymbol{\phi}_i]_{i=1}^{N_r}$ as follows:

$$\left(\boldsymbol{\phi}_i, \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) - \boldsymbol{\nabla}\cdot\nu\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right) + \boldsymbol{\nabla}p\right)_{L^2(\Omega)} = 0. \tag{2.33}$$

The following reduced order dynamical system is obtained after inserting the decomposition assumptions of Equation 2.31:

$$\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} - \boldsymbol{H}\boldsymbol{a}, \tag{2.34}$$

where $\boldsymbol{a}$ is the vector of reduced order degrees of freedom, and each of $\boldsymbol{B}, \boldsymbol{B_T}, \boldsymbol{C}$ and $\boldsymbol{H}$ is either a reduced order matrix or tensor. These terms are computed as follows:

$$(\boldsymbol{B})_{ij} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\boldsymbol{\phi}_j\right)_{L^2(\Omega)}, \tag{2.35}$$

$$(\boldsymbol{B_T})_{ij} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla}\boldsymbol{\phi}_j^T)\right)_{L^2(\Omega)}, \tag{2.36}$$

$$(\boldsymbol{C})_{ijk} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot (\boldsymbol{\phi}_j \otimes \boldsymbol{\phi}_k)\right)_{L^2(\Omega)}, \tag{2.37}$$

$$(\boldsymbol{H})_{ij} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla}\chi_j\right)_{L^2(\Omega)}. \tag{2.38}$$

It is important to mention that the convective non-linear term in the Navier–Stokes equations is approximated at the reduced order level through a third order tensor $\boldsymbol{C}$. This last approach in handling the non-linear term in the momentum equation could potentially increase the computational cost when the number of reduced velocity modes grows. Other techniques for the treatment of this term include EIM-DEIM methods [159, 20] and the Gappy-POD method [37]. At this stage, Equation 2.34 can be solved for the reduced degrees of freedom of the fluid dynamics fields. This system can be integrated for time values which are outside the time window in which snapshots were acquired, namely $[0, T]$. That case is referred to as extrapolation in time.

In the context of reduced order modeling, the notion of having two fully decoupled stages named the offline and the online stages is crucial. The offline stage represents the training stage which starts by sampling the parameter space. Then the FOM simulations are run and snapshots are acquired. The last step in the offline stage involves the computation of the POD modes, as well as any other term which depends on the POD modes such as the reduced order matrices and tensors. The offline stage is characterized in being of significant computational cost since the computations depend on the dimension of the FOM. On the other hand, the online stage involves fast computations which should not depend on the dimension of the FOM. In the POD-Galerkin ROM developed in this section, the online stage consists of solving Equation 2.34 for parameter values which could be new ones with respect to those used in the training of the model.

2.5 Momentum and Poisson Pressure Equations ROM

This sections presents a POD-Galerkin ROM which is based on the usage of both the momentum equation and the Poisson pressure equation at the reduced order level. This ROM approach is first presented in [137]. This POD-Galerkin ROM can be used only for the reduction of unsteady flows.

The POD-Galerkin ROM presented in the last section has shown lack of accuracy for what regards the reconstruction of the pressure field at the reduced order level. An accurate pressure approximation is imperative as many outputs of interest depend highly on the pressure field. Consequently, stabilization methods have been considered in order to reproduce the pressure field in an accurate fashion. In this thesis, two stabilization techniques are addressed. The first one exploits the Poisson equation for pressure at the reduced order level which makes, the separation of the pressure reduced degrees of freedom from the ones of the velocity, possible. The Poisson equation for pressure can be obtained by taking the divergence of the momentum equation in 2.1 and then taking advantage from the fact that the divergence of the velocity field is null. The resulting system is the following:

$$\begin{cases} \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = \boldsymbol{\nabla} \cdot \left[-p\mathbf{I} + \nu \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T \right) \right] & \text{in } \varOmega \times [0,T], \\ \varDelta p = -\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u})) & \text{in } \varOmega, \\ + \text{ Boundary conditions} & \text{on } \varGamma \times [0,T], \\ + \text{ Initial conditions} & \text{in } (\varOmega, 0). \end{cases} \tag{2.39}$$

We remark that the ROM developed in this section can not be used for the reduction of steady flows [84, 99]. In fact, among the possible choices for the pressure boundary condition which is required to render the PPE formulation of the NSE equivalent to the original one, we selected one which ensures that the velocity is divergence free only in the unsteady setting (see the remark in section 2 in [84]).

The reduced order decomposition assumptions in this case are:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},t;\boldsymbol{\mu}) &\approx \boldsymbol{u_r}(\boldsymbol{x},t;\boldsymbol{\mu}) = \textstyle\sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x}), \\ p(\boldsymbol{x},t;\boldsymbol{\mu}) &\approx p_r(\boldsymbol{x},t;\boldsymbol{\mu}) = \textstyle\sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i(\boldsymbol{x}), \end{aligned} \tag{2.40}$$

where one can see that new temporal coefficients denoted by $b_i(t;\boldsymbol{\mu})$ have been introduced for the approximation of the reduced pressure field. It is worth mentioning that the way the pressure modes are computed now is different to the previous section, where the SVD decomposition of the pressure correlation matrix $\boldsymbol{C^p} \in \mathbb{R}^{N_s \times N_s}$ has to be done. The entries of the matrix $\boldsymbol{C^p}$ are:

$$(\boldsymbol{C^p})_{ij} = \left(\boldsymbol{\mathcal{S}_p^i}, \boldsymbol{\mathcal{S}_p^j} \right)_{L^2(\varOmega)}, \tag{2.41}$$

then the pressure modes read as follows:

$$\chi_i = \frac{1}{N_s \boldsymbol{\lambda_i^p}} \sum_{j=1}^{N_s} \boldsymbol{\mathcal{S}_p^j} \boldsymbol{V_{ij}^p}, \tag{2.42}$$

where $\boldsymbol{\lambda^p}$ is the matrix containing in its diagonal the eigenvalues of the pressure correlation matrix $\boldsymbol{C^p}$ and $\boldsymbol{V^p}$ is the corresponding matrix of eigenvectors. At this point one may proceed in performing Galerkin projection of the equations. The momentum equation is projected onto the velocity POD modes, while the Poisson equation for pressure is projected onto the pressure POD space, these projections yield:

$$\left(\boldsymbol{\phi}_i, \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \left[-p\mathbf{I} + \nu \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T \right) \right] \right)_{L^2(\Omega)} = 0, \tag{2.43a}$$

$$\left(\nabla \chi_i, \nabla p + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u})\right)_{L^2(\Omega)} - \nu \left(\boldsymbol{n} \times \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \times \boldsymbol{u}\right)_\Gamma \; - (\chi_i, \boldsymbol{n} \cdot \boldsymbol{R}_t)_\Gamma = 0, \tag{2.43b}$$

where in the last equation, $\boldsymbol{R}$ is the initial velocity field. Substituting the reduced order approximations into the projected equations gives the following reduced order dynamical system:

$$\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a} - \boldsymbol{H} \boldsymbol{b}, \tag{2.44a}$$

$$\boldsymbol{D}\boldsymbol{b} + \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a} - \nu \boldsymbol{N} \boldsymbol{a} - \boldsymbol{L} = 0, \tag{2.44b}$$

where the new additional matrices and tensors are defined as follows:

$$(\boldsymbol{D})_{ij} = \left(\boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \chi_j\right)_{L^2(\Omega)}, \quad (\boldsymbol{G})_{ijk} = \left(\boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \cdot (\boldsymbol{\phi}_j \otimes \boldsymbol{\phi}_k)\right)_{L^2(\Omega)}, \tag{2.45}$$

$$(\boldsymbol{N})_{ij} = \left(\boldsymbol{n} \times \boldsymbol{\nabla} \chi_i, \boldsymbol{\nabla} \times \boldsymbol{\phi}_j\right)_\Gamma, \quad (\boldsymbol{L})_i = (\chi_i, \boldsymbol{n} \cdot \boldsymbol{R}_t)_\Gamma . \tag{2.46}$$

The reduced order system Equation 2.44 is a differential-algebraic system of equations (DAE). This DAE can be solved to obtain at the end the reduced order vector degrees of freedom for velocity and pressure named $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. As mentioned earlier the ROM approach presented in this section is restricted to work with unsteady flows. Therefore, a different ROM procedure is needed for the treatment of steady flows.

2.6 Momentum and Continuity Equations ROM using a Supremizer Stabilization Method

The supremizer enrichment approach introduced in [16] has been successful in stabilizing POD-Galerkin ROMs. This approach is based on enriching the velocity POD space by velocity-like modes which solve a supremizer problem associated with each of the pressure modes or snapshots. This procedure guarantees the fulfillment of a reduced order version of the inf-sup condition. As it will be shown shortly, employing the supremizer enrichment technique in the ROM formulation will allow the usage of the continuity equation at the reduced order level.

It is worth mentioning that there are two versions of the supremizer enrichment technique. The first one called the exact supremizer enrichment in which one solves the supremizer problem for each pressure mode χ_i obtaining at the end the velocity-like mode $\boldsymbol{s}(\chi_i)$. Then the supremizers $[\boldsymbol{s}(\chi_i)]_{i=1}^{N_S}$ are added to the velocity basis. In the second approach called approximate supremizer enrichment, the supremizer problem is solved for each pressure snapshot $\boldsymbol{\mathcal{S}_p^i}$ which gives the supremizer $\boldsymbol{s}(\boldsymbol{\mathcal{S}_p^i})$. Then a POD is applied on the supremizer snapshots matrix which yields the supremizer POD modes. These modes are then used to enrich the velocity POD space. If we adopt the first approach, the supremizer problem reads as follows:

$$\begin{cases} \Delta \boldsymbol{s}_i = -\boldsymbol{\nabla}\chi_i & \text{in } \Omega, \ \forall \chi_i \in \mathbb{V}^p_{POD}, \\ \boldsymbol{s}_i = \boldsymbol{0} & \text{on } \partial\Omega. \end{cases} \tag{2.47}$$

The supremizer space is then constructed:

$$\mathbb{V}^s_{POD} = [\boldsymbol{s}(\chi_1), \boldsymbol{s}(\chi_2), ..., \boldsymbol{s}(\chi_{N_p})]. \tag{2.48}$$

After that the velocity POD space can be enriched with the supremizer modes:

$$\tilde{\mathbb{V}}^u_{POD} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{N_u}] \oplus [\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_S}] \in \mathbb{R}^{N_u^h \times (N_u + N_S)}. \tag{2.49}$$

The enriched velocity POD space $\tilde{\mathbb{V}}^u_{POD}$ and its i-th basis $\tilde{\boldsymbol{\phi}}_i$ will still be called $\mathbb{V}^u_{POD}$ and $\boldsymbol{\phi}_i$, respectively, for making the notations easier. However, the dimension of the original velocity POD space N_u will still have its meaning, and if the supremizer approach is used in the ROM formulation then N_S will be mentioned explicitly.

The original velocity POD modes (see Equation 2.30) are a linear combination of the velocity snapshots which are divergence free. Therefore, the projection of the continuity equation onto the pressure modes would have made no sense since the velocity modes have zero divergence. Now after introducing the supremizer modes, one may utilize the continuity equation for the projection procedure. The velocity and pressure decomposition assumptions are the same as the ones used in the previous section in Equation 2.40, also the velocity and pressure POD modes are computed in the same way. The projection of the momentum and continuity equations reads as follows:

$$\left(\boldsymbol{\phi}_i, \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) - \boldsymbol{\nabla}\cdot\left[-p\mathbf{I} + \nu\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right)\right]\right)_{L^2(\Omega)} = 0, \tag{2.50a}$$

$$(\chi_i, \boldsymbol{\nabla}\cdot\boldsymbol{u})_{L^2(\Omega)} = 0. \tag{2.50b}$$

The reduced order dynamical system resulted from inserting the reduced approximations into Equation 2.50, is the following:

$$\boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b}, \tag{2.51a}$$

$$\boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}. \tag{2.51b}$$

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:

$$(\boldsymbol{M})_{ij} = \left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j\right)_{L^2(\Omega)}, \tag{2.52}$$

$$(\boldsymbol{P})_{ij} = \left(\chi_i, \boldsymbol{\nabla} \cdot \boldsymbol{\phi}_j\right)_{L^2(\Omega)}. \tag{2.53}$$

2.7 Non-homogeneous Dirichlet Boundary Conditions Treatment

In simulating physical fluid dynamics problems, it is often required to impose non-homogeneous Dirichlet boundary conditions on certain parts of the boundary. This might be the case in inlet-outlet problems, where it is natural to have non-homogeneous Dirichlet velocity condition at the inlet. In the context of PPDEs, it may be very well the case that this non-homogeneous boundary velocity vector is the parameter under study. These aspects make the treatment of the non-homogeneous Dirichlet boundary conditions essential for building an accurate reduced order model.

The methods employed for the enforcement of non-homogeneous boundary Dirichlet condition include the penalty method [30, 10, 21, 88, 131] and the lifting function method [59, 62, 73].

Before entering into the details of both methods, we introduce the following notations: let $\Gamma_D \subset \Gamma$ 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. It is important to clarify 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, in a two dimensional problem let $\boldsymbol{U}_{\Gamma_{D_1}} = (U_x^1, 0)$ and $\boldsymbol{U}_{\Gamma_{D_2}} = (U_x^2, 0)$ be the velocity vectors that must be imposed at the Dirichlet boundaries Γ_{D1} and Γ_{D2}, respectively. In this case there are two non-homogeneous Dirichlet boundary conditions to set and thus $N_{BC} = 2$. Let $U_{BC,i,j}$ be the non-zero value of the i-th component of the velocity vector to be imposed at the reduced order level at the j-th part of the Dirichlet boundary Γ_{Dj}. Denote by $\boldsymbol{U_{BC}}$ the vector of all scalar velocities $U_{BC,i,j}$, this vector has a dimension of N_{BC}, and its k-th element is called U_{BCk}.

2.7.1 The Lifting/Control Function Method

In the lifting/control function method, the enforcement of the non-homogeneous boundary condition is done by introducing a lifting function which carries the non-homogeneity. This is followed by homogenizing the velocity snapshots by subtracting from each snapshots a suitably scaled version of the lifting function. This results in a set of velocity snapshots which have homogeneous boundary conditions. The POD method is applied on the new set of velocity snapshots giving at the end homogeneous POD modes.

The non-homogeneous Dirichlet boundary conditions are removed as follows:

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{U_{BC}} \cdot \boldsymbol{\phi_L}, \tag{2.54}$$

where $\boldsymbol{\phi_L} \in \mathbb{R}^{N_{BC} \times N_u^h}$ 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. At this point, one can apply the POD method on the snapshots matrix:

$$\boldsymbol{\mathcal{S}_{\tilde{u}}} = \{\tilde{\boldsymbol{u}}(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., \tilde{\boldsymbol{u}}(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N_u^h \times N_s}. \tag{2.55}$$

In the online stage the boundary velocity vector given is called $\boldsymbol{U^*_{BC}}$ which may contain values different from the ones present in the original velocity snapshots. It is sought to approximate the reduced order velocity field corresponding to $\boldsymbol{U^*_{BC}}$, the ROM velocity field is approximated as follows:

$$\boldsymbol{u}(\boldsymbol{x}, \cdot; \boldsymbol{U^*_{BC}}) \approx \boldsymbol{U^*_{BC}} \cdot \boldsymbol{\phi_L} + \sum_{i=1}^{N_u} a_i(\cdot)\boldsymbol{\phi_i}(\boldsymbol{x}). \tag{2.56}$$

It can be seen above that the boundary velocity vector $\boldsymbol{U_{BC}}$ is assumed to be the parameter under question. However, in the presence of non-homogeneous boundary condition/s for the velocity field at a part of the boundary, the same described boundary treatment procedure has to be followed in all the following cases:

- When building a ROM for the reproduction and extrapolation in time without parameters (non-parametrized ROM).
- With the parametrized case with the boundary velocity vector being one of the parameters.
- With the parametrized case with the boundary velocity vector not being a parameter.

The way of choosing a suitable lifting function is problem dependent. In the case of the reduction of unsteady non-parametrized cases, where reduction aims to reproduce time snapshots and potentially extrapolate in time, a possible choice of the lifting function could be the average of the offline velocity snapshots. A more general approach for the generation of appropriate lifting functions is to solve linear potential flow problems with a unitary boundary condition for each non-homogeneous boundary condition to be set. These problems are steady ones, therefore, an iterative procedure with a tentative velocity field is carried out till convergence. While solving each of these potential flows, the value of the starting guessed velocity field at the Dirichlet boundary has to be zero everywhere except for one scalar entity where the lifting function is sought. The converged velocity field will be considered as the lifting function corresponding to the non-homogeneous boundary condition at the aforementioned entity. Besides the requirement of having unitary boundary condition, the lifting functions have to be divergence free fields.

The lifting/control function method can be adapted to work with the ROMs addressed in section 2.5 (PPE-ROM) and section 2.6 (SUP-ROM). On the other hand, adapting this method with the ROM mentioned in section 2.4 (U-ROM) is not straightforward. This is because of the fact that the reduced order degrees of freedom of the velocity are the same as the ones of the pressure. Therefore, the homogenization of the velocity snapshots in Equation 2.54 has to be accompanied with a procedure that obtains the corresponding pressure snapshots. This procedure is not easily defined.

As a consequence of the additional lifting function mode/modes, the dimension of the velocity POD space $\mathbb{V}^u_{POD}$ will be $N_u + N_{BC}$ and $N_u + N_S + N_{BC}$ for the PPE-ROM and the SUP-ROM, respectively.

2.7.2 The Penalty Method

The enforcement of the non-homogeneous boundary conditions in the penalty method is done by presenting a constraint in the reduced order dynamical system. This is done by adding a term in the reduced momentum equation which has zero value everywhere except on the Dirichlet boundary. This method has been initially implemented in the context of the finite element method in [10, 21]. In [59], the authors utilized the penalty method for the enforcement of non-homogeneous time-dependent Dirichlet boundary conditions for the case of the flow around a circular cylinder at Reynolds of 100. The POD-Galerkin ROM in [131] introduced the penalty method in order to account for the time-dependent Dirichlet boundary conditions, the authors presented a study of the accuracy and the stability of the method depending on the penalty parameters values. The penalty method has been used also in other ROMs such as in [87, 88, 81, 103, 137].

The method can be employed to all the ROMs mentioned in the previous sections. For example the SUP-ROM addressed in section 2.6 will be modified as follows:

$$\boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b} + \tau\Big(\sum_{k=1}^{N_{BC}}(U_{BC_k}\boldsymbol{D^k} - \boldsymbol{E^k}\boldsymbol{a})\Big), \tag{2.57a}$$

$$\boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}, \tag{2.57b}$$

where τ is a penalization factor whose value is set by sensitivity analysis, or by automatic tuning as recently presented in [142], where an iterative penalty method is presented. Generally speaking, having a higher value of τ leads to a stronger enforcement of the boundary conditions but might ill-condition the dynamical system. The newly introduced boundary matrices $\boldsymbol{E^k}$ and vectors $\boldsymbol{D^k}$ are defined as follows:

$$(\boldsymbol{E^k})_{ij} = \left(\boldsymbol{\phi_i}, \boldsymbol{\phi_j}\right)_{L^2(\Gamma_{D_k})}, \quad \text{for all} \quad k = 1, ..., N_{BC}, \tag{2.58}$$

$$(\boldsymbol{D^k})_i = (\boldsymbol{\phi_i})_{L^2(\Gamma_{D_k})}, \quad \text{for all} \quad k = 1, ..., N_{BC}. \tag{2.59}$$

Unlike in the lifting function method, the POD is done here directly on the non-homogeneous velocity snapshots.

2.8 Lift and Drag Forces Offline/Online Computations

In this thesis we are interested in recovering specific performance indicators which are important in engineering problems. This section addresses one of them which is the forces acting on the surface of bodies immersed in the flow.

The forces $\boldsymbol{F}$ acting on the surface of a body denoted by $\partial\Omega_f$ are given by the following surface integral:

$$\boldsymbol{F} = \int_{\partial\Omega_f} (2\mu\boldsymbol{\nabla}\boldsymbol{u} - p\boldsymbol{I})\boldsymbol{n}ds. \tag{2.60}$$

As mentioned earlier, having full decoupling between the offline and the online stages is a vital feature of efficient reduced order modeling. One may approximate the reduced order forces $\boldsymbol{F_r}$ by simply computing the following integral after reconstructing the reduced fields:

$$\boldsymbol{F_r} = \int_{\partial\Omega_f} (2\mu\boldsymbol{\nabla}\boldsymbol{u_r} - p_r\boldsymbol{I})\boldsymbol{n}ds, \tag{2.61}$$

but this would require accessing the original mesh which means that the computational cost of carrying out this integral will depend on N_h (the number of degrees of freedom of the FOM). Therefore, an alternative approach is needed for having an efficient reconstruction of the fluid dynamics forces.

The approach which is used throughout this work involves inserting the reduced order approximations in Equation 2.60. These approximations could be the uniform ones in Equation 2.31 or the non-uniform ones in Equation 2.40. Assuming the latter approximations, the forces are computed as follows:

$$\boldsymbol{F_r} = \int_{\partial\Omega_f} (2\mu\boldsymbol{\nabla}(\sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x})) - \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i\boldsymbol{I})\boldsymbol{n}ds, \tag{2.62}$$

$$\boldsymbol{F_r} = \int_{\partial\Omega_f} 2\mu\sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})\boldsymbol{n}ds - \int_{\partial\Omega_f}\sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i\boldsymbol{n}ds, \tag{2.63}$$

$$\boldsymbol{F_r} = \sum_{i=1}^{N_u} a_i(t;\boldsymbol{\mu})\int_{\partial\Omega_f} 2\mu\boldsymbol{\nabla}\boldsymbol{\phi}_i(\boldsymbol{x})\boldsymbol{n}ds - \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\int_{\partial\Omega_f}\chi_i\boldsymbol{n}ds. \tag{2.64}$$

Let:

$$\boldsymbol{\delta}_i = \int_{\partial \Omega_f} 2\mu \boldsymbol{\nabla} \boldsymbol{\phi}_i(\boldsymbol{x}) \boldsymbol{n} ds, \quad \text{for} \quad i = 1, ..., N_u, \tag{2.65a}$$

$$\boldsymbol{\theta}_j = \int_{\partial \Omega_f} \chi_j \boldsymbol{n} ds, \quad \text{for} \quad j = 1, ..., N_p, \tag{2.65b}$$

where each of $\boldsymbol{\nabla} \boldsymbol{\phi}_i(\boldsymbol{x})$ and χ_j can be viewed as a velocity and a pressure field, respectively. The terms (2.65a) and (2.65b) can be precomputed during the offline stage and then stored.

In the online stage when a new time and parameter values of t^* and $\boldsymbol{\mu}^*$, respectively, are introduced, the full order forces are computed as follows:

$$\boldsymbol{F}(t^*; \boldsymbol{\mu}^*) = \int_{\partial \Omega_f} (2\mu \boldsymbol{\nabla} \boldsymbol{u}(\boldsymbol{x}, t^*; \boldsymbol{\mu}^*) - p(\boldsymbol{x}, t^*; \boldsymbol{\mu}^*) \boldsymbol{I}) \boldsymbol{n} ds, \tag{2.66}$$

while the reduced order approximation is the following

$$\boldsymbol{F_r}(t^*; \boldsymbol{\mu}^*) = \sum_{i=1}^{N_u} a_i(t^*; \boldsymbol{\mu}^*) \boldsymbol{\delta}_i - \sum_{j=1}^{N_p} b_j(t^*; \boldsymbol{\mu}^*) \boldsymbol{\theta}_j. \tag{2.67}$$

It is important to underline the fact that the above formulas for the reduced forces are valid just in the case of affine parameter dependency.

2.9 Contents Summary

This section provides a summary of the contents of the current chapter. Firstly, this chapter has introduced in section 2.1 the mathematical problem which is the Navier–Stokes equations for incompressible fluids. The discretization employed at the full order level in this work is the finite volume discretization which has been addressed in subsection 2.1.1. The numerical algorithms which are used by the full order solver are then explained in section 2.2.

Then the attention shifts to the reduction methodologies utilized in this thesis. At first, the POD (section 2.3) as a method for generating reduced order spaces is addressed. Then, a group of reduced order models (ROMs) dedicated to the reduction of the Navier–Stokes equations are presented. These ROMs are based on different approaches when it comes to the reduced solutions. The first one (in section 2.4) uses a uniform approach in which only the momentum equation is needed for the determination of the reduced solutions. As for the second one (in section 2.5), it makes use of the Poisson equation for pressure at the reduced level. The latter ROM assumes that the velocity and pressure reduced solutions are different, where the usage of the momentum and the pressure equation permits to close the reduced dynamical system. The third ROM (in section 2.6) is based on a well-known stabilization approach in the ROM community, which is the supremizer stabilization method. This ROM enriches the velocity space by adding to it the supremizer modes, this allows to use continuity equation at the reduced order level.

In several applications, one may have non-homogeneous boundary conditions for the velocity field at the Dirichlet boundary. For this reason, the methods for dealing with such conditions at the reduced order level have been considered in this chapter in section 2.7. These methods are the lifting function method (subsection 2.7.1) and the penalty method (subsection 2.7.2). section 2.7 has addressed the differences between these two methods and the possibility of merging them with the three ROMs presented earlier.

Finally, this chapter has explained how the reduced forces acting on a certain surface can be computed in section 2.8. This procedure is recommended in reduced order modeling since it is essential for avoiding a computational cost which is dependent on the dimension of the FOM.

This chapter focused on laminar incompressible flows in finite volume setting. At this point, the tools needed to address turbulent flows are partially in place. In the next chapter, turbulence treatment at the full and the reduced order levels is going to be covered.

Chapter 3

Hybrid Projection-based/Data-driven ROMs for the Turbulence Treatment in the Incompressible Navier–Stokes Equations

This chapter focuses on the turbulence treatment of the incompressible Navier–Stokes equation at both the full and the reduced order levels. Firstly, the turbulence modeling part at the full order level is explained with direct reference to the methodologies available in OpenFOAM. The second part is devoted to the construction of versatile ROMs for turbulent flows with the help of the paradigm of hybrid ROMs. The hybrid ROMs considered in this work employ a data-driven technique for the approximation of the turbulent eddy viscosity. This data-driven technique is the interpolation using the Radial Basis Functions (RBF). The interpolation procedure is presented in two forms, in which different maps are constructed and approximated. Finally, other turbulent ROMs are presented.

Contents

3.1 Turbulence Modeling in the Full Order Model

In chapter 2 we have addressed the finite volume discretization of the NSE. In addition, the solution algorithms for both steady and unsteady flows have been presented. These aspects (the discretization technique and the solution algorithms) of the full order model have been discussed taking into account the methodologies available in the full order solver used in this thesis which is OpenFOAM. In similar fashion, we are going to approach the issue of turbulence modeling in the full order model, where we have to take into consideration what is already available in OpenFOAM when we build the reduced order model. There are several turbulence modeling strategies implemented in OpenFOAM. In this work we relied mainly on two of them which are the Reynolds Average Navier–Stokes (RANS) equations and the Large Eddy Simulation (LES). In this thesis we relied on the first option for treating turbulence at the full order level. It is important to mention that turbulence modeling is a rich field, where numerous models and techniques could be found in literature but here we aim at presenting and discussing a small set of them, which are coinciding or closely related with the models employed in this work's computations. The next subsection aims to introduce the idea of Reynolds averaging and the closure problem in turbulence. In subsection 3.1.2 the notion of the eddy viscosity models is introduced.

3.1.1 The Closure Problem and Reynolds Averaging

One of the main traits of turbulent flows is that several flow properties in such regime exhibit high frequency random oscillations both in the time and space domains. For such reason, Reynolds [122] introduced the concept of time averaging, in which all the fields are expressed as the sum of mean and fluctuating parts. In the majority of cases, the fluctuating component appears as a vibration around an equilibrium, or *average* flow solution. In the rather common case in which fluid dynamics simulations are only aimed at identifying the characteristics of the mean flow one would ideally want to be able to only to solve for the time-averaged part of the fluid dynamic variables. To this end, following the Reynolds procedure, after decomposing all the variables into a sum of mean and fluctuating terms, the momentum and continuity equations are time averaged giving rise to another system of equations. In such averaged system, the unknowns are the mean components of all the flow fields. However the non-linearity of the NSE leads to residual terms in which the fluctuations are still present in the new system, where the number of equations is not equal to the number of unknowns. This problem is called the *closure problem*. This section aims to give an idea about the time averaging of the NSE and the closure problem.

If we consider a generic fluid dynamics scalar field called $\sigma(\boldsymbol{x}, t)$, then the Reynolds decomposition assumption of $\sigma(\boldsymbol{x}, t)$ is the following:

$$\sigma(\boldsymbol{x}, t) = \bar{\sigma}(\boldsymbol{x}, t) + \sigma'(\boldsymbol{x}, t), \tag{3.1}$$

where $\bar{\sigma}(\boldsymbol{x},t)$ and $\sigma'(\boldsymbol{x},t)$ are the mean and the fluctuating parts, respectively. In the case of steady flows, the mean part $\bar{\sigma}$ is a spatial field without dependence on time that is $\bar{\sigma} = \bar{\sigma}(\boldsymbol{x})$. In this case the mean field is computed as follows:

$$\bar{\sigma}(\boldsymbol{x}) = \lim_{T\to\infty} \frac{1}{T} \int_t^{t+T} \sigma(\boldsymbol{x},\tau) d\tau. \tag{3.2}$$

As for the unsteady case, assuming that the fluctuating and the mean fields time scales are T_1 and T_2, respectively. Time averaging is done by calculating the following integral

$$\bar{\sigma}(\boldsymbol{x},t) = \frac{1}{T} \int_t^{t+T} \sigma(\boldsymbol{x},\tau) d\tau, \quad T_1 \leqslant T \leqslant T_2. \tag{3.3}$$

The overbar notation is used from here on to indicate any time-averaged quantity. We here recall that the time average of the mean field $\bar{\sigma}(\boldsymbol{x},t)$ is the mean field itself, while the time average of the fluctuating part $\sigma'(\boldsymbol{x},t)$ is zero.

The idea of Reynolds averaging can be used to form a new set of equations for the averaged part of each fluid dynamics variable starting from the NSE. Before doing that, deriving the formulas of the time averaging of non-linear quantities is needed. We first consider the time average of the product of two scalar quantities named ϕ and ψ:

$$\overline{\phi\psi} = \overline{(\bar{\phi}+\phi')(\bar{\psi}+\psi')} = \overline{\bar{\phi}\bar{\psi} + \bar{\phi}\psi' + \bar{\psi}\phi' + \phi'\psi'} = \bar{\phi}\bar{\psi} + \overline{\phi'\psi'}, \tag{3.4}$$

where one exploits the fact that the product of a mean quantity and a fluctuating quantity has zero mean. On the other hand, the quantity $\overline{\phi'\psi'}$ is not zero and this basically means that product of the means $\bar{\phi}\bar{\psi}$ is not equal to the mean of the products $\overline{\phi\psi}$. In fact the two quantities ϕ and ψ are called uncorrelated if $\overline{\phi'\psi'} = 0$, and otherwise they are correlated.

Once these instruments are in place, one may derive the Reynolds averaged NSE. Starting from the NSE equations 2.1 but written in the scalar form:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{3.5}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + 2\nu \frac{\partial \mathcal{S}_{ji}}{\partial x_j}, \tag{3.6}$$

where the Einstein summation convention has been used, and where $\mathcal{S}$ is the strain-rate tensor, defined by:

$$\mathcal{S}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{3.7}$$

The convective term can be written as follows:

$$u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial\left(u_i u_j\right)}{\partial x_j} - u_i \frac{\partial u_j}{\partial x_j} = \frac{\partial\left(u_i u_j\right)}{\partial x_j}, \tag{3.8}$$

where Equation 3.5 has been used, this will simplify the momentum equation to:

$$\frac{\partial u_i}{\partial t} + \frac{\partial \left(u_i u_j\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + 2\nu \frac{\partial \mathcal{S}_{ji}}{\partial x_j}. \tag{3.9}$$

The application of Reynolds time averaging to Equation 3.5 and Equation 3.9 leads to the Reynolds averaged equations, namely:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \tag{3.10}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \left(\bar{u}_i \bar{u}_j + \overline{u_i' u_j'}\right)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + 2\nu \frac{\partial \mathcal{S}_{ji}}{\partial x_j}. \tag{3.11}$$

We point out that the averaged continuity equation is identical to the original one, which also means that the fluctuating velocity component u_i' has zero divergence. As for the momentum equation, not all the fluctuating terms appearing in it have vanished. In fact, the term $\overline{u_i' u_j'}$ still appears in the momentum equation because of the non-linearity of the convective term.

The averaged momentum equation can be rewritten as

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + 2\nu \frac{\partial \left(\mathcal{S}_{ji} - \overline{u_i' u_j'}\right)}{\partial x_j}, \tag{3.12}$$

in which the term $\overline{u_i' u_j'}$ is often referred to as the Reynolds stress tensor $\mathcal{R}$, where $\mathcal{R}_{ij} = \overline{u_i' u_j'}$. The Reynolds stress tensor is a symmetric tensor with 6 unknown components. Therefore, the Reynolds averaging has brought 6 additional unknowns without any additional equation. The system composed by Equation 3.12 and Equation 3.10 is called the Reynolds Averaged Navier–Stokes (RANS) equations. The bottom line in turbulence modeling is to deal with the Reynolds stress tensor in order to close the system so as to compute the mean flow fields. This issue in turbulence modeling is the aforementioned *closure problem*.

3.1.2 The Eddy Viscosity Models (EVMs)

Over the years the closure problem has been addressed by several methodologies, characterized by different computational costs and accuracy of the results. Among others we mention methods which consists in deriving a transport equation for the Reynolds stress tensor components. Such closure models are obtained by taking moments of the NSE, that is by multiplying the NSE by a fluctuating quantity and then taking the mean. However, this procedure will result in additional non-linear terms containing fluctuation products averages which at the end increase again the number of unknowns. Specifically, these additional unknowns include a correlation term involving fluctuating velocity components coming from the convective term, and also other correlation terms from the diffusive part and a term which amounts to a fluctuating velocity-pressure correlation. Overall, the procedure brings in 22 new unknowns, which makes the closure problem even more apparent, and forces to approximate the unknown correlations in terms of the known mean flow fields. To avoid such complexity, several algorithms in RANS turbulence modeling are based on the Boussinesq eddy viscosity assumption [32]. The Boussinesq assumption (or hypothesis) states that the Reynolds stress tensor is proportional to the trace-less strain rate tensor $\mathcal{S}$, namely:

$$\mathcal{R}_{ij} = 2\nu_t \mathcal{S}_{ij} - \frac{2}{3} k \delta_{ij}, \tag{3.13}$$

where $k = \frac{1}{2}\overline{u_i' u_i'}$ is the turbulent kinetic energy and ν_t is the so-called artificial or eddy viscosity. The turbulence models which employ the Boussinesq eddy viscosity assumption are often called the Eddy Viscosity Models (EVMs). In these models, the eddy viscosity is approximated making use of either algebraic relations or transport-diffusion PDEs for other quantities which have an algebraic relationship with the eddy viscosity. The first group of EVMs is the algebraic models or the zero-equation models. As an example we mention the mixing length turbulence model [119]. First examples of PDEs-based model are represented by the Spalart–Allmaras model [135]. In such model one solves for a viscosity-like variable called $\tilde{\nu}$, where the eddy viscosity is related to $\tilde{\nu}$ through an algebraic relation. Richer EVMs are based on solving two additional transport-reaction-diffusion PDEs for specific turbulent flow variables. Among which we here mention the $k - \varepsilon$ and the $k - \omega$ models [85, 92]. In these models, k, ε and ω represent the turbulent kinetic energy, turbulent dissipation and the specific turbulent dissipation rate, respectively. The turbulent kinetic energy from the turbulent velocity fluctuations u_i' is measured per unit mass:

$$k = \frac{1}{2}\overline{u_i' u_i'} = \frac{1}{2}({u_1'}^2 + {u_2'}^2 + {u_3'}^2), \tag{3.14}$$

while turbulence dissipation rate ε is the rate at which turbulent kinetic energy is converted into thermal internal energy, and it is defined by:

$$\varepsilon = \nu \overline{\frac{\partial u_i'}{\partial x_k} \frac{\partial u_i'}{\partial x_k}}. \tag{3.15}$$

As for ω, it represents the conversion rate of turbulent kinetic energy into thermal internal energy per unit volume and time. It is defined implicitly in terms of k and ε, one of its varied definitions is:

$$\omega = \frac{\varepsilon}{k\beta*}, \tag{3.16}$$

where $\beta*$ is a constant equal to 0.09.

The first form of a two equations turbulence model was presented by Kolmogorov in [92] which is a $k - \omega$ model. Later on, other forms of the $k - \omega$ model were proposed with significant improvements such as in [125, 157, 158]. In [95] the standard $k - \varepsilon$ is presented. Another two equations model called the SST $k - \omega$ has been presented in [104], this turbulence model merges the $k - \omega$ model proposed by Wilcox and the $k - \varepsilon$ model. In fact, it has combined the advances in both standard version of the $k - \varepsilon$ and the $k - \omega$ models. The SST $k - \omega$ is used extensively in the CFD community, as it is a good compromise for flows with detachment and recirculation. This model also has been utilized in this thesis, for this reason we have chosen to report its complete turbulence equations.

The turbulent kinetic energy k equation is

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = P - \beta^* \omega k + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right], \tag{3.17}$$

and the second PDE for the specific turbulent dissipation rate ω is the following:

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \alpha \Omega_s^2 - \beta \omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \tag{3.18}$$

As for the terms and the constants which appear in the two equations above, they are defined as follows:

$$\begin{aligned}
&P = \min\left(\mathcal{R}_{ij} \frac{\partial u_i}{\partial x_j}, 10\beta^* \omega k \right), \quad W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right), \quad \Omega_s = \sqrt{2 W_{ij} W_{ij}}, \\
&\nu_t = \frac{a_1 k}{\max(a_1 \omega, \Omega_s F_2)}, \quad \phi = F_1 \phi_1 + (1 - F_1)\phi_2, \\
&F_1 = \tanh\left(\arg_1^4\right), \quad \arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\sigma_{\omega 2} k}{\mathrm{CD}_{k\omega} y^2} \right], \\
&\mathrm{CD}_{k\omega} = \max\left(2\rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-10} \right), \\
&F_2 = \tanh\left(\arg_2^2\right), \quad \arg_2 = \max\left(2\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \\
&\sigma_{k1} = 0.85, \quad \sigma_{w1} = 0.65, \quad \beta_1 = 0.075, \\
&\sigma_{k2} = 1.00, \quad \sigma_{w2} = 0.856, \quad \beta_2 = 0.0828, \\
&\beta^* = 0.09, \quad a_1 = 0.31.
\end{aligned}$$

Hence the RANS equations complemented by the SST $k - \omega$ turbulence model are written as follows:

$$
\begin{cases}
\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \boldsymbol{\nabla} \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}) = \boldsymbol{\nabla} \cdot \left[-\overline{p}\mathbf{I} + (\nu + \nu_t) \left(\boldsymbol{\nabla} \overline{\boldsymbol{u}} + (\boldsymbol{\nabla} \overline{\boldsymbol{u}})^T \right) \right] & \text{in } \Omega \times [0, T], \\
\boldsymbol{\nabla} \cdot \overline{\boldsymbol{u}} = 0 & \text{in } \Omega \times [0, T], \\
\nu_t = \frac{a_1 k}{\max(a_1 \omega, \Omega_s F_2)}, & \text{in } \Omega \times [0, T], \\
\frac{\partial k}{\partial t} + \overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla} k = P - \beta^* \omega k + \boldsymbol{\nabla} \cdot \left[(\nu + \sigma_k \nu_t) \boldsymbol{\nabla} k \right], & \text{in } \Omega \times [0, T], \\
\frac{\partial \omega}{\partial t} + \overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \omega = \alpha \Omega_s^2 - \beta \omega^2 + \boldsymbol{\nabla} \cdot \left[(\nu + \sigma_\omega \nu_t) \boldsymbol{\nabla} \omega \right] & \\
+2(1 - F_1) \frac{\sigma_{\omega 2}}{\omega} \boldsymbol{\nabla} \omega \cdot \boldsymbol{\nabla} k, & \text{in } \Omega \times [0, T], \\
+ \text{ Boundary conditions} & \text{on } \Gamma \times [0, T], \\
+ \text{ Initial conditions} & \text{in } (\Omega, 0).
\end{cases}
\tag{3.19}
$$

The RANS equations have been utilized extensively in literature for modeling turbulence in different fields. In [55] one may find an application of the RANS in aerodynamics. In the last work, the RANS equations have been used for modeling high-speed aerodynamic flow transition. In [89], the work employs the RANS equations for simulating steady turbulent problems in automotive engineering using the Spalart–Allmaras turbulence model. The RANS equations has been used also in studying wind dynamics in civil engineering, in [108] the authors present a three dimensional study involving the use of steady RANS for the prediction of mean wind pressure distributions on windward and leeward surfaces of a medium-rise building with and without balconies. In [53], the RANS equations with the $k - \varepsilon$ model are used to describe the hydrodynamic flow around the boat. The work [47] presents an application of the RANS equations for the design of sailing yachts.

3.2 Hybrid Projection-based/Data-driven ROM for Turbulent Flows

In section section 3.1 we have addressed turbulence modeling at the full order level. The focus now shifts to the treatment of turbulence at the reduced order level. We have mentioned that turbulence is simulated using the RANS equations with the help of a suitable eddy viscosity model. Thus, the construction of a turbulent ROM has to take into account the fact that the full order model involves several different turbulence variables. This is the case due to the abundant number of EVMs available to the user of OpenFOAM for closing the RANS equations.

The methodology which could be adapted at the reduced order level could involve decomposing each turbulence variables such as k, ε, ω or $\tilde{\nu}$ into a finite sum of spatial modes multiplied by temporal coefficients as was done with velocity and pressure in Equation 2.40. This is followed by carrying out the POD procedure onto the snapshots of each turbulence variable. After that, a Galerkin projection of the additional transport-diffusion PDEs can be done resulting in a reduced version of each turbulence PDE. A set of reduced equations is obtained from the projection of the momentum, the continuity and the turbulence additional equations. These reduced equations are then coupled and the reduced solution for velocity, pressure and each of the turbulence variables could be obtained by solving the coupled system. Yet, the last approach has several issues and is deemed inconvenient, in particular for the following reasons:

- The approach renders the task of solving the reduced order system in the online stage complex. This is because of the additional equations which have to be treated at the reduced order level, where the turbulent additional equations are characterized usually of having high level of non-linearity.

- The approach forces the creation and the maintenance of a ROM for each turbulence closure model available at the full order level. This customization of the reduced order model is clearly not practical, for instance a popular library such as OpenFOAM is well-supplied with various closure models. All of which would require a separate and specific reduced order model.

The last mentioned approach has been ruled out in this work due to the fact that the goal is to design unified reduction methodologies which work with multiple full order model solvers, each coming with its own specific implementation of several different turbulence models. To this end, the proposed approach in this thesis involves the extension of the decomposition assumption in Equation 2.40 only to the eddy viscosity ν_t without doing the same for the other turbulence variables such as k, ε, ω or $\tilde{\nu}$. This essentially means that a reduced order version of the eddy viscosity (ν_{t_r}) is introduced, namely:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) \approx \nu_{t_r}(\boldsymbol{x}, t; \boldsymbol{\mu}) = \sum_{i=1}^{N_{\nu_t}} g_i(t, \boldsymbol{\mu}) \eta_i(\boldsymbol{x}), \tag{3.20}$$

where $\eta_i(\boldsymbol{x})$ is the i-th eddy viscosity POD mode and $g_i(t, \boldsymbol{\mu})$ is the i-th coefficient of the POD expansion. If one considers the above expansion as an extension to those used in sections 2.5 and 2.6, then it can be seen that the reduced eddy viscosity assumes a different set of degrees of freedom to $a_i(t, \boldsymbol{\mu})$ and $b_i(t, \boldsymbol{\mu})$, which are the reduced velocity and pressure solutions, respectively. Indeed, in principle, $\boldsymbol{g}$ varies over time responding to variations of $\boldsymbol{a}$ and $\boldsymbol{b}$. This reflects at the reduced order level, the fact that the turbulent viscosity in every EVM depends on the mean flow field variables. At this point, one has to find a suitable way to compute the eddy viscosity reduced solution $[g_i(t, \boldsymbol{\mu})]_{i=1}^{N_{\nu_t}}$. Since the specific turbulence equations of each EVM will not be used, the approach chosen for the computation of the reduced eddy viscosity coefficients is based on data-driven methods and in particular interpolation with Radial Basis Functions [96, 105]. In detail, the reduced turbulent model treatment starts with computing the eddy viscosity modes making use of the snapshots method as explained in section 2.3. First, we define the eddy viscosity snapshots matrix:

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}} = \{\nu_t(\boldsymbol{x}, t_1; \boldsymbol{\mu}_1), ..., \nu_t(\boldsymbol{x}, t_{N_T}; \boldsymbol{\mu}_M)\} \in \mathbb{R}^{N^h_{\nu_t} \times N_s}, \tag{3.21}$$

where $\boldsymbol{\mathcal{S}}_{\nu_t}^{i}$ is the i-th column of the eddy viscosity snapshots matrix $\boldsymbol{\mathcal{S}}_{\nu_t}$. Then the eddy viscosity correlation matrix is computed

$$(\boldsymbol{C}^{\boldsymbol{\nu_t}})_{ij} = \left(\boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}}^{i}, \boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}}^{j}\right)_{L^2(\Omega)}. \tag{3.22}$$

The eddy viscosity modes are then expressed as:

$$\eta_i = \frac{1}{N_s \boldsymbol{\lambda}_i^{\boldsymbol{\nu_t}}} \sum_{j=1}^{N_s} \boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}}^{j} \boldsymbol{V}_{ij}^{\boldsymbol{\nu_t}}, \tag{3.23}$$

where the matrix $\boldsymbol{\lambda}^{\boldsymbol{\nu_t}}$ contains in its diagonal the eigenvalues of the matrix $\boldsymbol{C}^{\boldsymbol{\nu_t}}$ and $\boldsymbol{V}^{\boldsymbol{\nu_t}}$ is the matrix whose columns are the corresponding eigenvectors of $\boldsymbol{C}^{\boldsymbol{\nu_t}}$.

The next step consists into the training of the RBF using the data of the snapshots acquired. Later, during the online stage the coefficients $[g_i(t, \boldsymbol{\mu})]_{i=1}^{N_{\nu_t}}$ are interpolated.

We will extend the SUP-ROM presented in section 2.6 for the turbulent case with the decomposition assumption 3.20. This means that the momentum equation is projected onto the modes of the velocity and the continuity equation is projected onto the modes of pressure, with the usage of the supremizer stabilization approach. These projections will result in the following DAE:

$$\boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} + \boldsymbol{g}^T(\boldsymbol{C_{T1}} + \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b}, \tag{3.24a}$$

$$\boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}, \tag{3.24b}$$

where $\boldsymbol{g}$ is the vector containing the reduced order degrees of freedom of the eddy viscosity, and the new terms in the DAE above are defined as follows:

$$(\boldsymbol{C_{T1}})_{ijk} = \left(\boldsymbol{\phi}_i, \eta_j \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\boldsymbol{\phi}_k\right)_{L^2(\Omega)}, \tag{3.25}$$

$$(\boldsymbol{C_{T2}})_{ijk} = \left(\boldsymbol{\phi}_i, \boldsymbol{\nabla} \cdot \eta_j(\boldsymbol{\nabla}\boldsymbol{\phi}_k^T)\right)_{L^2(\Omega)}. \tag{3.26}$$

The system 3.24 has $N_u + N_p$ equations while the number of unknowns is $N_u + N_p + N_{\nu_t}$. This problem is solved, as mentioned, by employing an interpolation technique to obtain the vector $\boldsymbol{g}$. Thus for the vector $\boldsymbol{g}$ we are using a data-driven approach known in the literature as POD-I [151, 150, 127]. We would like to mention that the ROM in section 2.5 can be extended to the turbulent case as done above but the drawback of not working in the steady case will persist, therefore we have firstly extended the SUP-ROM. We remark that the system in 3.24 will have the penalty additional terms in case of using the penalty method for the treatment of the non-homogeneous boundary conditions.

The reduced order model that is put forward makes use of the projected momentum and continuity equations to obtain the reduced solution of the velocity and the pressure. On the other hand, it employs data-driven techniques for the approximation of the eddy viscosity solution manifold. These kind of ROMs are termed as Hybrid ROMs because they merge the classical projection-based methods with data-driven techniques. It is often common to label the projection-based methods with the term intrusive, while the term non-intrusive is used to describe the methods which employ the usage of the data. In this thesis we will refer to the ROM given by the reduced DAE in 3.24 as H-SUP-ROM.

The data-driven methods are helpful in approximating the maps between various terms and quantities which appear in the FOM formulation of the NSE. In this thesis, the data-driven method selected is, as mentioned, interpolation using RBF. This method is used for the purpose of approximating the coefficients $[g_i(t,\boldsymbol{\mu})]_{i=1}^{N_{\nu_t}}$ in 3.24. In order to do this, one has to make use of the data available for what concerns the eddy viscosity. This task can be done by properly choosing the maps needed to be approximated, or in other words, by choosing the suitable independent variable of the RBF interpolation. In the upcoming subsections, we are going to present two different methodologies for carrying the out the interpolation step with their corresponding hybrid ROM.

3.2.1 Hybrid ROM with RBF Interpolation Based on the Time-Parameter Values

In this subsection, we are going to assume that the independent variable of the RBF interpolation is the vector that merges the time and the parameter values. In order to render the methodology clear, we are going to set up the following conventions and notation, let $X_{\boldsymbol{\mu},t}$ be the set given by:

$$X_{\boldsymbol{\mu},t} = \mathcal{P}_M \bigtimes \{t_1, t_2, ..., t_{N_T}\}, \tag{3.27}$$

here $X_{\boldsymbol{\mu},t}$ is the Cartesian product of the set containing the time instants at which snapshots were acquired and the discretized parameter set. The set $X_{\boldsymbol{\mu},t}$ has a cardinality N_s and its i-th member will be denoted by $\boldsymbol{x}^i_{\boldsymbol{\mu},t}$. It can be seen that there is unique correspondence between the elements of $X_{\boldsymbol{\mu},t}$ and the columns of the matrix of snapshots (for all fields) used in the offline stage for the construction of the reduced basis.

The parameter sample introduced in the reduced order model at the online stage is denoted by $\boldsymbol{\mu}^*$. In order to have an accurate ROM result, the value of $\boldsymbol{\mu}^*$ should be close enough to the values of the parameter samples used in the offline phase. Let t^* be the time instant at which the ROM solution is sought, where $t_1 \leqslant t^* \leqslant t_{N_T}$. We refer to $\boldsymbol{z}^* = (t^*, \boldsymbol{\mu}^*)$ as the online time-parameter combined vector.

Let $g_{r,l}$ be the offline L^2 projection coefficient which results from the projection of the r-th eddy viscosity snapshot $\boldsymbol{\mathcal{S}}^r_{\boldsymbol{\nu_t}}$ onto the l-th eddy viscosity mode η_l, as follows:

$$g_{r,l} = (\boldsymbol{\mathcal{S}}^r_{\boldsymbol{\nu_t}}, \eta_l)_{L^2(\Omega)}, \quad \text{for} \quad r = 1, 2, ..., N_s \quad \text{and} \quad l = 1, 2, ..., N_{\nu_t}. \tag{3.28}$$

At this point, one may define the problem of the interpolation as follows. Given the set $X_{\boldsymbol{\mu},t}$, the corresponding eddy viscosity snapshots $[\boldsymbol{\mathcal{S}}^i_{\boldsymbol{\nu_t}}]_{i=1}^{N_s}$ and the corresponding coefficients $[g_{r,l}]_{r=1,l=1}^{N_s,N_{\nu_t}}$, approximate the vector $\boldsymbol{g}$ in (3.24) for the online time-parameter vector $\boldsymbol{z}^*$. The interpolation in this case is done on the scalar coefficients of the reduced eddy viscosity expansion $[g_i(t^*, \boldsymbol{\mu}^*)]_{i=1}^{N_{\nu_t}}$ (or denoted shortly by $[g_i(\boldsymbol{z}^*)]_{i=1}^{N_{\nu_t}}$), which means that one has to perform one interpolation for each of the N_{ν_t} modes used in the online stage.

It is worth remarking that the procedure described implies that each of the modes coefficients is interpolated independently from the others. Therefore, we may fix the eddy viscosity mode to be the L-th one η_L, where L is an arbitrary integer index, $1 \leqslant L \leqslant N_{\nu_t}$. The goal of the interpolation is to approximate the following map $[G_L]$ in $g_L = G_L(t, \boldsymbol{\mu})$:

$$G_L : \mathbb{R}^{q+1} \rightarrow \mathbb{R}, \quad \text{for any} \quad L = 1, 2, ..., N_{\nu_t}. \tag{3.29}$$

In order to approximate this map, firstly, we form the set of observations which consists of the coefficients of the L^2 projections of all the offline snapshots onto the eddy viscosity mode η_L, namely $\boldsymbol{Y}_L = [g_{r,L}]_{r=1}^{N_s} \in \mathbb{R}^{N_s}$. The goal will be to approximate the value of $g_L(\boldsymbol{z}^*) = g_L(t^*, \boldsymbol{\mu}^*)$.

Interpolation using RBF is based on the following formula:

$$G_L(\boldsymbol{z}) = \sum_{j=1}^{N_s} w_{L,j} \zeta_{L,j} \left(\left\| \boldsymbol{z} - \boldsymbol{x}^j_{\boldsymbol{\mu},t} \right\|_{L^2(\mathbb{R}^{q+1})} \right), \quad \text{for} \quad L = 1, 2, ..., N_{\nu_t}, \tag{3.30}$$

where $\zeta_{L,j}$ for $j = 1, ..., N_s$ are the RBF functions, $\boldsymbol{z} = (t, \boldsymbol{\mu})$, where here t refer to any time instant inside the snapshots window and $\boldsymbol{\mu}$ is a parameter value which lies in the parameter space $\mathcal{P}$, and $w_{L,j}$ are the weights of the radial basis functions which have to be computed during the training stage. In order to compute the weights, we will make use of the data obtained by the FOM:

$$G_L(\boldsymbol{x}^i_{\boldsymbol{\mu},t}) = g_{i,L}, \quad \text{for} \quad i = 1, 2, ..., N_s, \tag{3.31}$$

which implies that,

$$\sum_{j=1}^{N_s} w_{L,j} \zeta_{L,j} \left(\left\| \boldsymbol{x}^i_{\boldsymbol{\mu},t} - \boldsymbol{x}^j_{\boldsymbol{\mu},t} \right\|_{L^2(\mathbb{R}^{q+1})} \right) = g_{i,L}, \quad \text{for} \quad i = 1, 2, ..., N_s. \tag{3.32}$$

It is possible to rewrite the last equation as a linear system:

$$\boldsymbol{A}^{\zeta}_L \boldsymbol{w}_L = \boldsymbol{Y}_L, \tag{3.33}$$

where $(\boldsymbol{A}^{\zeta}_L)_{ij} = \zeta_{L,j} \left(\left\| \boldsymbol{x}^i_{\boldsymbol{\mu},t} - \boldsymbol{x}^j_{\boldsymbol{\mu},t} \right\|_{L^2(\mathbb{R}^{q+1})} \right)$, the linear system can be solved to obtain the weights $\boldsymbol{w}_L$, the latter will be stored in order to use them during the online stage. In this work, the linear system above is solved by a linear solver which is based on a QR decomposition with column pivoting, this solver has achieved relatively fast performance with acceptable accuracy.

In the online stage, the value of $\boldsymbol{g}(\boldsymbol{z}^*) = [g_i(\boldsymbol{z}^*)]_{i=1}^{N_{\nu_t}}$ may be computed as follows:

$$g_i(\boldsymbol{z}^*) \approx G_i(\boldsymbol{z}^*) = \sum_{j=1}^{N_s} w_{i,j}\zeta_{i,j}\left(\left\|\boldsymbol{z}^* - \boldsymbol{x}^j_{\boldsymbol{\mu},t}\right\|_{L^2(\mathbb{R}^{q+1})}\right), \quad \text{for} \quad i = 1, 2, ..., N_{\nu_t}. \tag{3.34}$$

The strategy of the RBF interpolation in this subsection has the deficiency of not being accurate for values of time t^* which lie outside the snapshots window $[0, T]$ (the RBF interpolant is trained with time values which lie inside the aforementioned range, and therefore, it is not logical to expect that it could give accurate approximation for values of t^* that do not fall in $[0, T]$). This could be considered as a major issue in several applications in reduced order modeling as extrapolation is generally regarded as a rather complex task compared to interpolation. In addition, the strategy of treating time as a part of the interpolation independent variable might not have physical sense. In fact, in several applications the flow could be periodic, which could make the absolute value of the time variable not relevant to the dynamics of the fluid fields. For such reason, we have tried to circumvent this problem by adopting a different methodology for the RBF interpolation, as explained in the next subsection.

3.2.2 Hybrid ROM with RBF Interpolation Based on the Velocity Projection Coefficients Values

The RBF interpolation methodology discussed in the last subsection is quite simple and straightforward. However, this methodology is limited to work with steady flows and unsteady flows without the capacity of carrying out extrapolation over time (outside the offline time window snapshots $[0, T]$). The H-SUP-ROM with the methodology presented in the last subsection has been put forth in [72], where it has been applied on a steady problem. This subsection suggests a different way of carrying out the RBF interpolation, in which the velocity projection coefficients (or a form of them as we will see) are to be the independent variable of the interpolation. The justification of this choice will be provided in the following discussion. The approach carried out in this subsection has been presented in [74].

Before addressing the interpolation independent variable choice though, it is important to introduce a modification to the way the eddy viscosity field was decomposed at the reduced order level in Equation 3.20. Primarily, we suggest to split the FOM eddy viscosity field as follows:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) = \overline{\nu_t}(\boldsymbol{x}; \boldsymbol{\mu}) + \nu_t'(\boldsymbol{x}, t; \boldsymbol{\mu}), \tag{3.35}$$

where basically we have divided the eddy viscosity into two contributions. The first one corresponds to the time-averaged viscosity field for the parameter $\boldsymbol{\mu}$, while the second term represents the time varying part. The rationale behind this proposal is based on the fact that the parameter variation is largely seen in the mean part $\overline{\nu_t}(\boldsymbol{x}; \boldsymbol{\mu})$. On the other hand, parameter changes do not affect to great extent the small oscillations in the field $\nu_t'(\boldsymbol{x}, t; \boldsymbol{\mu})$. In fact, we have noticed that, in the numerical examples discussed in this work, we may drop the dependency on the parameter for what concern the time varying field, i.e. $\nu_t'(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \nu_t'(\boldsymbol{x},t)$. This decomposition will turn out to be beneficial in the approximation of the reduced order eddy viscosity. The splitting of the eddy viscosity into two terms, one which is time dependent and the other which is parameter dependent, renders the approximation of ν_t an easier task.

The reduced order approximation of the ν_t is now modified as follows:

$$\nu_t(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \nu_{tr}(\boldsymbol{x},t;\boldsymbol{\mu}) = \sum_{i=1}^{M} \overline{g}_i(\boldsymbol{\mu})\overline{\eta}_i(\boldsymbol{x}) + \sum_{i=1}^{N_{\nu_t}} g_i(t)\eta_i(\boldsymbol{x}), \tag{3.36}$$

where the averaged part $\overline{\nu_t}(\boldsymbol{x};\boldsymbol{\mu})$ is approximated by the first sum in the above decomposition, and the second sum approximates the time varying part $\nu_t'(\boldsymbol{x},t)$. In this formulation, one may look at the fields $[\overline{\eta}_i]_{i=1}^M$ as additional eddy viscosity modes. The number of these fields is M which is the number of parameter samples used in the offline stage (or the dimension of the discretized parameter set $\mathcal{P}_M$). Each of these fields is computed by taking the time average over the offline snapshots which only correspond to one of the parameter samples inside $\mathcal{P}_M$. In other words, the field $\overline{\eta}_i$ is computed by taking the average of the eddy viscosity fields acquired at different time instants but correspond to the i-th parameter in $\mathcal{P}_M$. As for $[\overline{g}_i]_{i=1}^M$ they represent the degrees of freedom corresponding to the averaged fields, these coefficients are important for periodic regime problems. They are parameter dependent, and in case of testing the ROM in the online stage for the i-th parameter inside the offline parameter set (i.e. $\boldsymbol{\mu}^* = \boldsymbol{\mu}_i$), which is a reproduction test in the parameter space, then the vector $\overline{\boldsymbol{g}} = \boldsymbol{e}_i^M$, where $\boldsymbol{e}_i^M$ is the unit vector of dimension M consisting of zeros except for the its i-th component in which it has the value 1. The value of $\overline{\boldsymbol{g}}$ for cross validation tests (extrapolation in the parameter space) can be fixed by linear interpolation, for example if $\boldsymbol{\mu}^*$ lies at the distances d_1^* and d_2^* in the parameter space from its closest two offline parameter samples $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively (i.e. $\|\boldsymbol{\mu}^* - \boldsymbol{\mu}_1\|_{L^2(\mathbb{R}^q)} = d_1^*, \quad \|\boldsymbol{\mu}^* - \boldsymbol{\mu}_2\|_{L^2(\mathbb{R}^q)} = d_2^*, \quad d_1^* < \|\boldsymbol{\mu}^* - \boldsymbol{\mu}_i\|_{L^2(\mathbb{R}^q)}$ and $d_2^* < \|\boldsymbol{\mu}^* - \boldsymbol{\mu}_i\|_{L^2(\mathbb{R}^q)}$ for all $i = 3, 4, ...M$), then $\overline{\boldsymbol{g}}$ can be approximated as $\frac{d_2^*}{d_1^*+d_2^*}\boldsymbol{e}_1^M + \frac{d_1^*}{d_1^*+d_2^*}\boldsymbol{e}_2^M$. Finally the dynamical system of the H-SUP-ROM is modified as follows:

$$\boldsymbol{M}\dot{\boldsymbol{a}} = \nu(\boldsymbol{B}+\boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} + \overline{\boldsymbol{g}}^T(\overline{\boldsymbol{C_{T1}}}+\overline{\boldsymbol{C_{T2}}})\boldsymbol{a} + \boldsymbol{g}^T(\boldsymbol{C_{T1}}+\boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b}, \tag{3.37a}$$

$$\boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}, \tag{3.37b}$$

where the two new tensors are defined as:

$$(\overline{\boldsymbol{C_{T1}}})_{ijk} = \Big(\boldsymbol{\phi}_i, \overline{\eta}_j \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_k\Big)_{L^2(\Omega)}, \tag{3.38}$$

$$(\overline{\boldsymbol{C_{T2}}})_{ijk} = \Big(\boldsymbol{\phi}_i, \boldsymbol{\nabla}\cdot\overline{\eta}_j(\boldsymbol{\nabla}\boldsymbol{\phi}_k^T)\Big)_{L^2(\Omega)}. \tag{3.39}$$

At this point, we may come back to tackle the main issue in this subsection which is how we could choose the interpolation independent variable in such a way that leads to a hybrid ROM which could extrapolate over time. The suggested solution is by taking the velocity L^2

projection coefficients as the independent variable of the RBF interpolation. This choice is inspired by the fact that the eddy viscosity is ultimately depending on the mean velocity field and its gradient. The spirit of this choice is established on the fact that the eddy viscosity ν_t is a function of the time history of the velocity field $\boldsymbol{u}$. In other words, if we call the FOM eddy viscosity and the velocity at time t^n by ν_t^n and $\boldsymbol{u}^n$, respectively, then one may express the eddy viscosity as $\nu_t^n = \nu_t(\boldsymbol{u}^1, \boldsymbol{u}^2, ..., \boldsymbol{u}^n)$. In order to mimic the dependency between the velocity field and the eddy viscosity field at the reduced order level, we have to take into account the time evolution aspect of the last equation. This is done by assuming that the eddy viscosity reduced vector $\boldsymbol{g}$ is function of the velocity coefficients $\boldsymbol{a}$, along with their time derivative $\dot{\boldsymbol{a}}$, that is:

$$\boldsymbol{g}^n = \boldsymbol{g}^n(t^n) \approx \boldsymbol{g}^n(\boldsymbol{a}^n, \dot{\boldsymbol{a}}^n). \tag{3.40}$$

The last formula (3.40) represents an approximation of the relationship that exists between eddy viscosity and velocity fields. This approximation has been successfully used in the numerical examples that we considered in this work. However, the eddy viscosity is evolved through a PDE, albeit in many cases not directly but passing through an algebraic relationship with turbulence variables which in turn evolve through PDEs. Thus, it is possible that the approximation 3.40 could be eventually improved by including other fluid dynamics variables appearing in the turbulence model PDEs.

We proceed now to the training phase of the RBF in this formulation. This phase is conducted in the ROM offline stage. First, the L^2 projection coefficients of the velocity modes onto the snapshots are computed. Then, those corresponding to the supremizer are dropped, and the time derivatives of the projection coefficients are calculated using a backward Euler scheme. The velocity projection coefficients with their corresponding derivatives are put together to train the RBF. More in detail, the procedure starts from the solution snapshots, namely:

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{u}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{u}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{u}} \end{bmatrix}, \boldsymbol{\mathcal{S}}_{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{p}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{p}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{p}} \end{bmatrix}, \boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}} = \begin{bmatrix} \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_1},\boldsymbol{\nu_t}} \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_2},\boldsymbol{\nu_t}} \\ \vdots \\ \boldsymbol{\mathcal{S}}_{\boldsymbol{\mu_M},\boldsymbol{\nu_t}} \end{bmatrix}, \tag{3.41}$$

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

$$\begin{aligned} &\boldsymbol{a}^r_{\boldsymbol{\mu}_k,L^2} = [(\boldsymbol{\mathcal{S}}^{\boldsymbol{r}}_{\boldsymbol{\mu_k},\boldsymbol{u}}, \boldsymbol{\phi}_1)_{L^2(\Omega)}, ..., (\boldsymbol{\mathcal{S}}^{\boldsymbol{r}}_{\boldsymbol{\mu_k},\boldsymbol{u}}, \boldsymbol{\phi}_{N_u})_{L^2(\Omega)}], \\ &\text{for} \quad r = 1, 2, ..., N_T, \quad k = 1, 2, ..., M. \end{aligned} \tag{3.42}$$

Let

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{1},\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{a}^1_{\boldsymbol{\mu}_k,L^2} \\ \boldsymbol{a}^2_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \boldsymbol{a}^{N_T-1}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}, \boldsymbol{\mathcal{A}}_{\boldsymbol{2},\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{a}^2_{\boldsymbol{\mu}_k,L^2} \\ \boldsymbol{a}^3_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \boldsymbol{a}^{N_T}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}, \tag{3.43}$$

at this point, we can compute the time derivative vectors which are needed for the RBF interpolation by simply using the backward differentiation scheme as follows:

$$\dot{\boldsymbol{a}}^{r}_{\boldsymbol{\mu}_k,L^2} = \frac{\boldsymbol{a}^{r}_{\boldsymbol{\mu}_k,L^2} - \boldsymbol{a}^{r-1}_{\boldsymbol{\mu}_k,L^2}}{\Delta t_{\boldsymbol{\mu}_k}}, \quad \text{for} \quad r = 2,3,...,N_T, \quad k = 1,2,...,M, \tag{3.44}$$

where $\Delta t_{\boldsymbol{\mu}_k}$ is the time step at which snapshots were acquired for the parameter sample $\boldsymbol{\mu}_k$. This yields the following matrix of time derivative velocity coefficients:

$$\dot{\boldsymbol{\mathcal{A}}}_{\boldsymbol{k}} = \frac{\boldsymbol{\mathcal{A}}_{\boldsymbol{2,k}} - \boldsymbol{\mathcal{A}}_{\boldsymbol{1,k}}}{\Delta t_{\boldsymbol{\mu}_k}} = \begin{bmatrix} \dot{\boldsymbol{a}}^{2}_{\boldsymbol{\mu}_k,L^2} \\ \dot{\boldsymbol{a}}^{3}_{\boldsymbol{\mu}_k,L^2} \\ \vdots \\ \dot{\boldsymbol{a}}^{N_T}_{\boldsymbol{\mu}_k,L^2} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times N_u}. \tag{3.45}$$

In order to build the matrix which contain all the data observation needed to train the RBF, one has to merge the L^2 projection coefficients of velocity starting from the second time snapshot with the time derivative coefficients. This gives the following matrix:

$$\tilde{\boldsymbol{A}}_k = \begin{bmatrix} \boldsymbol{\mathcal{A}}_{\boldsymbol{2,k}} & \dot{\boldsymbol{\mathcal{A}}}_{\boldsymbol{k}} \end{bmatrix} \in \mathbb{R}^{(N_T-1)\times 2N_u}. \tag{3.46}$$

As for the eddy viscosity, one has to compute the projection coefficients of the eddy viscosity modes onto the snapshots corresponding to the time-varying part in Equation 3.35. In other words, the time-averaged part corresponding to the k-th parameter in the training set $\mathcal{P}_M$ which is $\overline{\nu_t}(\boldsymbol{x};\boldsymbol{\mu}_k)$ has to be subtracted from each eddy viscosity snapshot, as follows:

$$\begin{aligned} & g^{r}_{\boldsymbol{\mu}_k,i,L^2} = (\boldsymbol{\mathcal{S}}^{\boldsymbol{r}}_{\boldsymbol{\mu_k,\nu_t}} - \overline{\nu_t}(\boldsymbol{x};\boldsymbol{\mu}_k), \eta_i)_{L^2(\Omega)}, \quad \text{for} \quad r = 2,3,...,N_T, \\ & i = 1,2,...,N_{\nu_t} \quad \text{and} \quad k = 1,2,...,M. \end{aligned} \tag{3.47}$$

Let $\tilde{\boldsymbol{G}}_{i,k} \in \mathbb{R}^{(N_T-1)}$ be the vector containing the coefficients in (3.47) for a fixed i and k. The combined matrices and vectors for all parameter samples are denoted by $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{G}}_i$, respectively, and are defined as follows:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \tilde{\boldsymbol{A}}_1 \\ \tilde{\boldsymbol{A}}_2 \\ \vdots \\ \tilde{\boldsymbol{A}}_M \end{bmatrix} \in \mathbb{R}^{(N_s-M)\times 2N_u}, \tilde{\boldsymbol{G}}_i = \begin{bmatrix} \tilde{\boldsymbol{G}}_{i,1} \\ \tilde{\boldsymbol{G}}_{i,2} \\ \vdots \\ \tilde{\boldsymbol{G}}_{i,M} \end{bmatrix} \in \mathbb{R}^{(N_s-M)}. \tag{3.48}$$

The maps to be interpolated are the following $[G_i]_{i=1}^{N_{\nu_t}}$ in $g_i = G_i(\boldsymbol{a}, \dot{\boldsymbol{a}})$, where:

$$G_i : \mathbb{R}^{2N_u} \to \mathbb{R}, \quad \text{for} \quad i = 1,2,...,N_{\nu_t}. \tag{3.49}$$

This approximation is based on the interpolation points given in each row of the matrix $\tilde{\boldsymbol{A}}$ and the vector $\tilde{\boldsymbol{G}}_i$. The interpolation procedure relates the m-th row of $\tilde{\boldsymbol{A}}$ which is a vector called $\tilde{\boldsymbol{a}}^m_{L_2} \in \mathbb{R}^{2N_u}$ with the m-th element of $\tilde{\boldsymbol{G}}_i$ denoted by $[\tilde{G}_i]_m \in \mathbb{R}$.

In this formulation, the interpolation equation is the following

$$G_i(\tilde{\boldsymbol{a}}_{\text{online}}) = \sum_{j=1}^{N_s-M} w_{i,j}\zeta_{i,j}\left(\left\|\tilde{\boldsymbol{a}}_{\text{online}} - \tilde{\boldsymbol{a}}^j_{L_2}\right\|_{\mathbb{R}^{2N_u}}\right), \quad \text{for} \quad i = 1, 2, ..., N_{\nu_t}, \tag{3.50}$$

where $\tilde{\boldsymbol{a}}_{\text{online}} := [\boldsymbol{a}_{\text{online}}, \dot{\boldsymbol{a}}_{\text{online}}] \in \mathbb{R}^{2N_u}$ is a combination of the reduced velocity vector and its vector derivative. The FOM data is used to establish the following relation

$$G_i(\tilde{\boldsymbol{a}}^m_{L_2}) = [\tilde{G}_i]_m, \quad \text{for} \quad m = 1, 2, ..., N_s - M, \tag{3.51}$$

$$\sum_{j=1}^{N_s-M} w_{i,j}\zeta_{i,j}\left(\left\|\tilde{\boldsymbol{a}}^m_{L_2} - \tilde{\boldsymbol{a}}^j_{L_2}\right\|_{\mathbb{R}^{2N_u}}\right) = [\tilde{G}_i]_m, \quad \text{for} \quad m = 1, 2, ..., N_s - M. \tag{3.52}$$

In the online stage one can approximate $\boldsymbol{g}(\boldsymbol{a}^*, \dot{\boldsymbol{a}}^*) = [g_i(\boldsymbol{a}^*, \dot{\boldsymbol{a}}^*)]_{i=1}^{N_{\nu_t}}$:

$$g_i((\boldsymbol{a}^*, \dot{\boldsymbol{a}}^*)) \approx G_i([(\boldsymbol{a}^*, \dot{\boldsymbol{a}}^*)]) = \sum_{j=1}^{N_s-M} w_{i,j}\zeta_{i,j}\left(\left\|[(\boldsymbol{a}^*, \dot{\boldsymbol{a}}^*)] - \tilde{\boldsymbol{a}}^j_{L_2}\right\|_{\mathbb{R}^{2N_u}}\right). \tag{3.53}$$

We would like to remark that the H-SUP-ROM in its different versions discussed in this section is compatible with both the penalty method and the lifting function method. The interpolation discussed in this subsection could be done with the velocity coefficients coming from the lifting modes when the lifting function approach is employed. If that is the case, then the maps $[G_i]_{i=1}^{N_{\nu_t}}$ in $g_i = G_i(\boldsymbol{a}, \dot{\boldsymbol{a}})$ are:

$$G_i : \mathbb{R}^{2N_{BC}+2N_u} \to \mathbb{R}, \quad \text{for} \quad i = 1, 2, ..., N_{\nu_t}. \tag{3.54}$$

It is also important to underline that the homogenized set of velocity snapshots $\boldsymbol{\mathcal{S}}_{\tilde{\boldsymbol{u}}}$ in Equation 2.55 has to replace $\boldsymbol{\mathcal{S}}_{\boldsymbol{u}}$ in Equation 3.41 in case of the use of the lifting function method.

The H-SUP-ROM in Equation 3.37 can be solved by time integrating the dynamical system in the case of unsteady flows, or by simply solving the algebraic system in the case of steady flows. In this work, we used a Newton method for solving the reduced dynamical system. The Newton method computes the Jacobian in a numerical way. In the general unsteady case, it is recommended to choose a time step which is consistent with the one used during the FOM simulations. The time advancement scheme for computing the derivative of the reduced vector $\boldsymbol{a}$ is chosen as first order backward Euler scheme or a second order one.

3.3 Turbulent ROMs based on the Uniform-ROM and the PPE-ROM

In the last section, we have introduced the H-SUP-ROM proposed for the reduction of turbulent flows. The H-SUP-ROM is based on the projection of the momentum and continuity equations with the supremizer approach to obtain the reduced velocity and pressure degrees of freedom, while it employs a data-driven strategy for closing the system through approximation of the reduced eddy viscosity coefficients. In this section, we are going to address other relevant ROM formulations which may reduce turbulent flows under certain specific conditions.

The first turbulent ROM addressed here is an extension of the ROM addressed in 2.4 to which we refer as the Uniform-ROM or U-ROM. In this case, the extended ROM involves the approximation of the reduced eddy viscosity field through the use of the velocity degrees of freedom as it was done with the pressure field. In other words, the reduced eddy viscosity is given by:

$$\nu_t(\boldsymbol{x}, t; \boldsymbol{\mu}) \approx \nu_{tr}(\boldsymbol{x}, t; \boldsymbol{\mu}) = \sum_{i=1}^{N_r} a_i(t, \boldsymbol{\mu}) \eta_i(\boldsymbol{x}), \tag{3.55}$$

where the eddy viscosity modes $[\eta_i(\boldsymbol{x})]_{i=1}^{N_r}$ in this case are computed using the SVD decomposition of the velocity correlation matrix:

$$\eta_i = \frac{1}{N_s \boldsymbol{\lambda}_i^{\boldsymbol{u}}} \sum_{j=1}^{N_s} \boldsymbol{\mathcal{S}}_{\boldsymbol{\nu_t}}^{\boldsymbol{j}} \boldsymbol{V}_{ij}^{\boldsymbol{u}}. \tag{3.56}$$

The corresponding DAE for this turbulent ROM, which is only based on the projection of the momentum equation, is:

$$\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T(\boldsymbol{C} - \boldsymbol{C_{T1}} - \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H}\boldsymbol{a}, \tag{3.57}$$

where the vector $\boldsymbol{a}$ represents the degrees of freedom of the velocity, pressure and the eddy viscosity. This ROM can be used for the reduction of both steady and unsteady flows. It has also the advantage of being of low computational online cost, as the system 3.57 has relatively small size. However, as mentioned earlier, this formulation undermines the reduced pressure approximation. In addition, numerical evidences have demonstrated that this ROM is prone to long-term instabilities when the system 3.57 is integrated for long times (see section 4.3). Furthermore, the U-ROM has the limitation of not being compatible with the lifting function method described in subsection 2.7.1. The lifting function method requires a homogenization process for the velocity snapshots. The proposal of having unique set of reduced coefficients $[a_i(t, \boldsymbol{\mu})]_{i=1}^{N_r}$ is directly affected by the homogenization process of the velocity snapshots. The process of modifying the velocity snapshots suggests that such process should follow for the pressure and the eddy viscosity snapshots which is not well defined. In the next chapter, we will present a comparison between the turbulent U-ROM and the H-SUP-ROM with the penalty method used for the enforcement of the non-homogeneous boundary conditions.

At this point, one may proceed to the extension of the PPE-ROM presented in section 2.5. There are two different ways of extending such ROM. The first one is based on assuming that the reduced eddy viscosity field shares the same temporal coefficients with the velocity field as proposed in 3.56. In this formulation the velocity and the eddy viscosity fields will have the vector $\boldsymbol{a}$ as their reduced vector, while the pressure field will still be computed using different set of coefficients represented by the vector $\boldsymbol{b}$. This ROM is called Semi-Uniform-PPE-ROM or SU-PPE-ROM. We recall the reduced approximations of this ROM:

$$\overline{\boldsymbol{u}}(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \sum_{i=1}^{N_u} a_i(t,\boldsymbol{\mu})\boldsymbol{\phi}_i(\boldsymbol{x}), \quad \overline{p}(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \sum_{i=1}^{N_p} b_i(t;\boldsymbol{\mu})\chi_i(\boldsymbol{x}), \tag{3.58}$$

$$\nu_t(\boldsymbol{x},t;\boldsymbol{\mu}) \approx \sum_{i=1}^{N_u} a_i(t,\boldsymbol{\mu})\eta_i(\boldsymbol{x}), \tag{3.59}$$

with the POD modes computed as follows:

$$\boldsymbol{\phi}_i = \frac{1}{N_s\boldsymbol{\lambda}_i^{\boldsymbol{u}}}\sum_{j=1}^{N_s}\boldsymbol{u}_j\boldsymbol{V}_{ij}^{\boldsymbol{u}}, \quad \chi_i = \frac{1}{N_s\boldsymbol{\lambda}_i^{\boldsymbol{p}}}\sum_{j=1}^{N_s}p_j\boldsymbol{V}_{ij}^{\boldsymbol{p}}, \tag{3.60}$$

$$\eta_i = \frac{1}{N_s\boldsymbol{\lambda}_i^{\boldsymbol{u}}}\sum_{j=1}^{N_s}\nu_{tj}\boldsymbol{V}_{ij}^{\boldsymbol{u}}. \tag{3.61}$$

The FOM momentum and Poisson equations for the RANS turbulent modeling read as follow:

$$\begin{cases} \frac{\partial\overline{\boldsymbol{u}}}{\partial t} + \boldsymbol{\nabla}\cdot(\overline{\boldsymbol{u}}\otimes\overline{\boldsymbol{u}}) = \boldsymbol{\nabla}\cdot\left[-\overline{p}\mathbf{I} + (\nu+\nu_t)\left(\boldsymbol{\nabla}\overline{\boldsymbol{u}} + (\boldsymbol{\nabla}\overline{\boldsymbol{u}})^T\right)\right] & \text{in } \mathit{\Omega}\times[0,T], \\ \mathit{\Delta}\overline{p} = -\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\cdot(\overline{\boldsymbol{u}}\otimes\overline{\boldsymbol{u}})) + \boldsymbol{\nabla}\cdot\left[\boldsymbol{\nabla}\cdot\left(\nu_t\left(\boldsymbol{\nabla}\overline{\boldsymbol{u}} + (\boldsymbol{\nabla}\overline{\boldsymbol{u}})^T\right)\right)\right] & \text{in } \mathit{\Omega}, \\ + \text{ Boundary conditions} & \text{on } \mathit{\Gamma}\times[0,T], \\ + \text{ Initial conditions} & \text{in } (\mathit{\Omega},0). \end{cases} \tag{3.62}$$

The Galerkin projections of the momentum and the Poisson equations give the following DAE:

$$\dot{\boldsymbol{a}} = \nu(\boldsymbol{B}+\boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T(\boldsymbol{C} - \boldsymbol{C_{T1}} - \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{Hb}, \tag{3.63a}$$

$$\boldsymbol{Db} + \boldsymbol{a}^T(\boldsymbol{G} - \boldsymbol{C_{T3}} - \boldsymbol{C_{T4}})\boldsymbol{a} - \nu\boldsymbol{Na} - \boldsymbol{L} = 0. \tag{3.63b}$$

The additional tensors appearing in the reduced Poisson equation come from the divergence of the eddy viscosity term in the FOM momentum equation in Equation 3.19. Their entries are defined as:

$$(\boldsymbol{C_{T3}})_{ijk} = \left(\boldsymbol{\nabla}\chi_i, \eta_j \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\boldsymbol{\phi}_k\right)_{L^2(\Omega)}, \tag{3.64}$$

$$(\boldsymbol{C_{T4}})_{ijk} = \left(\boldsymbol{\nabla}\chi_i, \boldsymbol{\nabla} \cdot \eta_j (\boldsymbol{\nabla}\boldsymbol{\phi}_k^T)\right)_{L^2(\Omega)}. \tag{3.65}$$

The SU-PPE-ROM can not be used for the reduction of steady flows. This is related to the boundary conditions which have to be satisfied at the full order level. We have mentioned this issue in section 2.5, where it is related to the way the PPE is derived from the momentum and continuity equations of the NSE or the RANS. The derivation of an equivalent system to the one formed by the NSE requires an additional boundary condition for the divergence of the velocity or the pressure [84, 99]. In this work, the PPE formulation adopted involves an additional Neumann boundary condition for the pressure field. This latter condition is needed in order to make sure that the velocity field is divergence free for all time instants. The fulfillment of such condition can happen only in the general unsteady setting (see the remark in section 2 in [84]).

3.3.1 Hybrid ROM based on the PPE-ROM

In this subsection, we address the second choice for the extension of the PPE-ROM for the goal of reducing turbulent flows. The second methodology proposes to employ the reduced approximation of the eddy viscosity fields as the one used in the case of the H-SUP-ROM. That means decomposing ν_t as done in Equation 3.20 or in Equation 3.36, where this implies that the coefficients of the reduced eddy viscosity will be computed by the interpolation strategy explained in the last section. This ROM will be called H-PPE-ROM. Its DAE will read as follows:

$$\dot{\boldsymbol{a}} = \nu(\boldsymbol{B} + \boldsymbol{B_T})\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} + \overline{\boldsymbol{g}}^T(\overline{\boldsymbol{C_{T1}}} + \overline{\boldsymbol{C_{T2}}})\boldsymbol{a} + \boldsymbol{g}^T(\boldsymbol{C_{T1}} + \boldsymbol{C_{T2}})\boldsymbol{a} - \boldsymbol{H}\boldsymbol{b}, \tag{3.66a}$$

$$\boldsymbol{D}\boldsymbol{b} + \boldsymbol{a}^T\boldsymbol{G}\boldsymbol{a} - \overline{\boldsymbol{g}}^T(\overline{\boldsymbol{C_{T3}}} + \overline{\boldsymbol{C_{T4}}})\boldsymbol{a} - \boldsymbol{g}^T(\boldsymbol{C_{T3}} + \boldsymbol{C_{T4}})\boldsymbol{a} - \nu\boldsymbol{N}\boldsymbol{a} - \boldsymbol{L} = 0, \tag{3.66b}$$

the additional tensors $\overline{\boldsymbol{C_{T3}}}$ and $\overline{\boldsymbol{C_{T4}}}$ account for the parameter contribution of the eddy viscosity in the Poisson equation, they are calculated as:

$$(\overline{\boldsymbol{C_{T3}}})_{ijk} = \left(\boldsymbol{\nabla}\chi_i, \overline{\eta}_j \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\boldsymbol{\phi}_k\right)_{L^2(\Omega)}, \tag{3.67}$$

$$(\overline{\boldsymbol{C_{T4}}})_{ijk} = \left(\boldsymbol{\nabla}\chi_i, \boldsymbol{\nabla} \cdot \overline{\eta}_j (\boldsymbol{\nabla}\boldsymbol{\phi}_k^T)\right)_{L^2(\Omega)}. \tag{3.68}$$

Like the case of the PPE-ROM, its hybrid turbulent extension works just for non-stationary flows. The H-PPE-ROM has an advantage over the H-SUP-ROM in having less unknowns in the dynamical system because of the fact that the supremizer reduced degrees of freedom are not present in the H-PPE-ROM formulation. This often results in a slight advantage in terms of speed up during the online stage.

3.4 Contents Summary

In this section, we will give a recap of what has been presented in this chapter. The chapter is dedicated to the construction of turbulent ROMs for the incompressible Navier–Stokes equations. It starts with providing an overview of turbulence modeling at the full order level in section 3.1. The closure problem is addressed in subsection 3.1.1 and then the eddy viscosity models with the RANS formulation are considered in subsection 3.1.2.

Section 3.2 focuses on the development of a ROM which is capable of reducing turbulent flows. It presents a hybrid ROM which is based on both projection-based methods and data-driven techniques. The latter ROM is an extension to the one presented in section 2.6, and it employs interpolation using RBF for the approximation of the reduced eddy viscosity. Two different interpolation methodologies are presented in subsection 3.2.1 and subsection 3.2.2.

Finally, section 3.3 addresses other turbulent ROMs which are extended versions to the ones put forward in section 2.4 and section 2.5.

The main objective of this chapter is to provide a general reduction methodology for turbulent flows when the FOM is based on the RANS equations. This goal has been accomplished thanks to incorporating data-driven techniques in the ROM formulation.

The turbulent ROMs will be applied in the next chapter on different CFD problems. Comparisons on various fronts will be made.

Chapter 4

Applications and Numerical Results

In this chapter, the reduced order models developed in this thesis are tested on benchmark CFD cases. The reduction methodologies are applied on different CFD problems such as Uncertainty Quantification (UQ) problems and also turbulent problems. The results highlight the differences between the proposed reduction methods in several aspects. The first section of this chapter presents a numerical test for a UQ problem, in which the UQ technique employed is the non-intrusive Polynomial Chaos Expansion (PCE). In the same UQ problem, we will use the SUP-ROM developed in this work for the reduction of the parametrized problem. This test is aimed at evaluating the reliability and the accuracy of the SUP-ROM when considered as the input of the PCE instead of the FOM. This numerical test represents an interesting aspect in reduced order modeling for UQ problems. The second section addresses the application of the hybrid ROM constructed in the third chapter on a steady turbulent case. The results include the comparison with the turbulent uniform ROM introduced in the previous chapter. As for the third section, it deals with the unsteady case of the flow past a circular cylinder. This numerical case is considered in turbulent setting with Reynolds number $Re = O(10^5)$*. The hybrid and the non-hybrid turbulent ROMs are used for the reduction of this case. The results address the issues of long-time integration and ROM stability, accurate extrapolation in the parameter space and versatility in terms of the turbulent closure model employed at the FOM level. The finite volume C++ library OpenFOAM® (OF) [1] 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 [138].*

Contents

4.1 Applications of ROMs in CFD Uncertainty Quantification (UQ) Problems

In this section, we aim at merging the ROMs developed in this thesis with other approaches in Uncertainty Quantification (UQ). The objective of this numerical study is to limit the computational cost associated with the simulations required by the non-intrusive UQ algorithms. In fact, we aim at proposing a framework in which we provide the UQ algorithms with a computational asset that improves the performance of these algorithms. That asset is represented by the fast online computations performed by the ROMs developed in this work. To this end, the UQ algorithm chosen in this work is the non-intrusive Polynomial Chaos Expansion (PCE). Reduced order models have been used extensively in UQ problems, we refer the reader to the following works on ROMs in UQ available in literature [41, 40, 63].

In the first part of this section we will introduce physical problem of interest, on which we will apply the UQ and the ROM. Later we will address the PCE and we will give an overview of this UQ algorithm. Finally, we will present numerical results for the PCE and the SUP-ROM which are aimed at both comparing the two approaches and also to show that they could be combined for having better efficiency.

In CFD problems, input parameters uncertainties could affect the results of the numerical simulations resulting in a significant impact on the outputs of interest. In this section, we are going to study the methods which have been developed to evaluate these uncertainties. In particular, the method considered in this section is the non-intrusive Polynomial Chaos Expansion (PCE), which has been applied to the results of CFD simulations. In the context of the PCE method, random variables or random processes are represented in terms of orthogonal polynomials. The PCE method proposes the decomposition of the random variable into separable deterministic and stochastic components [100, 75]. The principal challenge in PCE consists in obtaining the deterministic coefficients of the expansion. In the framework on non-intrusive PCE, which is designed to require no changes to the CFD solver, the deterministic coefficients are computed in a post-processing phase which follows the simulations. In this case, a sampling based approach is employed for the identification of the coefficients as done in [80, 121]. In such settings, samples in the input parameter space are predefined and then numerical simulations are conducted for each sample. After that, the output of interest of the latter simulations is available, one may obtain the PCE coefficients, where the orthogonality of the polynomials is exploited to compute the deterministic coefficients in the expansion through integrals in the sampling space. However, the computational cost will grow significantly as the parameter space dimension increases, due to the fact that the sampling points chosen are quadrature points for such integrals. This represents a problem which has to be circumvented since the computational cost of the CFD simulations associated to the output evaluation at each sampling point is of considerable amount. In order to solve this problem, we relied in this work on a regression approach which is based on least squares minimization for the computation of the PCE coefficients.

In this work, we apply the PCE method to results obtained both with the full order solver and with the reduced order one. The goal of applying the PCE on the ROM results is to explore the possibility of offering further reductions of the computational cost. In addition, these tests have the objective of assessing how the PCE results are influenced by the use of the ROM developed in this work which is a POD-Galerkin based one. Thus, we will apply the POD-Galerkin ROM on the results obtained by solving the incompressible steady Navier–Stokes equations. The ROM-based PCE results will be compared to the original PCE results on the FOM data, and as a result, it will be possible to evaluate the ROM reliability as an output evaluator for the PCE.

In the next subsection we are going to introduce the physical problem, which is the flow around an airfoil, a classical benchmark in the field of aerospace engineering. Then we will explain the methodology on which the PCE is based. The output of interest in the considered problem is the lift coefficient C_l which comes from the non-dimensionalization of the lift force L. The latter is defined as the component of the fluid dynamic force (acting on the surface of the airfoil) in the direction perpendicular to the undisturbed flow direction. The interest is in approximating the C_l using both the ROM developed in section 2.6 and also the PCE method. As we mentioned, comparison of the results obtained by both techniques will be presented. Also, we will propose an idea of merging the two techniques which could reduce of the computational cost of the PCE.

4.1.1 The Physical Problem

The physical problem of interest is the two dimensional problem of the flow past an airfoil section with variable angle of attack and inflow velocity. In this problem, uncertainties arise from possible input parameters variations, where the input parameters consist of the inlet flow velocity and the angle of attack. The angle of attack in aerospace engineering is defined as the angle that lies between the flow velocity vector at infinite distance from the airfoil ($\boldsymbol{U}_\infty$) and the airfoil chord, see Figure 4.1. It is important from an engineering perspective to determine the angle at which the maximum value of the lifting force occurs (the lifting force is the forces component in the direction perpendicular to $\boldsymbol{U}_\infty$). After reaching that angle which is called the critical angle, the stall phenomenon takes place, where the lift forces acting on the surface of the airfoil suffer a typically sudden drop. Figure 4.2 shows lift coefficient, which is computed from the lift forces L as $C_l = \frac{L}{\frac{1}{2}\rho U^2 D}$, as a function of the airfoil angle of attack. The plot refers to the NACA $0012-64$ airfoil [3, 128], at a fixed Reynolds number of 10^6. The plot shows that the lift coefficient grows by increasing the angle of attack until the point when the flow separation occurs resulting in a loss of lift force. In the case of low Reynolds number flows, such as the ones considered in this section, this phenomenon occurs in mild fashion, unlike the more abrupt stalls manifested at higher velocities like the one in Figure 4.2. In such plot, the lift coefficient reaches its maximum value at an angle of 17 degrees before stall happens. The maximum lift coefficient is observed at higher values of the angle of attack in the case of laminar flows.

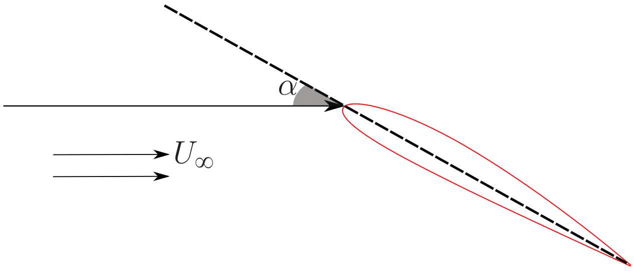


Figure 4.1: The angle of attack on an airfoil.

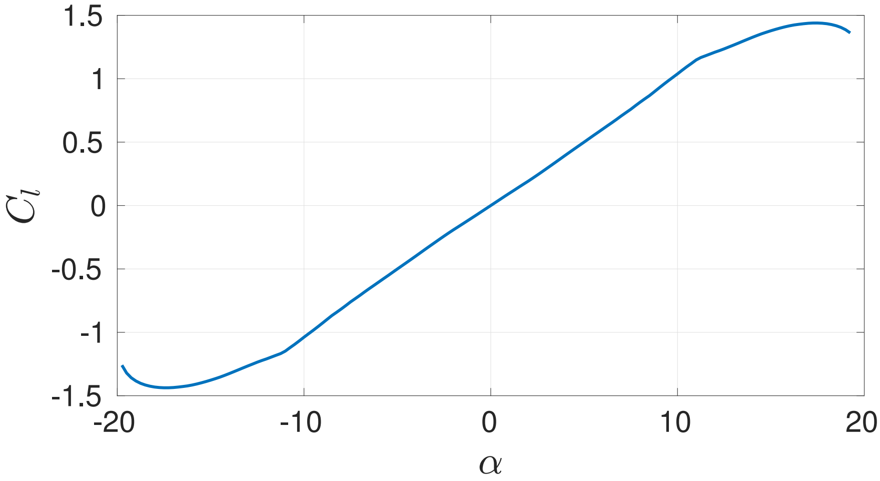


Figure 4.2: The lift coefficient curve for the airfoil NACA0012.

4.1.2 Non-Intrusive PCE

This subsection addresses the mathematical formulation of the PCE technique. In the context of Polynomial Chaos (PC) theory formulated by Wiener [155], real-valued multivariate Random Variables (RVs), such as the one considered in this work in the previous subsection (the lift coefficient C_l) can be decomposed into an infinite sum of separable deterministic coefficients and orthogonal polynomials [82]. Such polynomials are considered a group of stochastic terms which have dependency on mutually orthogonal Gaussian random variables. If we consider applying these polynomial to the output of interest (in this case the C_l), the decomposition assumption gives:

$$C_l{}^\star(\boldsymbol{\xi}) = \sum_{i=0}^{\infty} C_{li}\Psi_i(\boldsymbol{\xi}), \tag{4.1}$$

where the random variable $\boldsymbol{\xi} = (\alpha, U)$ expresses the uncertainty caused by the angle of attack and the inflow velocity. C_{li} is termed usually as the i^{th} stochastic mode of the expansion, while $\Psi_i(\boldsymbol{\xi})$ is the i^{th} polynomial. However, a truncation of the series above is usually done in practical applications, that is:

$$C_l{}^\star(\boldsymbol{\xi}) \approx \sum_{i=0}^{P} C_{li}\Psi_i(\boldsymbol{\xi}), \tag{4.2}$$

where here one may see that just the first $P + 1$ values are computed.

The orthogonal polynomials used in this work are called Hermite polynomials. These polynomials form a set of orthogonal basis functions in terms of Gaussian distribution [58]. In (4.2) $P + 1$ is the number of Hermite polynomials used in the expansion which depends on the order of the polynomials chosen and on the dimension n of the random variable vector $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_n\}$. In more details, the number P of Hermite polynomials of degree p in an n-dimensional space, is given by $P + 1 = \frac{(p+n)!}{p!n!}$ [58].

The main problem in the PCE approximation of RVs is the identification of the coefficients in the expansion, that is C_{li} in (4.1). This task can be done making use of different strategies, among them we mention the sampling based approaches and the quadrature method. In this work, we decided to use a sampling based approach which is based on the methodology proposed in [75]. The coefficients estimation begins by writing a discretized version of (4.2), namely

$$\begin{bmatrix} C^*_{l0} \\ C^*_{l1} \\ \vdots \\ C^*_{lN} \end{bmatrix} = \begin{bmatrix} \Psi_1(\boldsymbol{\xi}_0) & \Psi_2(\boldsymbol{\xi}_0) & \dots & \Psi_P(\boldsymbol{\xi}_0) \\ \Psi_1(\boldsymbol{\xi}_1) & \Psi_2(\boldsymbol{\xi}_1) & \dots & \Psi_P(\boldsymbol{\xi}_1) \\ \vdots & \ddots & & \\ \Psi_1(\boldsymbol{\xi}_N) & \Psi_2(\boldsymbol{\xi}_N) & \dots & \Psi_P(\boldsymbol{\xi}_N) \end{bmatrix} \begin{bmatrix} C_{l0} \\ C_{l1} \\ \vdots \\ C_{lP} \end{bmatrix},$$

where N is the number of the samples taken. In the case of $N = P + 1$, then the matrix in the system above becomes a square one which permits solving the linear system for obtaining the coefficients C_{li} given the known output coefficients C^*_{li}. However, what is practically common is that one may take a redundant number of samples and then the system can be solved in a least squares sense, as follows:

$$\boldsymbol{C_l} = (\boldsymbol{L}^T\boldsymbol{L})^{-1}\boldsymbol{L}^T\boldsymbol{C_l}^*, \tag{4.4}$$

where $\boldsymbol{L}$, $\boldsymbol{C_l}$ and $\boldsymbol{C_l}^*$ denote the rectangular matrix in (4.3), the PCE coefficients vector and output vector, respectively.

4.1.3 Application of PCE and the SUP-ROM

In this section, we are going to show the results of the SUP-ROM developed in section 2.6 in terms of the fluid dynamics fields and outputs of interest for the airfoil problem described in subsection 4.1.1. Then the results of PCE method on the main output of interest the lift coefficient are presented. Afterwards, we will make comparisons of the ROM and the PCE approaches for recovering the lift coefficient. Finally we will evaluate the performance of the UQ technique on the airfoil problem, both when FOM and ROM simulation results are used to feed the PCE algorithm. The tests carried out in this numerical example have two main goals. Firstly, we would like to make a comparison of the performance of the PCE and the ROM in approximating certain output of interest in such a CFD problem. The second goal is to study the potential merge of the two approaches for improving the accuracy and the efficiency.

4.1.3.1 The ROM Results

In this part we are going to test the SUP-ROM on the FOM results obtained for the airfoil problem described in subsection 4.1.1. We will start by describing the FOM settings. The computational domain is depicted in Figure 4.3, the FOM simulations were carried out using the simpleFoam solver in OpenFOAM, that is based on the SIMPLE algorithm which is designed for steady flows. The Gauss linear numerical scheme is employed for the approximation of gradients and Laplacian terms. We used a bounded Gauss upwind scheme for the convective term approximation. As for the linear solvers, a Preconditioned bi-conjugate gradient (PBiCG) solver is used for the velocity equations with Diagonal-based Incomplete LU (DILU) preconditioner, while the Geometric agglomerated Algebraic MultiGrid (GAMG) preconditioner with a GaussSeidel smoother is used for solving the pressure equation.

We recall that the parameters in this case are the airfoil angle of attack and the magnitude of the inflow velocity at the inlet. The training stage of the ROM demands an adequate number of FOM snapshots to be taken. Therefore, we decided to generate 520 snapshots which correspond to 520 different parameter samples. These samples are generated making use of the Latin Hyper Cube (LHC) [143] sampling algorithm. In order to generate the samples for the two parameters using the LHC sampling approach, one has to predefine the mean and the variance values of each parameter distribution. In all tests conducted in this numerical example, the mean values are 100 m/s and 0°, while the variances are equal to 20 $\mathrm{m}^2/\mathrm{s}^2$ and $300^{\circ 2}$ for velocity and angle of attack, respectively. The lift coefficient curve for this case is depicted in Figure 4.4. It can be deduced from this figure that the lift coefficient seems to be independent from the value of the inlet velocity, where different samples which have close values of the angle of attack but different inlet velocities, have resulted in equivalent value of C_l. Thus, we can conclude that input-output relationship can be considered as a curve in the C_l-α plane. We remark that this is an outcome of the fact that we consider the non-dimensional C_l as an output instead of the dimensional lift forces. In addition, the Reynolds number variation is not extremely wide which might have contributed to the last observation about the C_l-α curve.

After carrying out the full order simulations for the 520 samples, one may obtain the POD modes by applying the procedure explained in section 2.3. However, this problem features non-homogeneous boundary conditions at the Dirichlet boundary for the velocity field. Therefore, boundary treatment has to be taken into consideration. In the present test we have utilized the lifting function method in which the non-homogeneous boundary values in the velocity snapshots are transferred to one or more lifting velocity fields. In this case, we have two non-homogeneous boundary conditions to be set, i.e. $N_{BC} = 2$. The two conditions correspond to the x_1-component and the x_2-component of the velocity field at the inlet. The lifting functions $\boldsymbol{\phi}_{\boldsymbol{L}_{\boldsymbol{x}_1}}$ and $\boldsymbol{\phi}_{\boldsymbol{L}_{x_2}}$ are found by solving two linear potential flow problems with the boundary conditions at the inlet being $(1, 0)$ and $(0, 1)$, respectively, for $\boldsymbol{\phi}_{\boldsymbol{L}_{\boldsymbol{x}_1}}$ and $\boldsymbol{\phi}_{\boldsymbol{L}_{x_2}}$. At this point, the new velocity snapshots matrix with homogeneous values at the inlet can be computed. Then the POD velocity modes are obtained by applying the method of snapshots. Similar procedure is carried out for the pressure snapshots matrix giving the pressure POD modes. The cumulative eigenvalues of the correlation matrices built by the snapshots for the velocity and pressure are reported in Table 4.1. We list their values up to the fifteenth mode, and it can be seen that 5 modes allow for retaining 99.9% of the energy embedded in the system. After computing the pressure POD modes, one may solve the supremizer problem for each pressure mode resulting in the supremizer POD modes which will be used to enrich the velocity POD space [17]. The last step in the offline stage is to compute the reduced vectors, matrices and tensors which are needed to solve the system 2.51, and also for the computation of the reduced approximation of the forces (in Equation 2.65).

N Modes	$\boldsymbol{u}$	p
1	0.9033783	0.6497208
2	0.9978160	0.9649042
3	0.9985026	0.9926364
4	0.9990236	0.9986204
5	0.9993273	0.9996405
6	0.9995563	0.9998175
7	0.9996990	0.9999221
8	0.9998033	0.9999446
9	0.9998687	0.9999648
10	0.9999153	0.9999754
11	0.9999449	0.9999837
12	0.9999651	0.9999883
13	0.9999780	0.9999914
14	0.9999864	0.9999932
15	0.9999916	0.9999947

Table 4.1: Cumulative Eigenvalues of the correlation matrices for velocity and pressure.

At this point, all the offline computations are completed and it is possible to proceed to the online stage. The first test carried out is a reproduction test in the parameter space, where the same parameter samples used in the offline stage are also used in the online stage. Figure 4.5 depicts the results of the lift coefficient obtained by the SUP-ROM and compared to the FOM ones. These results are obtained using 10 modes for the reconstruction of velocity, pressure and supremizers. The figure shows that the ROM reconstruction of the C_l is not entirely accurate, as it can be noticed that the SUP-ROM C_l in the range $\alpha \in [-50, -30]$ is not closely reproducing its FOM counterpart. However, the approximation accuracy is improved significantly by adding more modes to the velocity POD space. Figure 4.6 shows the lift coefficient obtained by the SUP-ROM with 15 modes for the velocity and 10 for each of the pressure and supremizers. The ROM C_l in the last figure matches the FOM one to a higher degree. To measure the accuracy of the ROM reconstruction of the lift coefficients, we introduce the following L^2 relative error:

$$\varepsilon = 100 \frac{\sqrt{\sum_{t=1}^{n} (C_{l_t}^{FOM} - C_{l_t}^{ROM})^2}}{\sqrt{\sum_{t=1}^{n} (C_{l_t}^{FOM})^2}} \%, \tag{4.5}$$

where n is the number of sampling points, $C_{l_t}^{FOM}$ and $C_{l_t}^{ROM}$ are the $t-$th sample point of the

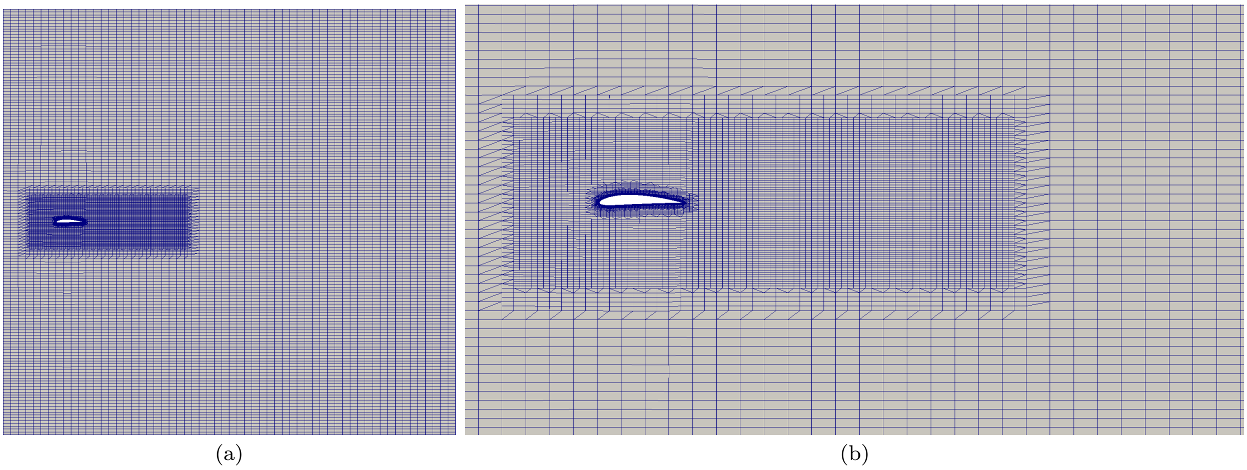


Figure 4.3: (a) The OpenFOAM mesh used in the simulations. (b) A picture of the mesh zoomed near the airfoil.

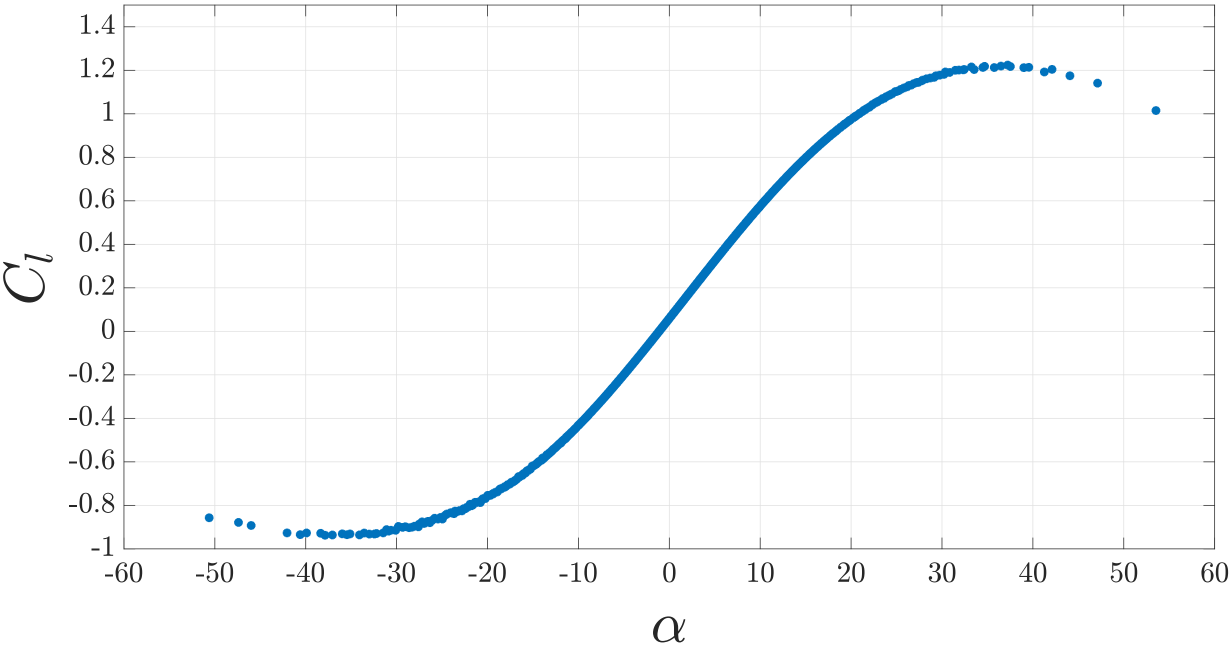


Figure 4.4: The FOM lift coefficient for the samples taken for the training phase.

lift coefficients for the FOM and the ROM, respectively. The values of ε for the SUP-ROM C_l in Figure 4.5 and Figure 4.6 are 3.5301 % and 1.5644 %, respectively.

We would like to mention that the settings of the numerical test carried out here is identical to the one presented in [73]. In such work, based on the same problem with identical parametric setting, it is claimed (in the section named *ROM results*) that building an accurate ROM requires aggregating samples from several Gaussian distributions which have different mean values of the angle of attack. In such way, it is possible to locate enough samples in the extreme angles of attack region, which is where the stall occurs. However, we can say that the aforementioned claim is only partially correct. In fact, further improvements in the lifting function technique used to deal with inflow BCs, resulted in a reduced model able to obtain low errors in the stall regions without the need of samples grouped in several Gaussian distributions centered in different parts of the C_l-α curve.

After having conducted a reproduction test in the parameter space, we increase the complexity

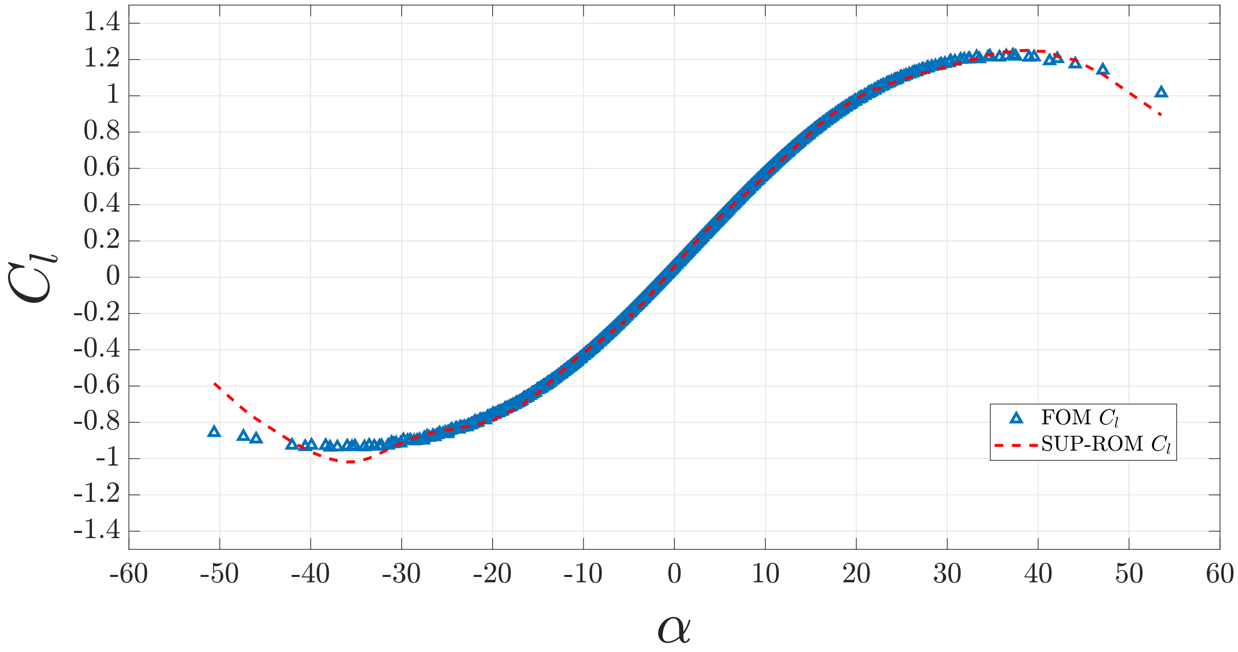


Figure 4.5: The first sampling case which is used for the training of the SUP-ROM : the full order lift coefficients curve versus the ROM reconstructed one with 10 modes used for each of velocity, pressure and supremizer fields.

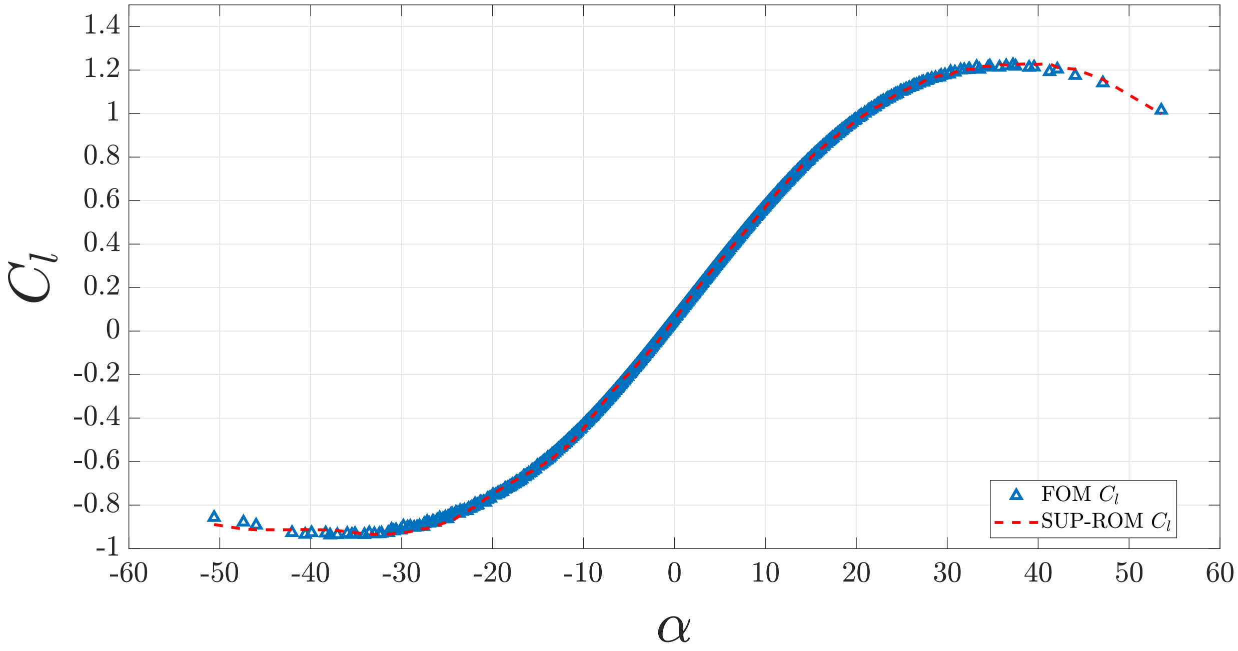


Figure 4.6: The first sampling case which is used for the training of the SUP-ROM : the full order lift coefficients curve versus the ROM reconstructed one with 15, 10 and 10 modes are used for velocity, pressure and supremizer fields respectively.

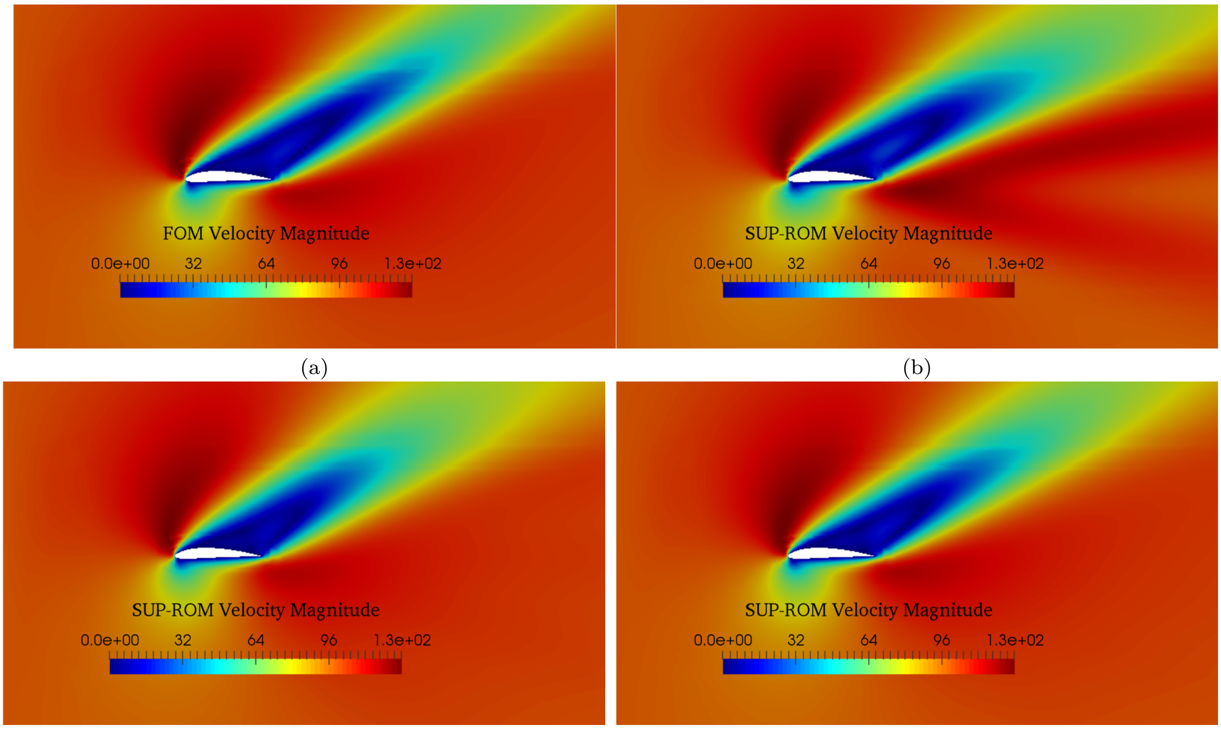


Figure 4.7: The full order velocity field for the parameter $\boldsymbol{\mu}^* = (98.8548$ m/s , 35.3141°) and a comparison with the reconstructed field by means of different number of modes for velocity, pressure and supremizer fields. (a) FOM field (b) ROM velocity field with 5 modes used for all variables. (c) ROM velocity field with 8 modes used for all variables. (d) ROM velocity field with 15, 10 and 10 modes used for velocity, pressure and supremizers, respectively.

by considering a general cross validation test in the parameter space. Therefore, we generated a new set of samples which contains 1000 sample points for the parameters space plot. This set is also obtained by making use of LHC with the same prescribed values of the means and the variances as in the first test.

Firstly, we present the results of the ROM fields in Figure 4.7. The plot in part (a) portrays contours of the FOM velocity field obtained for the parameter value $\boldsymbol{\mu}^* = (98.8548$ m/s , 35.3141°). The similar diagrams in (b), (c) and (d) depict different reconstructed surrogate fields obtained employing different number of modes for the same parameter value.

The lift coefficient curve for the 1000 new samples is depicted in Figure 4.8. The results of the SUP-ROM C_l approximation are shown in Figure 4.9, where the number of modes used in the online stage are 10 for each field in (a) and 15 modes for the velocity and 10 modes for each of the pressure and the supremizers in (b). The error values for the C_l in the last figures are 3.7659 % and 1.6079 % for (a) and (b), respectively. This last test has shown that the SUP-ROM is accurate also in the approximation of the flow fields and other outputs of interest for parameter values which were not used in the training stage.

4.1.3.2 The PCE Results

The aim of the present section is to evaluate the performance of the PCE algorithm implemented for the fluid dynamic problem at hand. To better describe the amount of simulations

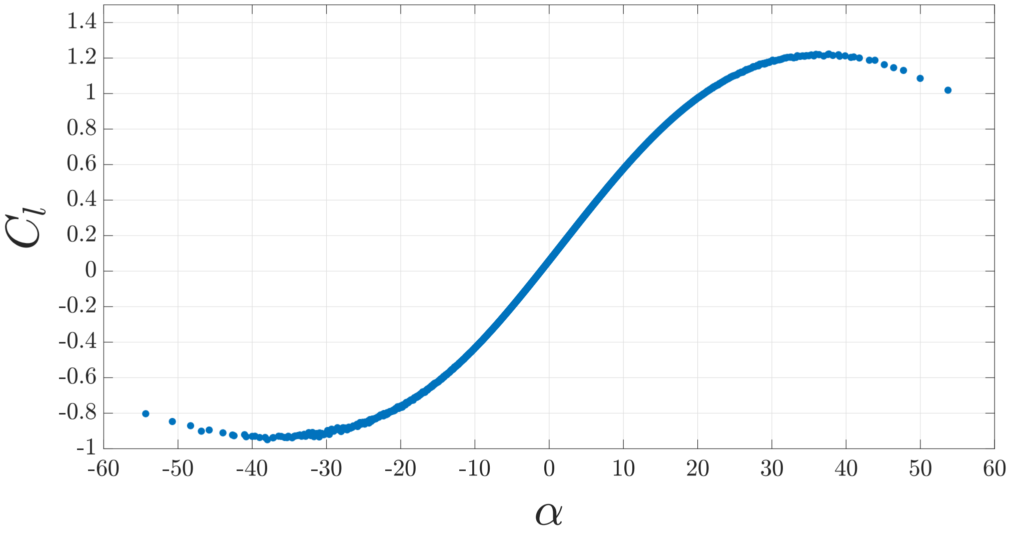
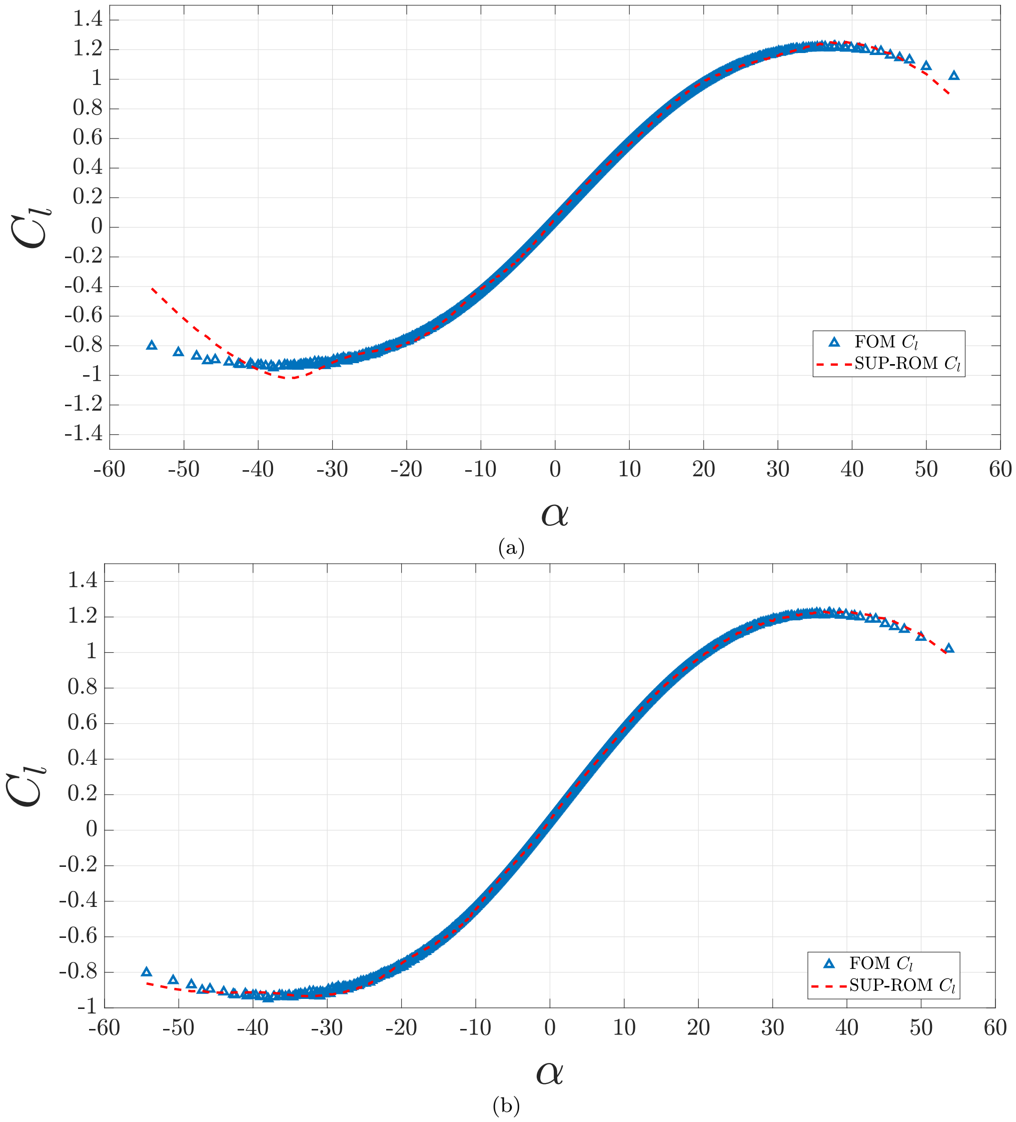


Figure 4.8: The FOM lift coefficient as a function of the angle of attack α for the second sampling group which is used for the cross validation test of the SUP-ROM.



(a)

(b)

Figure 4.9: A comparison between the FOM and the SUP-ROM reconstructed lift coefficients for the cross validation test (a) 10 modes are used for each of velocity, pressure and supremizer fields. (b) 15, 10 and 10 modes are used for velocity, pressure and supremizer fields, respectively.

carried out to both train and validate the UQ PCE model implemented, we present in Figure 4.10 a conceptual scheme of the simulation campaign carried out in this work.

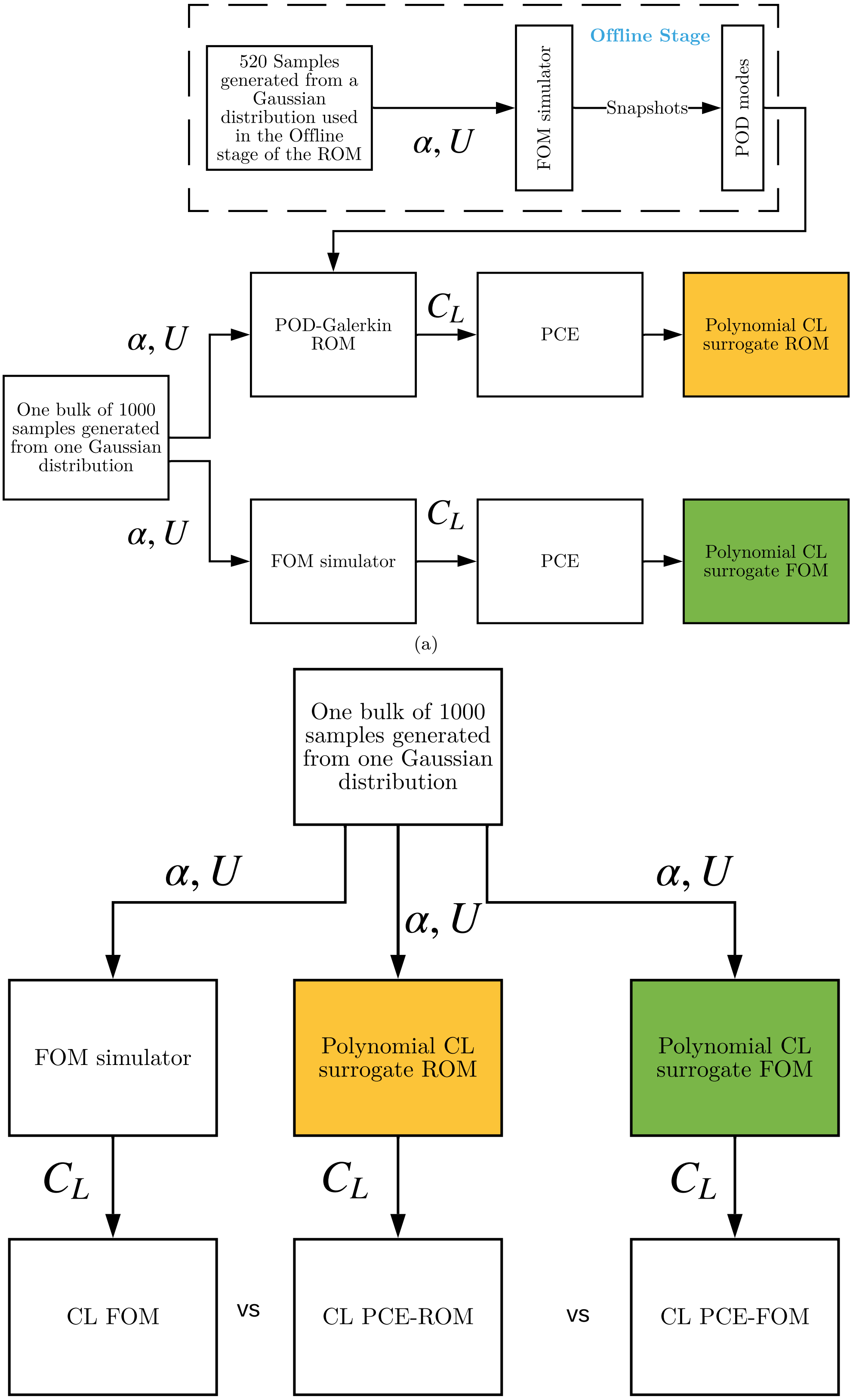
One of the main features of non intrusive PCE is that it can use any deterministic simulation software as a black box input source. We will then present different tests in which PCE has been fed with the output of fluid dynamic simulations based on models characterized by different fidelity levels. In a first test we have in fact generated a PCE based on the FOM, and evaluated its performance in a prediction test. The second test consisted in generating a PCE based on the SUP-ROM used in the previous section. The latter test allows for an evaluation of how the PCE results are affected when the expansion is based on a surrogate ROM model rather than the FOM one. Given the relatively high number of samples required for the PCE setup, it is in fact interesting to understand if the ROM can be used to reduce the computational cost associated with the evaluation of the solution output for a new sample, without a significant loss in terms of accuracy.

One of the main assumptions of the non intrusive PCE algorithm implemented is that of operating on Gaussian distributed input parameters. In fact, we have decided to apply PCE on the same sampling set that was used in the previous section for the cross validation test of the SUP-ROM. We recall that it contains 1000 samples which are generated by the LHC technique with the means of the inlet velocity and angle of attack being 100 m/s and 0°, while the variances are equal to 20 m^2/s^2 and $300^{\circ 2}$.

In the PCE methodology, one has to start by training the PCE (for determining the PCE coefficients). For this purpose, we decided to divide the 1000 samples block into two groups, each of which contains 500 samples. This is done by first ordering the samples according to the angle of attack, and then taking one sample for the training procedure and leaving the next one for the checking phase. In the first test, we use the FOM C_l values to feed the PCE algorithm. By solving system 4.3 in a least square sense, one may obtain the PCE reconstructed lift values for the 500 checking samples. In Figure 4.11, one may see the resulted PCE lift coefficient curve for the values of the angle of attack used as check samples. The last figure shows also the corresponding FOM C_l values. The error in this case is 2.4208 %. It can be deduced that the PCE has given satisfactory results when it was applied directly on the FOM results.

In the second test we replace the FOM results with the SUP-ROM data as input for PCE. Thus, the SUP-ROM C_l values were computed by running the SUP-ROM solver with 15 modes for the velocity and 10 modes for pressure and supremizers. After using the same 500 samples to compute the PCE coefficients, we used the PCE to predict the lift coefficients for the remaining 500 samples used for the check. We then compared the value of the predicted PCE coefficients in this case to both the SUP-ROM values and the FOM values.

The results of the aforementioned test are reported in Figure 4.12. The figure includes comparison of the PCE predicted C_l curve with both its SUP-ROM and FOM counterparts. The plots show a similar behavior of the PCE predictions obtained using the SUP-ROM and the FOM output data. By a quantitative standpoint, the PCE predictions present a 2.6522 %



Offline Stage
520 Samples generated from a Gaussian distribution used in the Offline stage of the ROM
α, U
FOM simulator
Snapshots
POD modes
α, U
POD-Galerkin ROM
C_L
PCE
Polynomial CL surrogate ROM
One bulk of 1000 samples generated from one Gaussian distribution
α, U
FOM simulator
C_L
PCE
Polynomial CL surrogate FOM
One bulk of 1000 samples generated from one Gaussian distribution
α, U
α, U
α, U
FOM simulator
Polynomial CL surrogate ROM
Polynomial CL surrogate FOM
C_L
C_L
C_L
CL FOM
vs
CL PCE-ROM
vs
CL PCE-FOM

(a)

(b)

Figure 4.10: The flowcharts describing the procedure followed in the numerical simulations for the UQ model generation and validation campaign respectively. The top scheme focuses on the procedure adopted for the generation of the UQ model, and in particular on the identification of the PCE coefficients. The polynomial surrogate based on the full order model (indicated in green) has been generated using 1000 Gaussian distributed samples in the α, U space. The same samples have been used to obtain the polynomial surrogate input-output relationship for the POD-Galerkin ROM (denoted by the yellow box). Note that the ROM used in this simulation campaign has been trained by means of 520 samples in the α, U space. Finally, the bottom flowchart illustrates the PCE validation campaign. Here, the same last 1000 samples have been used to obtain the corresponding output with the full order model, with the polynomial UQ surrogate trained with the FOM simulations (green box), and with the polynomial UQ surrogate trained with the ROM simulations (yellow box). For both polynomial UQ surrogates, the set of 1000 samples has been divided into two sets of 500 samples each, one set has been used for the identification of the PCE coefficients, while the second is used for the validation phase.

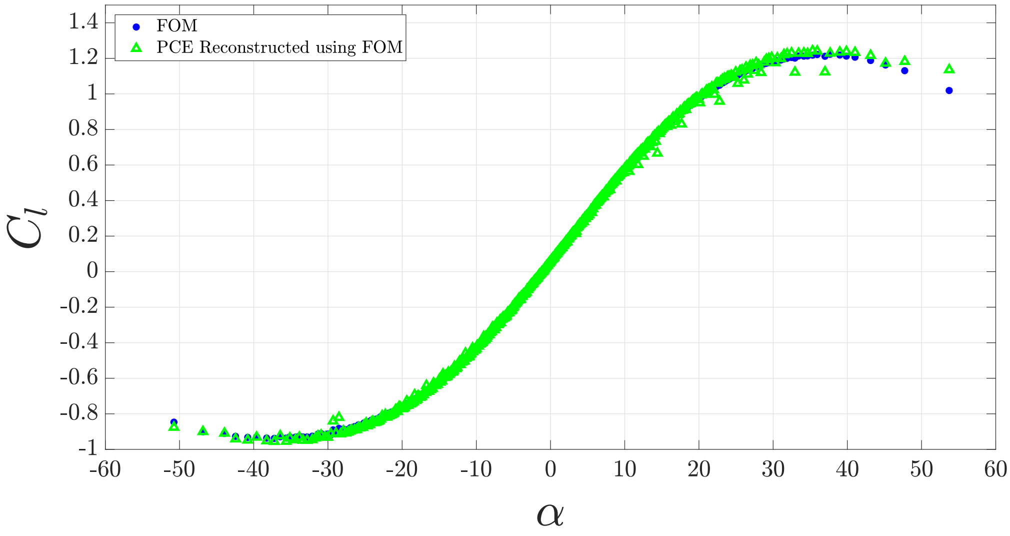


Figure 4.11: The PCE lift coefficient for the checking samples versus that computed by the FOM, the checking samples are 500, the PCE polynomials are of the fourth degree.

error with respect to the SUP-ROM predictions, while the L^2 norm of the error with respect to the FOM predictions is 2.8530 %. A summary of the comparisons made is reported in Table 4.2.

First data	Second data	Error
FOM	SUP-ROM	1.6079 %
FOM	PCE on FOM	2.4208 %
SUP-ROM	PCE on SUP-ROM	2.6522 %
FOM	PCE on SUP-ROM	2.8530 %

Table 4.2: A comparison between the relative error in L^2 norm for the results obtained from the SUP-ROM and the PCE, with the PCE being used on both the FOM and the SUP-ROM results. We remark that the number of POD modes used (if apply) are $15, 10$ and 10 for velocity, pressure and supremizer fields, respectively, for all cases. We underline also that 500 samples have been used for testing the PCE wherever it is used.

If we shift the attention to the computational side of the results addressed in the current and previous sections, we observe that the SUP-ROM has performed positively in that regard. In more details, the wall time consumed by the FOM in running the steady simulations for the 1000 parameter samples used in the cross validation test was 10185.5 s. On the other hand, the SUP-ROM with 15 modes of velocity and 10 modes for each of the pressure and the supremizers, has computed the lift and drag forces and also exported the reduced solutions in just 61.3889 s, which means that the SUP-ROM has a speed up of 165. It is worth remarking that the time taken by the FOM in simulating the 520 offline parameter samples is 5076.38 s. It can be, therefore, concluded that the SUP-ROM can be used as an accelerator for the PCE method, whenever the 0.43 % error increment associated with it are considered acceptable.

To summarize, in this numerical example, we studied two popular techniques that are used often in the fields of ROM and UQ which are the POD and PCE, respectively. The study aimed at comparing the accuracy of the two techniques in reconstructing the outputs of interest of viscous fluid dynamic simulations. We have concluded the work with combining the two approaches so as to exploit the ROM to speed up the many query problem needed to obtain the PCE coefficients. POD-Galerkin ROMs such as the SUP-ROM can be a reliable

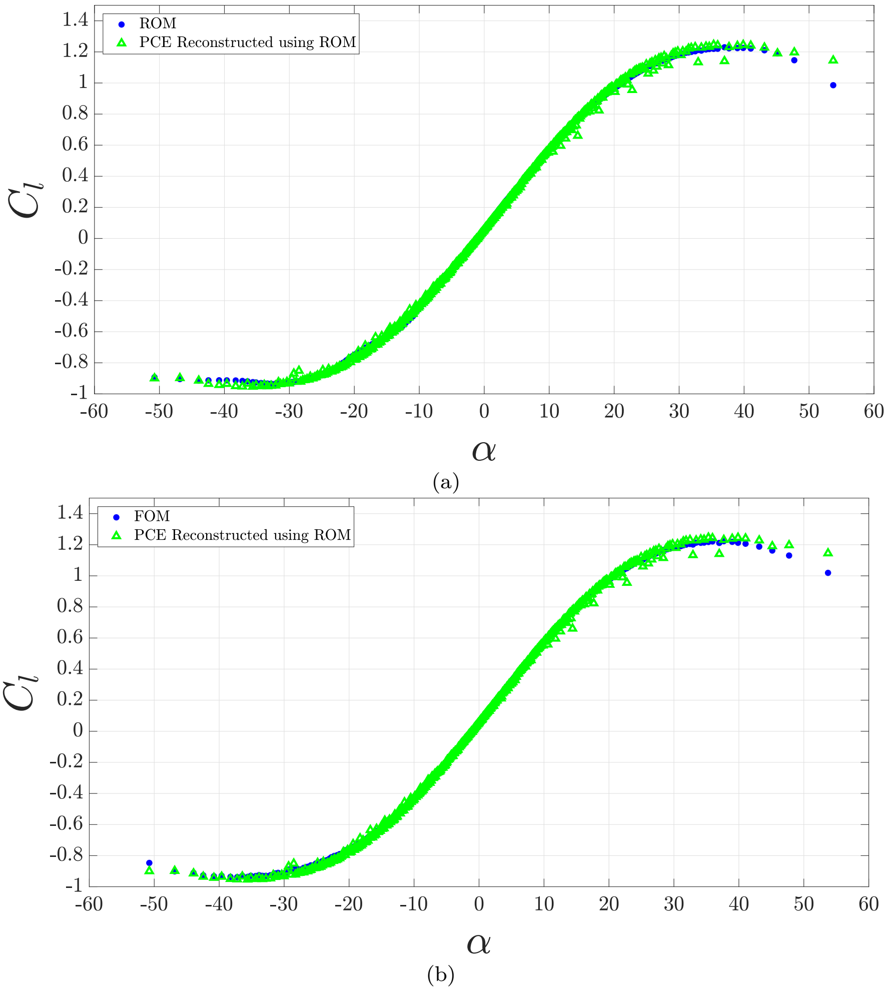


Figure 4.12: (a) The SUP-ROM lift coefficient versus the PCE lift coefficient curve when the PCE has been applied on the SUP-ROM output with 15, 10 and 10 modes used for velocity, pressure and supremizer fields respectively. (b) The FOM lift coefficient versus the PCE lift coefficient curve when PCE has been applied on the SUP-ROM output with same number of modes as in (a). In both graphs, the PCE coefficients were obtained with polynomial of the fourth grade.

output evaluator for the PCE, as the value of relative error PCE had when it was based on ROM results was 2.8530 % while the error was 2.4208 % when PCE was based on FOM outputs. The last result speaks positively for POD-Galerkin ROMs and makes them a valid tool to be possibly used in the field of uncertainty quantification.

4.2 A Steady Turbulent Case: Reynolds Parametrized Backward Step Case

In this section we are going to evaluate the performance of the hybrid ROM developed in section 3.2 on a benchmark test case in a steady setting. The problem considered is the backward step case. The computational domain is depicted in Figure 4.13. The boundary conditions for the velocity and the pressure fields are also indicated in the figure, which also reports the domain size and more specifically the lengths in Figure 4.13 are reported in terms of the characteristic length is $D = 1$ m, the physical viscosity is $\nu = 10^{-3}$ m^2/s. The inlet

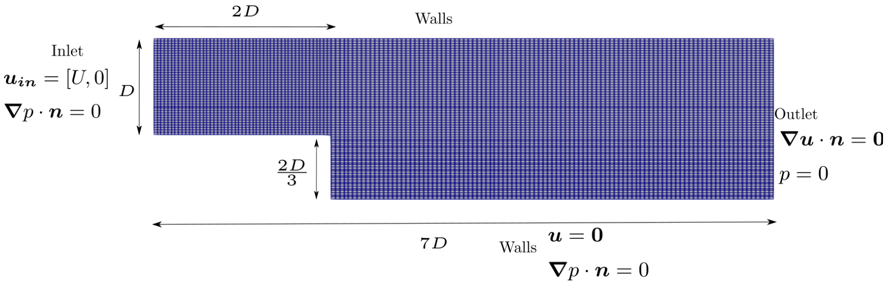

$2D$
Walls
Inlet
$\boldsymbol{u_{in}} = [U, 0]$
$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$
D
Outlet
$\boldsymbol{\nabla u} \cdot \boldsymbol{n} = \boldsymbol{0}$
$p = 0$
$\frac{2D}{3}$
$7D$
Walls
$\boldsymbol{u} = \boldsymbol{0}$
$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$

Figure 4.13: 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.

velocity U is varied between from 1 m/s to 25 m/s, this gives values of the Reynolds number which lie in the range $[1 \times 10^3, 2.5 \times 10^4]$.

.

The parameter considered in this problem is the velocity at the inlet U, or in other words the Reynolds number. The goal of this test is to investigate the performance of the hybrid ROM (the H-SUP-ROM) at high Reynolds number values. Therefore, we aim at reproducing the fluid dynamics fields using the H-SUP-ROM for several values of the parameter. In this test, we are also interested in assessing the H-SUP-ROM ability of reducing steady problems with the offline snapshots generated by different FOM turbulence closure models. To this end, we have considered FOM simulations carried out with both the $k - \varepsilon$ and the SST $k - \omega$ models. In this numerical example, we are going to compare the results obtained by the H-SUP-ROM to the FOM ones and also to the uniform ROM or the U-ROM addressed in section 3.3.

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 linear solver used for the velocity equation uses a symmetric Gauss Seidel smoother, while the pressure solver is based on the GAMG with GaussSeidel smoother.

100 snapshots were generated in the offline stage by running the FOM which utilizes the SIMPLE algorithm addressed in section 2.2. This has been done for the two FOMs based on both turbulence models considered. The 100 snapshots correspond to values of the inlet velocity which are equally distributed in the range mentioned above, i.e. $[1, 25]$ m/s.

In the present case, the non-homogeneous inlet boundary conditions are dealt with by the help of the penalty method. We remark that the ROM results were sensitive to the value of the penalization factor τ. Therefore, we have carried out a sensitivity analysis in order to determine the appropriate value of τ for both $k - \varepsilon$ and SST $k - \omega$ turbulence models. As explained in section 2.7, the penalty method allows for the application of the POD procedure directly on the original snapshots. After applying the SVD on the snapshots of the velocity, the pressure and the eddy viscosity fields, the POD modes are obtained. The cumulative eigenvalues decay for the three fluid dynamics variables is presented in Figure 4.14. The plot

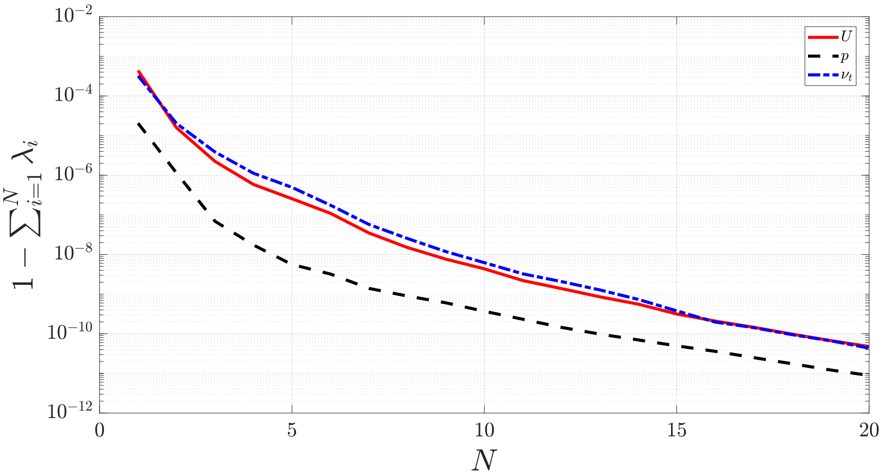


Figure 4.14: Cumulative ignored eigenvalues decay. In the plot, the solid red line refers to the velocity eigenvalues, the dashed black line indicates the pressure eigenvalues and the dash-dotted blue line finally refers to the eddy viscosity eigenvalues.

suggests that a small number of modes is already sufficient in order to retain the energetic information in the snapshots.

After obtaining the POD modes, the reduced vectors, matrices and tensors are also computed and stored. In order to evaluate the accuracy of the two ROMs considered, one has to consider taking a cross validation test for the parameter considered in this work. In practice, this consists in testing the ROM in the online stage with parameter values which were not present in the offline parameter training set. Therefore, a different set of 80 online parameters U_i^* (where $i = 1, ..., N_{test} = 80$) was generated. The online parameters are taken as equally distributed points in the range $[3, 20]$ m/s. This online sample set has values of the parameter which lie almost mid way between two offline parameter samples, and also it contains values which are very close to the offline parameter samples. It has to be recalled that the offline stage contains the training of the interpolation using the RBF for the approximation of the eddy viscosity reduced vector $\boldsymbol{g}$ in the H-SUP-ROM dynamical system in 3.24. The RBF interpolation strategy is simply based on approximating the maps between the offline parameter values U and the L^2 projection coefficients of the eddy viscosity modes onto the snapshots. This strategy is basically the one addressed in subsection 3.2.1. The interpolation using the RBF in this work has been carried out using the C++ library SPLINTER [61].

The online stage needs that a value of the reduced eddy viscosity vector $\boldsymbol{g}$ be available in order to solve the reduced system of the H-SUP-ROM. This value is obtained by the employing the RBF interpolant constructed in the offline stage. The RBF interpolant of each eddy viscosity mode will give an approximated value of its corresponding coefficient in the vector $\boldsymbol{g}$ as indicated in Equation 3.34. After having approximated $\boldsymbol{g}$ one may solve the algebraic system to obtain at the end the reduced velocity and pressure vector $\boldsymbol{a}$ and $\boldsymbol{b}$. Finally, the H-SUP-ROM fields for velocity, pressure and the eddy viscosity could be recovered. In the case of the U-ROM, the procedure is similar, where the dynamical system to be solved is the one defined by 3.57. Obviously, in the case of the U-ROM, the number of reduced unknowns is smaller than its counterpart in the H-SUP-ROM. In addition, the U-ROM formulation is purely intrusive, therefore, no interpolation procedure is needed.

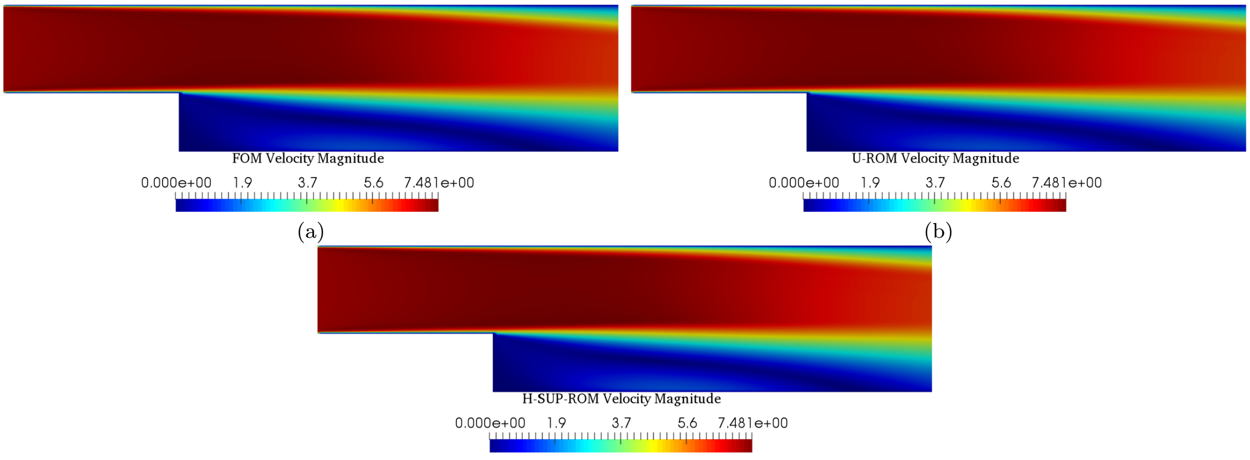


Figure 4.15: $k-\varepsilon$ turbulence model case, velocity fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM velocity, while in (b) one can see the U-ROM velocity, and finally in (c) we have the H-SUP-ROM velocity.

The first turbulence model considered is the $k-\varepsilon$ model. The FOM field corresponding to all online parameter values were computed for the sake of comparing the ROMs results to them. The reduced ROM fields obtained by both the U-ROM and the H-SUP-ROM for the online parameter value of $U^* = 7.0886$ m/s are compared to their FOM counterparts. The figures of the velocity, pressure and the eddy viscosity fields are presented in Figure 4.15, Figure 4.16 and Figure 4.17, respectively. The plots suggest that the reduced approximation of the FOM velocity field is acceptable from a qualitative viewpoint. The same can not be said for the pressure field. In order to have a clearer picture, we decided to introduce the following L^2 relative error defined as:

$$\varepsilon_u = \frac{\|\boldsymbol{u}-\boldsymbol{u}^*\|_{L^2(\Omega)}}{\|\boldsymbol{u}\|_{L^2(\Omega)}} \times 100\%, \varepsilon_p = \frac{\|p-p^*\|_{L^2(\Omega)}}{\|p\|_{L^2(\Omega)}} \times 100\%, \tag{4.6}$$

where $\boldsymbol{u}^*$ and p^* are general reduced order velocity and pressure fields, respectively. The values of the velocity relative error ε_u for the H-SUP-ROM and the U-ROM are 0.4444 % and 0.6522 %, respectively. On the other hand, the pressure errors ε_p are 0.3654 % and 20.9441 % for the H-SUP-ROM and the U-ROM, respectively. The last values indicate clearly that the H-SUP-ROM was accurate in reproducing the FOM pressure field, while the U-ROM has failed in giving satisfactory results in that regard.

At this point, we move to the next test which involves the use of the SST $k-\omega$ model turbulence model. The FOM simulation were run for the same offline and online parameter sets with equivalent FOM and ROM settings for the numerical schemes and the treatment of boundary conditions. We recall that the objective of this test is to assess how responsive the H-SUP-ROM and the U-ROM results are with respect to the turbulence model used for the FOM simulations. In Figure 4.18, Figure 4.19 and Figure 4.20, one may see the velocity, the pressure and the eddy viscosity fields, respectively, for the FOM, the U-ROM and the H-SUP-ROM for the online parameter value $U^* = 7.0886$ m/s. As in the previous case, it may be observed that the H-SUP-ROM was successful in recovering the fluid dynamics field from qualitative standpoint, while the same can not be said for the U-ROM. Considering the error values for this case, one may deduce that the pressure reconstruction by the U-ROM is

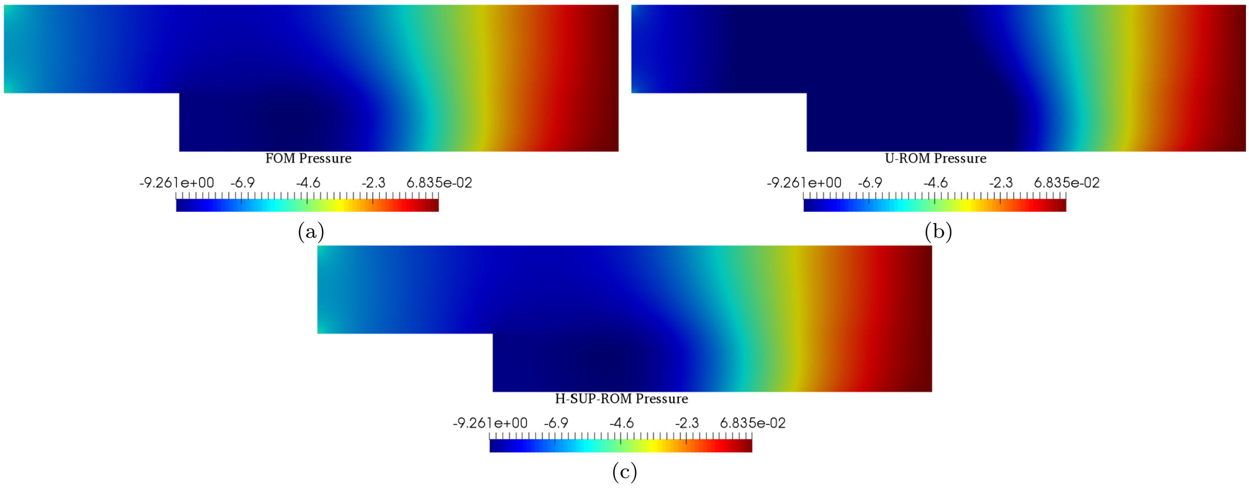


Figure 4.16: $k-\varepsilon$ turbulence model case, pressure fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM pressure, while in (b) one can see the U-ROM pressure, and finally in (c) we have the H-SUP-ROM pressure.

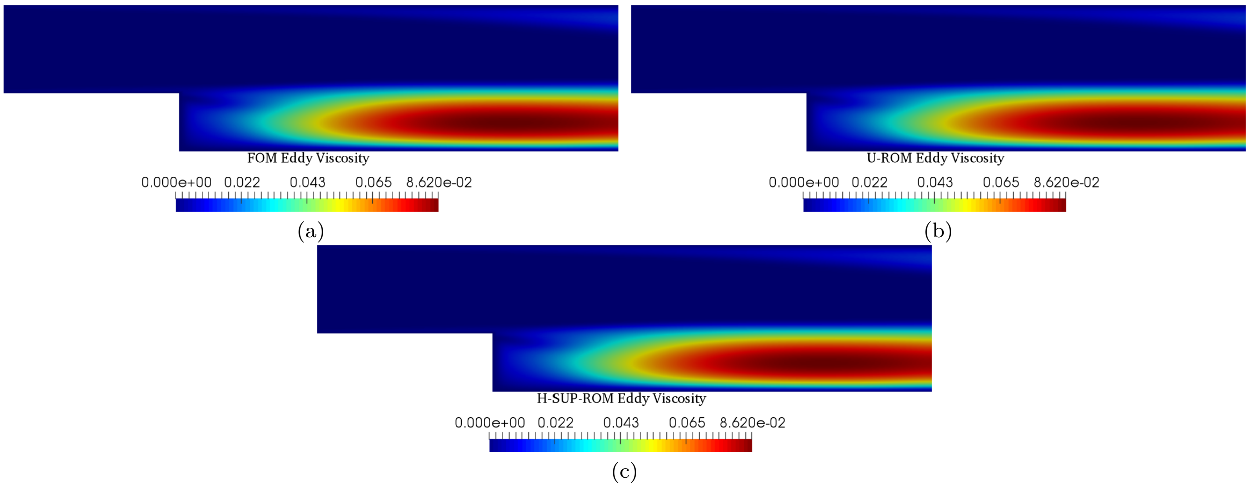
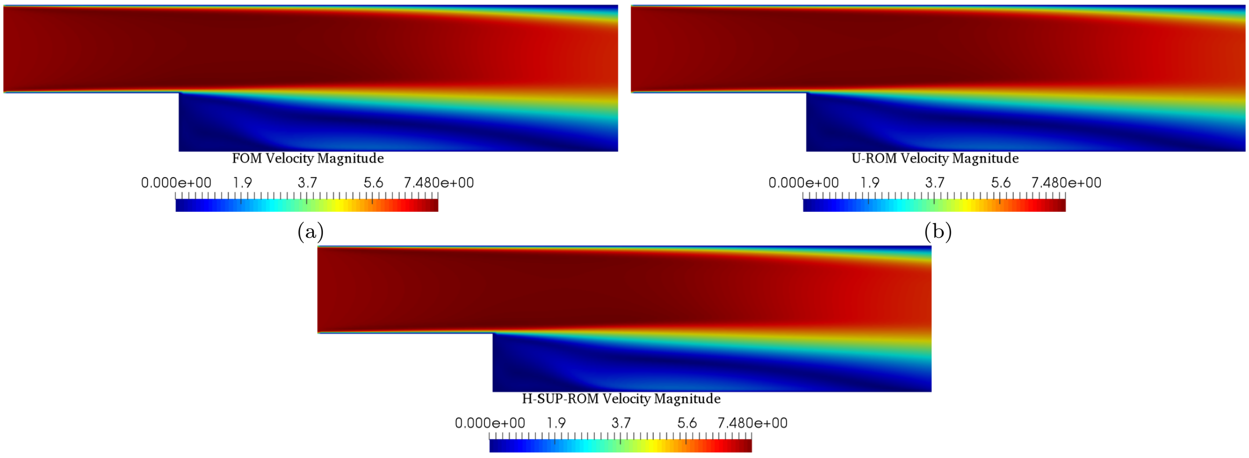


Figure 4.17: $k-\varepsilon$ turbulence model case, eddy viscosity fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM eddy viscosity, while in (b) one can see the U-ROM eddy viscosity, and finally in (c) we have the H-SUP-ROM eddy viscosity.

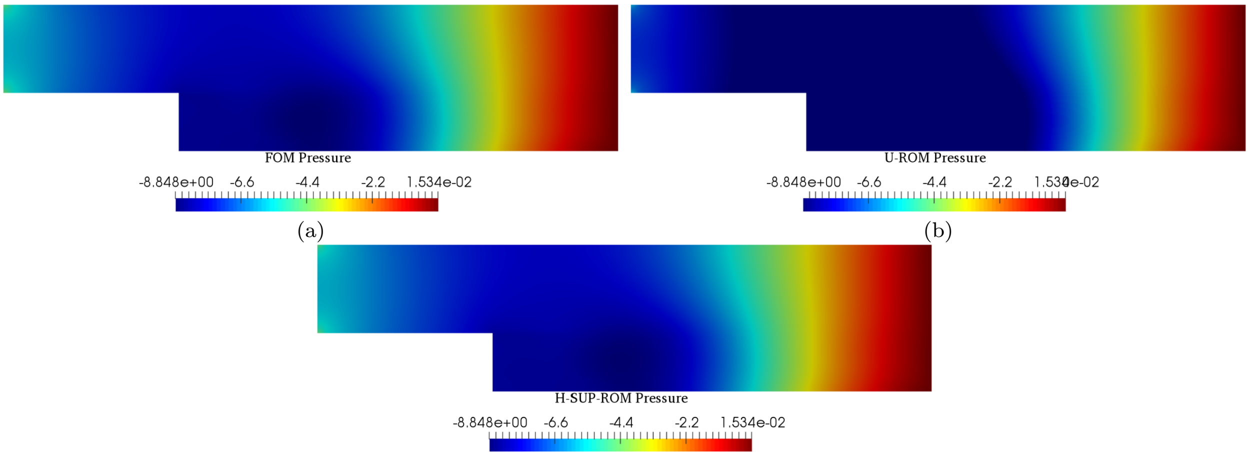
quite poor, as the error value ε_p is 22.3972 % for such intrusive model. On the contrary, the H-SUP-ROM has achieved a higher level of accuracy with $\varepsilon_p = 0.7329$ %. As for the velocity field errors, the U-ROM led to an error of $\varepsilon_u = 0.8177$ %, while the H-SUP-ROM gives a value of $\varepsilon_u = 0.8088$ %.

The reader may observe that the FOM fields obtained by different turbulence closure models have different values across the domain (except for the velocity). For the sake of having a clear picture of how accurate the reduction carried out by the hybrid ROM regardless of which turbulence modes was used at the full level, one may consider plotting the FOM and the H-SUP-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 result of this additional check can be seen in Figure 4.21, here this figure relates the FOM and the H-SUP-ROM pressure values for the distance of the points in space from the inlet when the second spatial variable x_2 is fixed at $\frac{5D}{6}$ which is half the height of the domain. The last figure shows the pressure distributions obtained by both the $k-\varepsilon$ and the SST $k-\omega$ models. Figure 4.21 proves that the H-SUP-ROM is able to capture the flow solution differences between the two different turbulence models. The latter results accomplishes one



(a) (b) (c)

Figure 4.18: 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 U-ROM velocity, and finally in (c) we have the H-SUP-ROM velocity.

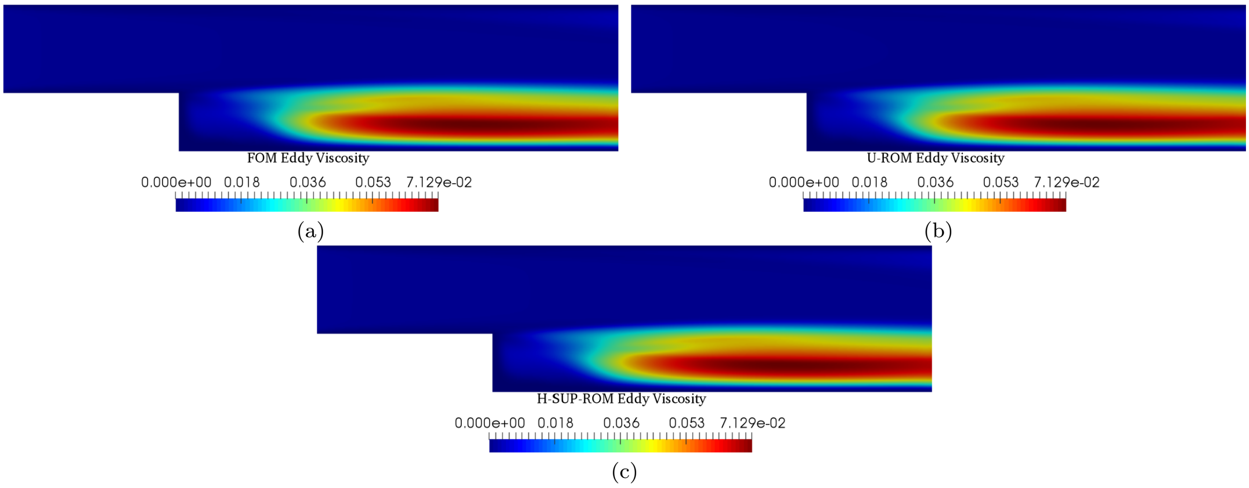


(a) (b) (c)

Figure 4.19: 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 U-ROM pressure, and finally in (c) we have the H-SUP-ROM pressure.

of the main goals of the hybrid ROM proposed in this work.

The final result in this section is for the H-SUP-ROM convergence analysis which is shown Figure 4.22. These figures 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 H-SUP-ROM results exhibit fast convergence to the FOM solution for both $k-\varepsilon$ 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 grows, the gain in accuracy becomes only marginal compared to the ν_t field interpolation error.



(a)

(b)

(c)

Figure 4.20: SST $k-\omega$ turbulence model case, eddy viscosity fields for the value of the parameter $U = 7.0886$ m/s: (a) shows the FOM eddy viscosity, while in (b) one can see the U-ROM eddy viscosity, and finally in (c) we have the H-SUP-ROM eddy viscosity.

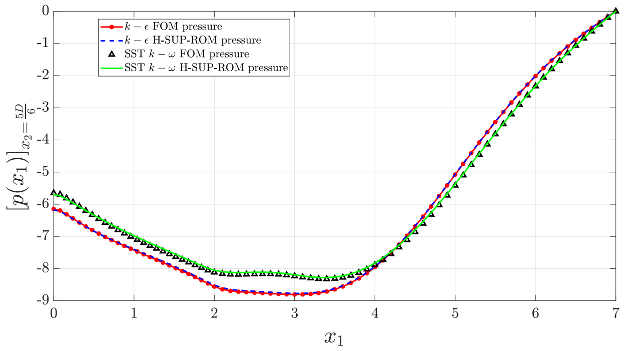
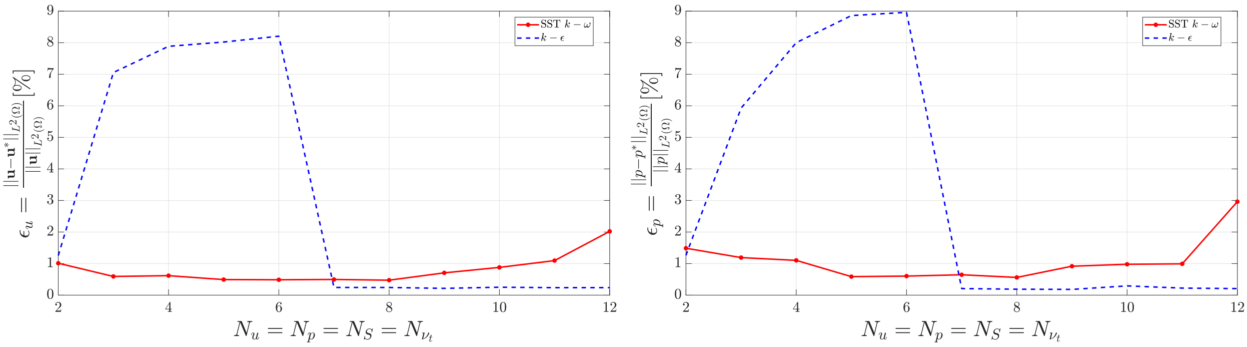


Figure 4.21: The pressure fields obtained using both $k-\varepsilon$ and SST $k-\omega$ turbulence models and the H-SUP-ROM ones. The plot is for the pressure value along the x_1 direction keeping the value of x_2 fixed at half the maximum height.



(a)

(b)

Figure 4.22: 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 H-SUP-ROM models obtained with two different turbulence models at the full order level which are $k-\varepsilon$ 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 .

4.3 An Unsteady Turbulent Case: Reynolds Parametrized Flow Past a Circular Cylinder

In this subsection, the reduction methodologies are tested on a turbulent case. The problem under study is the classical CFD benchmark of the unsteady flow past a circular cylinder. We refer the reader to [162, 163, 129] for a deep analysis of this problem. This problem has been used for testing reduced order methods in numerous works in the literature, see for example [137, 106, 160, 34, 65]. As the vortex shedding past circular cylinders is an inherently two-dimensional phenomenon, she case is studied in two dimensions. The computational grid and the flow domain are illustrated in Figure 4.23. The figure also shows the boundary conditions set for the velocity and the pressure fields, in which all the distances are expressed in terms of the characteristic length of the problem the diameter of the cylinder $D = 1$ m. The physical kinematic viscosity ν is equal to 10^{-4} m^2/s and the velocity at the inlet is horizontal and uniform with magnitude U_{in}. The number of cells in the mesh is equal to 11644. The case is parametrized through the Reynolds number Re by changing the velocity at the inlet U_{in}. The latter varies inside the range of $[7.5, 12]$ m/s which corresponds to $Re \ \in [7.5 \times 10^4, 1.2 \times 10^5]$. The simulations will run and evolve in time until a final periodic regime is fully developed.

The simulations are carried out using the non-steady solver named pimpleFoam which is based on merging the SIMPLE and the PISO algorithms described in section 2.2. The pimpleFoam solver in OpenFOAM has the capacity of adapting the time steps in a way which assures that the maximum Courant number CFL [45, 46] does not exceed a prescribed value which is in this case has been set to $\mathrm{CFL}_{\mathrm{max}} = 0.9$. In relation to the time advancing schemes used in this case, the backward Euler scheme is utilized for the computation of the time derivative of the velocity field. As for the spatial gradients, a Gauss linear scheme is employed. The convective term has been approximated with a 2nd order bounded Gauss upwind divergence scheme which utilizes upwind interpolation weights, with an explicit correction based on the local cell gradient. Gauss linear scheme is used for the discretization of the diffusive term. The values of the relaxation factors α_u and α_p are fixed at 0.7 and 0.3, respectively. Only one non-orthogonal corrector iteration is used for dealing with the non-orthogonality of the mesh. As for the linear solvers, a GaussSeidel smoother solver is used for the velocity equation, whereas the GAMG solver with GaussSeidel smoother is used for solving the pressure equation. Turbulence treatment at the full order level is carried out by the usage of the SST $k - \omega$ turbulence model.

The first step of the offline stage is to sample the parameter space. Therefore, ten samples for the horizontal velocity at the inlet are taken inside the range $[7.5, 12]$ m/s. The next task is to take time snapshots for each parameter value after making sure that the flow reaches the final periodic regime. We emphasize that the objective of the reduction in this numerical example is to reproduce the fields and other related quantities only for the final periodic regime. As for the time snapshots, they have to be taken wisely by covering $1.5 - 2$ solution cycles of the periodic regime. This is vital for having POD modes which are representative of the fluid dynamics fields across the periodic solution cycle. Thus, in order to properly choose the time instants at which snapshots will be acquired, one has to compute the time period of the final periodic regime. This has been done by performing Fourier analysis on the FOM time signal of lift and drag fluid dynamic forces acting on the cylinder's surface. The lift coefficient C_l is obtained by the lift force L as $C_l = \frac{L}{\frac{1}{2}\rho U^2 D}$. To illustrate the aforementioned procedure, Figure 4.24 depicts the lift coefficient time signal for the case of the $U_{in} = 10$ m/s. This time signal is computed with non-uniform time steps, therefore an interpolation procedure on a uniform time grid has been done in order to allow for the Fast Fourier Transform (FFT) algorithm. After doing so, the FFT resulted in a vortex shedding period of 0.4299 s. This period corresponds to a Strouhal number [144] value of 0.2326. We recall that the Strouhal number is a non-dimensional number associated with the frequency $St = \frac{fD}{U}$. The value obtained is relatively close to the experimental one of approximately 0.20 [31]. Once the time period is computed, one may proceed by simulating enough cycles of the solution. In this case, 1.2 additional seconds were simulated with a fixed time step of 0.0003 s, and snapshots of the fluid dynamics fields were saved each 0.006 s, which results in a total of 200 snapshots for this parameter sample.

The procedure of acquiring the snapshots has been repeated for each parameter sample in the training set. The number of snapshots taken for each parameter sample is $N_T = 200$ giving a total number of snapshots $N_s = 2000$. Table 4.3 reports the time steps and the time intervals in which snapshots were taken for each value of the sampled parameter. The table suggests that the time step and the time window in which snapshots were saved vary as a result of the change in the frequency of vortex shedding of the system, which is in turn dependent on the asymptotic velocity U_{in}.

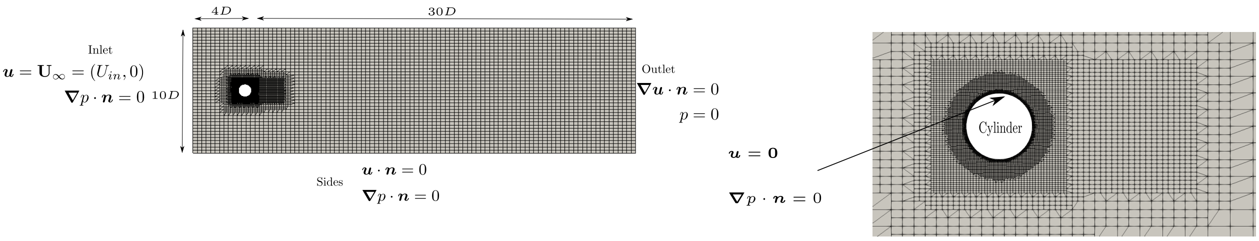
The enforcement of non-homogeneous boundary conditions at the inlet is carried out with the help of the penalty method [59, 131]. The POD procedure is therefore applied directly on the snapshots matrices of the velocity, pressure and the eddy viscosity. In that regard, Figure 4.25 shows the cumulative eigenvalues decay of the correlation matrices of the three fluid dynamics variables. Afterwards, the supremizer problem has been solved for each pressure mode [16], resulting in the supremizer modes which have been used for the enrichment of the velocity POD space.

The first attempt for the reduction of this unsteady parameterized problem will be carried out using the Uniform-ROM and the H-SUP-ROM. At a later stage, we will also investigate the other turbulent ROMs which are based on the Poisson pressure equation. However, most of the tests carried out in this section will be focused on the H-SUP-ROM because of its general reduction formulation (it may be used for both steady and unsteady flows).

Firstly, we recall that in the formulation of the H-SUP-ROM, an interpolation procedure for the approximation of the eddy viscosity coefficient vector $\boldsymbol{g}$ is required. This procedure has to be performed at each time step t^* of the simulation corresponding to the online parameter value U_{in}^*. In fact, the vector $\boldsymbol{g}(t^*, U_{in}^*)$ can be computed by employing one of the two interpolation strategies addressed in sections 3.2.1 and 3.2.2. In the first strategy introduced in subsection 3.2.1, the interpolation has to be done with respect to the combined time-parameter vector. However, this choice restricts the online time integration to be confined in the window where snapshots have been acquired (for example the 1.2 seconds for the case of $U_{in} = 10$ m/s or the window of 1 second length for $U_{in} = 12$ m/s). In order to allow time extrapolation in the ROM formulation, we decided to employ the second strategy addressed in subsection 3.2.2 in which the vector $\boldsymbol{g}$ is obtained through RBF interpolation from the reduced order velocity coefficients vectors of $\boldsymbol{a}$ and $\dot{\boldsymbol{a}}$ (Equation 3.40). As the values of the components of the vector time signal $\boldsymbol{a}(t)$ oscillate between a minima and a maxima over time, the replacement of of t by $\boldsymbol{a}(t)$ as the RBF interpolation independent variable has in fact the advantage of allowing extrapolation. Nonetheless, this is true if the values of the vector $\boldsymbol{a}$ components obtained during the ROM time integration fall within the bounds of the FOM snapshots. Consequently, it is evident that the accuracy of such interpolation outside the offline snapshots window is highly dependent on how close the current solution vector $\boldsymbol{a}$ is to the vectors of the L^2 projection coefficients (see Equation 3.42) used in the offline stage for training the RBF.

We would like to remark that in this problem we applied the splitting assumption in Equation 3.35. Thus, the interpolation using RBF is actually aimed at obtaining the coefficients of the expansion of the fluctuating reduced eddy viscosity field which approximates the FOM one $\nu_t'(\boldsymbol{x}, t)$. As a result, the contribution of the parameter (which is U_{in}) is present only in the time-averaged part $\overline{\nu_t}(\boldsymbol{x}; \boldsymbol{\mu})$. At reduced order level, the vector $\overline{\boldsymbol{g}}$ is dependent on the parameter. In this example, we have $M = 10$ which corresponds to the number of eddy viscosity time-averaged fields, which were computed as the average of the 200 time snapshots corresponding to each parameter sample. In the online stage, the vector $\overline{\boldsymbol{g}}(U_{in}^*)$ is computed by linear interpolation, while the vector $\boldsymbol{g}(t^*)$ (notice that there is no dependency on the parameter) is obtained from the 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 $\boldsymbol{g}(0)$ are obtained from the inlet velocity parameter using a linear interpolation (based on the values of the initial L^2 projection vectors of $\boldsymbol{a}(0, U_{in})$, $\boldsymbol{b}(0, U_{in})$ and $\boldsymbol{g}(0)$). We also mention that in the RBF interpolation, a shape parameter of the RBF functions has to be tuned. In this work the chosen RBFs are always of the Gaussian type, where their formula is $\zeta_{i,j}\left(\|\cdot\|_{\mathbb{R}^{2N_u}}\right) = exp(-r^2\|\cdot\|^2_{\mathbb{R}^{2N_u}})$, where r is the shape parameter of the RBF, which plays an important role for the accuracy of the interpolation. In order to determine a suitable value for this parameter, we relied on the cross validation algorithm named Leave-One-Out Cross Validation (LOOCV) [51]. This choice turned out to be sufficient for the tests conducted in this chapter.

At this point, we proceed to illustrate the details of the first numerical test, which is a cross validation or an extrapolation test in the parameter space. The test is aimed at reconstructing the time history of the fluid dynamics fields for the online parameter value $U_{in}^* = 7.75$ m/s, which was not used for training the ROM. Also, one of the objectives of this test is to obtain the fluid dynamic forces for time values which go far beyond the offline snapshots time window. After carrying out the offline phase, all the reduced vectors, matrices and tensors which appear in the dynamical systems of the U-ROM and H-SUP-ROM (in 3.57 and 3.37, respectively) are saved for later use in the online stage. The online stage involves solving the dynamical systems mentioned for the vector $\boldsymbol{a}$ in the case of the U-ROM, and for the two vectors $\boldsymbol{a}$ and


4D
30D
Inlet
$\boldsymbol{u} = \mathbf{U}_\infty = (U_{in}, 0)$
$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$
10D
Outlet
$\boldsymbol{\nabla u} \cdot \boldsymbol{n} = 0$
$p = 0$
Sides
$\boldsymbol{u} \cdot \boldsymbol{n} = 0$
$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$
$\boldsymbol{u} = \boldsymbol{0}$
$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$
Cylinder

(a) (b)

Figure 4.23: (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.

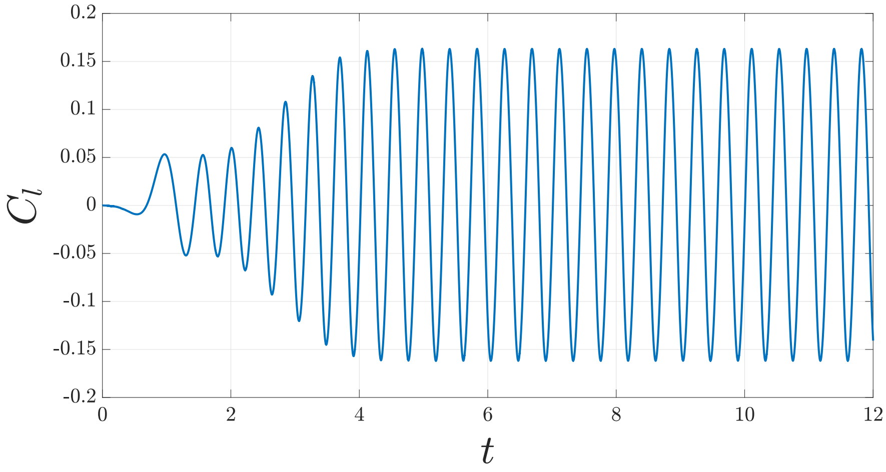


Figure 4.24: The lift coefficient curve for parameter sample $U_{in} = 10$ m/s.

$\boldsymbol{b}$ in the case of the H-SUP-ROM. The initial reduced vectors are approximated as the average of the ones corresponding to the first two offline parameter samples since $U_{in}^* = 7.75$ m/s lies between them. We recall that the reduced vector $\overline{\boldsymbol{g}}(U_{in}^*)$ is time independent and therefore it does not change during the time integration of the dynamical system. The latter vector has been approximated as $0.5\boldsymbol{e}_1^M + 0.5\boldsymbol{e}_2^M$ (recall that $\boldsymbol{e}_i^M$ is the unit vector of dimension M and contains zero elements everywhere except in its i-th component in which it has the value 1). The fields obtained by solving the reduced systems of both ROMs are then compared the FOM ones. The FOM simulator has been run for enough time to reach the periodic regime, before extending the simulation at fixed time step of 0.0004 s. FOM snapshots were taken every 0.008 s, and the total simulation time is 8 s which encompassed 13 solution periods. We emphasize that the last FOM snapshots mentioned are computed just for the sake of making the comparison with the obtained ROM fields, and these FOM fields were obviously not used during the offline stage for the computation of the POD modes. It is also important that special attention is given to the phase of the FOM snapshots. In more details, the phase of the first snapshot acquired for all the parameter samples is recommended to be the same. This is also the case for the FOM snapshots computed for the online parameter for the ROM assessment. It is assumed that the starting time of the online simulations is equal to 0 in all the tests considered in this subsection.

The results for the velocity, pressure and the eddy viscosity fields computed by the FOM, the U-ROM and the H-SUP-ROM at the time instant $t = 2.8$ s are depicted in Figure 4.26, Figure 4.27 and Figure 4.28, respectively. The results show clearly that the velocity field

Parameter sample : U_{in} **in** m/s	**FOM time step** in s	**Snapshot acquiring time** in s
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 4.3: Offline parameter samples and the corresponding snapshots data

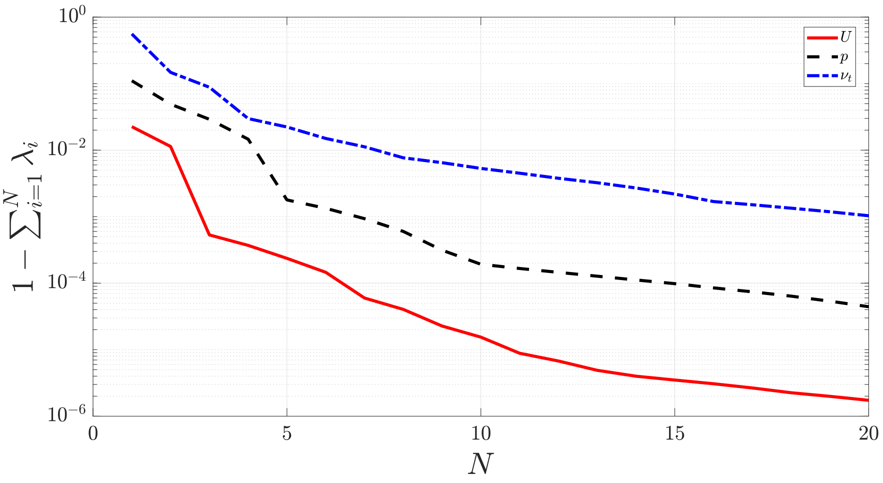
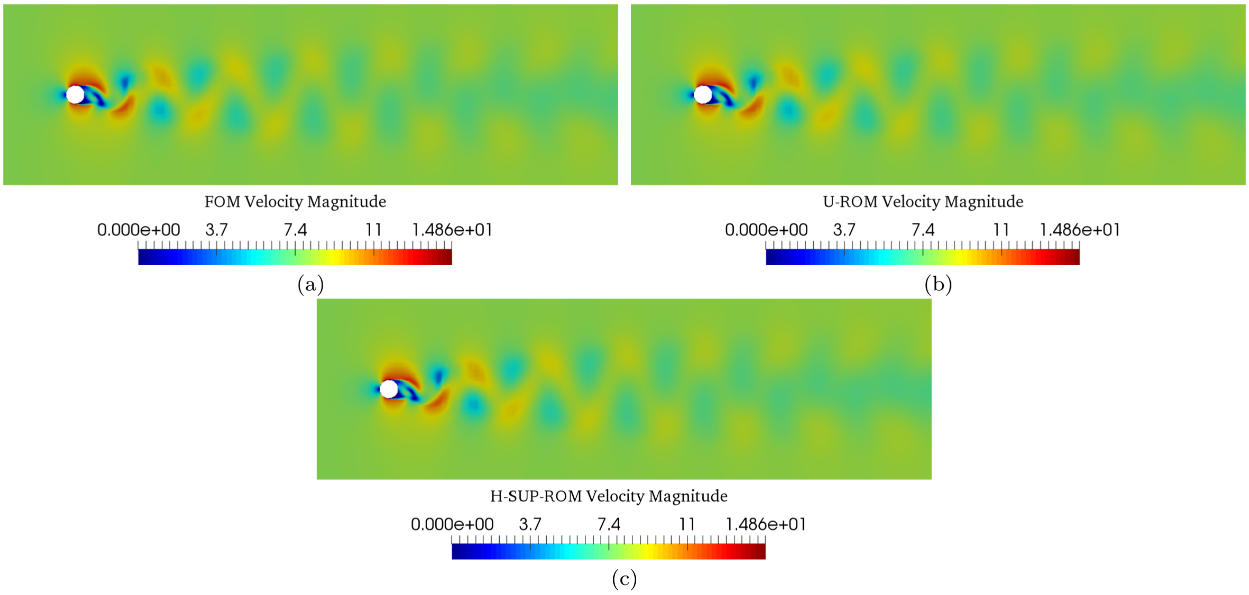


Figure 4.25: Cumulative ignored eigenvalues decay. In the plot, the solid red line refers to the velocity eigenvalues, the dashed black line indicates the pressure eigenvalues and the dash-dotted blue line finally refers to the eddy viscosity eigenvalues.

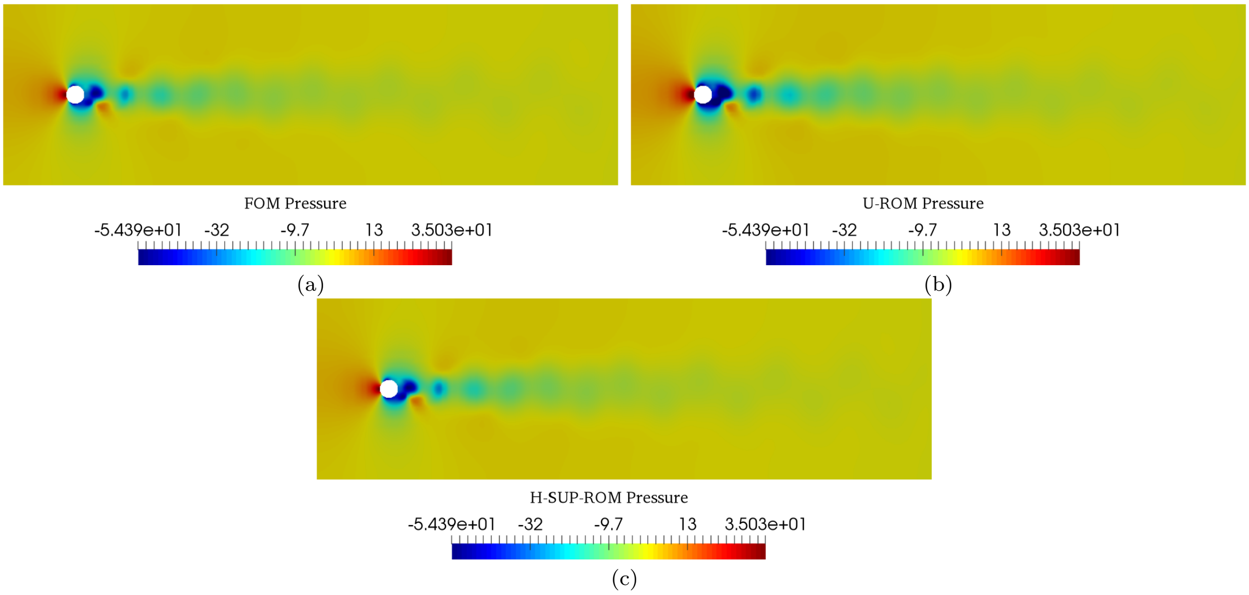
has been reconstructed accurately by both the U-ROM and the H-SUP-ROM. This can be seen at both the qualitative and the quantitative levels. The L^2 relative errors for both the U-ROM and the H-SUP-ROM are in fact 1.3553 % and 0.6954 %, respectively. As for the pressure field, the accuracy of the U-ROM is considerably poor in contrast to the case of the hybrid ROM. The value of the L^2 relative error for the pressure reduced field is 33.0963 % for the U-ROM, and it is 4.8085 % for the H-SUP-ROM. The reduced pressure fields figure illustrates certainly that the U-ROM has failed in approximating the FOM pressure field. This lack of accuracy can be evidently seen in the region close to the surface of the cylinder, which clearly affects the accuracy of the reduced approximation of the lift and drag forces. On the other hand, the H-SUP-ROM reconstruction of the pressure field is acceptable. We would also like to remark that the number of modes used in the online stage in the case of the U-ROM is $N_r = 14$, where as for the H-SUP-ROM, the online modes setting is $N_u = 20$ and $N_p = N_S = N_{\nu_t} = 10$. These choices of the number of the online modes have given the best results in terms of L^2 relative error for both ROMs in question among the other attempted possible choices.

In order to have a better assessment of the accuracy achieved by both ROMs, it is particularly useful plotting the time evolution of the L^2 relative error associated with the approximation



(a) (b) (c)

Figure 4.26: 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 U-ROM velocity with $N_r = 14$, and finally in (c) we have the H-SUP-ROM velocity with $N_u = 20$ and $N_p = N_S = N_{\nu_t} = 10$.



(a) (b) (c)

Figure 4.27: 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 U-ROM pressure with $N_r = 14$, and finally in (c) we have the H-SUP-ROM pressure with $N_u = 20$ and $N_p = N_S = N_{\nu_t} = 10$.

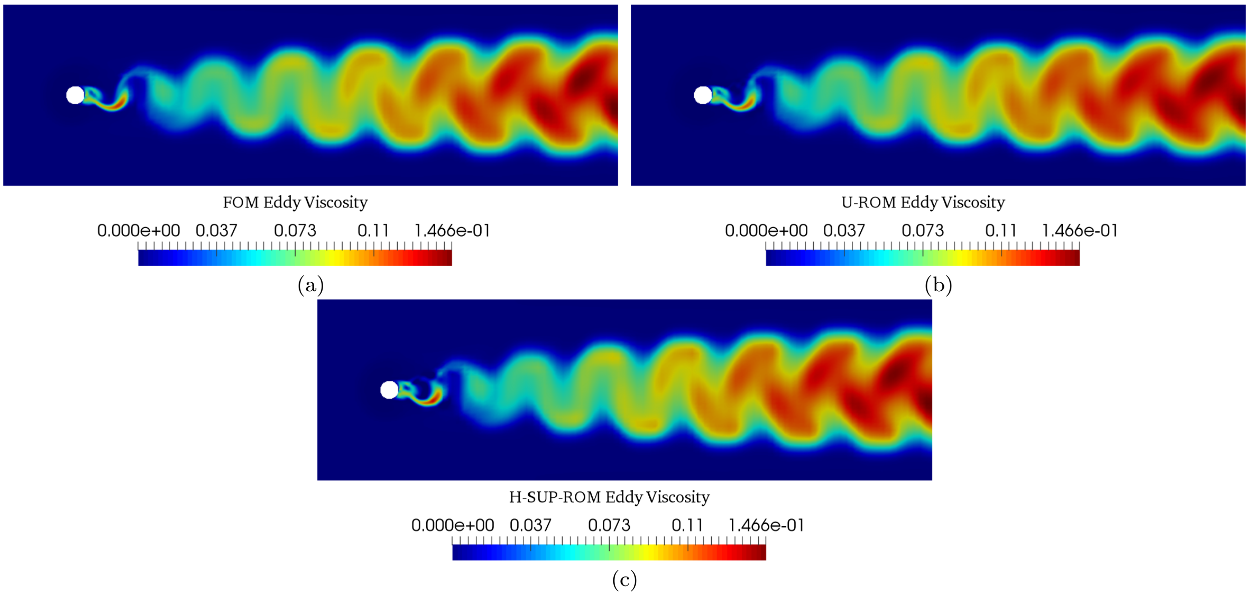


Figure 4.28: Eddy viscosity fields for the parameter value $U_{in} = 7.75$ m/s at $t = 2.8$ s: (a) shows the FOM eddy viscosity, while in (b) one can see the U-ROM eddy viscosity with $N_r = 14$, and finally in (c) we have the H-SUP-ROM eddy viscosity with $N_u = 20$ and $N_p = N_S = N_{\nu_t} = 10$.

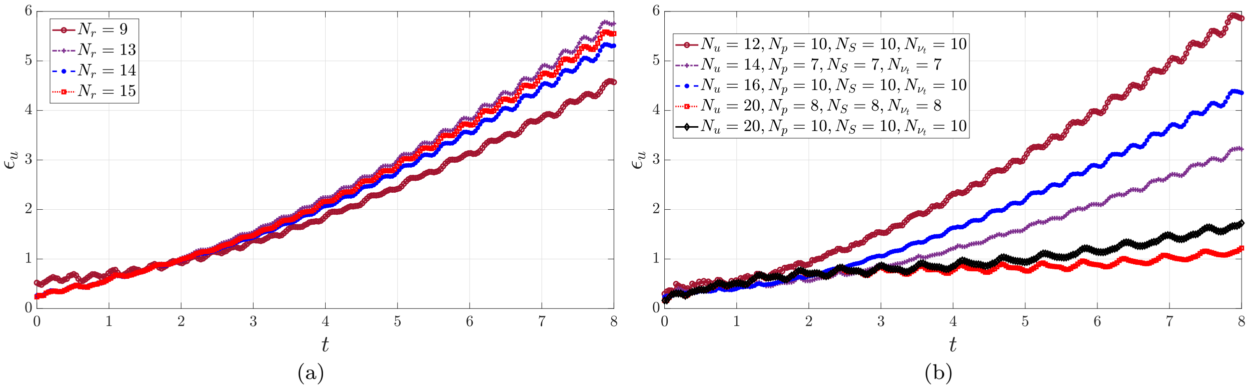


Figure 4.29: The time evolution of the L^2 relative errors of the velocity reduced approximations for both the U-ROM and the H-SUP-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the U-ROM model. Figure (b) depicts the case of the H-SUP-ROM model. The error values in both graphs are in percentages.

of each field. Such plot for the case of the velocity field is depicted in Figure 4.29, while Figure 4.30 shows the error curve for the pressure field. In reduced order modeling, one should expect that the behavior of the error as function of the time is going to be increasing. However, ROMs which are able to contain the growth of such error during the integration of the dynamical system are considered more accurate and favorable. In this regard, it can be appreciated from the error figures that the H-SUP-ROM has given better results in terms of curbing the error values, where one can see that by increasing the number of velocity modes N_u the maximum values of ε_u tend to decrease. Also, it may be well observed from Figure 4.30 that the pressure reduced approximation is not accurate for the U-ROM, while the H-SUP-ROM has given satisfactory pressure results.

The next result in this test is related to one of the most important performance indicators for engineers when dealing with problems such as the crossflow cylinder. This performance indicator is the fluid dynamics forces acting on the surface of the cylinder. These forces depend on the values of the velocity and pressure fields in specific local areas. Thus, evaluating

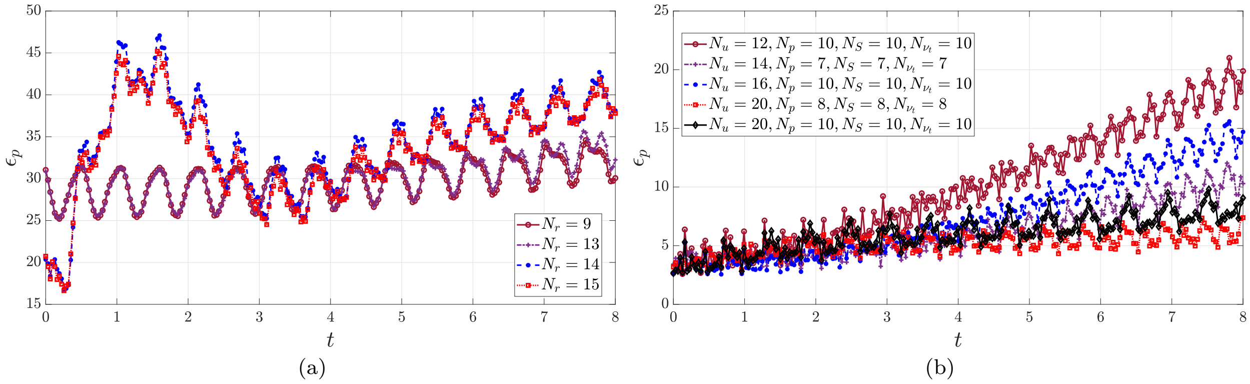


Figure 4.30: The time evolution of the L^2 relative errors of the pressure reduced approximations for both the U-ROM and the H-SUP-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the U-ROM model. Figure (b) depicts the case of the H-SUP-ROM model. The error values in both graphs are in percentages.

the accuracy of the ROM through the global error evaluators discussed so far is not enough, especially when there is an interest in recovering the forces acting on the surface of a body immersed in the flow. 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 H-SUP-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 H-SUP-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 modeling community it is recommended to refrain from such procedure, 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 the online fluid dynamic forces computation has been laid out in section 2.8.

The resulting lift coefficients curves computed by both the U-ROM and the H-SUP-ROM are compared to the FOM one. Figure 4.31 shows the time signal of the lift coefficient obtained by all of the FOM, the U-ROM and the H-SUP-ROM for the parameter value $U^*_{in} = 7.75$ m/s. In (a), one may see the full history of the lift coefficient for the time range considered in this test, that is $[0, 8]$ s. The figure on the right in (b) depicts the values of the lift coefficient curves just in the segment containing the last two seconds. The last figures apparently illustrate the fact that the H-SUP-ROM outperforms the U-ROM. In fact, it can be concluded from the previous figures that the U-ROM has an instability problem when it comes to long-time integration. On the other hand, the H-SUP-ROM C_l curve is matching its FOM counterpart to a good degree. The number of reduced modes employed in the online stage is still the same as mentioned for the fields results reported above.

In order to have a better assessment of the reconstruction of the lift coefficient by both ROMs through a quantitative point of view, we introduce the following L^2 relative percentage error, in the integration time interval $[T1, T2]$, between the FOM C_l and its reduced approximations by the both ROMs, namely

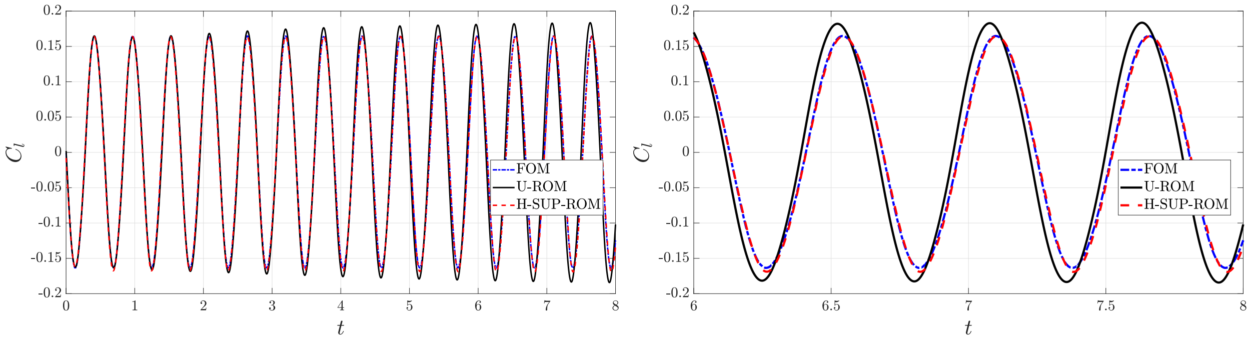
$$\varepsilon_{C_L} = \frac{\|C_l(t) - C_l{}^*(t)\|_{L_2(T_1,T_2)}}{\|C_l(t)\|_{L_2(T_1,T_2)}} \times 100\%, \tag{4.7}$$

where $C_l(t)$ is the time signal of the values of the FOM lift coefficients at all time instants between T_1 and T_2. On the other hand $C_l{}^*(t)$ is the time evolution of the lift coefficients computed by the reduced order model — whether the U-ROM or the H-SUP-ROM. Referring back to Figure 4.31, we calculated the error metric defined above for the full time range of $[0, 8]$ and also for its corresponding first half $[0, 4]$. The values of the U-ROM ε_{C_L} for the two time intervals are 16.7095 % and 5.1252 %, for the full range and its first half, respectively. Instead, the H-SUP-ROM approximation of the C_l has given two close error values for these ranges, which are 3.5792 % and 4.0504 %. The latter values of the C_l error demonstrate in fact that the U-ROM has a clear stability problem for long-time integration problems. Such instabilities have not been observed in the case of the hybrid model. In Figure 4.32, one may see the behavior of the error mentioned above as a function of the number of modes used in the online stage. It can be noticed from the previous figures that the H-SUP-ROM has been capable of recovering the FOM force coefficient with satisfying level of accuracy. Actually, the error values obtained by the H-SUP-ROM approach values as low as 3 %, while the U-ROM C_l is consistently above 16 % off the FOM values.

A further aspect of the results shown in Figure 4.32 will be investigated for the sake of having a better understanding of the ROMs accuracy. This is done by trying to attribute how much of the L^2 error is caused by imprecise reproduction of the amplitude or the frequency of the lift coefficient oscillations. To this end, we define the relative peak error ε_{peak} as follows:

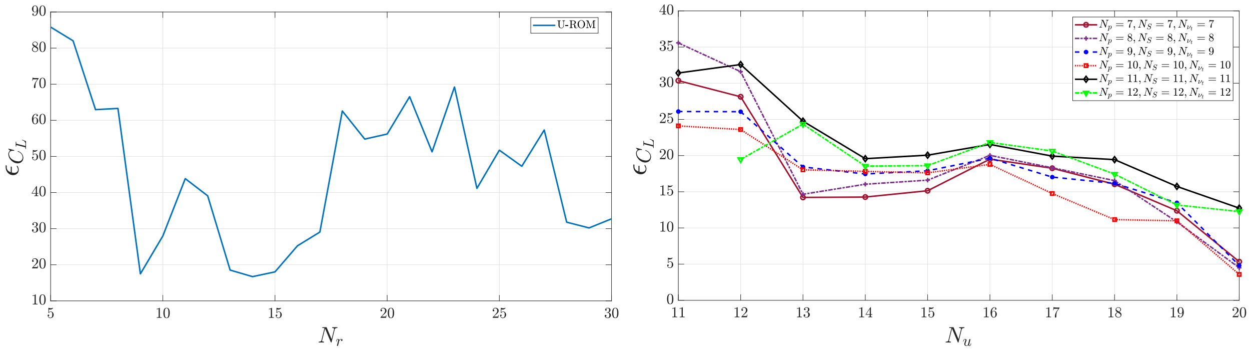
$$\varepsilon_{n,peak} = \frac{PK_{n,FOM} - PK_{n,*}}{PK_{n,FOM}} \times 100\%, \tag{4.8}$$

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 U-ROM or H-SUP-ROM C_l peak. Figure 4.33 shows the relative peak error for both ROMs. The relative peak error values computed for each of the 29 peaks the time interval $[0, 8]$ s are presented in Figure 4.33. It has to be remarked that the peak errors are presented for the values of the number of online modes used for the U-ROM N_r which has given the best results for the relative L^2 lift error analysis shown previously. Figure 4.33 indicates that the relative error of the U-ROM increases with time until it reaches values as high as $10 - 20$ %. Unlike the case of the U-ROM, the hybrid ROM has lower values of the relative peak errors which are below 3.5 % for several modal truncation order for velocity, pressure, supremizers and the eddy viscosity.



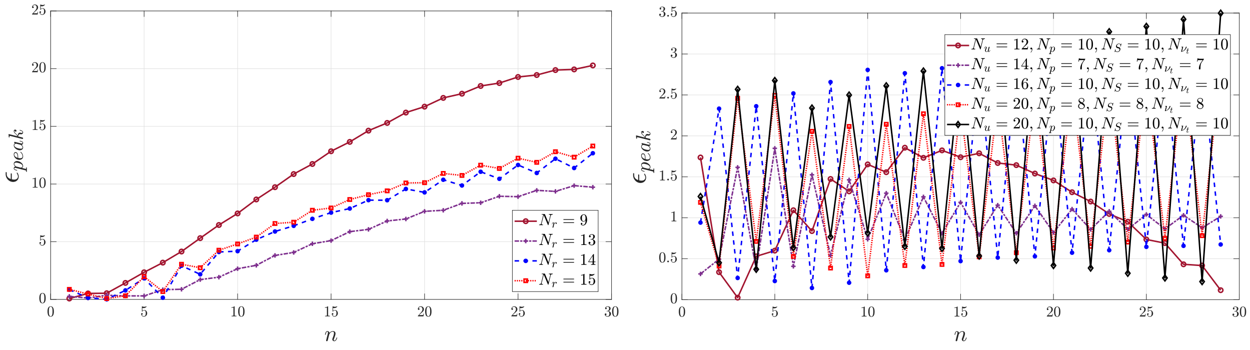
(a) (b)

Figure 4.31: 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 U-ROM and the H-SUP-ROM lift coefficients histories : (a) the full range is shown (b) the last 2 s C_l is shown.



(a) (b)

Figure 4.32: 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 U-ROM and the H-SUP-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 U-ROM and the H-SUP-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the U-ROM model, where N_r is the number of modes used in the online stage for all variables (by construction of the U-ROM it is not possible to choose different number of online modes for the reduced variables). Figure (b) depicts the case of the H-SUP-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.



(a) (b)

Figure 4.33: 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 U-ROM and the H-SUP-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 U-ROM and the H-SUP-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the U-ROM model, where N_r is the number of modes used in the online stage for all variables (by construction of the U-ROM it is not possible to choose different number of online modes for the reduced variables). Figure (b) depicts the case of the H-SUP-ROM model. The error values in both graphs are in percentages.

After having shown the results of a cross validation test for both the U-ROM and the H-SUP-ROM, now we proceed by testing the two other turbulent ROMs addressed in section 3.3 on the same problem and for the same cross validation test. These ROMs are named as the SU-PPE-ROM and the H-PPE-ROM. Both ROMs are based on the use of the Poisson equation for pressure at the reduced order level for the stabilization of the pressure field. The two ROMs can be used just for the reduction of unsteady flows, where due to the formulation of the boundary conditions they can not be extended to the steady case. The difference between these two ROMs lies in the fact that the SU-PPE-ROM is an intrusive ROM while the H-PPE-ROM is a hybrid one. The H-PPE-ROM employs an equivalent procedure to the H-SUP-ROM when it comes to the approximation of the eddy viscosity field.

As mentioned above, we will perform a cross validation test for the SU-PPE-ROM and the H-PPE-ROM which is identical to the one conducted for the U-ROM and the H-SUP-ROM. Therefore, the online parameter value is still $U_{in}^* = 7.75$ m/s. The SU-PPE-ROM is applied with the use of the penalty method for the treatment of the non-homogeneous boundary conditions. On the contrary, this time the lifting function method will be utilized to enforce the boundary conditions for the hybrid ROM (the H-PPE-ROM in this case). The lifting function chosen in this case corresponds to the solution of a potential flow problem with the inlet velocity being fixed at $U_{in} = 1$ m/s. The problem is solved by running an iterative steady solver and when the solver completes the iterative procedure, the velocity field obtained at the end is considered as the additional lifting mode/function which will be added to the velocity POD modes. The next step involves the homogenization of the velocity snapshots using this lifting function, this gives new 2000 velocity snapshots which will be stored. The POD method is then applied on the latter velocity snapshots resulting at the end in the velocity POD modes. All the reduced matrices, vectors and tensors which appear in the dynamical systems of both ROMs in Equation 3.66 and Equation 3.63 are computed and stored. The initial velocity, pressure and eddy viscosity reduced vectors are set in the exact way as in the previous test.

The first results shown are those of the lift coefficient C_l for the same range considered in the previous test, that is $[0, 8]$ s. Figure 4.34 depict the lift coefficients curves obtained by both PPE ROMs and the FOM. It can be appreciated from the last figure that the SU-PPE-ROM has done a better job in matching the FOM for the time history of the C_l signal compared to the U-ROM in Figure 4.31. However, it can also be seen that the H-PPE-ROM has given slightly better fitting of the FOM C_l. It has to be mentioned that the number of modes used in the online stage for both ROMs represents in effect the best combination in terms of accuracy measured by the error ε_{C_L} in Equation 4.7. In Figure 4.34, the number of velocity and pressure modes is equal to 9 for the SU-PPE-ROM, while the number of velocity, pressure and eddy viscosity modes is equal to 15 in the case of the H-PPE-ROM. The results from a quantitative perspective show that the SU-PPE-ROM has an error $\varepsilon_{C_L} = 5.0525$ % for the time range $[0, 4]$ s and $\varepsilon_{C_L} = 12.4255$ % for the full time range. On the other hand, the H-PPE-ROM corresponding errors are 2.2445 % and 4.3678 %. In order to better assess the accuracy of both models, one has to consider doing a convergence analysis for the online truncation modes, and also to investigate the errors committed in the approximation of the

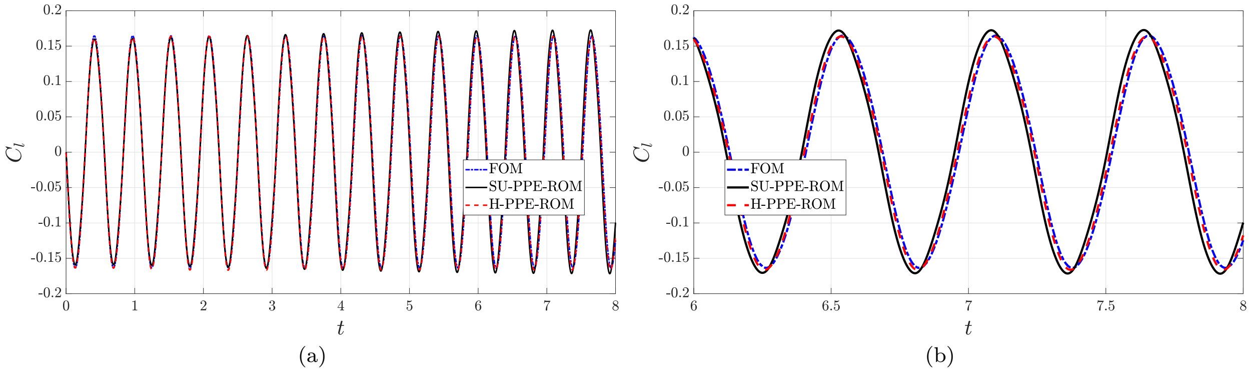


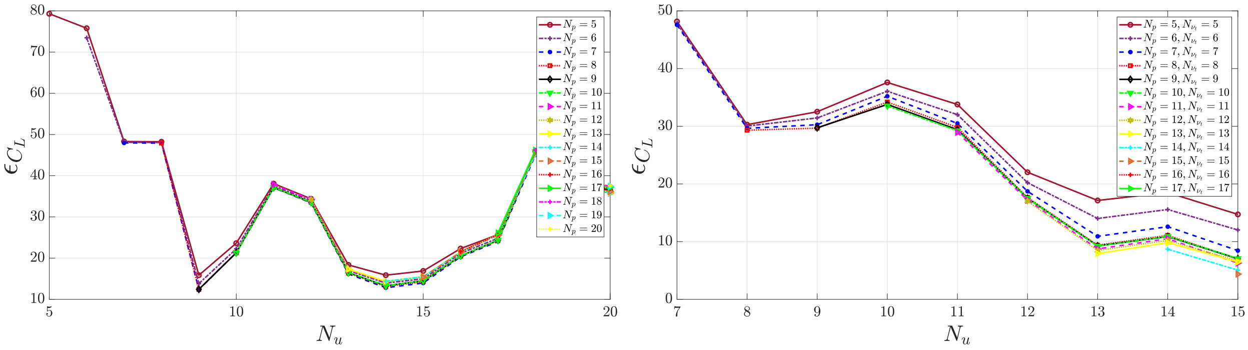
Figure 4.34: Lift coefficients curves for the cross validation test done for the parameter value $U_{in} = 7.75$ m/s for the time range $[0, 8]$ s, the figure shows the FOM, the SU-PPE-ROM and the H-PPE-ROM lift coefficients histories : (a) the full range is shown (b) the last 2 s C_l is shown.

peaks of the C_l curve.

The SU-PPE-ROM has given significantly better results in terms of accuracy and stability compared to the U-ROM. Nevertheless, the constraint of having temporal coefficients shared between velocity and eddy viscosity could be problematic in specific settings. For example, if 20 and 12 modes are needed for an accurate reconstruction of the velocity and the eddy viscosity fields, respectively, then this formulation would restrict the ROM solver to use 8 additional unnecessary eddy viscosity modes for the solution of the reduced DAE. This is likely to cause stability problems in solving the DAE.

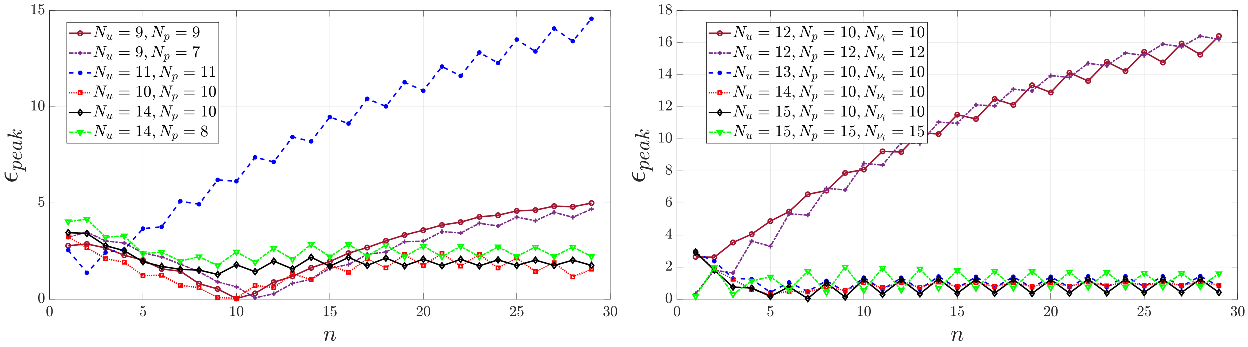
The convergence analysis is presented in Figure 4.35, where one can see that the error values for the SU-PPE-ROM reaches a minimum when $N_u = N_p = 9$ which is the case depicted in Figure 4.34, for a total number of degrees of freedom of 18. The error has a similar behavior when compared to the one depicted in Figure 4.32 for the U-ROM, where it can be seen that there are two minima for the error curve occurring at truncation values of 9 and 14 for the velocity and the pressure modes, and the error values remain consistently above 10 %. As for the H-PPE-ROM, it has achieved small error values when the number of the velocity modes grows up to $13 - 15$ modes. The values of the error are lower than 10 % for several truncation settings, for example it is around 8 % when 13 modes are employed for all reduced variables. It has to be remarked that in the H-PPE-ROM, the N_u variable includes already the lifting function added for the treatment of the boundary conditions at the reduced order level.

The results in Figure 4.36 report the peaks error for the PPE ROMs for different settings of the online modes. Again, here it can be observed that compared to the U-ROM, the SU-PPE ROM has achieved better results which speaks to the necessity of having a separate set of reduced coefficients for the pressure field. The values of the peaks errors (for the SU-PPE ROM) have reached as low as 2 % for certain choices of the online truncation. The last figure illustrates also that the H-PPE-ROM has been successful in recovering the peaks of the C_l time signal with errors lower than 2 % for online truncations with $N_u = 13, 14, 15$. Another important result is the one reported in Figure 4.37 and Figure 4.38, where one can see the time evolution of the L^2 error for both ROMs in question. It can be appreciated from the previous figures that the H-PPE-ROM outperforms the SU-PPE-ROM for the error committed in the



(a) (b)

Figure 4.35: 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 SU-PPE-ROM and the H-PPE-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 U-ROM and the H-PPE-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the SU-PPE-ROM model, where N_u is the number of modes used in the online stage for both the velocity and the eddy viscosity, while N_p is the number of modes used for the pressure field. Figure (b) depicts the case of the H-PPE-ROM model, where one can see the error values varying the number of modes used for the velocity (including the lifting velocity mode) with different fixed settings for the two other variables (the pressure and the eddy viscosity). The error values in both graphs are in percentages.



(a) (b)

Figure 4.36: 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 SU-PPE-ROM and the H-PPE-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 SU-PPE-ROM and the H-PPE-ROM models for the time range $[0, 8]$ s : (a) shows the error curve for the U-ROM model, where N_u is the number of modes used in the online stage for both the velocity and the eddy viscosity, while N_p is the number of modes used for the pressure field. Figure (b) depicts the case of the H-PPE-ROM model. The error values in both graphs are in percentages.

approximation of the pressure field. In fact, in the case of the SU-PPE-ROM, the error values at the end of the time integration are above 10 % for all the different truncations. On the other hand, the corresponding error value for the H-PPE-ROM reaches values of $4 - 7$ % at the final time of the reduced simulations for certain choices of the online modes. It can be also deduced that the H-PPE-ROM has performed better in that regard than the H-SUP-ROM (see the corresponding figure in 4.30), where it can be noticed that the pressure error growth in the case of PPE-based hybrid ROM is more suppressed. The justification of that could be that the use of the Poisson equation for pressure at the reduced level has introduced a higher level of consistency between the ROM formulation and the FOM one. In fact, the FOM constructs a pressure equation from the momentum and the continuity equation as explained in section 2.2. This procedure is mimicked by the H-PPE-ROM. Nonetheless, we have to mention that in general the FOM solution algorithm differs from the ROM one, in that the FOM one is based on an iterative segregated approach, while the ROM one uses a coupled approach.

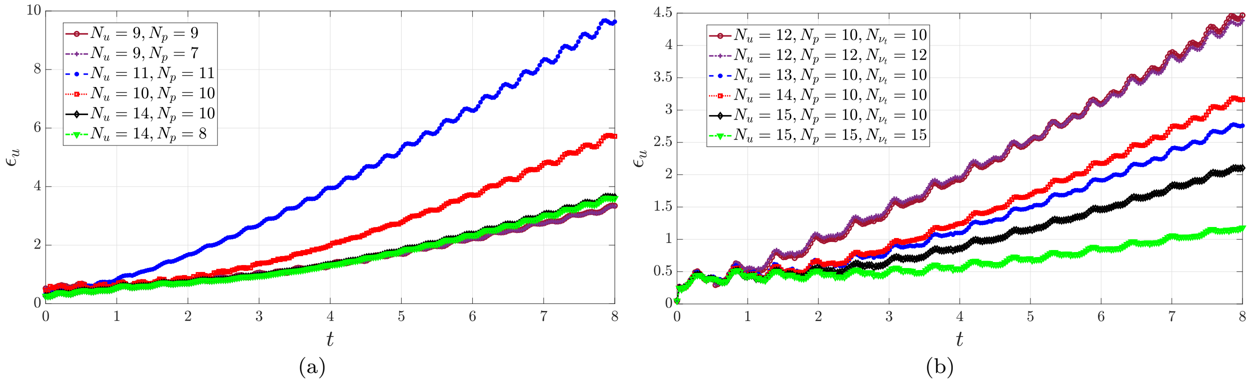


Figure 4.37: The time evolution of the L^2 relative errors of the velocity reduced approximations for both the SU-PPE-ROM and the H-PPE-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the SU-PPE-ROM model. Figure (b) depicts the case of the H-PPE-ROM model. The error values in both graphs are in percentages.

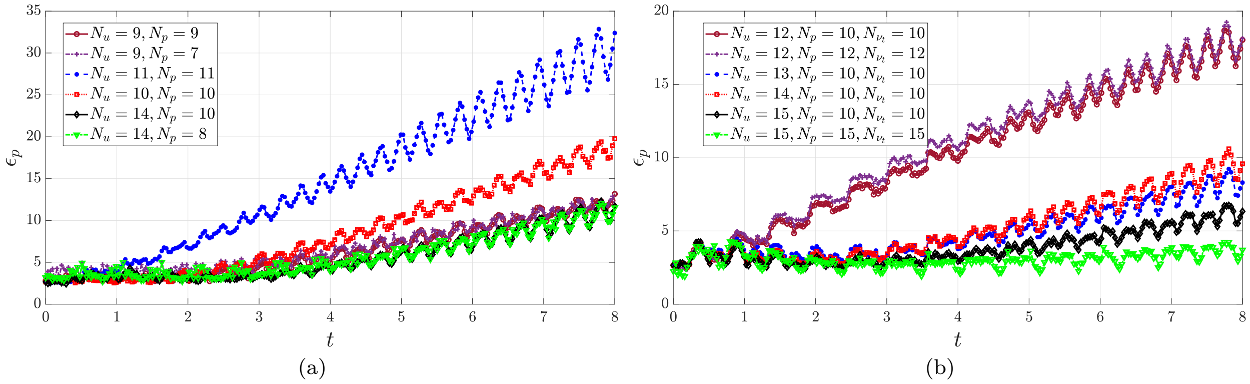


Figure 4.38: The time evolution of the L^2 relative errors of the pressure reduced approximations for both the SU-PPE-ROM and the H-PPE-ROM models. The curves correspond to the case run with the parameter value $U_{in} = 7.75$ m/s : (a) shows the error curve for the SU-PPE-ROM model. Figure (b) depicts the case of the H-PPE-ROM model. The error values in both graphs are in percentages.

To summarize the results given by the four ROMs tested in the current problem, we report Table 4.4 which compares the performance of all the ROMs in terms of the accuracy and the efficiency. In the table SU stands for the speed up, where here it is calculated as $SU = \frac{t_{off}}{t_{online}}$, with t_{off} being the wall time needed to run the full order simulations and export the snapshots, and t_{online} is the corresponding online time. In the cross validation test considered for $U_{in} = 7.75$ m/s, the value of t_{off} is 1540.766 s. This corresponds to the wall time of running the simulations in parallel on 6 processors, along with the time needed to reconstruct the fields. We recall that the results reported in the table are for the best choices of the online modes for the error ε_{C_L} in the full time integration range $[0, 8]$ s.

The ROM	ε_{C_L}	$\max_n \varepsilon_{n,peak}$	**SU**	**DoF**
The U-ROM	16.7095 %	12.67 %	31.80	$N_r = 14$
The SU-PPE-ROM	12.4255 %	4.995 %	30.16	$N_u = N_p = 9$
The H-SUP-ROM	3.5792 %	3.498 %	9.34	$N_u = 20,\ N_p = N_S = N_{\nu_t} = 10$
The H-PPE-ROM	4.3678 %	1.59 %	12.86	$N_u = N_p = N_{\nu_t} = 15$

Table 4.4: Summary of the accuracy and the efficiency results for the ROMs considered in the problem of the flow around the cylinder.

The cross validation test for $U^*_{in} = 7.75$ m/s is now concluded. The objective of the next test is to measure the accuracy of the hybrid ROM (in this case the H-SUP-ROM was chosen) for higher Reynolds number and also for longer time extrapolation intervals. Therefore, we

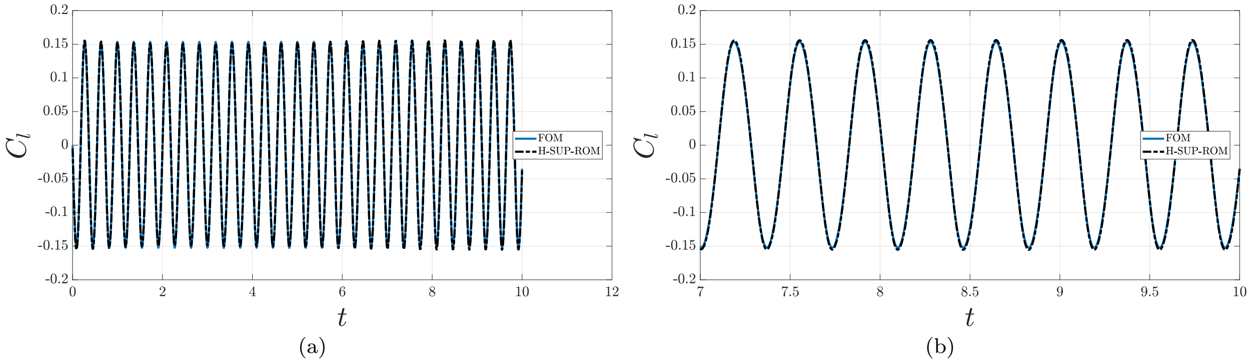


Figure 4.39: Lift coefficients curves for the cross validation test done for the parameter value $U_{in} = 11.75$ m/s for the time range $[0, 10]$ s, the figure shows the FOM and the H-SUP-ROM lift coefficients histories : (a) the full range is shown (b) the last 3 s history of C_l is shown.

choose the inlet parameter online sample to be $U^*_{in} = 11.75$ m/s, the time range for which the ROM will be run is $[0, 10]$ s. This time interval has 27 solution cycles which is almost double the number of solution cycles contained in the $[0, 8]$ s considered in the previous case. The results include the time history of the lift coefficient C_l in addition to its corresponding L^2 relative error named ε_{C_L} and also the peaks error and the reduced approximation of the time period.

As in the first test, the FOM simulator was run enough time to reach the periodic regime. Then the simulation was extended for other 10 s starting from a phase which is equivalent to the one set for the offline samples. The additional 10 seconds were simulated with time step equal to 0.00025 s. The formulation and the reduction strategy of the H-SUP-ROM are the same as the ones set in the first numerical test in this section. The C_l H-SUP-ROM curve is reproduced using 12 modes for the velocity and 10 modes for each of pressure, supremizers and eddy viscosity. Figure 4.39 shows the results for the C_l curves over all the whole time range considered in part (a), while the figure in (b) depicts the lift curves just for the last 3 seconds. These figures clearly suggest that the H-SUP-ROM has been successful in dealing with such high value of the Reynolds number which is equal to 1.75×10^5. They also prove the ability of the hybrid ROM in tackling the issue of long-time integration, as they indicate that qualitatively speaking the matching between the FOM and the H-SUP-ROM was not harmed even in the last solution cycles. The value of the error ε_{C_L} (as defined in Equation 4.7) is 1.9486 %. The peaks of the FOM C_l curve were recovered with maximum error of 2.0526 % for all the 55 peaks present in the curve. Finally, the average time period computed in by the FOM solver is about 0.3641 s, while the average time period approximated by the hybrid ROM is about 0.3642 s. One may conclude that the 0.2 % relative error in the approximation of the time period is mainly caused by amplitude inaccuracies rather than being a result of incorrect frequency reconstructions.

After having tested the accuracy and the reliability of the hybrid ROM proposed in this thesis, it remains to verify that the hybrid ROM meets one important objective which is the versatility of the ROM framework. This feature of the ROM is measured in terms of its applicability for different turbulent full order closure models without having to readapt itself each time the turbulence FOM modeling is changed. For the sake of testing this feature, we

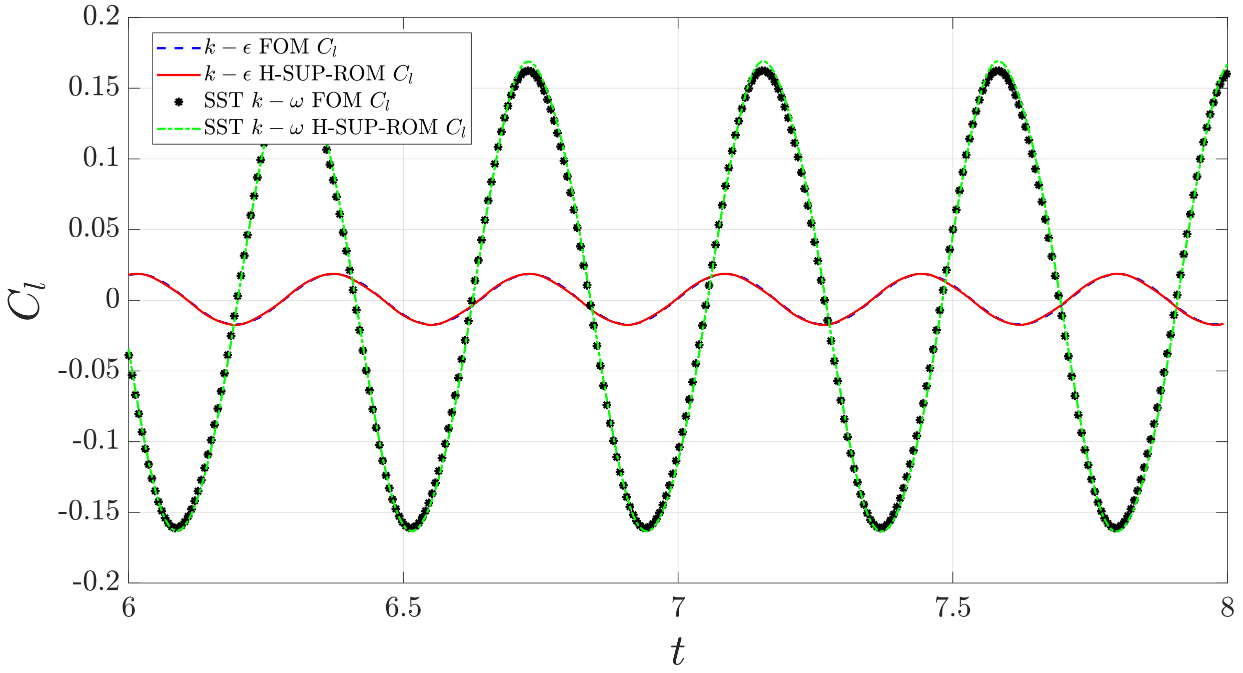


Figure 4.40: The lift coefficient curves obtained using both $k-\varepsilon$ and SST $k-\omega$ turbulence models and the H-SUP-ROM ones. The case considered is a non-parametrized one with $U_{in} = 10$ m/s corresponding to $Re = 10^5$. The plot is for the time range $t \in [6,8]$, the H-SUP-ROM achieved relative L^2 errors (over the range $t \in [0,8]$) which are less than 5 % in both cases.

have done a final test in which we generated FOM snapshots from two different turbulence closure models which are the the SST $k-\omega$ and the $k-\varepsilon$ models. The case under study is still the cylinder case but without parameters, therefore, the goal of the reduction is just to reproduce the time snapshots and extrapolate in time. The Reynolds number in this case is $Re = 10^5$, the FOM simulation were carried out for both turbulence models till the regime solution is fully developed. Then the acquirement of snapshots was done with time rate of 1.2 s and 1.6 s for $k-\varepsilon$ and SST $k-\omega$ models, respectively. The hybrid ROM chosen is the H-SUP-ROM with the penalty method for the treatment of the boundary conditions. The H-SUP-ROM was run for 8 seconds for both turbulence models. The resulted FOM lift coefficient for both closure models and the corresponding ROM ones are depicted in Figure 4.40. It is evident from the graph that the H-SUP-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. The value of the errors defined in Equation 4.7 for both turbulence models are below 5 % and the reduced approximation of time period is accurate in both cases with relative errors below 1 %.

4.4 Concluding Remarks

In this chapter we have presented the results of the application of various POD-Galerkin ROMs on different CFD problems. These POD-Galerkin ROMs have been developed in this thesis for the goal of reducing problems discretized by the finite volumes method. In addition, a number of these ROMs was specifically constructed for the reduction of turbulent flows. The results shown in this chapter had been presented in the following works [137, 73, 72, 74].

The applications considered in this chapter included a problem in uncertainty quantification and two turbulent problems. The uncertainty quantification (UQ) problem is the classical problem of the flow past an airfoil in steady state and parametrized settings. In this problem, we applied the SUP-ROM developed in section 2.6 together with the non-intrusive polynomial chaos expansion (PCE) algorithm. We have shown that the SUP-ROM could be used as an input evaluator for the PCE. This use of the SUP-ROM offers significant level of reduction of the computational cost associated with PCE computations.

In the second and third sections, we have considered two turbulent problems. The first one is the steady problem of the backward step, while the second one is the flow past a circular cylinder. In these problems, we applied the hybrid ROMs developed in this thesis. The hybrid ROMs have given accurate results for the reconstruction of the fluid dynamic fields and other outputs of interest such as the lift coefficient. In addition, they provided acceptable levels of speed up which speaks for their efficiency. Finally, the hybrid ROMs proved being sensitive to the FOM results obtained by the use of different turbulence closure models. The latter result accomplishes one of the important goals set in this thesis.

Chapter 5

Conclusions and Outlook

This chapter presents the conclusions which are drawn from the results of the work done in this thesis. It also gives an idea of possible future extensions which could enhance or complete the work presented in this thesis.

Contents

5.1 Concluding remarks

This manuscript has dealt with reduced order models for parametrized flows in the finite volume setting. The work presented in this thesis has two main goals:

- The development of ROMs techniques specifically tailored for finite volumes discretization schemes. This is particularly relevant since the FVM is widespread to simulate industrial problems.
- The implementation of ROMs designed for the reduction of turbulent flow problems. As these types of flows are ubiquitous in real-world applications, there is in fact an increasing demand to simulate them efficiently.

In the following, we summarize the methodologies developed during the preparation of this work:

- In chapter 2, we have developed several reduced order methods for the reduction of the steady and unsteady NSE. These ROMs are based on POD-Galerkin projection approach using different strategies. The first ROM exploits only the momentum equations and it assumes that the same set of coefficients for the reduced velocity and pressure solutions. The second ROM utilizes a pressure Poisson equation (PPE), which is derived by taking the divergence of the momentum equation and then exploiting the free divergence constraint on the velocity. The latter ROM assumes two different sets of coefficients for the reduced velocity and pressure solutions, and computes them exploiting both the momentum and the pressure equations at the reduced order level. The third ROM presented in this work employs the supremizer stabilization method [16], which consists in the enrichment of the velocity POD space in order to fulfill a reduced version of the inf-sup condition.
- In order to treat non-homogeneous Dirichlet conditions at the inlet boundary, we employed two different methods, the lifting function method and the penalty method. The lifting function method introduces a framework in which the non-homogeneity is transferred from the velocity fields to one or more velocity lifting functions. These lifting functions are considered as additional velocity modes which are added to the velocity POD space. Such fields can be obtained by various approaches and the suitable choice is problem dependent. In this work, the lifting modes are computed by solving a potential flow problem. We used also a penalty method which add a constraint to the momentum equation in order to fulfill the Dirichlet condition at the boundary.
- We have developed specific method in order to ensure an efficient offline/online decoupling for the computation of forces acting on body surfaces. The employed approach consists in the precomputation of a set of reduced matrices which are then employed together with velocity and pressure reduced solutions for the goal of computing the reduced forces. This last procedure does not involve any access to the original FOM mesh.

- The turbulence modeling at both the FOM and the ROM levels is addressed in chapter 3. In section 3.1, we give an idea about RANS turbulence modeling used in this work which is based on the employment of the Boussinesq assumption. The ROM developed to mimic the RANS equations complemented by eddy viscosity models (EVMs) merges classical projection-based methods with data-driven techniques. The goal was to develop a unified approach which could reduce turbulent RANS problems regardless of the EVM used by the FOM. Making use of the observation that all EVMs are based on the eddy viscosity field, the offline phase of the turbulent ROM includes the reduction of such field. At the online stage, the reduced coefficients of the eddy viscosity are obtained through radial basis functions (RBF) interpolation based on other reduced flow variables or time-parameter values. The hybrid ROM proposed in section 3.2 is based on the supremizer stabilization method. We also proposed in section 3.3 other turbulent ROMs. In section 3.3, an extension to the uniform ROM (proposed in section 2.4) to turbulent flows is presented. Also, we extended the PPE-ROM (proposed in section 2.5) by two different ways. The first one assumes that the reduced eddy viscosity and the reduced velocity solutions are the same. In the second approach, we proposed a hybrid ROM which is based on the PPE approach with separate reduced solutions for velocity, pressure and eddy viscosity.

After having summarized the methodologies followed in this thesis, we proceed to the conclusions which can be drawn from the numerical tests conducted in chapter 4. We report the summary of the numerical tests and the conclusions in the following points:

- In chapter 4, we applied the ROMs developed in this thesis on various problems. The first one proposed in section 4.1 is an uncertainty quantification (UQ) problem. The considered physical problem is the two dimensional flow past an airfoil section. Both the magnitude of the inlet velocity and the angle of attack of the airfoil are variable and have been the parameters considered. The objective of applying the ROM techniques on such a problem was that of evaluating possible reductions of non-intrusive polynomial chaos expansion (PCE) algorithm in UQ. The PCE requires to operate on specific input results which are usually computed by the full order solver. However, we tried to assess whether the PCE results could be affected, if one feeds the PCE with the ROM results instead of the FOM ones. We have shown that the SUP-ROM (developed in section 2.6) has given accurate results when used as the input evaluator for the PCE algorithm. These last results demonstrate that POD-Galerkin ROMs are a reliable surrogate input source for UQ algorithms and that they could be used for offering further reduction of the computational cost.

- In section 4.2, we considered the turbulent flow over a backward facing step. The parameterization is based on the Reynolds number ($Re = O(10^4)$) through variations of the inlet velocity. The solutions of the hybrid ROM based on the supremizer stabilization method (the H-SUP-ROM) have been compared to those obtained with both the FOM and the U-ROM (described in section 3.3). The results confirm that the H-SUP-ROM is able to reduce the turbulent problem with good accuracy even when different EVMs are employed at the FOM stage. On the contrary, the U-ROM which leads to good reduction of the velocity, has failed to provide satisfactory results for the pressure field.

- In section 4.3, we considered the unsteady of the turbulent flow around the circular cylinder. Also in this case the parameterization is based on the Reynolds number ($Re = O(10^5)$) changing the inlet velocity. The interest was in reproducing the fluid dynamics fields associated with the final periodic regime, along with other outputs of interest such as the lift coefficient time history. As in the steady problem, we considered both the H-SUP-ROM and the U-ROM for the reduction of the problem. The results shown in this case include the fluid dynamic fields, the time evolution of the L^2 relative error of the velocity and the pressure fields, the lift coefficient curve and its corresponding L^2 relative error over time as a function of the number of modes and the graph of the lift peaks relative errors as a function of the number of modes. These results have suggested that the H-SUP-ROM provides accurate results, and that it outperforms the U-ROM in several aspects. Moreover, the H-SUP-ROM proved capable of obtaining stable solutions, as the error with respect to the FOM solution does not significantly grow over time across several solution cycles. Finally, the comparison with FOM results generated with different turbulence models confirmed that the H-SUP-ROM is sensitive with respect to the EVM used at the FOM level.

- A further investigation was aimed at evaluating the accuracy gains obtained by means of the turbulence treatment. In fact, the H-SUP-ROM and the U-ROM differ in two aspects which are the pressure treatment and the turbulence treatment. Thus, we developed two PPE-ROMs with and without hybrid turbulence treatment, namely the SU-PPE-ROM and the H-PPE-ROM, respectively. This allowed for an evaluation of how much error in the reconstruction of the FOM fields is due to the pressure reconstruction technique used, and how much error is instead dependent on the eddy viscosity coefficients reduced approximation. The lift coefficients results for the unsteady cylinder case indicate that consistent performance improvement is already obtained when the SU-PPE-ROM is introduced to properly treat the pressure at the reduced level. An even further improvement is then achieved when the hybrid turbulence treatment is introduced. This is especially true when several solution cycles are simulated. A final comparison has been carried out between the solutions of H-SUP-ROM and H-PPE-ROM, which share the same turbulence model but are based on different pressure treatments. The results suggest that H-PPE-ROM is slightly more accurate than the H-SUP-ROM. This could be attributed to the fact the pressure Poisson equation employed in the H-PPE-ROM mimics the one used by the FOM FV solver introducing a higher level of consistency between the solvers.

5.2 Outlooks and Perspectives

Finally, we suggest some possible future extensions of the work carried out in this thesis.

- In this work, we have used data-driven techniques for the approximation of the eddy viscosity in the proposed turbulent ROMs. In particular, we have resorted to RBF to interpolate the eddy viscosity coefficients based on time-parameter or the reduced velocity coefficients. When the interpolation variable is the time-parameter, possible alternatives to RBF could be Dynamic Mode Decomposition (DMD) which is designed to study the time evolution of a system and would allow for time extrapolation. On the other hand, in cases when the reduced eddy viscosity vector interpolation based on the velocity vector, it would be particularly interesting to employ the Artificial Neural Networks (ANNs) [29] which is particularly suited for multi-dimensional input-output maps.

- The applications of the POD-Galerkin ROMs proposed in this thesis is particularly intriguing in physical problems which show significant qualitative changes of behavior depending on parameter values. In particular, we are currently working on the problem of the aerodynamic flow past an airfoil section, in which hysteresis occurs due to angle of attack variations across the stall region. Capturing this phenomenon not only at the FOM level but also at the reduced order model is of significant value.

- A possible way to improve the reduced order models developed in this work could be that of reproducing at the reduced level the segregated approaches like SIMPLE or PIMPLE used in OpenFOAM solvers instead of the fully coupled approach currently used.

List of Figures

List of Tables

Bibliography

[1] OpenFOAM website. https://openfoam.org/. Accessed: 13-10-2017.

[2] Star ccm+ users manual. http://www.cd-adapco.com/products/star-ccm/documentation.

[3] Abbott, I. *Theory of wing sections : including a summary of airfoil data*. Dover Publications, Mineola, 1999.

[4] Akhtar, I., Nayfeh, A. H., and Ribbens, C. J. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theoretical and Computational Fluid Dynamics 23*, 3 (2009), 213–237.

[5] Akkari, N., Casenave, F., and Moureau, V. Time Stable Reduced Order Modeling by an Enhanced Reduced Order Basis of the Turbulent and Incompressible 3D Navier–Stokes Equations. *Mathematical and Computational Applications 24*, 2 (Apr. 2019), 45.

[6] Alla, A., and Kutz, J. N. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing 39*, 5 (Jan. 2017), B778–B796.

[7] Amsallem, D., and Farhat, C. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering 91*, 4 (Feb 2012), 358–377.

[8] ANSYS. Ansys fluent - cfd software | ansys, 2016.

[9] Aubry, N., Holmes, P., Lumley, J. L., and Stone, E. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics 192*, -1 (Jul 1988), 115.

[10] Babuška, I. The finite element method with penalty. *Mathematics of Computation 27*, 122 (May 1973), 221–221.

[11] Bader, E., Kärcher, M., Grepl, M. A., and Veroy, K. Certified Reduced Basis Methods for Parametrized Distributed Elliptic Optimal Control Problems with Control Constraints. *SIAM Journal on Scientific Computing 38*, 6 (Jan 2016), A3921–A3946.

[12] Baiges, J., Codina, R., and Idelsohn, S. Reduced-order modelling strategies for the finite element approximation of the incompressible Navier-Stokes equations. *Computational Methods in Applied Sciences 33* (2014), 189–216.

[13] Balajewicz, M., and Dowell, E. H. Stabilization of projection-based reduced order models of the Navier–Stokes. *Nonlinear Dynamics 70*, 2 (Aug 2012), 1619–1632.

[14] Balajewicz, M., Tezaur, I., and Dowell, E. Minimal subspace rotation on the Stiefel manifold for stabilization and enhancement of projection-based reduced order models for the compressible Navier–Stokes equations. *Journal of Computational Physics 321* (2016), 224 – 241.

[15] Balajewicz, M. J., Dowell, E. H., and Noack, B. R. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation. *Journal of Fluid Mechanics 729* (July 2013), 285–308.

[16] Ballarin, F., Manzoni, A., Quarteroni, A., and Rozza, G. Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Engineering 102*, 5 (Nov 2014), 1136–1161.

[17] Ballarin, F., Manzoni, A., Quarteroni, A., and Rozza, G. 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.

[18] Ballarin, F., Rebollo, T. C., Ávila, E. D., Mármol, M. G., and Rozza, G. Certified Reduced Basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height. *Computers & Mathematics with Applications 80*, 5 (Sept. 2020), 973–989.

[19] Ballarin, F., and Rozza, G. POD-Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids 82*, 12 (Jun 2016), 1010–1034.

[20] Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique 339*, 9 (Nov 2004), 667–672.

[21] Barrett, J. W., and Elliott, C. M. Finite element approximation of the dirichlet problem using the boundary penalty method. *Numerische Mathematik 49*, 4 (July 1986), 343–366.

[22] Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., and Silveira, L. M. *Model Order Reduction*, vol. 1-3. De Gruyter, Berlin, Boston, 2020.

[23] Benner, P., Gugercin, S., and Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review 57*, 4 (Jan 2015), 483–531.

[24] Benner, P., Ohlberger, M., Pater, A., Rozza, G., and Urban, K. *Model Reduction of Parametrized Systems.*, vol. 17 of *MS&A series.* Springer, 2017.

[25] Benosman, M., Borggaard, J., San, O., and Kramer, B. Learning-based robust stabilization for reduced-order models of 2D and 3D Boussinesq equations. *Applied Mathematical Modelling 49* (Sept. 2017), 162–181.

[26] Bergmann, M., Bruneau, C.-H., and Iollo, A. Enablers for robust POD models. *Journal of Computational Physics 228*, 2 (2009), 516–538.

[27] Bergmann, M., Bruneau, C.-H., and Iollo, A. Improvement of reduced order modeling based on POD. In *Computational Fluid Dynamics 2008*. Springer Berlin Heidelberg, 2009, pp. 779–784.

[28] Berselli, L. C., Iliescu, T., and Layton, W. J. *Mathematics of large eddy simulation of turbulent flows*. Springer Science & Business Media, 2005.

[29] Bishop, C. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.

[30] Bizon, K., and Continillo, G. Reduced order modelling of chemical reactors with recycle by means of POD-penalty method. *Computers & Chemical Engineering 39* (Apr 2012), 22–32.

[31] Blevins, R. D., and Coughran, C. S. 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.

[32] Boussinesq, J. des eaux courantes. Memoires presentes par divers savants a l'Academic des Sciences de l'Institut National de France. *Tome XXIII*, 1 (1877).

[33] Burkardt, J., Gunzburger, M., and Lee, H.-C. POD and CVT-based reduced-order modeling of navier–stokes flows. *Computer Methods in Applied Mechanics and Engineering 196*, 1-3 (Dec 2006), 337–355.

[34] Caiazzo, A., Iliescu, T., John, V., and Schyschlowa, S. A numerical investigation of velocity–pressure reduced order models for incompressible flows. *Journal of Computational Physics 259* (Feb. 2014), 598–616.

[35] Carlberg, K., Barone, M., and Antil, H. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics 330* (Feb 2017), 693–734.

[36] Carlberg, K., Bou-Mosleh, C., and Farhat, C. 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 (Oct 2010), 155–181.

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

[38] Chacón, T. R., Delgado, E. Á., Gómez, M. M., Ballarin, F., and Rozza, G. On a certified Smagorinsky reduced basis turbulence model. *SIAM Journal on Numerical Analysis 55*, 6 (Jan. 2017), 3047–3067.

[39] Chacón, T. R., Delgado, E. Á., Gómez, M. M., and Rubino, S. Assessment of self-adapting local projection-based solvers for laminar and turbulent industrial flows. *Journal of Mathematics in Industry 8*, 1 (June 2018).

[40] Chen, P., Quarteroni, A., and Rozza, G. Comparison between reduced basis and stochastic collocation methods for elliptic problems. *Journal of Scientific Computing 59*, 1 (2014), 187–216.

[41] Chen, P., Quarteroni, A., and Rozza, G. Reduced basis methods for uncertainty quantification. *SIAM/ASA Journal on Uncertainty Quantification 5*, 1 (2017), 813–869.

[42] Chinesta, F., Huerta, A., Rozza, G., and Willcox, K. Model reduction methods. *Encyclopedia of Computational Mechanics Second Edition* (2017), 1–36.

[43] Couplet, M., Basdevant, C., and Sagaut, P. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics 207*, 1 (Jul 2005), 192–220.

[44] Couplet, M., Sagaut, P., and Basdevant, C. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. *Journal of Fluid Mechanics 491* (Sep 2003), 275–284.

[45] Courant, R., Friedrichs, K., and Lewy, H. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen 100*, 1 (Dec 1928), 32–74.

[46] Courant, R., Friedrichs, K., and Lewy, H. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development 11*, 2 (Mar 1967), 215–234.

[47] Cowles, G. W., Parolini, N., and Sawley, M. L. Numerical simulation using RANS-based tools for America's Cup design. *Proceedings of the 16th Chesapeake Sailing Yacht Symposium* (2003).

[48] Demo, N., Tezzele, M., Gustin, G., Lavini, G., and Rozza, G. 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* (2018), IOS Press, Amsterdam, The Netherlands, pp. 212–219.

[49] Drohmann, M., Haasdonk, B., and Ohlberger, M. Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. *SIAM Journal on Scientific Computing 34*, 2 (2012), A937–A969.

[50] Dumon, A., Allery, C., and Ammar, A. Proper General Decomposition (PGD) for the resolution of Navier-Stokes equations. *Journal of Computational Physics 230*, 4 (2011), 1387–1407.

[51] Fasshauer, G. E., and Zhang, J. G. On choosing "optimal" shape parameters for RBF approximation. *Numerical Algorithms 45*, 1-4 (Mar. 2007), 345–368.

[52] Fick, L., Maday, Y., Patera, A. T., and Taddei, T. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *Journal of Computational Physics 371* (Oct. 2018), 214–243.

[53] Formaggia, L., Miglio, E., Mola, A., and Parolini, N. Fluid–structure interaction problems in free surface flows: Application to boat dynamics. *International Journal for Numerical Methods in Fluids 56*, 8 (2008), 965–978.

[54] Formaggia, L., Mola, A., Parolini, N., and Pischiutta, M. A three-dimensional model for the dynamics and hydrodynamics of rowing boats. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology 224*, 1 (Sept. 2009), 51–61.

[55] Fu, S., and Wang, L. RANS modeling of high-speed aerodynamic flow transition with consideration of stability theory. *Progress in Aerospace Sciences 58* (Apr. 2013), 36–59.

[56] Galletti, B., Bruneau, C. H., Zannetti, L., and Iollo, A. Low-order modelling of laminar flow regimes past a confined square cylinder. *Journal of Fluid Mechanics 503* (Mar 2004), 161–170.

[57] Georgaka, S., Stabile, G., Rozza, G., and Bluck, M. J. Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems. *Communications in Computational Physics 27*, 1 (June 2020), 1–32.

[58] Ghanem, R. G., and Spanos, P. D. *Stochastic finite elements: a spectral approach.* Courier Corporation, North Chelmsford, Massachusetts, 2003.

[59] Graham, W. R., Peraire, J., and Tang, K. Y. Optimal control of vortex shedding using low-order models. Part I–open-loop model development. *International Journal for Numerical Methods in Engineering 44*, 7 (Mar. 1999), 945–972.

[60] Grimberg, S., Farhat, C., and Youkilis, N. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *Journal of Computational Physics 419* (Oct. 2020), 109681.

[61] Grimstad, B., et al. SPLINTER: a library for multivariate function approximation with splines. http://github.com/bgrimstad/splinter, 2015. Accessed: 2015-05-16.

[62] Gunzburger, M. D., Peterson, J. S., and Shadid, J. N. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Computer Methods in Applied Mechanics and Engineering 196*, 4-6 (Jan. 2007), 1030–1047.

[63] Gunzburger, M. D., Peterson, J. S., and Shadid, J. N. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Computer methods in applied mechanics and engineering 196*, 4 (2007), 1030–1047.

[64] Guo, M., and Hesthaven, J. S. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer Methods in Applied Mechanics and Engineering 341* (Nov 2018), 807–826.

[65] Guo, M., and Hesthaven, J. S. Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering 345* (Mar 2019), 75–99.

[66] Haasdonk, B., and Ohlberger, M. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In *Proc. 5th International Symposium on Finite Volumes for Complex Applications*, pp. 471–478.

[67] Haasdonk, B., and Ohlberger, M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *Mathematical Modelling and Numerical Analysis 42*, 2 (2008), 277–302.

[68] Haasdonk, B., and Ohlberger, M. Reduced basis method for explicit finite volume approximations of nonlinear conservation laws. American Mathematical Society, 2009, pp. 605–614.

[69] Haasdonk, B., Ohlberger, M., and Rozza, G. A Reduced Basis Method for Evolution Schemes with Parameter-Dependent Explicit Operators. *ETNA, Electronic Transactions on Numerical Analysis 32* (2008), 145–161.

[70] Hesthaven, J., and Ubbiali, S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics 363* (Jun 2018), 55–78.

[71] Hesthaven, J. S., Rozza, G., and Stamm, B. Certified Reduced Basis Methods for Parametrized Partial Differential Equations. *Springer Briefs in Mathematics* (2016).

[72] Hijazi, S., Ali, S., Stabile, G., Ballarin, F., and Rozza, G. In *FEF 2017 Selected Contributions*, H. van Brummelen, A. Corsini, S. Perotto, and G. Rozza, Eds., vol. 132 of *Lecture Notes in Computational Science and Engineering.* Springer International Publishing, 2020, ch. The Effort of Increasing Reynolds Number in Projection-Based Reduced Order Methods: From Laminar to Turbulent Flows, pp. 245–264.

[73] Hijazi, S., Stabile, G., Mola, A., and Rozza, G. In *QUIET Selected Contributions*, M. D'Elia, M. Gunzburger, and G. Rozza, Eds., vol. 137 of *Lecture Notes in Computational Science and Engineering.* Springer International Publishing, 2020, ch. Non-intrusive Polynomial Chaos Method Applied to Full-Order and Reduced Problems in Computational Fluid Dynamics: A Comparison and Perspectives, pp. 217–240.

[74] Hijazi, S., Stabile, G., Mola, A., and Rozza, G. Data-driven POD-Galerkin reduced order model for turbulent flows. *Journal of Computational Physics 416* (Sept. 2020), 109513.

[75] Hosder, S., Walters, R., and Perez, R. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In *44th AIAA aerospace sciences meeting and exhibit* (2006), p. 891.

[76] Iliescu, T., and Wang, Z. Variational multiscale proper orthogonal decomposition: Navier–Stokes equations. *Numerical Methods for Partial Differential Equations 30*, 2 (Dec. 2013), 641–663.

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

[78] Ionita, A. C., and Antoulas, A. C. Data-driven parametrized model reduction in the loewner framework. *SIAM Journal on Scientific Computing 36*, 3 (Jan 2014), A984–A1007.

[79] Issa, R. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics 62*, 1 (1986), 40–65.

[80] Isukapalli, S. S. Uncertainty analysis of transport-transformation models. *Unpublished PhD diss. New Brunswick, NJ: Rutgers, The State University of New Jersey, Department of Chemical and Biochemical Engineering* (1999).

[81] Janon, A., Nodet, M., and Prieur, C. Certified reduced-basis solutions of viscous Burgers equation parametrized by initial and boundary values. *ESAIM: Mathematical Modelling and Numerical Analysis 47*, 2 (Jan. 2013), 317–348.

[82] Janya-Anurak, C. *Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems Using Bayesian Uncertainty Quantification Based on Generalized Polynomial Chaos*, vol. 31. KIT Scientific Publishing, 2017.

[83] Jasak, H. *Error analysis and estimation for the finite volume method with applications to fluid flows.* PhD thesis, Imperial College, University of London, 1996.

[84] Johnston, H., and Liu, J.-G. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics 199*, 1 (2004), 221 – 259.

[85] Jones, W., and Launder, B. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer 15*, 2 (Feb 1972), 301–314.

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

[87] Kalashnikova, I., and Barone, M. F. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *International Journal for Numerical Methods in Engineering 83*, 10 (2010), 1345–1375.

[88] Kalashnikova, I., and Barone, M. F. 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 (May 2012), 1337–1362.

[89] Karpouzas, G. K., Papoutsis–Kiachagias, E. M., Schumacher, T., de Villiers, E., Giannakoglou, K. C., and Othmer, C. Adjoint optimization for vehicle external aerodynamics. *International Journal of Automotive Engineering 7*, 1 (2016), 1–7.

[90] Kean, K., and Schneier, M. Error analysis of supremizer pressure recovery for POD based reduced-order models of the time-dependent navier–stokes equations. *SIAM Journal on Numerical Analysis 58*, 4 (Jan. 2020), 2235–2264.

[91] Khalak, A., and Williamson, C. Motions, forces and mode transitions in vortex-induced vibrations at low mass-damping. *Journal of fluids and Structures 13*, 7 (1999), 813–851.

[92] Kolmogorov, A. N. Equations of turbulent motion of an incompressible fluid. *Izvestia Academy of Sciences, USSR 6*, 1,2 (1942), 55–58.

[93] Kunisch, K., and Volkwein, S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis 40*, 2 (2002), 492–515.

[94] Landahl, M. T., and Mollo-Christensen, E. *Turbulence and Random Processes in Fluid Mechanics.* Cambridge University Press, Cambridge, Sept. 1992.

[95] Launder, B., and Spalding, D. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering 3*, 2 (Mar. 1974), 269–289.

[96] Lazzaro, D., and Montefusco, L. B. Radial basis functions for the multivariate interpolation of large scattered data sets. *Journal of Computational and Applied Mathematics 140*, 1-2 (Mar 2002), 521–536.

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

[98] LeGresley, P., and Alonso, J. Investigation of non-linear projection for POD based reduced order models for aerodynamics. In *39th Aerospace Sciences Meeting and Exhibit* (Jan. 2001), American Institute of Aeronautics and Astronautics, Reno, NV, U.S.A.

[99] Liu, J.-G., Liu, J., and Pego, R. L. Stable and accurate pressure approximation for unsteady incompressible viscous flow. *Journal of Computational Physics 229*, 9 (May 2010), 3428–3453.

[100] Loève, M. *Probability Theory I.* Springer New York, 1977.

[101] Loiseau, J.-C., and Brunton, S. L. Constrained sparse Galerkin regression. *Journal of Fluid Mechanics 838* (Jan 2018), 42–67.

[102] Lombardi, M., Parolini, N., Quarteroni, A., and Rozza, G. Numerical simulation of sailing boats: Dynamics, FSI, and shape optimization. In *Springer Optimization and Its Applications*. Springer US, 2012, pp. 339–377.

[103] Lorenzi, S., Cammi, A., Luzzi, L., and Rozza, G. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Computer Methods in Applied Mechanics and Engineering 311* (Nov 2016), 151–179.

[104] Menter, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal 32*, 8 (Aug 1994), 1598–1605.

[105] Micchelli, C. A. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation 2*, 1 (Dec 1986), 11–22.

[106] Mohebujjaman, M., Rebholz, L. G., Xie, X., and Iliescu, T. Energy balance and mass conservation in reduced order models of fluid flows. *Journal of Computational Physics 346* (Oct. 2017), 262–277.

[107] Moin, P., and Mahesh, K. Direct Numerical Simulation: A tool in turbulence research. *Annual Review of Fluid Mechanics 30*, 1 (Jan 1998), 539–578.

[108] Montazeri, H., and Blocken, B. CFD simulation of wind-induced pressure coefficients on buildings with and without balconies: Validation and sensitivity analysis. *Building and Environment 60* (Feb. 2013), 137–149.

[109] Mou, C., Koc, B., San, O., and Iliescu, T. Data-driven variational multiscale reduced order models. *ArXiv abs/2002.06457* (2020).

[110] Moukalled, F., Mangani, L., and Darwish, M. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*, 1st ed. Springer Publishing Company, New York, Incorporated, 2015.

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

[112] Noack, B. R., and Cordier, L. Low-dimensional Galerkin model of a laminar shear-layer. *Tech. Rep.2002-01* (2002).

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

[114] Noack, B. R., and Eckelmann, H. A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder. *Physics of Fluids 6*, 1 (1994), 124–143.

[115] Noack, B. R., Papas, P., and Monkewitz, P. A. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics 523* (01 2005), 339–365.

[116] Östh, J., Noack, B. R., Krajnović, S., Barros, D., and Borée, J. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *Journal of Fluid Mechanics 747* (Apr. 2014), 518–544.

[117] Patankar, S., and Spalding, D. 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.

[118] Peherstorfer, B., and Willcox, K. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering 291* (Jul 2015), 21–41.

[119] Prandtl, L. *Uber die ausgebildete Turbulenz*, vol. 5. ZAMM, 1925.

[120] Quarteroni, A., Manzoni, A., and Negri, F. *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing, New York, 2016.

[121] Reagana, M. T., Najm, H. N., Ghanem, R. G., and Knio, O. M. Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection. *Combustion and Flame 132*, 3 (2003), 545–555.

[122] Reynolds, O. IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. (A.) 186* (Dec. 1895), 123–164.

[123] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics 641* (Nov 2009), 115.

[124] Rubino, S. Numerical analysis of a projection-based stabilized POD-ROM for incompressible flows. *SIAM Journal on Numerical Analysis 58*, 4 (Jan. 2020), 2019–2058.

[125] Saffman, P. G. A model for inhomogeneous turbulent flow. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 317*, 1530 (June 1970), 417–433.

[126] Sagaut, P. *Large Eddy Simulation for Incompressible Flows.* Springer Science & Business Media, Berlin Heidelberg, 2006.

[127] Salmoiraghi, F., Scardigli, A., Telib, H., and Rozza, G. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics 32*, 4-5 (May 2018), 233–247.

[128] Sarkar, S., Witteveen, J., Loeven, A., and Bijl, H. Effect of uncertainty on the bifurcation behavior of pitching airfoil stall flutter. *Journal of Fluids and Structures 25*, 2 (feb 2009), 304–320.

[129] Schäfer, M., and Turek, S. The benchmark problem 'flow around a cylinder'flow simulation with high performance computers ii. *Notes on Numerical Fluid Mechanics 52* (1996), 547–566.

[130] Schmid, P. J. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics 656* (Jul 2010), 5–28.

[131] Sirisup, S., and Karniadakis, G. Stability and accuracy of periodic flow solutions obtained by a POD-penalty method. *Physica D: Nonlinear Phenomena 202*, 3-4 (Mar. 2005), 218–237.

[132] Sirovich, L. Turbulence and the dynamics of coherent structures. II. symmetries and transformations. *Quarterly of Applied Mathematics 45*, 3 (Oct 1987), 573–582.

[133] Sirovich, L. Turbulence and the Dynamics of Coherent Structures part I: Coherent Structures. *Quarterly of Applied Mathematics 45*, 3 (1987), 561–571.

[134] Smagorinsky, J. General Circulation Experiments with the Primitive Equations . *Monthly Weather Review 91*, 3 (Mar. 1963), 99–164.

[135] Spalart, P., and Allmaras, S. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit* (Jan 1992), American Institute of Aeronautics and Astronautics.

[136] Stabile, G., Ballarin, F., Zuccarino, G., and Rozza, G. A reduced order variational multiscale approach for turbulent flows. *Advances in Computational Mathematics 45*, 5-6 (June 2019), 2349–2368.

[137] Stabile, G., Hijazi, S., Mola, A., Lorenzi, S., and Rozza, G. 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 (Dec 2017), 210–236.

[138] Stabile, G., and Rozza, G. ITHACA-FV - In real Time Highly Advanced Computational Applications for Finite Volumes. http://www.mathlab.sissa.it/ithaca-fv. Accessed: 2018-01-30.

[139] Stabile, G., and Rozza, G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids 173* (Sep 2018), 273–284.

[140] Stabile, G., Zancanaro, M., and Rozza, G. Efficient geometrical parametrization for finite-volume-based reduced order methods. *International Journal for Numerical Methods in Engineering 121*, 12 (Feb. 2020), 2655–2682.

[141] Star, K., Stabile, G., Georgaka, S., Belloni, F., Rozza, G., and Degroote, J. POD-Galerkin reduced order model of the Boussinesq approximation for buoyancy-driven enclosed flows. In *Building theory and applications : proceedings of M&C 2019* (2019), American Nuclear Society (ANS), Illinois, pp. 2452–2461.

[142] Star, S. K., Stabile, G., Belloni, F., Rozza, G., and Degroote, J. Extension and comparison of techniques to enforce boundary conditions in finite volume pod-galerkin reduced order models for fluid dynamic problems. *ArXiv abs/1912.00825* (2019).

[143] Stein, M. Large sample properties of simulations using latin hypercube sampling. *Technometrics 29*, 2 (May 1987), 143–151.

[144] Strouhal, V. Ueber eine besondere art der tonerregung. *Annalen der Physik und Chemie 241*, 10 (1878), 216–251.

[145] Tallet, A., Allery, C., Leblond, C., and Liberge, E. 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 (May 2015), 909–932.

[146] Tezzele, M., Demo, N., Gadalla, M., Mola, A., and Rozza, G. In *Technology and Science for the Ships of the Future*, A. Marinò and V. Bucci, Eds., vol. 0. IOS Press, Amsterdam, The Netherlands, 2018, ch. Model Order Reduction by Means of Active Subspaces and Dynamic Mode Decomposition for Parametric Hull Shape Design Hydrodynamics, p. 569–576.

[147] Ullman, S., and Lang, J. A POD-Galerkin Reduced Model with Updated Coefficients for Smagorinsky LES.

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

[149] Volkwein, S. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz 4*, 4 (2013).

[150] Walton, S., Hassan, O., and Morgan, K. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Applied Mathematical Modelling 37*, 20-21 (Nov. 2013), 8930–8945.

[151] Wang, Y., Yu, B., Cao, Z., Zou, W., and Yu, G. 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 (Aug 2012), 4827–4836.

[152] Wang, Z., Akhtar, I., Borggaard, J., and Iliescu, T. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *Journal of Computational Physics 230*, 1 (Jan. 2011), 126–146.

[153] Wang, Z., Akhtar, I., Borggaard, J., and Iliescu, T. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods in Applied Mechanics and Engineering 237–240* (2012), 10 – 26.

[154] Washabaugh, K. M., Zahr, M. J., and Farhat, C. On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. In *54th AIAA Aerospace Sciences Meeting* (Jan. 2016), American Institute of Aeronautics and Astronautics.

[155] Wiener, N. The homogeneous chaos. *American Journal of Mathematics 60*, 4 (1938), 897–936.

[156] Wilcox, D. *Turbulence Modeling for CFD*. No. v. 1 in Turbulence Modeling for CFD. DCW Industries, La Canada, California, U.S.A, 2006.

[157] Wilcox, D. C. Numerical study of separated turbulent flows. In *7th Fluid and PlasmaDynamics Conference* (June 1974), American Institute of Aeronautics and Astronautics.

[158] Wilcox, D. C. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal 26*, 11 (Nov. 1988), 1299–1310.

[159] Xiao, D., Fang, F., Buchan, A., Pain, C., Navon, I., Du, J., and Hu, G. Non linear model reduction for the Navier Stokes equations using residual DEIM method. *Journal of Computational Physics 263* (2014), 1 – 18.

[160] Xie, X., Mohebujjaman, M., Rebholz, L. G., and Iliescu, T. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing 40*, 3 (Jan 2018), B834–B857.

[161] Zahr, M. J., and Farhat, C. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *International Journal for Numerical Methods in Engineering 102*, 5 (Dec. 2014), 1111–1135.

[162] Zdravkovich, M. M. *Flow around Circular Cylinders: Volume 1: Fundamentals*, vol. 350. Cambridge University Press, 1997.

[163] Zdravkovich, M. M. *Flow around Circular Cylinders: Volume 2: Applications*, vol. 2. Oxford University Press, 2003.

[164] Zimmermann, R., and Görtz, S. Non-linear reduced order models for steady aerodynamics. *Procedia Computer Science 1*, 1 (May 2010), 165–174.