

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

**Data Enhanced Reduced Order Methods
for Turbulent Flows**



Advisors

Prof. Claudio Canuto
Prof. Gianluigi Rozza

Co-Advisors

Dr. Giovanni Stabile
Dr. Andrea Mola

Candidate

Anna Ivagnes

Anno Accademico 2020-2021

*To my aunt Anna and my uncle Totò.
To my cousin Francesco,
you will always be sculpted in my heart.*

Summary

This Thesis focuses on the combination of reduced order models and data-driven techniques applied to the study of turbulent flows in order to improve the pressure and velocity accuracy of standard reduced order methods. The full order model is based on the incompressible Navier–Stokes equations; the reduced order model is constructed by means of Proper Orthogonal Decomposition with Galerkin approach. The available data are used to construct different correction/closure terms, which are added to the reduced equations in order to model the interaction between the resolved and the unresolved scales.

Both supremizer enrichment approach and Poisson equation approach have been considered for pressure treatment at the reduced level. The effect of the data-driven correction terms on both resolution systems has been studied and compared.

The numerical investigation of the turbulent flow past a circular cylinder at $Re = 50000$ shows that our method yields significantly more accurate velocity and pressure approximations than the standard reduced order method. In particular, the velocity correction introduced in the momentum equation improves both the velocity and pressure fields. In addition, the pressure corrections developed, which — to the best of my knowledge — have been introduced here for the first time, further improve the pressure accuracy.

Acknowledgements

First of all, I would like to start by thanking my advisor, Prof. Canuto, for proposing me the internship and Thesis in collaboration with the SISSA *mathLab* group and for believing in my capacities.

I really want to thank Prof. Rozza for giving me the opportunity of being part of his beautiful research group. He has always been available to listen to my doubts and to share with me his opinions and suggestions. He has shown an incredible ability to integrate humanity and working life and he has demonstrated to be sensitive to the problems of his students.

A sincere thank you goes to my co-advisors Dr. Giovanni Stabile and Dr. Andrea Mola, who were available literally every day to discuss with me the results of the work and to give me suggestions and hints for this Thesis. I really thank them also for having encouraged and supported me in the months I passed in SISSA.

I also want to thank Prof. Iliescu, from Virginia Tech, for the advises and suggestions he gave us during the months of Thesis production.

I thank all my colleagues in SISSA for immediately integrating me in the group and for making me feel at home. A huge thank you goes in particular to Martina, thank you for being that friend with whom I can literally share everything, just like we have always known each other. Thank you also Pierfrancesco, my colleague and friend in the Thesis production — and in the after work evenings, too.

I would like to thank all my friends for never making me feel alone, even if we were 1 thousand kilometers apart. Thank you to the friends of a lifetime, Gabriella, Maria Celeste, Chiara and Anna Lu, I know I can count on you everywhere and every time. Thank you Sara, the Opossum one, for being always by my side in the hilarious and in the sad moments, you will always remain the best flatmate and companion of adventures. Thank you to all my bachelor friends for sincerely supporting me out of any competition and envy, I miss all of you and the time spent in Lecce.

Thank you Andrea, for being one of few standing points of my life. Thank you for being my boyfriend and my best friend at the same time; thank you also for supporting me with enthusiasm in all my choices, even if sometimes they mean a physical distance from you. Thank you finally for introducing in my everyday life moments of lightness and frivolity.

A thank you goes to my big sister Sara, who has always been my biggest supporter in her personal way. I can feel your closeness even if you're far from me, and I can hear your life advises even if you don't say anything.

I would like to thank my parents, with whom I spent every moment of every day in the period of the pandemic, which coincides with most of my master courses. It was not a good period, but after all we came out strengthened from it. Thank you for giving me suggestions without enforcing anything, and always supporting me in every choice I make.

Finally, I want to dedicate my Thesis to my uncle Totò and my aunt Anna, the strongest people I've ever known. I think of you and Francesco every day, he will always be a piece of my heart.

Anna Ivagnes, Trieste, July 2021

Contents

List of Tables	9
List of Figures	10
Introduction	17
1 Full Order Model: incompressible Navier–Stokes Equations	21
1.1 Finite Volume Discretization for the NSE	22
1.1.1 The pressure-based solver for the incompressible NSE	23
1.2 Reynolds Averaged Navier–Stokes (RANS) approach	24
1.2.1 Eddy Viscosity Models	26
2 Reduced Order Methods for the NSE	29
2.1 The Proper Orthogonal Decomposition	29
2.2 POD-Galerkin Reduced Order Methods	30
2.2.1 Reduced Order Method with Supremizer Approach (SUP-ROM)	31
2.2.2 Reduced Order Method with Poisson Equation for Pressure (PPE-ROM)	32
2.2.3 Boundary Conditions Treatment	33
2.3 Turbulence Modelling	34
3 A Novel Approach based on Data-Driven Reduced Order Methods (DD-ROMs)	37
3.1 Data-driven velocity corrections in the SUP-ROM approach	39
3.1.1 Data-driven correction for velocity	40
3.1.2 Constrained data-driven correction for velocity	42
3.1.3 Three-Scales Data-Driven Variational Multiscale ROMs (3S-DD-VMS-ROM)	43
3.1.4 Data-driven corrections for pressure	45
3.2 Data-driven corrections for the PPE-ROM approach	46
3.2.1 Data-driven correction for term \mathbf{Db}	47
3.2.2 Data-driven correction for term $\mathbf{a}^T \mathbf{Ga}$	50
3.2.3 Combined data-driven corrections proposals	50

4	Numerical Results	55
4.1	The Case Study: Unsteady Flow around a Circular Cylinder	55
4.2	Modal Decomposition Effectiveness	59
4.3	Analysis of the POD Galerkin SUP-ROM without corrections	60
4.4	Effect of velocity correction in the SUP-ROM approach	63
4.4.1	Velocity correction: the influence of R	63
4.4.2	Velocity correction: constrained and not constrained	67
4.4.3	Velocity correction: extrapolation efficiency	71
4.4.4	Velocity correction: the 3S-DD-VMS-ROM approach	74
4.5	Effect of pressure corrections in the SUP-ROM approach	75
4.6	Analysis of the POD Galerkin PPE-ROM without corrections	77
4.7	Effect of corrections in the PPE-ROM approach	79
4.7.1	Evaluation of the correction term τ_D	79
4.7.2	Effect of pressure corrections in the PPE-ROM	80
4.7.3	Combined effect of velocity and pressure corrections in the PPE-ROM	82
4.8	Comparison: the turbulence modelling, the data-driven corrections and the combined effect	84
4.8.1	Turbulence inclusion in the SUP-ROM approach	84
4.8.2	Turbulence inclusion in the PPE-ROM approach	87
4.8.3	Combined effect of turbulence modelling and correction terms for different mode regimes	88
4.8.4	Graphical results	91
4.8.5	Analysis of the computational cost of simulations	97
5	Conclusions and Outlooks	99

List of Tables

- 4.1 Optimal number of singular values retained in the truncated SVD, varying the number of modes considered in online simulations. Both optimization problems for the standard and constrained case are taken into account. The dynamical system is evolved according to an implicit Euler time scheme or to a second order implicit time scheme. 66
- 4.2 Computational time evaluated for some simulations carried out making use of the supremizer approach. 97
- 4.3 Computational time evaluated for some simulations carried out making use of a Poisson approach. 97

List of Figures

4.1	(a) The mesh used in simulations. (b) The mesh zoomed around the cylinder. Image taken from [1].	56
4.2	Decay of the eigenvalues corresponding to velocity, pressure and supremizer modes.	60
4.3	Percentage reconstruction errors for different numbers of modes.	60
4.4	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p = N_{sup}$. In simulations time integration is carried out by means of a first order time scheme.	61
4.5	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$ and $N_{sup} > N_p$. In simulations time integration is carried out by means of a first order time scheme.	61
4.6	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p = N_{sup}$. In simulations time is evolved with a second order time scheme.	62
4.7	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$ and $N_{sup} > N_p$. In simulations time is evolved with a second order time scheme.	62
4.8	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. The correction term included is the unconstrained velocity correction.	64
4.9	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. The correction term included is the constrained velocity correction.	64
4.10	Variation of the metric $\varepsilon_u(L^2)$ varying the number of singular values retained in the singular value decomposition.	65
4.11	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$, including the simulation with the exact correction. The optimal R and R_c are considered.	65

4.12	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity constrained and not.	67
4.13	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not.	67
4.14	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 10$ and $N_{sup} = 15$. Results without any correction term, with the correction term for velocity constrained and not.	68
4.15	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the standard velocity correction is added to the dynamical system and the time is evolved according to a first order implicit scheme.	68
4.16	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the constrained velocity correction is added to the dynamical system and the time is evolved according to an Euler implicit scheme.	69
4.17	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the standard velocity correction is added to the dynamical system and the time is evolved according to a second order implicit scheme.	69
4.18	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the constrained velocity correction is added to the dynamical system and the time is evolved according to a second order implicit scheme.	70
4.19	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the standard and constrained correction term for velocity. The optimal R and R_c are both 5.	71
4.20	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 9 and 4 respectively.	72
4.21	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 3 and 4, respectively.	73

4.22	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 9 and 6, respectively.	73
4.23	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 10$, $N_{sup} = 15$. Results without any correction term, with the correction term for velocity not constrained and with the correction term using 2 scales for resolved ROM are showed.	74
4.24	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity only, with both the correction terms for velocity and pressure are displayed.	75
4.25	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results with and without the exact pressure corrections are presented.	76
4.26	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$. In simulations time is evolved with a first order time scheme.	78
4.27	Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$. In simulations time is evolved with a second order time scheme.	78
4.28	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results are displayed for the cases: without any correction (\ominus), with the exact correction term τ_D (∇), with the linear ansatz (\diamond) and the quadratic ansatz (\oplus) for the correction τ_D . The previous cases are compared with the reconstruction error (\diamond).	79
4.29	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections (\ominus); with the correction τ_D (∇); with the correction τ_G (\diamond); with both τ_D and τ_G found from two disjoint least squares problems (\oplus), in Case 1 and 2 of section 3.2.3 (\boxminus and \star , respectively). Results are compared with the reconstruction errors, referred to the projected fields (\diamond).	81
4.30	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections (\ominus); with the velocity correction τ_u (∇); with the corrections τ_u and τ_D (\diamond); with τ_u and τ_G (\oplus), with τ_u , τ_D and τ_G in Case 3 of section 3.2.3 (\boxminus). Results are compared with the reconstruction errors, referred to the projected fields (\diamond).	82

4.31	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections (\ominus); with τ_u , τ_D and τ_G disjoint (∇); with the velocity and pressure corrections in Cases 1, 2 and 3 of section 3.2.3 (\diamond , \ominus and \boxminus , respectively). Results are compared with the reconstruction errors, referred to the projected fields (\diamond).	83
4.32	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections (\ominus); with τ_u , τ_D and τ_G disjoint (∇); with the velocity and pressure corrections in Cases 1, 2 and 3 of section 3.2.3 (\diamond , \ominus and \star , respectively). Results are compared with the reconstruction errors, referred to the projected fields (\diamond).	84
4.33	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$ and a first order time scheme. Results include the following cases: without any data-driven term (\ominus); with velocity constrained correction (∇); with turbulence term (\diamond); with both velocity correction and turbulence term (\ominus); projection (\diamond).	85
4.34	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$ and a second order time scheme. Results include the following cases: without any data-driven term (\ominus); with velocity constrained correction (∇); with turbulence term (\diamond); with both velocity correction and turbulence term (\ominus); projection (\diamond).	86
4.35	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 5$ and a first order time scheme. Results include the following cases: without any data-driven term (\ominus); with all corrections (∇); with turbulence term (\diamond); with corrections and turbulence term (\ominus); projection (\diamond).	87
4.36	Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 5$ and a second order time scheme. Results include the following cases: without any data-driven term (\ominus); with all corrections (∇); with turbulence term (\diamond); with corrections and turbulence term (\ominus); projection (\diamond).	87
4.37	Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the SUP-ROM with a first order time scheme. The cases represented are the following: without any data-driven term (\ominus); with turbulence modelling, without corrections (∇); without turbulence modelling, with standard and constrained correction (\diamond and \ominus , respectively); with both turbulence modelling and constrained correction (\boxminus); projection (\diamond).	89

4.38	Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the SUP-ROM with a second order time scheme. The cases represented are the following: without any data-driven term (—○—); with turbulence modelling, without corrections (—▽—); without turbulence modelling, with standard and constrained correction (—◇— and —◇—, respectively); with both turbulence modelling and constrained correction (—⊞—); projection (—◇—).	90
4.39	Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the PPE-ROM with a first order time scheme. The cases represented are the following: without any data-driven term (—○—); with turbulence modelling, without corrections (—▽—); without turbulence modelling, with corrections (—◇—); with both turbulence modelling and corrections (—⊞—); projection (—◇—).	90
4.40	Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the PPE-ROM with a second order time scheme. The cases represented are the following: without any data-driven term (—○—); with turbulence modelling, without corrections (—▽—); without turbulence modelling, with corrections (—◇—); with both turbulence modelling and corrections (—⊞—); projection (—◇—).	91
4.41	Representation of the pressure field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.	93
4.42	Representation of the velocity magnitude field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.	94
4.43	Representation of the absolute errors on the pressure field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.	95
4.44	Representation of the absolute errors on the velocity magnitude field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.	96

Introduction

The present Master Thesis investigates the application of data-driven techniques to projection - based Reduced Order Methods for the incompressible Navier–Stokes Equations. The main goal is that of improving the accuracy of the reduced approximation of velocity and pressure fields making use of data-driven terms in the governing equations of the online problem.

The project of the Thesis has been carried out in collaboration with *Scuola Internazionale Superiore di Studi Avanzati (SISSA)* of Trieste and it is inserted in the *Accurate ROMs for Industrial Applications (ARIA)* framework. The work is also related to the project ERC *AROMA-CFD (European Research Council: Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics)* and to the MIUR PRIN project *NA_FROM-PDEs (Numerical Analysis of Full and Reduced Order Methods for Partial Differential Equations)*.

The Full Order fluid dynamic model considered in this Thesis is the incompressible Reynolds-Averaged Navier–Stokes Equations. As well-known, the resolution of the numerical problem associated to the incompressible NSE at the Full Order level is particularly expensive in terms of computational cost and CPU time. Instead, Reduced Order Models exploit information obtained from a series of full order simulations carried out in a typically expensive *offline phase*, to set up a lower dimensional problem that can be solved in a limited amount of time with minor computational effort in an *online phase*. Based on such an offline/online paradigm, the Reduced Order Methods (ROMs) have been widely applied to fluid flow simulations to reduce the computational cost of the resulting parametric numerical simulations in the context of many-query problems, such as in the recent works [2, 3].

In particular, this Thesis focuses on ROMs constructed with a POD-Galerkin method (POD-Galerkin ROMs) [4, 5, 6, 7]. These ROMs are built by using the general Galerkin framework: compute a set of basis functions $\{\phi_1, \dots, \phi_r\}$, $\{\chi_1, \dots, \chi_q\}$ (modes), identified applying the Proper Orthogonal Decomposition to a series of solutions (or *snapshots*) of the Full Order problem; reconstruct the unknown solution as a linear combination of these modes, $\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^r a_i(t) \phi_i(\mathbf{x})$, $p(\mathbf{x}, t) = \sum_{i=1}^q b_i(t) \chi_i(\mathbf{x})$; project the FOM governing equations onto the space spanned by these modes. The resulting Galerkin ROM (G-ROM) is a system of equations in which the unknowns are the coefficients in the linear combination ansatzes used above. The main difference between the ROMs and the classical Galerkin methods is that the ROM basis is a global data-driven basis — since it is constructed from the available numerical data obtained from the Full Order simulations

— whereas the classical Galerkin methods do not generally use data to build the basis. Since the ROM basis exploits knowledge of prior solutions of the problem, it allows in most applications for good accuracy with a significant reduction of the resolution system degrees of freedom.

Although ROMs have been already successfully used in numerous fluid flow applications, there are significant challenges in the development of fast, stable and robust ROMs engineering for CFD problems. In particular, in the numerical simulation of turbulent flows a standard ROM approximation could require hundreds of ROM modes, which would significantly increase the ROM computational cost. To ensure a low ROM computational cost, ROMs are generally built with relatively few basis functions. Yet, in presence of turbulence the standard G-ROM approach would yield inaccurate approximations. One popular approach for increasing the accuracy of the standard G-ROM in high Reynolds number regime is to add an extra term to the reduced equations, named *correction/closure* term. The same terminology has been used in classical Large Eddy Simulation (LES) [8, 9].

The role of the ROM closure term is to model the effect of the discarded ROM modes, $\{\phi_{r+1}, \dots\}$, $\{\chi_{q+1}, \dots\}$, on the ROM dynamics.

There are three main approaches to model the correction/closure term at the reduced order level:

- Functional modelling, in which physical insight is used to build the model.
- Structural modelling, in which mathematical methods (e.g., expansions and asymptotics) are used to build the model.
- Data-driven modelling, in which available numerical or experimental data is used to build the model.

The present Thesis exclusively focuses on data-driven modelling of the correction/closure terms, and it can be considered the expansion of previous works, such as [1], [10] and [11].

The past work by Hijazi et al. [1] focused on an hybrid data-driven approach, with the introduction of a data-driven turbulence term in the equations of a POD-Galerkin reduced model. Different and totally new data-driven correction terms are developed in this Thesis, taking inspiration from works proposed by the group of Prof. Traian Iliescu, such as [10, 11, 12]. Such previous literature, only focused on Large Eddy Simulation equations and only included a correction for the nonlinear term related to the momentum equation. In the context of the present Thesis, data-driven correction/closure terms are instead applied to RANS simulation, and are extended to also improve the pressure field approximation.

The case study considered to test the efficiency of the approach developed is that of the unsteady turbulent flow around a cross-flow circular cylinder. The reduced order model is treated with two different pressure/velocity coupling techniques, namely the supremizer enrichment approach (SUP-ROM) and the Poisson approach (PPE-ROM). The first technique consists in the addition of velocity modes — called supremizer modes — in the velocity POD space in order to fulfill the *inf-sup* condition [13, 14]; the second

approach is based on replacing the continuity equation with the Poisson equation for pressure [8, 15].

As mentioned, in the framework of the ROM-Galerkin models considered, the reduced approximation of velocity and pressure fields is obtained through an offline-online procedure applied to the case study of the turbulent flow past a circular cylinder.

In this work, the offline phase Full Order solutions are computed making use of the C++ open source software *OpenFOAM* [16]. POD is then applied to extract modal information from the previously computed snapshots, making use of the functions of *ITHACA-FV* (In real Time Highly Advanced Computational Applications for Finite Volumes), a C++ library based on the finite volume solver *OpenFOAM* and developed at the *mathLab* group in SISSA ([17, 18, 19]). Finally, a specified *Python* software has been developed for the online stage of the procedure. Here, the reduced equations are solved both in the standard SUP-ROM and PPE-ROM approaches. In this work, several different data-driven corrections are added to the reduced systems for both formulations so as to evaluate their effect in terms of reduced fields accuracy, measured by means of relative errors of the reduced velocity and pressure fields with respect to their FOM counterparts.

The data-driven correction methods by Iliescu et al. ([10, 11, 12]), here applied for the first time to RANS equations model, result in an improvement especially in the approximation of the velocity field. Unfortunately, the same is not observed for pressure. This motivation has led to the formulation of new data-driven pressure correction terms at the reduced level, which are able to significantly increase the reduced pressure field accuracy. Finally, the data-driven approach developed in the present work is also first compared and then coupled with the turbulence modelling approach developed by Hijazi et al. in [1].

The present Thesis has the following structure:

- Chapter 1: Full Order Model for the incompressible NSE. In this Chapter, the general Full Order Model and the finite volume discretization is described. The Reynolds Averaged Navier–Stokes approach (RANS) and the eddy viscosity turbulence models are then presented.
- Chapter 2: Reduced Order Methods. The Reduced Order Method is here explained and performed on the Full Order system, and the methodology of the Proper Orthogonal Decomposition (POD) is explained. The two different approaches of SUP-ROM and PPE-ROM are then presented. The problem of the introduction of a turbulence model at the reduced order level is addressed.
- Chapter 3: The Novel Data-Driven approach applied to the Reduced Order Models. In this chapter the alternative data-driven formulation, the heart of the Thesis, is presented; all the methods used to find the correction/closure terms added to the reduced NSE are explained in detail.
- Chapter 4: Results. The case study used to test the methods is here presented in its physics and computational setting; the approximated fields for velocity and pressure are displayed and compared to those obtained in the Full Order Model and in the standard reduced order setting.

- Chapter 5: Conclusions and Outlooks. This Chapter retraces the logical passages followed in the numerical simulations carried out in the online phase.

Chapter 1

Full Order Model: incompressible Navier–Stokes Equations

In the present Chapter, the Full Order Model is formulated starting from the Navier–Stokes Equations (NSE) for incompressible flows. We consider the fluid domain $\Omega \in \mathbb{R}^d$ with $d = 1, 2$ or 3 , Γ is defined as its boundary; $t \in [0, T]$ is the time variable, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the flow velocity vector field and $p = p(\mathbf{x}, t)$ is the normalized pressure scalar field divided by the fluid density, ν is the fluid kinematic viscosity. The strong form of the non-dimensionalized NSE is the following.

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \nabla p & \text{in } \Omega \times [0, T], \\ \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } \Omega \times [0, T], \\ + \text{boundary conditions} & \text{on } \Gamma \times [0, T], \\ + \text{initial conditions} & \text{in } (\Omega, 0). \end{array} \right. \quad (1.1)$$

In System (1.1), the first equation is the momentum equation, corresponding to a momentum conservation law. The acceleration of the fluid $\frac{\partial \mathbf{u}}{\partial t}$ is expressed as a combination of different contributions: the viscosity effect $\nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$, the convective non linear effect $-\nabla \cdot (\mathbf{u} \otimes \mathbf{u})$, the effect of the gradient of pressure $-\nabla p$, deriving from the divergence of the hydrostatic part of the Cauchy stress tensor.

The analytical solution for System (1.1) can be found just in specific ideal cases, otherwise the system is numerically solved. In order to distinguish different types of flows the adimensional Reynolds number is considered. As in (1.2), it is defined as a quotient between inertial effects, represented by the characteristic velocity U and the characteristic length L , and viscous effects, expressed by the kinematic viscosity ν .

$$Re = \frac{\rho UL}{\mu} = \frac{UL}{\nu}. \quad (1.2)$$

A threshold for the Reynolds number can be identified depending on the working problem; above this threshold the inertial effects are prevalent with respect to the viscous effects and the flow begins to show a turbulent behaviour.

A turbulent flow is typically characterized by the following properties:

- the formation of three-dimensional vortexes, belonging to different scales;
- a chaotic, unstable and irregular movement of the fluid particles, governed by statistical laws;
- a multiscale dissipative process, leading to an energy cascade from the highest to the smallest scales, called *Kolmogorov scales*.

1.1 Finite Volume Discretization for the NSE

In the computation of the full order solutions of Navier–Stokes Equations, the software *OpenFOAM* employs the finite volume method. The finite volume method ([20]) is a mathematical technique that converts the partial differential equations (the NSE in our case) defined on differential volumes in algebraic equations defined on finite volumes.

The first preliminary step of this method is a polyhedral discretization of the domain, in order to define the finite control volumes. The second step is to consider the integral form of the NSE over each finite volume and to discretize them.

The momentum equation in integral form on a single control element of the domain V_i can be written in the following form:

$$\int_{V_i} \frac{\partial \mathbf{u}}{\partial t} dV + \int_{V_i} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) dV - \int_{V_i} \nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) dV + \int_{V_i} \nabla p dV = \mathbf{0}. \quad (1.3)$$

All the terms appearing in Equation (1.3) are transformed into integral surfaces according to the divergence theorem and are evaluated at the faces of the discretized volumes. We consider ∂V_i the set of all the boundaries of the control volume, \mathbf{S}_f the surface vector related to face f ; p_f and \mathbf{u}_f are the pressure and velocity fields evaluated at the centre of face f .

The pressure gradient in Equation (1.3) is evaluated in the following way:

$$\int_{V_i} \nabla p dV = \int_{\partial V_i} p d\mathbf{S} \approx \sum_f \mathbf{S}_f p_f.$$

The convective term is evaluated as follows:

$$\int_{V_i} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) dV = \int_{\partial V_i} (\mathbf{u} \otimes \mathbf{u}) \cdot d\mathbf{S} \approx \sum_f \mathbf{u}_f \otimes \mathbf{u}_f \cdot \mathbf{S}_f = \sum_f (\mathbf{u}_f \cdot \mathbf{S}_f) \mathbf{u}_f = \sum_f F_f \mathbf{u}_f,$$

where F_f is the mass flux through the face f .

The diffusion term is discretized in the following way:

$$\int_{V_i} \nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) dV = \int_{\partial V_i} \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \cdot d\mathbf{S} \approx \sum_f \nu_f (\nabla \mathbf{u})_f \cdot \mathbf{S}_f, \quad (1.4)$$

where $(\nabla \mathbf{u})_f$ and ν_f are the gradient of the velocity field and the viscosity evaluated at the center of face f , respectively. In particular the scalar product in (1.4) is computed as:

$$(\nabla \mathbf{u})_f \cdot \mathbf{S}_f = |\mathbf{S}_f| \frac{\mathbf{u}_N - \mathbf{u}_P}{|\mathbf{d}|},$$

where \mathbf{u}_N and \mathbf{u}_P are the velocities evaluated at the centers of two adjacent volumes and \mathbf{d} is the vector connecting the centers of the cells [21].

1.1.1 The pressure-based solver for the incompressible NSE

The solver used in our simulations by *OpenFOAM* to solve the discretized system is *pimpleFoam* and it involves the PIMPLE algorithm. The PIMPLE algorithm [20] uses a segregated pressure-based approach and it is the combination of SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) [22] and PISO (Pressure Implicit with Splitting of Operators) [23] algorithms. For instance, the work [24] propose a reduced order model which is fully consistent with the SIMPLE algorithm.

First of all, the discretized system of the NSE can be written in matricial form in the following way:

$$\begin{bmatrix} [\mathbf{A}_u] & [\nabla(\cdot)] \\ [\nabla \cdot (\cdot)] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \quad (1.5)$$

where the matrix \mathbf{A}_u is defined such that:

$$\mathbf{A}_u \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T).$$

The momentum equation in (1.5) can be written as follows:

$$\mathbf{u} + [\mathbf{A}_u^{-1}][\nabla(\cdot)][p] = \mathbf{0}. \quad (1.6)$$

By taking the divergence of Equation (1.6), and considering that $[\nabla \cdot (\cdot)]\mathbf{u} = 0$, a Poisson equation for pressure is obtained:

$$[\nabla \cdot (\cdot)][\mathbf{A}_u^{-1}][\nabla(\cdot)][p] = 0.$$

Since the matrix \mathbf{A}_u can be dense and hard to be inverted, the following decomposition is introduced:

$$[\mathbf{A}_u] = [\mathbf{D}_u] + [\mathbf{L}\mathbf{U}_u],$$

where matrix $[\mathbf{D}_u]$ is a diagonal matrix, whereas the matrix $[\mathbf{L}\mathbf{U}_u]$ is the off-diagonal part of matrix $[\mathbf{A}_u]$. The resulting system is:

$$\begin{bmatrix} [\mathbf{D}_u] & [\nabla(\cdot)] \\ [\nabla \cdot (\cdot)] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} -[\mathbf{L}\mathbf{U}_u][\mathbf{u}] \\ 0 \end{bmatrix}.$$

The momentum equation can be written in a discretized version for each control volume as follows:

$$a_P^u \mathbf{u}_P = \mathbf{H}(\mathbf{u}) - \nabla p, \quad (1.7)$$

where: a_P^u is the vector of diagonal coefficients evaluated at a generic cell centre, $\mathbf{H}(\mathbf{u})$ contains the off-diagonal coefficients and any right hand side contributions.

The non-linearity contained in the convective term and included in term $\mathbf{H}(\mathbf{u})$ of equation (1.7) is solved in this case by considering $\mathbf{H}(\mathbf{u}_P^{N-1})$ at the r.h.s., in which \mathbf{u}_P^{N-1} is the solution computed at the previous time step.

The SIMPLE algorithm consists in the following steps:

1. start with an initial guess for pressure and velocity;
2. Momentum predictor step: solve the discretized momentum equation with the guessed pressure field;
3. Pressure correction step: solve the Poisson equation with the available velocity field and correct the pressure field;
4. repeat until convergence.

The PISO algorithm involves two correction steps for pressure without re-discretizing the momentum equation. In the first correction step, a conservative velocity field is obtained; in the second correction step, a more physical pressure field is obtained.

The PIMPLE algorithm is essentially a repeated PISO algorithm: the number of times the PISO procedure is iterated is the number of outer corrections of the PIMPLE algorithm.

1.2 Reynolds Averaged Navier–Stokes (RANS) approach

At the full order level the Navier–Stokes Equations are solved by the software *OpenFOAM* [16], following the discretization techniques and the pressure-based solver described in Section (1.1). Different models have been used in literature to characterize turbulent flows; in this Thesis we consider the RANS model (Reynolds Averaged Navier–Stokes equations).

The RANS model is based on the *Reynolds decomposition*, proposed for the first time by Osborne Reynolds [25]. The *Reynolds decomposition* arose from the observation that many properties in turbulent flows are characterized by random oscillations both in space and time. In particular, the decomposition considers each flow field as the sum of its mean and its fluctuating parts. Considering a generic field $\sigma(\mathbf{x}, t)$, with $\mathbf{x} = (x, y, z)$, the Reynolds decomposition can be written as:

$$\sigma(\mathbf{x}, t) = \bar{\sigma}(\mathbf{x}, t) + \sigma'(\mathbf{x}, t),$$

where $\bar{\sigma}(\mathbf{x}, t)$ is the mean field. In the case of steady flows the mean flow is just dependent on space:

$$\bar{\sigma}(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \sigma(\mathbf{x}, s) ds. \quad (1.8)$$

The expression (1.8) can be written since in steady flows the scales of the mean field and of the fluctuating field are far between each other. In the case of an unsteady flow, both scales (named T_1 and T_2) have to be considered in the definition of the mean field. In the unsteady case the mean field is dependent on both space and time and it is defined as follows.

$$\bar{\sigma}(\mathbf{x}, t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \sigma(\mathbf{x}, s) ds, \quad T_1 \leq T \leq T_2.$$

In the RANS model the Navier–Stokes equations are rewritten just taking the Reynolds time-average of the equations.

First of all, the NSE in (1.1) are written using the Einstein notation:

$$\begin{cases} \frac{\partial u_i}{\partial x_i} = 0, \\ \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + 2\nu \frac{\partial \mathbf{E}_{ij}}{\partial x_j}, \end{cases} \quad (1.9)$$

where \mathbf{E} is the strain rate tensor, describing the rate of stretching and shearing, and it is defined as:

$$\mathbf{E}_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

System (1.9) is rewritten, taking into account the following properties:

$$\begin{aligned} \frac{\partial \overline{u'_i}}{\partial x_i} &= 0, \\ \overline{u_i u_j} &= \overline{u_i} \overline{u_j} + \overline{u'_i u'_j}. \end{aligned} \quad (1.10)$$

From the first property in (1.10), the continuity equation in (1.9) is written as follows:

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0. \quad (1.11)$$

The convective term is expressed as follows:

$$u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial (u_i u_j)}{\partial x_j} - u_i \frac{\partial u_j}{\partial x_j} = \frac{\partial (u_i u_j)}{\partial x_j}.$$

By replacing the convective term in the momentum equation in (1.9), the new momentum equation is:

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + 2\nu \frac{\partial \mathbf{E}_{ij}}{\partial x_j}. \quad (1.12)$$

Taking the time-average of equation (1.12), the following momentum equation is obtained:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial(\bar{u}_i \bar{u}_j + \overline{u'_i u'_j})}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + 2\nu \frac{\partial \bar{\mathbf{E}}_{ij}}{\partial x_j}, \quad (1.13)$$

where the averaged strain rate tensor becomes:

$$\bar{\mathbf{E}}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right).$$

Considering the equality (1.11), we also have:

$$\frac{\partial(\bar{u}_i \bar{u}_j)}{\partial x_j} = \bar{u}_i \frac{\partial \bar{u}_j}{\partial x_j} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j}.$$

Then the momentum equation in (1.13) can be finally written as follows:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial(2\nu \bar{\mathbf{E}}_{ij} - \mathcal{R}_{ij})}{\partial x_j}, \quad (1.14)$$

where $\mathcal{R}_{ij} = \overline{u'_i u'_j}$ is the Reynolds stress tensor. Equations (1.11) and (1.14) form the RANS equations. The Reynolds stress tensor is a feature of the turbulent flows and it is a symmetric tensor. Since the Reynolds stress tensor adds 6 additional unknowns to the RANS system, it is not a closed system. The main aim of the introduction of a turbulence modelling is the one of finding a closure model for the system. This is done by adding equations which define the additional unknowns of the system.

1.2.1 Eddy Viscosity Models

In the full order simulation, the closure problem is solved by considering an *Eddy viscosity model*. This model is based on the Boussinesq hypothesis, stating that the Reynolds stress tensor is proportional to the strain rate tensor:

$$\mathcal{R}_{ij} = 2\nu_t \mathbf{E}_{ij} - \frac{2}{3} \kappa \delta_{ij}, \quad (1.15)$$

in which $\kappa = \frac{1}{2} \overline{u'_i u'_i}$ is the turbulent kinetic energy, ν_t is the eddy viscosity. The Eddy Viscosity Models are based on the assumption (1.15) and are aimed to find a closure model to the RANS equations. Two examples of eddy viscosity models are the $\kappa - \varepsilon$ and $\kappa - \omega$ models, where ε is the turbulent dissipation and ω is the specific turbulent dissipation rate and it is dependent on κ and ε . The definitions of ε and ω are here reported:

$$\varepsilon = \nu \overline{\frac{\partial u'_i}{\partial x_k} \frac{\partial u'_i}{\partial x_k}},$$

$$\omega = \frac{\varepsilon}{\kappa \beta^*},$$

where $\beta^* = 0.09$.

The $\kappa - \varepsilon$ turbulence model consists in adding two transport partial differential equations to the full order formulation, one for κ and one for ε , respectively; this model has been presented in [26]. In a similar way, the $\kappa - \omega$ model introduces two transport equations for κ and ω and it has been first proposed in [27]. The SST $\kappa - \omega$ model, presented in [28], is used in this Thesis and it combines the standard $\kappa - \omega$ and the $\kappa - \varepsilon$ models. The equations of the SST $\kappa - \omega$ model and all constants appearing are defined in [1]. The final formulation, including the RANS equations and the transport equations for κ and ω , is the following:

$$\left\{ \begin{array}{ll} \frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = \nabla \cdot [-\bar{p}\mathbf{I} + (\nu + \nu_t) (\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T)] & \text{in } \Omega \times [0, T], \\ \nabla \cdot \bar{\mathbf{u}} = 0 & \text{in } \Omega \times [0, T], \\ \nu_t = \frac{a_1 \kappa}{\max(a_1 \omega, \Omega_s F_2)} & \text{in } \Omega \times [0, T], \\ \frac{\partial \kappa}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \kappa = P - \beta^* \omega \kappa + \nabla \cdot [(\nu + \sigma_k \nu_t) \nabla \kappa] & \text{in } \Omega \times [0, T], \\ \frac{\partial \omega}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \omega = \alpha \Omega_s^2 - \beta \omega^2 + \nabla \cdot [(\nu + \sigma_\omega \nu_t) \nabla \omega] & \text{in } \Omega \times [0, T], \\ + 2(1 - F_1) \frac{\sigma_\omega^2}{\omega} \nabla \omega \cdot \nabla \kappa & \text{in } \Omega \times [0, T], \\ + \text{Boundary conditions} & \text{on } \Gamma \times [0, T], \\ + \text{Initial conditions} & \text{in } (\Omega, 0), \end{array} \right. \quad (1.16)$$

where the following constants appear:

$$\begin{aligned} P &= \min \left(\mathcal{R}_{ij} \frac{\partial u_i}{\partial x_j}, 10\beta^* \omega \kappa \right), \quad \Omega_s = \sqrt{2E_{ij}E_{ij}}, \\ F_1 &= \tanh \arg_1^4, \quad \arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\sigma_\omega k}{\text{CD}_{\kappa\omega} y^2} \right], \\ \text{CD}_{\kappa\omega} &= \max \left(2\rho\sigma_\omega^2 \frac{1}{\omega} \frac{\partial \kappa}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-10} \right), \\ F_2 &= \tanh \arg_2^2, \quad \arg_2 = \max \left(2 \frac{\sqrt{\kappa}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \\ \sigma_{k1} &= 0.85, \quad \sigma_{\omega1} = 0.65, \quad \beta_1 = 0.075, \quad \sigma_{k2} = 1.00, \quad \sigma_{\omega2} = 0.856, \quad \beta_2 = 0.0828, \\ \beta^* &= 0.09, \quad a_1 = 0.31. \end{aligned}$$

Chapter 2

Reduced Order Methods for the NSE

In the previous Chapter, the Full Order Model computed in the solver *OpenFOAM* is presented. The discretization techniques and the pressure-based solvers implemented in *OpenFOAM* for the NSE have been displayed, together with the turbulence treatment with the RANS approach.

In the present Chapter, the Proper Orthogonal Decomposition (POD) technique is explained and applied to the discretized NSE in order to find the velocity and pressure modes and the correspondent coefficients. The computation of the solutions of the Full Order Model and the Proper Orthogonal Decomposition represent the offline phase of the procedure followed in this project.

In this Chapter, the basic theory of the Reduced Order Methods is also presented and applied to the incompressible NSE.

2.1 The Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) is a technique generally used to decompose a field depending on the variables that influences its behaviour. The method of snapshots is the technique used in this Thesis to generate the POD reduced order space. The method consists in computing the FOM solutions for different time instants $\{t_j\}_{j=1}^{N_T}$: each of the Full Order solution is called FOM snapshot.

The Proper Orthogonal Decomposition is applied on the Full Order snapshots matrices, given by the following matrices:

$$\begin{aligned}\mathcal{S}_u &= \{\mathbf{u}(\mathbf{x}, t_1), \dots, \mathbf{u}(\mathbf{x}, t_{N_T})\} \in \mathbb{R}^{N_u^h \times N_T}, \\ \mathcal{S}_p &= \{p(\mathbf{x}, t_1), \dots, p(\mathbf{x}, t_{N_T})\} \in \mathbb{R}^{N_p^h \times N_T},\end{aligned}$$

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

Now we consider \mathcal{S}_u^n and \mathcal{S}_p^n the n -th velocity and pressure snapshots, obtained at the n -th time step.

The POD procedure consists in the research for the velocity and pressure POD spaces, by solving two optimization problems [1]:

$$\begin{cases} \mathbb{V}_{\text{POD}}^u = \arg \min_{i=1, \dots, N_T} \frac{1}{N_T} \sum_{n=1}^{N_T} \|\mathcal{S}_u^n - \sum_{i=1}^{N_u} (\mathcal{S}_u^n, \phi_i)_{L^2(\Omega)} \phi_i\|_{L^2(\Omega)}^2 \quad \forall N_u = 1, \dots, N_T, \\ \mathbb{V}_{\text{POD}}^p = \arg \min_{i=1, \dots, N_T} \frac{1}{N_T} \sum_{n=1}^{N_T} \|\mathcal{S}_p^n - \sum_{i=1}^{N_p} (\mathcal{S}_p^n, \chi_i)_{L^2(\Omega)} \chi_i\|_{L^2(\Omega)}^2 \quad \forall N_p = 1, \dots, N_T. \end{cases} \quad (2.1)$$

The POD spaces, solutions of (2.1), are $\mathbb{V}_{\text{POD}}^u = \text{span}\{[\phi_i]_{i=1}^{N_u}\}$ and $\mathbb{V}_{\text{POD}}^p = \text{span}\{[\chi_i]_{i=1}^{N_p}\}$, with $N_u \ll N_u^h$ and $N_p \ll N_p^h$. The POD basis modes, $[\phi_i]_{i=1}^{N_u}$ for velocity and $[\chi_i]_{i=1}^{N_p}$ for pressure, can be equivalently found by solving an eigenvalue problem [7]. In particular the correlation matrices are:

$$(\mathbf{C}^u)_{ij} = (\mathcal{S}_u^i, \mathcal{S}_u^j)_{L^2(\Omega)}, \quad (\mathbf{C}^p)_{ij} = (\mathcal{S}_p^i, \mathcal{S}_p^j)_{L^2(\Omega)}, \quad \text{where } \mathbf{C}^u \in \mathbb{R}^{N_T \times N_T} \text{ and } \mathbf{C}^p \in \mathbb{R}^{N_T \times N_T}.$$

Each mode can be computed as follows [1]:

$$\phi_i = \frac{1}{N_T \lambda_{ii}^u} \sum_{j=1}^{N_T} \mathcal{S}_u^j \mathbf{V}_{ij}^u, \quad \chi_i = \frac{1}{N_T \lambda_{ii}^p} \sum_{j=1}^{N_T} \mathcal{S}_p^j \mathbf{V}_{ij}^p,$$

where \mathbf{V}^u and \mathbf{V}^p are the matrices whose columns are the eigenvectors of the correlation matrices, whereas $\boldsymbol{\lambda}^u$ and $\boldsymbol{\lambda}^p$ are diagonal matrices whose entries are the eigenvalues of the correlation matrices.

Once the velocity and pressure modes are computed, the solution can be projected into a certain number of modes and for any number of modes considered it will be the optimal solution, i.e. the one that minimizes the error in the particular norm considered. In our case the projection of the FOM solution into the POD space is the one that minimizes the L^2 norm.

In the Chapter dedicated to results, each of the solutions obtained in the online phase will be compared to the best possible solution, which is the POD projection into the number of modes considered in each specific case.

2.2 POD-Galerkin Reduced Order Methods

In this Section, the online part of the procedure is explained, in particular we focus on POD-Galerkin Reduced Order Methods [29]. The ROM technique is based on the assumption that an approximated solution of the velocity and pressure fields can be written as a linear combination of the POD modes:

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &\approx \mathbf{u}_r(\mathbf{x}, t) = \sum_{i=1}^r a_i(t) \phi_i(\mathbf{x}), \\ p(\mathbf{x}, t) &\approx p_r(\mathbf{x}, t) = \sum_{i=1}^q b_i(t) \chi_i(\mathbf{x}), \end{aligned} \quad (2.2)$$

where r and q are the reduced number of modes considered to approximate velocity and pressure, respectively. The approximated expressions of the fields are then replaced in the incompressible NSE and the ROM procedure is applied to incompressible NSE. This procedure consists in projecting the momentum equation into the velocity modes and the continuity equation into the pressure modes.

After the projection is performed, a dynamical system is obtained, whose unknowns are the velocity and pressure vectors of coefficients; the components of those vectors are the coefficients related to velocity and pressure modes, as follows:

$$\mathbf{a} = (a_i)_{i=1}^r, \quad \mathbf{b} = (b_i)_{i=1}^q.$$

An approach considered in [1] was that of solving the momentum equation, having as unknowns the coefficients of the velocity vector \mathbf{a} with the hypothesis that $\mathbf{b} = \mathbf{a}$. In this case, the results obtained for pressure are poor. Since the pressure field is used to compute a lot of outputs, such as lift and drag forces, two stabilization techniques are considered in this Thesis in order to obtain a better reconstruction of the pressure field. The two different velocity-pressure coupling approaches for the ROM are the following:

- the SUP-ROM: a reduced order method in which the momentum equation and the continuity equation are solved in the reduced order space, introducing additional modes for the velocity space in order to satisfy the *inf-sup* condition and to avoid stability issues of the solution;
- the PPE-ROM: a reduced order method in which the continuity equation is replaced by the Poisson equation for pressure; it allows a better reconstruction of the pressure field.

2.2.1 Reduced Order Method with Supremizer Approach (SUP-ROM)

In the SUP-ROM approach the POD spaces computed for velocity and pressure are the following:

$$\mathbb{V}_{\text{POD}}^u = \text{span}\{[\phi_i]_{i=1}^{N_u} \oplus [\mathbf{s}_i]_{i=1}^{N_{sup}}\}, \quad \mathbb{V}_{\text{POD}}^p = \text{span}\{[\chi_i]_{i=1}^{N_p}\},$$

where some additional velocity modes called *supremizer* modes are considered [14]. Each of these additional modes is associated to each of the pressure modes, according to the following expression:

$$\begin{cases} \Delta \mathbf{s}_i = -\nabla \chi_i & \text{in } \Omega, \\ \mathbf{s}_i = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.3)$$

The supremizer problem is here expressed in strong form; details on the derivation can be found in [30] and [31].

Two strategies can be adopted in order to fulfill the *inf-sup* condition: the *exact* and the *approximated* supremizer enrichment [13, 14]. In the *exact* approach, problem (2.3) is solved for each pressure basis χ_i in the POD space. In the *approximated* approach, which is the one here adopted, the problem (2.3) is solved for each pressure snapshot $p(t)$

and the snapshot matrix of supremizer is assembled. Then the supremizer basis functions $(\mathbf{s}_i)_{i=1}^{N_{sup}}$ are found through a POD procedure [14].

From now on we consider: $[\phi_i]_{i=N_u}^{N_u+N_{sup}} = [\mathbf{s}_i]_{i=1}^{N_{sup}}$.

This approach is called *supremizer enrichment*, it was first introduced in [30] and then also treated in [13]: the aim of this technique is to fulfill the *inf-sup* condition.

The incompressible NSE equations (1.1) are then rewritten by replacing the approximations of velocity and pressure fields (2.2). The momentum equation is then projected into the velocity modes and the continuity equation into the pressure modes, as follows:

$$\begin{cases} \left(-\frac{\partial \mathbf{u}_r}{\partial t} - \nabla \cdot (\mathbf{u}_r \otimes \mathbf{u}_r) + \nabla \cdot \nu (\nabla \mathbf{u}_r + (\nabla \mathbf{u}_r)^T) - \nabla p_r, \phi_i \right)_{L^2(\Omega)} = 0 & \text{for } i = 1, \dots, N_u + N_{sup}, \\ (\nabla \cdot \mathbf{u}_r, \chi_i)_{L^2(\Omega)} = 0 & \text{for } i = 1, \dots, N_p. \end{cases} \quad (2.4)$$

Considering \mathbf{a} the vector of coefficients for velocity field and \mathbf{b} the vector of coefficients for pressure field, system (2.4) becomes the following dynamical system:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b}, \\ \mathbf{P}\mathbf{a} = \mathbf{0}. \end{cases} \quad (2.5)$$

Matrices appearing in system (2.5) are defined in the following way:

$$\begin{aligned} (\mathbf{M})_{ij} &= (\phi_i, \phi_j)_{L^2(\Omega)}, & (\mathbf{B})_{ij} &= (\phi_i, \nabla \cdot \nabla \phi_j)_{L^2(\Omega)}, & (\mathbf{B}_T)_{ij} &= (\phi_i, \nabla \cdot (\nabla \phi_j)^T)_{L^2(\Omega)}, \\ (\mathbf{C})_{ijk} &= (\phi_i, \nabla \cdot (\phi_j \otimes \phi_k))_{L^2(\Omega)}, & (\mathbf{H})_{ij} &= (\phi_i, \nabla \chi_j)_{L^2(\Omega)}, & (\mathbf{P})_{ij} &= (\chi_i, \nabla \cdot \phi_j)_{L^2(\Omega)}. \end{aligned}$$

2.2.2 Reduced Order Method with Poisson Equation for Pressure (PPE-ROM)

The second approach we followed is the PPE-ROM approach, in which the continuity equation in the reduced system (2.5) is replaced by the Poisson equation for pressure, obtained by taking the divergence of the momentum equation and taking into account the fact that the velocity field is divergence-free. This approach was firstly proposed in [32] and then re-proposed in [18] and [14] in a finite volume setting.

The following equation for pressure is obtained at the full order level:

$$\Delta p = -\nabla \cdot (\nabla \cdot (\mathbf{u} \otimes \mathbf{u})) \quad \text{in } \Omega. \quad (2.6)$$

Equation (2.6) is the Poisson equation for pressure, that induces the so-called PPE formulation for the full order NSE:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \nabla p & \text{in } \Omega \times [0, T], \\ \Delta p = -\nabla \cdot (\nabla \cdot (\mathbf{u} \otimes \mathbf{u})) & \text{in } \Omega, \\ + \text{initial conditions} & \text{in } (\Omega, 0), \\ + \text{boundary conditions} & \text{on } \Gamma \times [0, T], \\ \frac{\partial p}{\partial \mathbf{n}} = -\nu \mathbf{n} \cdot (\nabla \times \nabla \times \mathbf{u}) - \mathbf{n} \cdot \mathbf{R}_t & \text{on } \Gamma \times [0, T]. \end{cases} \quad (2.7)$$

The last equation in (2.7) is a Neumann boundary condition for the Poisson equation for pressure [14]. In this case, the velocity and pressure POD spaces are found without introducing additional supremizer modes:

$$\begin{aligned}\mathbb{V}_{\text{POD}}^u &= \text{span}\{[\phi_i]_{i=1}^{N_u}\}, \\ \mathbb{V}_{\text{POD}}^p &= \text{span}\{[\chi_i]_{i=1}^{N_p}\},\end{aligned}\tag{2.8}$$

where $N_u \ll N_u^h$ and $N_p \ll N_p^h$, $[\phi_i]_{i=1}^{N_u}$ are the velocity POD modes, $[\chi_i]_{i=1}^{N_p}$ are the pressure POD modes.

A Galerkin projection procedure is carried out in order to obtain the approximated fields:

$$\begin{aligned}\mathbf{u}(\mathbf{x}, t) &\approx \mathbf{u}_r(\mathbf{x}, t) = \sum_{i=1}^{N_u} a_i(t) \phi_i(\mathbf{x}), \\ p(\mathbf{x}, t) &\approx p_r(\mathbf{x}, t) = \sum_{i=1}^{N_p} b_i(t) \chi_i(\mathbf{x}).\end{aligned}\tag{2.9}$$

Then the momentum equation in (2.7) is projected onto velocity modes and the pressure Poisson equation in (2.7) is projected onto pressure modes. Projected NSE are written as follows:

$$\begin{cases} \left(-\frac{\partial \mathbf{u}_r}{\partial t} - \nabla \cdot (\mathbf{u}_r \otimes \mathbf{u}_r) + \nabla \cdot \nu (\nabla \mathbf{u}_r + (\nabla \mathbf{u}_r)^T) - \nabla p_r, \phi_i \right)_{L^2(\Omega)} = 0 & \text{for } i = 1, \dots, N_u, \\ \left(\nabla p_r + \nabla \cdot (\mathbf{u}_r \otimes \mathbf{u}_r), \nabla \chi_i \right)_{L^2(\Omega)} - \nu (\mathbf{n} \times \nabla \chi_i, \nabla \times \mathbf{u}_r)_\Gamma - (\mathbf{n} \cdot \mathbf{R}_t, \chi_i)_\Gamma = 0 & \text{for } i = 1, \dots, N_p. \end{cases}\tag{2.10}$$

In the second equation of (2.10) the vector \mathbf{R} is such that:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{R}(\mathbf{x}) \text{ on } \Gamma_{\text{Inlet}},$$

where Γ_{Inlet} is the inlet of the boundary domain. In the specific test case considered in the results Chapter, the velocity conditions at the inlet do not change as time evolves and the term $(\mathbf{n} \cdot \mathbf{R}_t, \chi_i)_\Gamma$ is identically null.

Considering $\mathbf{a} = (a_i)_{i=1}^{N_u}$ and $\mathbf{b} = (b_i)_{i=1}^{N_p}$, the following dynamical system is obtained:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_\Gamma)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b}, \\ \mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} = \mathbf{0}. \end{cases}\tag{2.11}$$

Additional matrices appearing in system (2.11) are defined in the following way:

$$\begin{aligned}(\mathbf{D})_{ij} &= (\nabla \chi_i, \nabla \chi_j)_{L^2(\Omega)}, & (\mathbf{G})_{ijk} &= (\nabla \chi_i, \nabla \cdot (\phi_j \otimes \phi_k))_{L^2(\Omega)}, \\ (\mathbf{N})_{ij} &= (\mathbf{n} \times \nabla \chi_i, \nabla \phi_j)_\Gamma, & (\mathbf{L})_{ij} &= (\chi_i, \mathbf{n} \cdot \mathbf{R}_t)_\Gamma.\end{aligned}$$

2.2.3 Boundary Conditions Treatment

In the simulations of our fluid dynamics problems it is necessary to impose non-homogeneous Dirichlet boundary conditions. These conditions have to be considered in the dynamical system solved in the online phase. Different methods have been developed in order to include the boundary conditions into the system, for example the *penalty method* [33, 34, 35, 36, 37], which is the one used in this Thesis.

The Penalty Method

We consider the Dirichlet boundary $\Gamma_D \subset \Gamma$ and it can be decomposed into separate boundaries: $\Gamma_{D1}, \Gamma_{D2}, \dots, \Gamma_{Dk}$. We call N_{BC} the number of boundary conditions imposed on some parts of the boundary; each boundary condition corresponds to the non-zero component of the velocity vector on each part of the boundary considered. In particular, the notation $(U_{BC})_{i,j}$ indicates the non-zero value of the i -th component of the velocity vector in the part Γ_{Dj} of the boundary. We consider the vector \mathbf{U}_{BC} as the vector of the scalar velocities $(U_{BC})_{ij}$ of dimension N_{BC} .

The dynamical system obtained for the ROM with the supremizer enrichment (2.5) becomes the following system, by adding the boundary conditions:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{BC}} (U_{BC,k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right), \\ \mathbf{P}\mathbf{a} = \mathbf{0}. \end{cases} \quad (2.12)$$

The term $\tau \left(\sum_{k=1}^{N_{BC}} (U_{BC,k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right)$ in (2.12) is a constraint expressing the Dirichlet non-homogeneous boundary conditions: this way of enforcing the boundary conditions is called *penalty method* [38]. In particular, N_{BC} is the number of velocity boundary conditions on k different parts of the Dirichlet boundary; $U_{BC,k}$ is the velocity non-zero component at the k -th part of the Dirichlet boundary; τ is a penalization factor, matrices \mathbf{E}^k and vectors \mathbf{D}^k are defined as:

$$(\mathbf{E}^k)_{ij} = (\phi_i, \phi_j)_{L^2(\Gamma_{D_k})}, \quad (\mathbf{D}^k)_i = (\phi_i)_{\Gamma_{D_k}}, \quad \text{for all } k = 1, \dots, N_{BC}.$$

In a similar way, the dynamical system obtained for the ROM with the supremizer enrichment (2.11) becomes the following system, by adding the boundary conditions:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{BC}} (U_{BC,k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right), \\ \mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} = \mathbf{0}. \end{cases} \quad (2.13)$$

2.3 Turbulence Modelling

In the previous Section, a dynamical system both for the SUP-ROM and the PPE-ROM approaches is obtained. However, in systems (2.12) and (2.13) a turbulence modelling needs to be included. At the Full Order Level the turbulence was included by adding to RANS equations the transport equations for κ and ω , as in (1.16). At the reduced order level an approximation of the eddy viscosity term can be included in equations by introducing a reduced order version of the eddy viscosity [1], as follows:

$$\nu_t(\mathbf{x}, t) \approx \nu_{tr}(\mathbf{x}, t) = \sum_{i=1}^{N_{\nu_t}} g_i(t) \eta_i(\mathbf{x}),$$

where $\eta_i(\mathbf{x})$ is the i -th eddy viscosity POD mode and $g_i(t)$ is the correspondent coefficient. Before including the turbulence term in the dynamical system, the eddy viscosity

coefficients need to be evaluated. The approach for the computation of the eddy viscosity coefficients used in [1] involved a data-driven technique based on Radial Basis Functions [39, 40]. In particular, in the offline phase the eddy viscosity POD modes are found by solving an eigenvalue problem, similarly to the velocity and the pressure modes:

$$\eta_i = \frac{1}{N_T \lambda_{ii}^{\nu_t}} \sum_{j=1}^{N_T} \mathcal{S}_{\nu_t}^j \mathbf{V}_{ij}^{\nu_t},$$

where the diagonal matrix $\boldsymbol{\lambda}^{\nu_t}$ has on its entries the eigenvalues of the correlation matrix \mathbf{C}^{ν_t} , \mathbf{V}^{ν_t} is the matrix whose columns are the eigenvectors of \mathbf{C}^{ν_t} , \mathcal{S}_{ν_t} is the eddy viscosity snapshot matrix.

Adding the turbulence terms to system (2.12), the new dynamical system is the following:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} + \mathbf{g}^T(\mathbf{C}_{T1} + \mathbf{C}_{T2})\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{BC}} (U_{BC,k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right), \\ \mathbf{P}\mathbf{a} = \mathbf{0}, \end{cases} \quad (2.14)$$

where the new matrices appearing are defined as:

$$\begin{cases} (\mathbf{C}_{T1})_{ijk} = (\phi_i, \eta_j \nabla \cdot \nabla \phi_k)_{L^2(\Omega)}, \\ (\mathbf{C}_{T2})_{ijk} = (\phi_i, \nabla \cdot \eta_j (\nabla \phi_k)^T)_{L^2(\Omega)}. \end{cases}$$

When a PPE approach is considered, the FOM momentum and Poisson equation, according to the RANS turbulent model, are written as follows:

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

Consequently the dynamical system (2.13) evolves in the following form:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} + \mathbf{g}^T(\mathbf{C}_{T1} + \mathbf{C}_{T2})\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{BC}} (U_{BC,k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right), \\ \mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} - \mathbf{g}^T(\mathbf{C}_{T3} + \mathbf{C}_{T4})\mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} = \mathbf{0}, \end{cases} \quad (2.15)$$

where:

$$(\mathbf{C}_{T3})_{ijk} = (\nabla \chi_i, \eta_j \nabla \cdot \nabla \phi_k)_{L^2(\Omega)}, \quad (\mathbf{C}_{T4})_{ijk} = (\nabla \chi_i, \nabla \cdot \eta_j (\nabla \phi_k)^T)_{L^2(\Omega)}.$$

In the dynamical systems defined in (2.14) and (2.15), the number of unknowns is composed by: $N_u + N_{sup}$ (in (2.14)) and N_u (in (2.15)) coefficients for velocity, N_p coefficients for pressure, N_{ν_t} coefficients for the eddy viscosity. However, the number of equations is $N_u + N_{sup} + N_p$ in (2.14) and $N_u + N_p$ in (2.15). Thus, there are more unknowns than equations and the system is not closed in both cases of supremizer and Poisson approach.

The vector \mathbf{g} should be computed with data-driven methods in order to close the reduced system: the coefficients $[g_i(t)]_{i=1}^{N_{\nu_t}}$ are approximated considering the mapping $\mathbf{g} = f(\mathbf{a})$ through *interpolation* or *regression* techniques.

In the first approach the reduced eddy viscosity coefficients are interpolated with Radial Basis Functions [39, 40]. This technique was exploited in [1], following the POD-I approach [41, 42, 43]. In the second approach, which is the one used in this Thesis, the reduced eddy viscosity coefficients are computed with a regression technique starting from the velocity coefficients $[a_i]_{i=1}^{N_u}$ through a fully-connected neural network.

Chapter 3

A Novel Approach based on Data-Driven Reduced Order Methods (DD-ROMs)

In this Chapter, the Reduced Order Methods and the techniques described in Section 2.3 are compared and combined with other data-driven techniques, which were previously addressed in [10, 11, 12]. The reduced order momentum equation in the standard Galerkin-ROM (G-ROM) formulation can be written in the form:

$$\dot{\mathbf{a}} = f(\mathbf{a}), \quad (3.1)$$

where the unknowns are $\mathbf{a} = (a_1(t), a_2(t), \dots, a_r(t))$ and the reduced velocity field is $\mathbf{u}_r = \sum_{i=1}^r a_i(t) \phi_i(\mathbf{x})$. In the numerical simulation of turbulent flows, a standard ROM approximation could require hundreds or even thousands of ROM modes, which would significantly increase the ROM computational cost. To ensure a low ROM computational cost, ROMs are generally built with relatively few basis functions. The data-driven techniques addressed for the G-ROM formulation consist in the addition of one or more extra terms to the momentum equation in order to include the contribution of the unresolved modes. Equation (3.1) is rewritten as follows:

$$\dot{\mathbf{a}} = f(\mathbf{a}) + \boldsymbol{\tau}(\mathbf{a}),$$

where $\boldsymbol{\tau}(\mathbf{a})$ is called *closure* or *correction* term and has the aim of describing the interactions between the resolved modes and the unresolved modes. The closure/correction term is found by solving an optimization problem between the exact term and an ansatz which is proposed to approximate it. The procedure followed to compute the approximated term inherited the method used in [44].

In particular, two reduced regimes can be distinguished depending on the number of modes resolved in the online phase:

- the *under-resolved* regime, using fewer ROM modes than the number required to accurately approximate the dynamics of the given system. In this situation, the

terms added to the equation are called *closure terms*, which is the same terminology used for LES [45, 9, 8].

- the *marginally-resolved* regime, which is an intermediate regime, between the *under-resolved* regime and the *fully resolved* regime (i.e., when the number of modes is enough to represent the underlying dynamics). When we are in this situation, the terms added to the equation are called *correction terms* [46].

In the simulations carried out in this Thesis, the regime considered is the *marginally-resolved* regime. In this case, the number of ROM basis functions is enough to represent the main features of the underlying dynamics, but the standard G-ROM yields inaccurate approximations especially in terms of pressure, as can be seen in [1]. In order to better approximate the pressure field, the models described in 2.2.1 and 2.2.2 are analysed in this Thesis as stabilization techniques for the standard G-ROM formulation.

The framework considered for both approaches is that of *data-driven filtered ROM* (DDF-ROM) [10]. The *filtered ROM* formulation is obtained by applying a spatial filter to the Full Order Model (the incompressible NSE in our case), following a typical LES-ROM approach [9, 47, 2]. The spatial filter applied to the FOM is the ROM projection itself, as described in [10] and [11]. The ROM projection of velocity and pressure onto the first r and q modes, respectively, is seen as a spatial filter of the full order variables such that, for fixed r and q :

$$(\bar{\mathbf{u}}^r, \boldsymbol{\phi}_i) = (\mathbf{u}, \boldsymbol{\phi}_i) \forall i = 1, \dots, r, \quad (3.2)$$

$$(\bar{p}^r, \chi_i) = (p, \chi_i) \forall i = 1, \dots, q. \quad (3.3)$$

The number of modes taken into account in (3.2) is $r = N_u + N_{sup}$ when supremizer modes are added, and $r = N_u$ when supremizer modes are not considered; the number of modes for pressure in (3.3) is $q = N_p$. The POD-Galerkin ROM procedure is then applied to the NSE. The system obtained with the stabilization techniques is in the following form:

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a}, \mathbf{b}), \\ g(\mathbf{a}, \mathbf{b}) = \mathbf{0}, \end{cases} \quad (3.4)$$

where $\mathbf{a} = (a_1(t), \dots, a_r(t))$ and $\mathbf{b} = (b_1(t), \dots, b_q(t))$. The system (3.4) is rewritten adding the closure/correction terms in the following way:

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a}, \mathbf{b}) + \boldsymbol{\tau}_1(\mathbf{a}, \mathbf{b}), \\ g(\mathbf{a}, \mathbf{b}) + \boldsymbol{\tau}_2(\mathbf{a}, \mathbf{b}) = \mathbf{0}, \end{cases} \quad (3.5)$$

where the terms $\boldsymbol{\tau}_i(\mathbf{a}, \mathbf{b})$, $i = 1, 2$ are new correction/closure terms developed in this Thesis. Since the velocity correction significantly improved the results for velocity but not for pressure, new correction terms depending on pressure have been developed in order to obtain a better reconstruction of the pressure field as well. Since in the supremizer approach the terms depending on pressure are linear terms, their effect on the results is not as evident as the effect of the velocity correction term. Therefore, the Poisson approach is addressed since it allows to define and propose for the first time some pressure correction terms, totally or partially depending on pressure reduced coefficients.

3.1 Data-driven velocity corrections in the SUP-ROM approach

In this Section, the data-driven approach introduced in [10, 11] is applied to the reduced NSE when a supremizer enrichment is considered.

The reduced system (2.4) can be reformulated as a system of spatially filtered NSE:

$$\begin{cases} \left(-\frac{\partial \bar{\mathbf{u}}_d^r}{\partial t}, \boldsymbol{\phi}_i \right) + \nu (\nabla \cdot (\nabla \bar{\mathbf{u}}_d^r + (\nabla \bar{\mathbf{u}}_d^r)^T), \boldsymbol{\phi}_i) - ((\bar{\mathbf{u}}_d^r \cdot \nabla) \bar{\mathbf{u}}_d^r, \boldsymbol{\phi}_i) - (\nabla \bar{p}_d^r, \boldsymbol{\phi}_i) + \\ \quad + c_u (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) + c_{p(1)} (\boldsymbol{\tau}_{p(1)}^{\text{SFS}}, \boldsymbol{\phi}_i) = 0 \text{ for } i = 1, \dots, r, \\ (\nabla \cdot \bar{\mathbf{u}}_d^r, \chi_i) + c_{p(2)} (\boldsymbol{\tau}_{p(2)}^{\text{SFS}}, \chi_i) = 0 \text{ for } i = 1, \dots, q, \end{cases} \quad (3.6)$$

where the following correction terms for velocity and pressure have been introduced:

$$\begin{aligned} \boldsymbol{\tau}_u^{\text{SFS}} &= - \left(\overline{(\mathbf{u}_d \cdot \nabla) \mathbf{u}_d}^r - (\bar{\mathbf{u}}_d^r \cdot \nabla) \bar{\mathbf{u}}_d^r \right), \\ \boldsymbol{\tau}_{p(1)}^{\text{SFS}} &= - \left(\overline{\nabla p_d}^r - \nabla \bar{p}_d^r \right), \\ \boldsymbol{\tau}_{p(2)}^{\text{SFS}} &= \overline{\nabla \cdot \mathbf{u}_d}^r - \nabla \cdot \bar{\mathbf{u}}_d^r. \end{aligned} \quad (3.7)$$

In system (3.6), c_u , $c_{p(1)}$ and $c_{p(2)}$ are parameters such that:

- $c_u = 1$ if the first correction term in (3.7) is added to the system (3.6), and $c_u = 0$ if not;
- $c_{p(1)} = 1$ if the second correction term in (3.7) is added to the momentum equation in (3.6), and $c_{p(1)} = 0$ otherwise;
- $c_{p(2)} = 1$ if the third correction term in (3.7) is added to the continuity equation in (3.6), and $c_{p(2)} = 0$ otherwise.

The parameters c_u , $c_{p(1)}$, $c_{p(2)}$ are added to the system in order to evaluate the influence of each correction term in the reduced system. The aim is to analyse the effect of each data-driven term singularly, but also to evaluate the solutions of the system when more than one correction terms are considered.

In the spatially filtered NSE, the following hypotheses are introduced:

- in order to reduce the computational effort $\mathbf{u} \approx \mathbf{u}_d = \sum_{i=1}^d a_i(t) \boldsymbol{\phi}_i(\mathbf{x})$ and $p \approx p_d = \sum_{i=1}^{d_p} b_i(t) \chi_i(\mathbf{x})$. In this Thesis, $d = d_p = 50$: the projection in the filtered NSE (3.6) is carried out not starting from the FOM fields, but starting from the fields reconstructed using a number d and d_p of modes, where d and d_p are smaller than the rank of the snapshots matrix, as in [10];
- Since the ROM projection is used as a spatial filter:

$$\bar{\mathbf{u}}_d^r = \mathbf{u}_r \text{ and } \bar{p}_d^r = p_r,$$

where $r \ll d$ and $q \ll d_p$.

Then the following terms are introduced:

$$\begin{aligned} \boldsymbol{\tau}^u & \text{ such that } \tau_i^u = \left(\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i \right), \\ \boldsymbol{\tau}^{p(1)} & \text{ such that } \tau_i^{p(1)} = \left(\boldsymbol{\tau}_{p(1)}^{\text{SFS}}, \boldsymbol{\phi}_i \right), \\ \boldsymbol{\tau}^{p(2)} & \text{ such that } \tau_i^{p(2)} = \left(\boldsymbol{\tau}_{p(2)}^{\text{SFS}}, \chi_i \right). \end{aligned} \quad (3.8)$$

Adding the correction terms and the boundary conditions, the dynamical system (2.5) becomes:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_\Gamma)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right) + c_u \boldsymbol{\tau}^u + c_{p(1)} \boldsymbol{\tau}^{p(1)}, \\ \mathbf{P}\mathbf{a} + c_{p(2)} \boldsymbol{\tau}^{p(2)} = \mathbf{0}. \end{cases} \quad (3.9)$$

The new system (3.9) is not a closed system because the correction terms depend on fields \mathbf{u}_d and p_d . In order to close the problem a data-driven modelling is adopted, as in [10] and [11]. The key problem is to find the approximated expressions:

$$\boldsymbol{\tau}^u \approx \boldsymbol{\tau}^u(\mathbf{a}), \quad \boldsymbol{\tau}^{p(1)} \approx \boldsymbol{\tau}^{p(1)}(\mathbf{b}), \quad \boldsymbol{\tau}^{p(2)} \approx \boldsymbol{\tau}^{p(2)}(\mathbf{a}).$$

3.1.1 Data-driven correction for velocity

In this Section, only the correction term for velocity $\boldsymbol{\tau}^u$ is considered in system (3.9) in order to evaluate its influence on the dynamical system first, i.e. we consider: $c_u = 1$, $c_{p(1)} = c_{p(2)} = 0$.

The correction term for velocity is modelled as in [10] using the ansatz:

$$\boldsymbol{\tau}^u(\mathbf{a}) = \tilde{\mathbf{A}}\mathbf{a} + \mathbf{a}^T \tilde{\mathbf{B}}\mathbf{a}, \quad (3.10)$$

where $\tilde{\mathbf{B}}$ is a 3-entries tensor.

To find $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ the following optimization problem is solved:

$$\min_{\substack{\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}, \\ \tilde{\mathbf{B}} \in \mathbb{R}^{r \times r \times r}}} \sum_{j=1}^M \|\boldsymbol{\tau}^{\text{exact}}(t_j) - \boldsymbol{\tau}^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2, \quad (3.11)$$

where M time instances are considered to build the correction term and the term $\boldsymbol{\tau}^{\text{exact}}(t_j)$ is computed starting from the snapshot vectors $\mathbf{a}_d^{\text{snap}}(t_j)$, satisfying the conditions:

$$a_{d_i}^{\text{snap}}(t_j) = (\mathbf{u}_d(t_j), \boldsymbol{\phi}_i) \quad \forall i = 1, \dots, d, j = 1, \dots, M.$$

The exact correction term is evaluated as follows:

$$\boldsymbol{\tau}^{\text{exact}}(t_j) = \left(-\overline{(\mathbf{a}_d^{\text{snap}}(t_j))^T \mathbf{C}_d \mathbf{a}_d^{\text{snap}}(t_j)^r} \right) - \left(-(\mathbf{a}_r^{\text{snap}}(t_j))^T \mathbf{C}_r^{\text{snap}}(t_j) \right),$$

where the tensor $\mathbf{C}_d \in \mathbb{R}^{d \times d \times d}$ is defined in the following way.

$$\mathbf{C}_d_{ijk} = \left(\boldsymbol{\phi}_i, \nabla \cdot (\boldsymbol{\phi}_j \otimes \boldsymbol{\phi}_k) \right).$$

The approximated term is evaluated as in (3.10), but starting from $\mathbf{a}_r^{\text{snap}}(t_j)$ at each time step j :

$$\boldsymbol{\tau}^{\text{ansatz}}(t_j) = \tilde{A}\mathbf{a}_r^{\text{snap}}(t_j) + (\mathbf{a}_r^{\text{snap}}(t_j))^T \tilde{B}\mathbf{a}_r^{\text{snap}}(t_j)$$

The optimization problem (3.11) is rewritten as a least squares problem following a procedure similar to that used in [44]. In particular, the following terms are defined:

- the snapshot matrix $\hat{X} \in \mathbb{R}^{M \times r}$. Indicating with $\hat{X}_{j,\cdot}$ the j -th row of the matrix, we have:

$$\hat{X}_{j,\cdot} = \mathbf{a}_r^{\text{snap}}(t_j);$$

- r vectors $\mathbf{a}^{(1)}(t_j), \dots, \mathbf{a}^{(r)}(t_j)$ for each time step, such that:

$$\mathbf{a}^{(i)}(t_j) = a_i^{\text{snap}}(t_j) \begin{bmatrix} a_0^{\text{snap}}(t_j) \\ a_1^{\text{snap}}(t_j) \\ \dots \\ a_i^{\text{snap}}(t_j) \end{bmatrix} \in \mathbb{R}^i \text{ for } i = 1, \dots, r,$$

where $a_i^{\text{snap}}(t_j)$ is the i -th component of the snapshot vector at time step j ;

- r different matrices $\hat{X}^{(1)}, \dots, \hat{X}^{(r)}$, with $\hat{X}^{(i)} \in \mathbb{R}^{M \times i}$ such that:

$$\hat{X}_{j,\cdot}^{(i)} = \mathbf{a}^{(i)}(t_j);$$

- the matrix $\mathbf{R} \in \mathbb{R}^{M \times r}$ such that:

$$\mathbf{R}_{j,\cdot} = \boldsymbol{\tau}^{\text{exact}}(t_j) \quad \forall j = 1, \dots, M.$$

The optimization problem (3.11) can be expressed in the following way:

$$\min_{\substack{\tilde{A} \in \mathbb{R}^{r \times r}, \\ \tilde{B} \in \mathbb{R}^{r \times r \times r}}} \|\mathbf{R} - \hat{X}\tilde{A}^T - \sum_{i=1}^r \hat{X}^{(i)}(\tilde{B}^{(i)})^T\|_F^2, \quad (3.12)$$

where $\tilde{B}^{(i)}$ are blocks of the tensor \tilde{B} of dimension $i \times i$, and the norm considered in the minimization is the *Frobenius* norm. In a more compact form the optimization problem (3.12) can be written as follows:

$$\min_{\mathbf{O}} \|\mathbf{R} - \mathbf{D}\mathbf{O}^T\|_F^2, \quad (3.13)$$

where:

$$\mathbf{D} = [\hat{X}, \hat{X}^{(1)}, \hat{X}^{(2)}, \dots, \hat{X}^{(r)}], \quad \mathbf{O} = [\tilde{A}, \tilde{B}^{(1)}, \tilde{B}^{(2)}, \dots, \tilde{B}^{(r)}]$$

The problem (3.13) can be also seen as a set of r optimization problems:

$$\min_{\mathbf{o}_i, i=1, \dots, r} \|\mathbf{r}_i - \mathbf{D}\mathbf{o}_i\|_{L^2(\Omega)}^2,$$

where \mathbf{o}_i is the i -th row of matrix \mathbf{O} and \mathbf{r}_i is the i -th column of matrix \mathbf{R} . As in [10] and [44], the problem (3.13) is ill-conditioned since matrix \mathbf{D} has a very high condition

number. In order to solve this issue, a truncated singular value decomposition is applied to matrix \mathbf{D} , just as in step 6 of Algorithm 1 in [10].

After \tilde{A} and \tilde{B} are found from the least squares problem, the dynamical system to be solved is:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right) + \tilde{A}\mathbf{a} + \mathbf{a}^T \tilde{B}\mathbf{a}, \\ \mathbf{P}\mathbf{a} = \mathbf{0}. \end{cases} \quad (3.14)$$

The ill-conditioning of the least squares problem leads to an ill-conditioning of the dynamical system (3.14). In order to fix this problem step 5 of Algorithm 1 of article [12] is applied: the number R of singular values retained for matrix \mathbf{D} is the optimal one, i.e. the one that minimizes the error metric:

$$\varepsilon_u(L^2) = \sum_{j=1}^M \|\mathbf{u}_{\text{sol}}(t_j) - \mathbf{u}_r(t_j)\|_{L^2(\Omega)}, \quad (3.15)$$

where $\mathbf{u}_{\text{sol}}(t_j) = \sum_{i=1}^r a_i(t_j) \phi_i$ is found from the solution of the dynamical system at each time step and $\mathbf{u}_r(t_j)$ is the projection of the full order solution on the POD space generated by r modes.

3.1.2 Constrained data-driven correction for velocity

The least squares problem for velocity can be also solved using a different approach, as the one used in [11]. In this approach the matrices \tilde{A} and \tilde{B} inherit the following physical properties:

- $\mathbf{a}^T \tilde{A}\mathbf{a} \leq 0$, i.e. \tilde{A} negative semi-definite;
- $\mathbf{a}^T (\mathbf{a}^T \tilde{B}\mathbf{a}) = 0$, i.e. \tilde{B} skew-symmetric.

The optimization problem becomes:

$$\min_{\substack{\tilde{A} \in \mathbb{R}^{r \times r}, \\ \tilde{B} \in \mathbb{R}^{r \times r \times r}, \\ \mathbf{a}^T \tilde{A}\mathbf{a} \leq 0, \\ \mathbf{a}^T (\mathbf{a}^T \tilde{B}\mathbf{a}) = 0}} \sum_{j=1}^M \|\boldsymbol{\tau}^{\text{exact}}(t_j) - \boldsymbol{\tau}^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2. \quad (3.16)$$

As showed in [11] and in the results Chapter, the constrained method is more efficient than the not constrained one when the number of modes considered for velocity and pressure is very small; as the number of modes increases, the not constrained method seems to produce better results in terms of percentage error of velocity and pressure. We will refer to the number of singular values retained in the truncated SVD as R_c in the results Chapter.

3.1.3 Three-Scales Data-Driven Variational Multiscale ROMs (3S-DD-VMS-ROM)

The correction term for velocity $\boldsymbol{\tau}^u$ can be also built starting from a *variational multiscale* (VMS) methodology. According to this *LES-inspired* approach, the POD modes obtained in the offline phase can be divided into three categories: (i) resolved large scales, (ii) resolved small scales, (iii) unresolved scales, as in article [12]. The *VMS-ROM* closure term $\boldsymbol{\tau}^u$ should capture the interactions between all the classes of scales. In Section 3.1.1, the full order velocity field is considered as the sum of a resolved ROM component \mathbf{u}_r and of an unresolved ROM component, as follows:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_r + \mathbf{u}' = \sum_{i=1}^r a_i(t) \boldsymbol{\phi}_i(\mathbf{x}) + \sum_{i=r+1}^d a_i(t) \boldsymbol{\phi}_i(\mathbf{x}). \quad (3.17)$$

The closure term used in Section 3.1.1 is built solving the least squares problem (3.11) and it models the interaction between the resolved ROM modes $\{\boldsymbol{\phi}_i\}_{i=1}^r$ and the unresolved modes $\{\boldsymbol{\phi}_i\}_{i=r+1}^d$, without considering the interaction within the two types of resolved scales. In this Section all the three scales are considered, with r_1 large resolved modes, $r - r_1$ small resolved modes, $d - r$ unresolved modes. The velocity field is written as follows:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_L + \mathbf{u}_S + \mathbf{u}' = \sum_{i=1}^{r_1} a_i(t) \boldsymbol{\phi}_i(\mathbf{x}) + \sum_{i=r_1+1}^r a_i(t) \boldsymbol{\phi}_i(\mathbf{x}) + \sum_{i=r+1}^d a_i(t) \boldsymbol{\phi}_i(\mathbf{x}).$$

Two vectors of coefficients can be distinguished: the one which describes the large resolved scales $\mathbf{a}_L = (a_i)_{i=1}^{r_1}$, and the one of the small resolved scales $\mathbf{a}_S = (a_i)_{i=r_1+1}^r$.

The momentum equation of system (3.6) can be written for both the large and the small resolved ROM components, just considering the closure term $(\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i)$. In particular, one can write:

$$\begin{aligned} & \left(-\frac{\partial \mathbf{u}_L}{\partial t}, \boldsymbol{\phi}_i \right) + \nu \left(\nabla \cdot \left(\nabla \mathbf{u}_L + (\nabla \mathbf{u}_L)^T \right), \boldsymbol{\phi}_i \right) - ((\mathbf{u}_L \cdot \nabla) \mathbf{u}_L, \boldsymbol{\phi}_i) - (\nabla p_L, \boldsymbol{\phi}_i) + \\ & \quad + (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) = 0 \text{ for } i = 1, \dots, r_1, \\ & \left(-\frac{\partial \mathbf{u}_S}{\partial t}, \boldsymbol{\phi}_i \right) + \nu \left(\nabla \cdot \left(\nabla \mathbf{u}_S + (\nabla \mathbf{u}_S)^T \right), \boldsymbol{\phi}_i \right) - ((\mathbf{u}_S \cdot \nabla) \mathbf{u}_S, \boldsymbol{\phi}_i) - (\nabla p_S, \boldsymbol{\phi}_i) + \\ & \quad + (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) = 0 \text{ for } i = r_1 + 1, \dots, r. \end{aligned}$$

The term $\boldsymbol{\tau}_u^{\text{SFS}}$ is defined as: $\boldsymbol{\tau}_u^{\text{SFS}} = - \left(\overline{(\mathbf{u}_d \cdot \nabla) \mathbf{u}_d}^r - ((\mathbf{u}_L + \mathbf{u}_S) \cdot \nabla) (\mathbf{u}_L + \mathbf{u}_S) \right)$.

The following two terms are defined:

$$\begin{aligned} \boldsymbol{\tau}_L & \in \mathbb{R}^{r_1} \text{ such that } \tau_{L_i} = (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) \text{ for } i = 1, \dots, r_1, \\ \boldsymbol{\tau}_S & \in \mathbb{R}^{r-r_1} \text{ such that } \tau_{S_i} = (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) \text{ for } i = r_1, \dots, r. \end{aligned}$$

The terms $\boldsymbol{\tau}_L$ and $\boldsymbol{\tau}_S$ are modelled with the same ansatz used in Section 3.1.1:

$$\boldsymbol{\tau}_L(\mathbf{a}_L) = \tilde{A}_L \mathbf{a}_L + \mathbf{a}_L^T \tilde{B}_L \mathbf{a}_L, \quad \boldsymbol{\tau}_S(\mathbf{a}_S) = \tilde{A}_S \mathbf{a}_S + \mathbf{a}_S^T \tilde{B}_S \mathbf{a}_S,$$

where $\tilde{A}_L \in \mathbb{R}^{r_1 \times r_1}$, $\tilde{B}_L \in \mathbb{R}^{r_1 \times r_1 \times r_1}$, $\tilde{A}_S \in \mathbb{R}^{(r-r_1) \times (r-r_1)}$, $\tilde{B}_S \in \mathbb{R}^{(r-r_1) \times (r-r_1) \times (r-r_1)}$.

The matrices above mentioned are evaluated through the following two least squares problems:

$$\begin{aligned} \min_{\tilde{A}_L, \tilde{B}_L} \|\boldsymbol{\tau}_L^{\text{exact}} - (\tilde{A}_L \mathbf{a}_L^{\text{snap}} + (\mathbf{a}_L^{\text{snap}})^T \tilde{B}_L \mathbf{a}_L^{\text{snap}})\|_{L^2(\Omega)}, \\ \min_{\tilde{A}_S, \tilde{B}_S} \|\boldsymbol{\tau}_S^{\text{exact}} - (\tilde{A}_S \mathbf{a}_S^{\text{snap}} + (\mathbf{a}_S^{\text{snap}})^T \tilde{B}_S \mathbf{a}_S^{\text{snap}})\|_{L^2(\Omega)}. \end{aligned} \quad (3.18)$$

The least squares problems (3.18) are solved separately just as in Section 3.1.1; the problems are different from those explained in [12], in which matrices are applied to the vector \mathbf{a} , including all the coefficients related to the resolved ROM scales. The least squares problems evolve in the following forms:

$$\begin{aligned} \min_{\mathbf{o}_{L_i}, i=1, \dots, r_1} \|\mathbf{r}_{L_i} - \mathbf{D}_L \mathbf{o}_{L_i}\|_{L^2(\Omega)}^2, \\ \min_{\mathbf{o}_{S_i}, i=1, \dots, r-r_1} \|\mathbf{r}_{S_i} - \mathbf{D}_S \mathbf{o}_{S_i}\|_{L^2(\Omega)}^2, \end{aligned}$$

where:

- \mathbf{r}_{L_i} and \mathbf{r}_{S_i} are the i -th columns of matrices $\mathbf{R}_L \in \mathbb{R}^{M \times r_1}$ and $\mathbf{R}_S \in \mathbb{R}^{M \times (r-r_1)}$ such that $\mathbf{R}_{L_{j,\cdot}} = \boldsymbol{\tau}_L(t_j)$ and $\mathbf{R}_{S_{j,\cdot}} = \boldsymbol{\tau}_S(t_j)$;
- $\mathbf{D}_L = [\hat{X}_L, \hat{X}_L^{(1)}, \dots, \hat{X}_L^{(r_1)}]$ and $\mathbf{D}_S = [\hat{X}_S, \hat{X}_S^{(1)}, \dots, \hat{X}_S^{(r-r_1)}]$.

Matrices $\hat{X}_L \in \mathbb{R}^{M \times r_1}$ and $\hat{X}_S \in \mathbb{R}^{M \times (r-r_1)}$ are the snapshots matrices for the two ROM scales, such that:

$$\hat{X}_{L_{j,\cdot}} = \mathbf{a}_L^{\text{snap}}(t_j), \quad \hat{X}_{S_{j,\cdot}} = \mathbf{a}_S^{\text{snap}}(t_j);$$

- \mathbf{o}_{L_i} and \mathbf{o}_{S_i} are the i -th rows of matrices \mathbf{O}_L and \mathbf{O}_S respectively, where:

$$\mathbf{O}_L = [\tilde{A}_L, \tilde{B}_L^{(1)}, \dots, \tilde{B}_L^{(r_1)}], \quad \mathbf{O}_S = [\tilde{A}_S, \tilde{B}_S^{(1)}, \dots, \tilde{B}_S^{(r-r_1)}].$$

Matrices $\tilde{B}_L^{(i)}$ and $\tilde{B}_S^{(i)}$ are blocks of matrices \tilde{B}_L and \tilde{B}_S , respectively.

Then the following dynamical system is solved:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right) + \begin{bmatrix} \tilde{A}_L \mathbf{a}_L + \mathbf{a}_L^T \tilde{B}_L \mathbf{a}_L \\ \tilde{A}_S \mathbf{a}_S + \mathbf{a}_S^T \tilde{B}_S \mathbf{a}_S \end{bmatrix}, \\ \mathbf{P}\mathbf{a} = \mathbf{0}. \end{cases} \quad (3.19)$$

where $\mathbf{a} = [\mathbf{a}_L \quad \mathbf{a}_S]^T$. As in the previous Section, a truncated SVD is applied to matrices \mathbf{D}_L and \mathbf{D}_S in order to avoid the ill-conditioning of system (3.19). In simulations the

number of singular values and the value of r_1 should be the optimal ones, i.e. the one that minimizes the error metric (3.15), as in Algorithm 2 of [12]. The following minimization problem should be solved:

$$\min_{r_1, R_L, R_S} \varepsilon_u(L^2),$$

where R_L and R_S are the singular values retained for \mathbf{D}_L and \mathbf{D}_S , respectively.

3.1.4 Data-driven corrections for pressure

In the present Section, the aim is to introduce new correction or closure terms involving also the pressure modes. In particular, the combined effect of the closure terms for velocity and pressure is analysed, i.e. in system (3.9), we have $c_u = c_{p(1)} = c_{p(2)} = 1$. As it will be widely discussed in the Chapter of results, the reason for proposing a new correction term involving pressure is that the pressure field is not improved when only a velocity correction is considered.

In this Section, the focus is on the evaluation of the closure expressions for the new terms $\boldsymbol{\tau}^{p(1)}$ and $\boldsymbol{\tau}^{p(2)}$. From the expressions (3.8) and following a procedure similar to that one used in Section 3.1.1, the expressions of ansatzes for the correction terms are written as:

$$\begin{aligned} \boldsymbol{\tau}^{p(1)}(\mathbf{b}) &= \tilde{H}\mathbf{b}, \\ \boldsymbol{\tau}^{p(2)}(\mathbf{a}) &= \tilde{P}\mathbf{a}. \end{aligned}$$

Two alternatives are here studied in order to find the matrices \tilde{H} and \tilde{P} .

1. Solve two different and disjoint optimization problems, one for each correction term:

$$\begin{aligned} \min_{\tilde{H} \in \mathbb{R}^{r \times q}} \|\boldsymbol{\tau}_{p(1)}^{\text{exact}} - \boldsymbol{\tau}_{p(1)}^{\text{ansatz}}\|_{L^2(\Omega)}^2, \\ \min_{\tilde{P} \in \mathbb{R}^{q \times r}} \|\boldsymbol{\tau}_{p(2)}^{\text{exact}} - \boldsymbol{\tau}_{p(2)}^{\text{ansatz}}\|_{L^2(\Omega)}^2. \end{aligned} \quad (3.20)$$

where:

$$\begin{aligned} \boldsymbol{\tau}_{p(1)}^{\text{exact}}(t_j) &= \left(-\overline{\mathbf{H}_d \mathbf{b}_{d_p}^{\text{snap}}(t_j)^r} \right) - \left(-(\mathbf{H} \mathbf{b}_q^{\text{snap}}(t_j)) \right), \\ \boldsymbol{\tau}_{p(2)}^{\text{exact}}(t_j) &= \left(\overline{\mathbf{P}_d \mathbf{a}_d^{\text{snap}}(t_j)^r} \right) - (\mathbf{P} \mathbf{a}_r^{\text{snap}}(t_j)), \\ \boldsymbol{\tau}_{p(1)}^{\text{ansatz}} &= \tilde{H} \mathbf{b}_q^{\text{snap}}(t_j), \\ \boldsymbol{\tau}_{p(2)}^{\text{ansatz}} &= \tilde{P} \mathbf{a}_r^{\text{snap}}(t_j). \end{aligned}$$

Matrices \mathbf{H}_d and \mathbf{P}_d appearing in the exact terms are defined as:

$$(\mathbf{H}_d)_{ij} = (\boldsymbol{\phi}_i, \nabla \chi_j)_{L^2(\Omega)}, \quad (\mathbf{P}_d)_{ij} = (\chi_i, \nabla \cdot \boldsymbol{\phi}_j)_{L^2(\Omega)}.$$

2. Solve a unique optimization problem to find both terms:

$$\min_{\substack{\tilde{H} \in \mathbb{R}^{r \times q}, \\ \tilde{P} \in \mathbb{R}^{q \times r}}} \|\boldsymbol{\tau}_p^{\text{exact}} - \boldsymbol{\tau}_p^{\text{ansatz}}\|_{L^2(\Omega)}^2, \quad (3.21)$$

where:

$$\boldsymbol{\tau}_p^{\text{exact}} = [\boldsymbol{\tau}_{p(1)}^{\text{exact}}, \boldsymbol{\tau}_{p(2)}^{\text{exact}}], \quad \boldsymbol{\tau}_p^{\text{ansatz}} = [\boldsymbol{\tau}_{p(1)}^{\text{ansatz}}, \boldsymbol{\tau}_{p(2)}^{\text{ansatz}}].$$

In both cases the matrices of the least squares problems are ill-conditioned and the truncated singular value decomposition should be applied to solve this issue. As in Section 3.1.1, the number of singular values retained in each optimization problem is chosen in order to minimize the error metric $\varepsilon_p(L^2)$, defined as:

$$\varepsilon_p(L^2) = \sum_{j=1}^M \|p_{\text{sol}}(t_j) - p_q(t_j)\|_{L^2(\Omega)}, \quad (3.22)$$

where $p_{\text{sol}}(t_j) = \sum_{i=1}^q b_i(t_j)\chi_i$ is found from the solution of the dynamical system at each time step, and $p_q(t_j)$ is the projection of the full order pressure on the space generated by q modes. The reasons for choosing a different error metric involving the pressure field is that the pressure corrections are here introduced to improve the accuracy of the pressure field. The number of singular values retained in problem 1 is called R_{p1} for the first and R_{p2} for the second optimization problem. The number of singular values retained in the third optimization problem 3.21 is $R_{p\text{tot}}$. The dynamical system obtained with the introduction of the new terms is the following one:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T\mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k}\mathbf{D}^k - \mathbf{E}^k\mathbf{a}) \right) + \tilde{\mathbf{A}}\mathbf{a} + \mathbf{a}^T\tilde{\mathbf{B}}\mathbf{a} + \tilde{\mathbf{H}}\mathbf{b}, \\ \mathbf{P}\mathbf{a} + \tilde{\mathbf{P}}\mathbf{a} = \mathbf{0}. \end{cases} \quad (3.23)$$

As we will see in the results Chapter, the correction terms $\boldsymbol{\tau}_{p(1)}$ and $\boldsymbol{\tau}_{p(2)}$ do not improve the solutions with respect to those obtained without corrections. For this reason, in Section 3.2, the only correction term added to the momentum equation will be $\boldsymbol{\tau}_u(\mathbf{a})$.

For the construction of the closure terms for pressure and velocity the supremizer modes are not considered since they induce instability in the reduced dynamical system. For this reason in the least squares problems $r' = N_u$ instead of $r = N_u + N_{\text{sup}}$ is considered.

3.2 Data-driven corrections for the PPE-ROM approach

In this Section, the approach considering the Poisson equation for pressure is analysed. This approach induces the definition of new correction terms, corresponding to terms in the Poisson equation, as showed in (3.24), where in this case $r = N_u$ and $q = N_p$.

$$\begin{cases} \left(-\frac{\partial \bar{\mathbf{u}}_d^r}{\partial t}, \boldsymbol{\phi}_i \right) + \nu \left(\nabla \cdot (\nabla \bar{\mathbf{u}}_d^r + (\nabla \bar{\mathbf{u}}_d^r)^T), \boldsymbol{\phi}_i \right) - ((\bar{\mathbf{u}}_d^r \cdot \nabla) \bar{\mathbf{u}}_d^r, \boldsymbol{\phi}_i) - (\nabla \bar{p}_d^r, \boldsymbol{\phi}_i) + \\ \quad + c_u (\boldsymbol{\tau}_u^{\text{SFS}}, \boldsymbol{\phi}_i) = 0 \text{ for } i = 1, \dots, r, \\ \left(\nabla \bar{p}_d^r, \nabla \chi_i \right) + (\nabla \cdot (\bar{\mathbf{u}}_d^r \otimes \bar{\mathbf{u}}_d^r), \nabla \chi_i) - \nu (\nabla \times \bar{\mathbf{u}}_d^r, \mathbf{n} \times \nabla \chi_i)_\Gamma - (\mathbf{n} \cdot \mathbf{R}_{d_t}^r, \chi_i)_\Gamma + \\ \quad + c_D (\boldsymbol{\tau}_D^{\text{SFS}}, \nabla \chi_i) + c_G (\boldsymbol{\tau}_G^{\text{SFS}}, \nabla \chi_i) = 0 \text{ for } i = 1, \dots, q. \end{cases} \quad (3.24)$$

The new exact closure terms introduced in the system (3.24) are:

$$\begin{aligned}\boldsymbol{\tau}_D^{\text{SFS}} &= \overline{\nabla p_d^r} - \nabla \bar{p}_d^r, \\ \boldsymbol{\tau}_G^{\text{SFS}} &= \overline{\nabla \cdot (\mathbf{u}_d \otimes \mathbf{u}_d)^r} - \nabla \cdot (\bar{\mathbf{u}}_d^r \otimes \bar{\mathbf{u}}_d^r).\end{aligned}$$

The parameters c_D and c_G added to the system (3.24) are equal to 1 only if the correction to which they are referred is considered in simulations. In particular:

- $c_D = 1$ if the correction corresponding to the term $\mathbf{D}\mathbf{b}$ is added in the Poisson equations for pressure, and $c_D = 0$ otherwise;
- $c_G = 1$ if the correction corresponding to the term $\mathbf{a}^T \mathbf{G}\mathbf{a}$ is added to the Poisson equation and it's null otherwise.

Then, the following terms are introduced:

$$\begin{aligned}\boldsymbol{\tau}^D \text{ such that } \tau_i^D &= \left(\boldsymbol{\tau}_D^{\text{SFS}}, \nabla \chi_i \right), \\ \boldsymbol{\tau}^G \text{ such that } \tau_i^G &= \left(\boldsymbol{\tau}_G^{\text{SFS}}, \nabla \chi_i \right).\end{aligned}\tag{3.25}$$

Adding the correction terms defined in (3.25), the dynamical system (2.11) becomes:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right) + c_u \boldsymbol{\tau}^u, \\ \mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} + c_D \boldsymbol{\tau}^D + c_G \boldsymbol{\tau}^G = \mathbf{0}. \end{cases}\tag{3.26}$$

In order to close the system (3.26) the following approximated expressions need to be found:

$$\boldsymbol{\tau}^u \approx \boldsymbol{\tau}^u(\mathbf{a}), \quad \boldsymbol{\tau}^D \approx \boldsymbol{\tau}^D(\mathbf{b}), \quad \boldsymbol{\tau}^G \approx \boldsymbol{\tau}^G(\mathbf{a}, \mathbf{b}).$$

3.2.1 Data-driven correction for term $\mathbf{D}\mathbf{b}$

For the term $\boldsymbol{\tau}^D$ the following ansatzes have been proposed and added into the reduced system:

1. $\boldsymbol{\tau}^D(\mathbf{b}) = \tilde{D}\mathbf{b}$, with a linear approximation;
2. $\boldsymbol{\tau}^D(\mathbf{b}) = \tilde{D}\mathbf{b} + \mathbf{b}^T \tilde{B}_P \mathbf{b}$, with a quadratic approximation, similar to the one proposed for the velocity data-driven correction.

Case 1: the linear correction

In order to compute the matrix \tilde{D} , the following optimization problem should be solved:

$$\min_{\tilde{D} \in \mathbb{R}^{q \times q}} \sum_{j=1}^M \|\boldsymbol{\tau}_D^{\text{exact}}(t_j) - \boldsymbol{\tau}_D^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2,\tag{3.27}$$

where the term $\boldsymbol{\tau}^{\text{exact}}(t_j)$ is computed starting from the snapshot vectors $\mathbf{b}_d^{\text{snap}}(t_j)$, satisfying the conditions:

$$b_{d_i}^{\text{snap}}(t_j) = (p_d(t_j), \chi_i) \quad \forall i = 1, \dots, d, j = 1, \dots, M.$$

The exact correction term is evaluated as follows:

$$\boldsymbol{\tau}_D^{\text{exact}}(t_j) = \left(\overline{\mathbf{D}_d \mathbf{b}_d^{\text{snap}}(t_j)^q} \right) - \mathbf{D} \mathbf{b}_q^{\text{snap}}(t_j),$$

where the matrix $\mathbf{D}_d \in \mathbb{R}^{d \times d}$ is defined in the following way:

$$(\mathbf{D}_d)_{ij} = (\nabla \chi_i, \nabla \chi_j), \quad \text{with } i, j = 1, \dots, d.$$

We recall that $d_p = d = 50$ in all simulations performed in this Thesis. For this reason, in all equations d_p is replaced by d for the sake of simplicity.

The approximated term is evaluated at each time step j :

$$\boldsymbol{\tau}_D^{\text{ansatz}}(t_j) = \tilde{D} \mathbf{b}_q^{\text{snap}}(t_j). \quad (3.28)$$

The optimization problem (3.27) is rewritten as a least squares problem where the following terms are defined:

- the snapshot matrix $\hat{X}_P \in \mathbb{R}^{M \times q}$. Indicating with $\hat{X}_{P_{j,\cdot}}$ the j -th row of the matrix, we have:

$$\hat{X}_{P_{j,\cdot}} = \mathbf{b}_q^{\text{snap}}(t_j);$$

- the matrix $\mathbf{R}_D \in \mathbb{R}^{M \times q}$ such that:

$$\mathbf{R}_{D_{j,\cdot}} = \boldsymbol{\tau}_D^{\text{exact}}(t_j) \quad \forall j = 1, \dots, M.$$

The optimization problem (3.27) is rewritten as:

$$\min_{\tilde{D} \in \mathbb{R}^{q \times q}} \|\mathbf{R}_D - \hat{X}_P \tilde{D}^T\|_F^2. \quad (3.29)$$

The problem (3.29) can be also seen as a set of q optimization problems, as follows:

$$\min_{\mathbf{o}_i, i=1, \dots, q} \|\mathbf{r}_i - \hat{X}_P \mathbf{o}_i\|_{L^2(\Omega)}^2,$$

where \mathbf{o}_i is the i -th row of matrix \tilde{D} and \mathbf{r}_i is the i -th column of matrix \mathbf{R}_D . As in [10] and [44], the problem (3.29) is ill-conditioned since the matrix \mathbf{D} has a very high condition number and, as done in the previous Sections, a truncated singular value decomposition is applied to matrix \mathbf{D} .

After \tilde{D} is found from the least squares problem, the Poisson equation in the dynamical system is:

$$\mathbf{D} \mathbf{b} + \mathbf{a}^T \mathbf{G} \mathbf{a} + \tilde{D} \mathbf{b} - \nu \mathbf{N} \mathbf{a} - \mathbf{L} = \mathbf{0}. \quad (3.30)$$

The ill-conditioning of the least squares problem leads to an ill-conditioning of the dynamical system considered. In order to fix this problem, the number R of singular values retained for matrix \mathbf{X}_P is the optimal one, i.e. the one that minimizes the error metric 3.22. The error metric to be minimized for terms in the Poisson equation concerns the pressure field. The reason for this choice is that the correction term $\tilde{A}\mathbf{a} + \mathbf{a}^T \tilde{B}\mathbf{a}$ has a role in improving the results mainly for the velocity field; it is experimentally showed that the correction terms added in the Poisson equation have no effect on the velocity field, but they significantly improve the pressure accuracy results.

Case 2: the quadratic correction

The linear correction described in the previous Section did not give good results in terms of improvement with respect to the standard ROM; it is also not a good approximation of the exact correction term, as shown in the results Chapter. For this reason, in this Section a quadratic correction is analysed too.

The correction term for $\mathbf{D}\mathbf{b}$ is modelled using the ansatz:

$$\boldsymbol{\tau}_D^{\text{ansatz}}(\mathbf{b}) = \tilde{D}\mathbf{b} + \mathbf{b}^T \tilde{B}_P \mathbf{b}, \quad (3.31)$$

where \tilde{B}_P is a 3-entries tensor.

To find \tilde{D} and \tilde{B}_P a similar procedure to the one presented in Section 3.1.1 is considered. The following terms are defined:

- q vectors $\mathbf{b}^{(1)}(t_j), \dots, \mathbf{b}^{(q)}(t_j)$ for each time step, such that:

$$\mathbf{b}^{(i)}(t_j) = b_i^{\text{snap}}(t_j) \begin{bmatrix} b_0^{\text{snap}}(t_j) \\ b_1^{\text{snap}}(t_j) \\ \dots \\ b_i^{\text{snap}}(t_j) \end{bmatrix} \in \mathbb{R}^i \text{ for } i = 1, \dots, q,$$

where $b_i^{\text{snap}}(t_j)$ is the i -th component of the snapshot vector at time step j ;

- q different matrices $\hat{X}_P^{(1)}, \dots, \hat{X}_P^{(q)}$, with $\hat{X}_P^{(i)} \in \mathbb{R}^{M \times i}$ such that:

$$\hat{X}_{P_j}^{(i)} = \mathbf{b}^{(i)}(t_j).$$

The optimization problem (3.27) can be expressed in the following way:

$$\min_{\substack{\tilde{D} \in \mathbb{R}^{q \times q}, \\ \tilde{B}_P \in \mathbb{R}^{q \times q \times q}}} \|\mathbf{R}_D - \hat{X}_P \tilde{D}^T - \sum_{i=1}^q \hat{X}_P^{(i)} (\tilde{B}_P^{(i)})^T\|_F^2, \quad (3.32)$$

where $\tilde{B}_P^{(i)}$ are blocks of the tensor \tilde{B}_P of dimension $i \times i$. In a more compact form the optimization problem (3.32) can be written as follows:

$$\min_{\mathbf{O}_D} \|\mathbf{R} - \mathbf{D}_P \mathbf{O}^T\|_F^2, \quad (3.33)$$

where:

$$\mathbf{D}_P = [\hat{X}_P, \hat{X}_P^{(1)}, \hat{X}_P^{(2)}, \dots, \hat{X}_P^{(q)}], \quad \mathbf{O}_D = [\tilde{D}, \tilde{B}_P^{(1)}, \tilde{B}_P^{(2)}, \dots, \tilde{B}_P^{(q)}].$$

The problem (3.33) can be also seen as a set of q optimization problems.

$$\min_{\mathbf{o}_{D_i}, i=1, \dots, q} \|\mathbf{r}_i - \mathbf{D}_P \mathbf{o}_{D_i}\|_{L^2(\Omega)}^2,$$

where \mathbf{o}_{D_i} is the i -th row of matrix \mathbf{O}_D and \mathbf{r}_{P_i} is the i -th column of matrix \mathbf{R}_P . As in the previous corrections, a truncated singular value decomposition is applied to matrix \mathbf{D}_P and the number of singular values retained is the one that optimizes the error metric (3.22).

The Poisson equation of the dynamical system becomes:

$$\mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} + \tilde{D}\mathbf{b} + \mathbf{b}^T \tilde{B}_P \mathbf{b} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} = \mathbf{0}. \quad (3.34)$$

3.2.2 Data-driven correction for term $\mathbf{a}^T \mathbf{G}\mathbf{a}$

For the term τ^G a similar ansatz to the one proposed in Section 3.1.1 has been considered:

$$\tau^G(\mathbf{a}) = \tilde{G}_A \mathbf{a} + \mathbf{a}^T \tilde{G}_B \mathbf{a}$$

The procedure followed to find matrices \tilde{G}_A and \tilde{G}_B is exactly the same explained in Section 3.1.1 for the computation of matrices \tilde{A} and \tilde{B} , but in this case the exact term is expressed as:

$$\tau_G^{\text{exact}} = \overline{(\mathbf{a}_d^{\text{snap}}(t_j))^T \mathbf{G}_d \mathbf{a}_d^{\text{snap}}(t_j)} - (\mathbf{a}_r^{\text{snap}}(t_j))^T \mathbf{G}_r \mathbf{a}_r^{\text{snap}}(t_j),$$

where tensor $\mathbf{G}_d \in \mathbb{R}^{d \times d \times d}$ is defined as:

$$\mathbf{G}_{d_{ijk}} = (\nabla \chi_i, \nabla \cdot (\phi_j \otimes \phi_k)), \quad \text{with } i, j, k = 1, \dots, d$$

The Poisson equation considering just the correction for the term containing tensor \mathbf{G} can be rewritten as:

$$\mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} + \tilde{G}_A \mathbf{a} + \mathbf{a}^T \tilde{G}_B \mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} = \mathbf{0}. \quad (3.35)$$

3.2.3 Combined data-driven corrections proposals

Other attempts have been done in order to provide a more compact ansatz in equations. In particular, we have tried to approximate more than one correction terms with a unique ansatz. The following ansatzes have been considered and simulated:

1. Joint correction to $\mathbf{D}\mathbf{b}$ and $\mathbf{a}^T \mathbf{G}\mathbf{a}$, with the following ansatz:

$$\tau_{\text{DG}}^{\text{joint}}(\mathbf{a}, \mathbf{b}) = \tilde{D}_{pg} \mathbf{b} + \mathbf{a}^T \tilde{B}_{pg} \mathbf{a}.$$

2. Joint correction to $\mathbf{D}\mathbf{b}$ and $\mathbf{a}^T \mathbf{G}\mathbf{a}$ as a function of vector of coefficients $\mathbf{ab} = (\mathbf{a}, \mathbf{b}) \in \mathbb{R}^{N_u + N_p}$, with the following ansatz:

$$\tau_{\text{DG}}^{\text{joint}, ab}(\mathbf{ab}) = \tilde{I}_A \mathbf{ab} + \mathbf{ab}^T \tilde{I}_B \mathbf{ab}.$$

3. Joint correction to $\mathbf{D}\mathbf{b}$ and $\mathbf{a}^T\mathbf{G}\mathbf{a}$ in Poisson equation and $\mathbf{a}^T\mathbf{C}\mathbf{a}$ in momentum equation, using the following ansatz:

$$\tau_{\text{DCG}}^{\text{joint},ab}(\mathbf{a}\mathbf{b}) = \tilde{J}_A\mathbf{a}\mathbf{b} + \mathbf{a}\mathbf{b}^T\tilde{J}_B\mathbf{a}\mathbf{b}.$$

In all previous cases just one least squares problem is solved to find the unknown matrices appearing in the ansatz expression.

Case 1

In the first case the least squares problem to be solved is:

$$\min_{\substack{\tilde{D}_{pg} \in \mathbb{R}^{q \times q}, \\ \tilde{B}_{pg} \in \mathbb{R}^{q \times r \times r}}} \sum_{j=1}^M \|\tau_{\text{DG}}^{\text{exact}}(t_j) - \tau_{\text{DG}}^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2, \quad (3.36)$$

where the exact correction term is:

$$\tau_{\text{DG}}^{\text{exact}}(t_j) = \tau_D^{\text{exact}}(t_j) + \tau_G^{\text{exact}}(t_j) \quad \forall j = 1, \dots, M, \quad (3.37)$$

and the ansatz is:

$$\tau_{\text{DG}}^{\text{joint}}(\mathbf{a}, \mathbf{b}) = \tilde{D}_{pg}\mathbf{b} + \mathbf{a}^T\tilde{B}_{pg}\mathbf{a}.$$

In this case the optimization problem (3.36) becomes:

$$\min_{\substack{\tilde{D}_{pg} \in \mathbb{R}^{q \times q}, \\ \tilde{B}_{pg} \in \mathbb{R}^{q \times r \times r}}} \|\mathbf{R}_{\text{DG}} - \hat{X}_P\tilde{D}_{pg}^T - \sum_{i=1}^q \hat{X}^{(i)}(\tilde{B}_{pg}^{(i)})^T\|_F^2, \quad (3.38)$$

where $\tilde{B}_{pg}^{(i)}$ are blocks of the tensor \tilde{B}_{pg} of dimension $i \times i$. Matrix \mathbf{R}_{DG} in (3.38) is defined as:

$$\mathbf{R}_{\text{DG},j,\cdot} = \tau_{\text{DG}}^{\text{exact}}(t_j) \quad \forall j = 1, \dots, M.$$

Problem (3.38) is seen as follows:

$$\min_{\mathbf{O}_{\text{DG}}} \|\mathbf{R}_{\text{DG}} - \mathbf{D}_{\text{DG}}\mathbf{O}_{\text{DG}}^T\|_F^2, \quad (3.39)$$

where:

$$\mathbf{D}_{\text{DG}} = [\hat{X}_P, \hat{X}^{(1)}, \hat{X}^{(2)}, \dots, \hat{X}^{(q)}], \quad \mathbf{O}_{\text{DG}} = [\tilde{D}_{pg}, \tilde{B}_{pg}^{(1)}, \tilde{B}_{pg}^{(2)}, \dots, \tilde{B}_{pg}^{(q)}].$$

The problem (3.39) is decomposed into q different optimization problems and a truncated singular value decomposition is applied to matrix \mathbf{D}_{DG} . The error metric minimized is still the one involving pressure (3.22), since terms involved in the correction are contained in the Poisson equation for pressure and they're supposed to improve the results for pressure. Adding the correction term obtained using this approach, the Poisson equation becomes:

$$\mathbf{D}\mathbf{b} + \mathbf{a}^T\mathbf{G}\mathbf{a} - \nu\mathbf{N}\mathbf{a} - \mathbf{L} + \tilde{D}_{pg}\mathbf{b} + \mathbf{a}^T\tilde{B}_{pg}\mathbf{a} = \mathbf{0}.$$

Case 2

In this Section, a particular case is considered. Since term $\mathbf{D}\mathbf{b}$ depends just on pressure modes, whereas term $\mathbf{a}^T\mathbf{G}\mathbf{a}$ depends on both velocity and pressure modes, the two terms can be merged into a unique least squares problem involving the total vector of coefficients $\mathbf{ab} = (\mathbf{a}, \mathbf{b})$. In particular, calling $r_{\text{tot}} = r + q$ the least squares problem is:

$$\min_{\substack{\tilde{I}_A \in \mathbb{R}^{q \times r_{\text{tot}}}; \\ \tilde{I}_B \in \mathbb{R}^{q \times r_{\text{tot}} \times r_{\text{tot}}}}} \sum_{j=1}^M \|\tau_{\text{DG}}^{\text{exact}}(t_j) - \tau_{\text{joint, ab}}^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2, \quad (3.40)$$

where the ansatz for the correction is:

$$\tau_{\text{joint, ab}}^{\text{ansatz}}(\mathbf{ab}) = \tilde{I}_A \mathbf{ab} + \mathbf{ab}^T \tilde{I}_B \mathbf{ab}.$$

Then the following terms are defined:

- the matrix $\hat{ab} \in \mathbb{R}^{M \times r_{\text{tot}}}$ such that $\hat{ab}_{j,\cdot} = \mathbf{ab}_r^{\text{snap}}(t_j) = (\mathbf{a}_r^{\text{snap}}(t_j), \mathbf{b}_r^{\text{snap}}(t_j))$;
- r_{tot} vectors $\mathbf{ab}^{(1)}(t_j), \dots, \mathbf{ab}^{(r_{\text{tot}})}(t_j)$ for each time step, such that:

$$\mathbf{ab}^{(i)}(t_j) = ab_i^{\text{snap}}(t_j) \begin{bmatrix} ab_0^{\text{snap}}(t_j) \\ ab_1^{\text{snap}}(t_j) \\ \dots \\ ab_i^{\text{snap}}(t_j) \end{bmatrix} \in \mathbb{R}^i \text{ for } i = 1, \dots, r_{\text{tot}},$$

where $ab_i^{\text{snap}}(t_j)$ is the i -th component of the snapshot joint vector at time step j . In this way mixed terms of pressure and velocity snapshot coefficients are also taken into account;

- the matrix $\mathbf{D}_{ab} = [\hat{ab}, \hat{ab}^{(1)}, \dots, \hat{ab}^{(r_{\text{tot}})}]$;
- the matrix $\mathbf{O}_{ab} = [\tilde{I}_A, \tilde{I}_B^{(1)}, \dots, \tilde{I}_B^{(r_{\text{tot}})}]$.

The optimization problem (3.40) then becomes:

$$\min_{\mathbf{O}_{ab}} \|\mathbf{R}_{\text{DG}} - \mathbf{D}_{ab} \mathbf{O}_{ab}^T\|_F^2. \quad (3.41)$$

Problem (3.41) can also be decomposed into a set of q different least squares problems and as in the other cases the truncated SVD is applied.

Adding the correction term to the Poisson equation, it becomes:

$$\mathbf{D}\mathbf{b} + \mathbf{a}^T\mathbf{G}\mathbf{a} - \nu\mathbf{N}\mathbf{a} - \mathbf{L} + \tilde{I}_A \mathbf{ab} + \mathbf{ab}^T \tilde{I}_B \mathbf{ab} = \mathbf{0}.$$

Case 3

In the last case, the parameters in system (3.26) are such that $c_u = c_D = c_G = 1$, i.e. all the correction terms are considered in the reduced system. In this case a unique least

squares problem is solved in order to find the three data-driven terms. The optimization problem is:

$$\min_{\substack{\tilde{J}_A \in \mathbb{R}^{r_{\text{tot}} \times r_{\text{tot}}}; \\ \tilde{J}_B \in \mathbb{R}^{r_{\text{tot}} \times r_{\text{tot}} \times r_{\text{tot}}}}} \sum_{j=1}^M \|\boldsymbol{\tau}_{DCG,ab}^{\text{exact}}(t_j) - \boldsymbol{\tau}_{DCG,ab}^{\text{ansatz}}(t_j)\|_{L^2(\Omega)}^2. \quad (3.42)$$

The exact term is defined as follows:

$$\boldsymbol{\tau}_{DCG,ab}^{\text{exact}}(t_j) = (\boldsymbol{\tau}_u^{\text{exact}}(t_j), \boldsymbol{\tau}_D^{\text{exact}}(t_j) + \boldsymbol{\tau}_G^{\text{exact}}(t_j)) \quad \forall j = 1, \dots, M.$$

The ansatz is:

$$\boldsymbol{\tau}_{DCG,ab}^{\text{ansatz}}(\mathbf{ab}) = \tilde{J}_A \mathbf{ab} + \mathbf{ab}^T \tilde{J}_B \mathbf{ab},$$

where matrices $\tilde{J}_A \in \mathbb{R}^{r_{\text{tot}} \times r_{\text{tot}}}$ and $\tilde{J}_B \in \mathbb{R}^{r_{\text{tot}} \times r_{\text{tot}} \times r_{\text{tot}}}$ are computed through a similar procedure to the one used in Case 2. The final correction to be inserted in the system can be divided in two vectors:

$$\tilde{J}_A \mathbf{ab} + \mathbf{ab}^T \tilde{J}_B \mathbf{ab} = (\mathbf{J}_1, \mathbf{J}_2) \quad \text{where } \mathbf{J}_1 \in \mathbb{R}^{N_u}, \mathbf{J}_2 \in \mathbb{R}^{N_p}.$$

The new dynamical system with data-driven terms becomes:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}} = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a} - \mathbf{a}^T \mathbf{C}\mathbf{a} - \mathbf{H}\mathbf{b} + \tau \left(\sum_{k=1}^{N_{\text{BC}}} (U_{\text{BC},k} \mathbf{D}^k - \mathbf{E}^k \mathbf{a}) \right) + \mathbf{J}_1, \\ \mathbf{D}\mathbf{b} + \mathbf{a}^T \mathbf{G}\mathbf{a} - \nu \mathbf{N}\mathbf{a} - \mathbf{L} + \mathbf{J}_2 = \mathbf{0}. \end{cases} \quad (3.43)$$

Chapter 4

Numerical Results

This Chapter is dedicated to the presentation and discussion of the numerical simulations results. The case study considered is that of the turbulent incompressible flow around a circular cylinder. This test case is a common benchmark for unsteady turbulent flows, given its simple posure and yet significant practical applications. In the offline-online procedure, the offline phase was carried out making use of the C++ open source software *OpenFOAM* [16]; the reduced system of equations has been assembled with the C++ library *ITHACA-FV* [17, 18, 19]. The online part consists in a series of simulations, in which the data-driven terms examined in Chapter 3 are computed and included in the reduced systems; the online part is carried out in a specifically developed *Python* script.

The results of the FOM simulations are compared with the results obtained by solving the reduced order dynamical systems with and without the correction terms.

4.1 The Case Study: Unsteady Flow around a Circular Cylinder

The case study analysed is that of the unsteady incompressible flow past a circular cylinder. It is a well-known case study in literature, analysed in detail in [48, 49, 50, 51]. The case study has been simulated with reduced order techniques in the past, for instance in [18, 52, 10, 53, 54]. Given the inherent 2D nature of the vortex shedding phenomenon, the case is studied in two dimensions and the mesh used is composed by 11644 cells. The mesh and the boundary conditions set for velocity and pressure are represented in Figure 4.1 [1]. The diameter of the cylinder is $D = 1$ m, the fluid kinematic viscosity $\nu = 1 \times 10^{-4} \text{ m}^2 \text{ s}^{-1}$ and the velocity at the inlet is horizontal and fixed at $U_{in} = 5 \text{ m s}^{-1}$, which corresponds to $Re = 5 \times 10^4$. The penalty parameter τ appearing in the boundary conditions in (2.12) and (2.13) is set to $\tau = 1000$. This parameter is usually set according to a sensitivity analysis, as discussed in [55], but a too large value for τ can lead to an ill-conditioning of the dynamical system [1].

As mentioned, the software *OpenFOAM* is used to evaluate the full order fields for the offline stage. In particular, we made use of the unsteady solver *pimpleFoam*, which implements the PIMPLE algorithm, and of the $\kappa - \omega$ two equations model for the turbulence

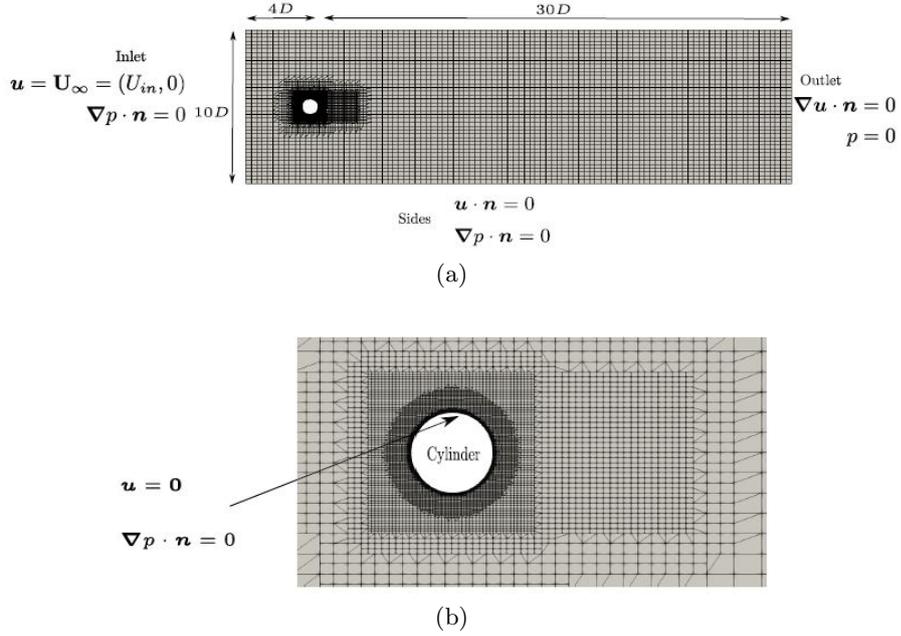


Figure 4.1: (a) The mesh used in simulations. (b) The mesh zoomed around the cylinder. Image taken from [1].

treatment. The time step considered in the simulations is 0,0002 s, but the time snapshots are taken every 20 time steps, i.e. every 0,004 s. The simulation last 100 seconds.

Then 5000 time snapshots, corresponding to the time window [79.992,99.992], are selected. The Proper Orthogonal Decomposition is performed through the C++ library *ITHACA-FV* on the aforementioned interval: the POD modes for velocity, pressure, and supremizer fields (in the case of SUP-ROM approach), are obtained from the snapshot matrices.

In the online phase, the reduced order dynamical system is computed and solved. In this stage different existing and newly proposed data-driven approaches are considered to compare their results to those obtained with the approach presented in [1]. To better guide the reader through the numerical tests and their rationale, a brief summary of their evolution throughout this Thesis work is here provided.

1. A preliminary step is to analyse the POD results. An analysis on the eigenvalues decay for pressure and velocity is carried out in order to find which are the *under-resolved* and *marginally-resolved* regimes. The projected solutions for different numbers of modes are studied in terms of relative errors of velocity and pressure fields with respect to the high fidelity solutions. We will refer to these errors as *reconstruction errors*.
2. Secondly, the solution of the SUP-ROM given by system (3.9), where a supremizer approach is applied without any correction term, is studied. In particular, the stability issues caused by the supremizer fields are discussed.

3. The data-driven methods proposed in [10, 11, 46] are then applied and fitted to the SUP-ROM approach. Correction/closure terms depending on the reduced order velocity coefficients are added to the momentum equation in system (3.9), and the influence of term $\boldsymbol{\tau}_u(\mathbf{a}^i)$, which we refer to as correction term for velocity, is analysed. The dynamical system (3.9) with $c_u = 1$ and $c_{p(1)} = c_{p(2)} = 0$ produces a better approximation of the velocity field than the formulation without any correction term. However, in these tests the reproduction of the pressure field is not significantly improved with respect to the solution of the system without the velocity correction term.
4. Since in several applications it is important to obtain an accurate pressure field prediction, new pressure corrections for the SUP-ROM approach are also introduced. To this end, the dynamical system (3.9) with $c_u = c_{p(1)} = c_{p(2)} = 1$ is solved, making use of both the velocity and the pressure correction/closure terms. As will be documented, the results still show no improvement of the pressure field. One of the possible reasons for this is that, as it is the case for its velocity counterpart, the pressure correction is mainly designed to surrogate effects of nonlinear terms. Since no nonlinear terms involve pressure, the corresponding correction/closure term is less effective than the velocity correction.
5. The next step is to consider a different approach where a dedicated equation for pressure is considered, in particular the PPE-ROM is taken into account. In the PPE formulation, new pressure correction terms for the Poisson equation are added and different ansatzes are proposed. The results lead to an evident improvement of the reduced pressure field.
6. The last step of our analysis is to compare the results obtained with the SUP-ROM and the PPE-ROM approach with those obtained in previous works [1], which was focused on a different data-driven approach where the reduced eddy viscosity field was introduced in the reduced NSE. Finally also the combination of the two data-driven approaches is studied and implemented, leading to even better results for the pressure and the velocity fields.

For the supremizer approach the most general form in which also the turbulent data-driven terms are introduced is the following:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}}^i = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a}^i - (\mathbf{a}^i)^T \mathbf{C}\mathbf{a}^i - \mathbf{H}\mathbf{b}^i + c_u \boldsymbol{\tau}_u(\mathbf{a}^i, \mathbf{b}^i) & + \\ + c_{p(1)} \boldsymbol{\tau}_{p(1)}(\mathbf{a}^i, \mathbf{b}^i) + c_t ((\mathbf{g}^i)^T (\mathbf{C}_{T1} + \mathbf{C}_{T2}) \mathbf{a}^i) & \text{at each } i = 1, \dots, M, \\ \mathbf{P}\mathbf{a}^i + c_{p(2)} \boldsymbol{\tau}_{p(2)}(\mathbf{a}^i, \mathbf{b}^i) = \mathbf{0} & \text{at each } i = 1, \dots, M, \end{cases} \quad (4.1)$$

where M is the total number of time steps in the online phase. In most of the simulations we will consider $M = 500$, corresponding to 2 s; in order to analyse the time extrapolation efficiency of the method we will consider $M = 2000$ in some simulations, corresponding to 8 s.

For the Poisson approach the dynamical system with the addition of the turbulent

terms is the following:

$$\begin{cases} \mathbf{M}\dot{\mathbf{a}}^i = \nu(\mathbf{B} + \mathbf{B}_T)\mathbf{a}^i - (\mathbf{a}^i)^T \mathbf{C}\mathbf{a}^i - \mathbf{H}\mathbf{b}^i + c_u \boldsymbol{\tau}_u(\mathbf{a}^i, \mathbf{b}^i) & + \\ + c_t ((\mathbf{g}^i)^T (\mathbf{C}_{T1} + \mathbf{C}_{T2})\mathbf{a}^i) & \text{at each } i = 1, \dots, M, \\ \mathbf{D}\mathbf{b}^i + (\mathbf{a}^i)^T \mathbf{G}\mathbf{a}^i - \nu \mathbf{N}\mathbf{a}^i - \mathbf{L} + c_t ((\mathbf{g}^i)^T (\mathbf{C}_{T3} + \mathbf{C}_{T4})\mathbf{a}^i) & + \\ + c_D \boldsymbol{\tau}_D(\mathbf{a}^i, \mathbf{b}^i) + c_G \boldsymbol{\tau}_G(\mathbf{a}^i, \mathbf{b}^i) = \mathbf{0} & \text{at each } i = 1, \dots, M. \end{cases} \quad (4.2)$$

In systems (4.1) and (4.2), the parameter c_t is introduced to add or remove the turbulence treatment. More specifically, when $c_t = 1$ the eddy viscosity turbulence model is added to the reduced system.

As for the time integration two different approaches are taken into account; the time derivative $\dot{\mathbf{a}}^i$ appearing in the momentum equation in (4.1) and (4.2) is computed following two implicit time schemes, namely:

- implicit Euler time scheme:

$$\dot{\mathbf{a}}^i = \frac{\mathbf{a}^i - \mathbf{a}^{i-1}}{\Delta t}, \quad (4.3)$$

where Δt is the time interval and i indicates the time step at which the coefficients are evaluated;

- implicit second order time scheme:

$$\begin{aligned} \dot{\mathbf{a}}^i &= \frac{\mathbf{a}^i - \mathbf{a}^{i-1}}{\Delta t}, \text{ for } i = 1, 2, \\ \dot{\mathbf{a}}^i &= \frac{3\mathbf{a}^i - 4\mathbf{a}^{i-1} + \mathbf{a}^{i-2}}{2\Delta t}, \text{ for } i > 2. \end{aligned} \quad (4.4)$$

It is worth remarking that the second order time scheme corresponds to the scheme implemented in *OpenFOAM* and used to solve the full order problem.

The results are usually displayed in terms of the percentage $L^2(\Omega)$ error of the absolute value of velocity and pressure. The errors are evaluated with respect to the full order fields and compared with the reconstruction errors. The projection of the full order solution is the best possible result which can be achieved with a given amount of modes. Thus, the solution of the reduced system cannot improve with respect to that projection.

The percentage errors with respect to the full order fields at each j -th time step are evaluated in the following way:

$$\varepsilon_u(t_j) = \frac{\|\mathbf{u}_r^{abs}(\mathbf{x}, t_j) - \mathbf{u}_d^{abs}(\mathbf{x}, t_j)\|_{L^2(\Omega)}}{\|\mathbf{u}_d^{abs}(\mathbf{x}, t_j)\|_{L^2(\Omega)}}, \quad \varepsilon_p(t_j) = \frac{\|p_r(\mathbf{x}, t_j) - p_d(\mathbf{x}, t_j)\|_{L^2(\Omega)}}{\|p_d(\mathbf{x}, t_j)\|_{L^2(\Omega)}}. \quad (4.5)$$

In order to compute the errors defined in (4.5), the following vectors are considered:

- the reduced order field of the velocity, taken in its absolute value $\mathbf{u}_r^{abs}(\mathbf{x}, t_j)$ at each j -th time step. It is evaluated taking the norm of the reduced vector field for velocity at each time step, expressed as follows:

$$\mathbf{u}_r(\mathbf{x}, t_j) = \sum_{i=1}^r a_i(t_j) \phi_i(\mathbf{x}),$$

where the coefficients $a_i(t_j)$ are the solutions of the dynamical systems (4.1) (in the supremizer approach) and (4.2) (in the Poisson approach). The number of modes for velocity is $r = N_u + N_{sup}$ in the supremizer approach, and $r = N_u$ in the Poisson approach;

- the approximated full order field of the absolute value of velocity $\mathbf{u}_d^{abs}(\mathbf{x}, t_j)$, at each time step. It is evaluated starting from the first d modes, where $d = 100$ when a supremizer approach is considered, and $d = 50$ when a Poisson approach is considered. The absolute value of the velocity at each cell and at each time step is evaluated starting from the following approximated field:

$$\mathbf{u}_d(\mathbf{x}, t_j) = \sum_{i=1}^d a_i^{\text{snap}}(t_j) \phi_i(\mathbf{x}).$$

When a supremizer approach is considered: $\{\phi_i\}_{i=51}^{100} = \{\mathbf{s}_i(\chi_i)\}_{i=1}^{50}$ are the supremizer modes;

- the reduced order field for pressure $p_r(\mathbf{x}, t_j)$ computed as follows:

$$p_r(\mathbf{x}, t_j) = \sum_{i=1}^q b_i(t_j) \chi_i(\mathbf{x}),$$

where the coefficients $(b_i)_{i=1}^q$ are computed as the solution of the reduced dynamical system;

- the approximated full order field of pressure $p_d(\mathbf{x}, t_j)$, computed starting from the first $d_p = d = 50$ modes for pressure:

$$p_d(\mathbf{x}, t_j) = \sum_{i=1}^{d_p} b_i^{\text{snap}}(t_j) \chi_i(\mathbf{x}).$$

4.2 Modal Decomposition Effectiveness

In the present Section, an analysis of the eigenvalues corresponding to the velocity, supremizer and pressure modes is carried out, in order to capture the number of modes retaining most of the system's energy. In Figure 4.2 the plot of the eigenvalues decay shows that the first mode of both velocity and pressure retains more than the 90% of the system's energy.

Looking at Figure 4.2, 3, 4, 5 or 6 modes for velocity and pressure can be attributed to the *marginally-resolved* regime, which will be the main interest of our analysis.

The projection of the solution into the reduced order POD space is the best possible approximation among the solutions belonging to the same reduced space. Figure 4.3 displays the percentage L^2 reconstruction errors of velocity and pressure referred to different number of modes. As expected, the approximations of velocity and pressure fields become more accurate as the number of modes generating the POD space increases.

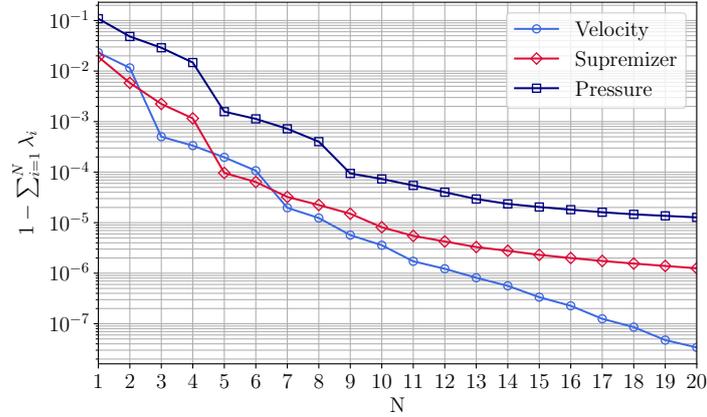


Figure 4.2: Decay of the eigenvalues corresponding to velocity, pressure and supremizer modes.

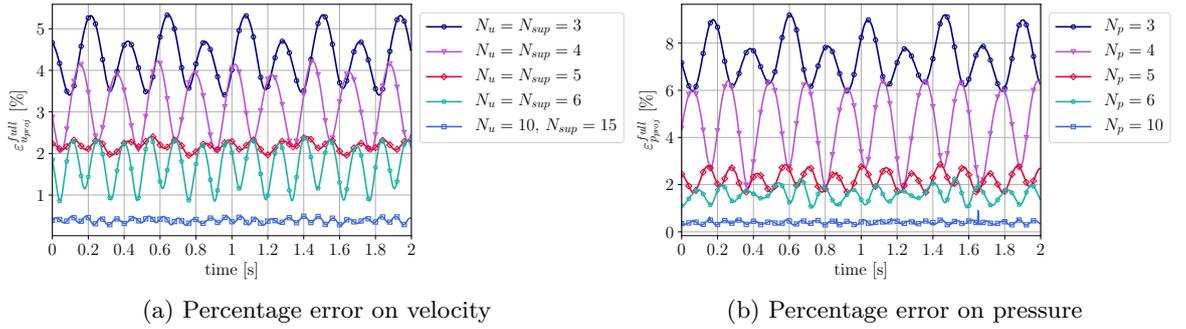


Figure 4.3: Percentage reconstruction errors for different numbers of modes.

4.3 Analysis of the POD Galerkin SUP-ROM without corrections

This Section is dedicated to the preliminary analysis of the solutions of the dynamical system which is not including data-driven corrections and turbulence model terms.

Figures 4.4 and 4.5 display the errors of the absolute value of velocity and of pressure with respect to the high-fidelity solutions, as a result of a 1st order time evolution scheme. Different combinations of number of modes are considered; the expectation is that the precision of the standard Galerkin-ROM would improve as the number of modes is increased. However, the *approximated* supremizer approach is characterized by stability issues particularly affecting the pressure field. As can be appreciated from Figure 4.4(b) when the number of velocity and pressure modes is greater than 10 and the number of supremizer modes is $N_{sup} = N_u = N_p$, the reduced pressure solution error significantly

increases. These issues are not as severe when $N_{sup} > N_p$, as shown in Figure 4.5. Considering for instance $N_u = N_p = 10$ and $N_{sup} = 15$, or $N_u = N_p = 20$ and $N_{sup} = 40$ results show an acceptable accuracy. Unfortunately, the stability issues are still evident when $N_u = N_p = 30$ and $N_{sup} = 50$.

This problem comes from the fact that the supremizer *approximated* procedure has been adopted, and it would be solved with an *exact* procedure [13, 17].

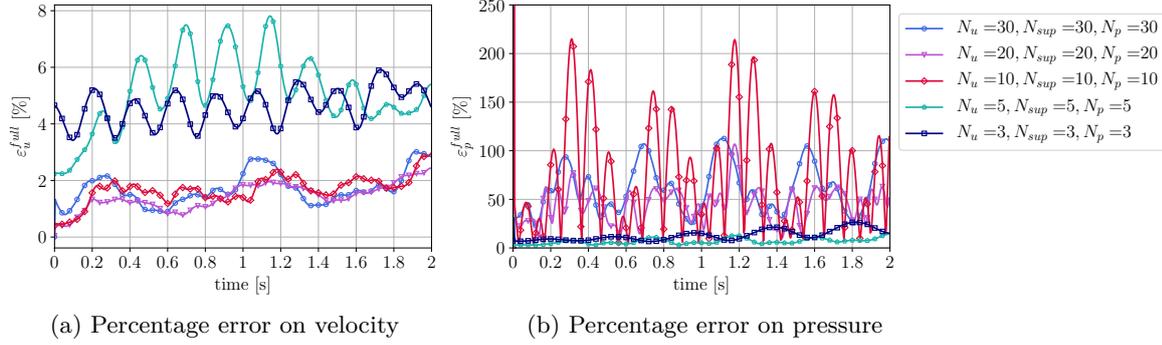


Figure 4.4: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p = N_{sup}$. In simulations time integration is carried out by means of a first order time scheme.

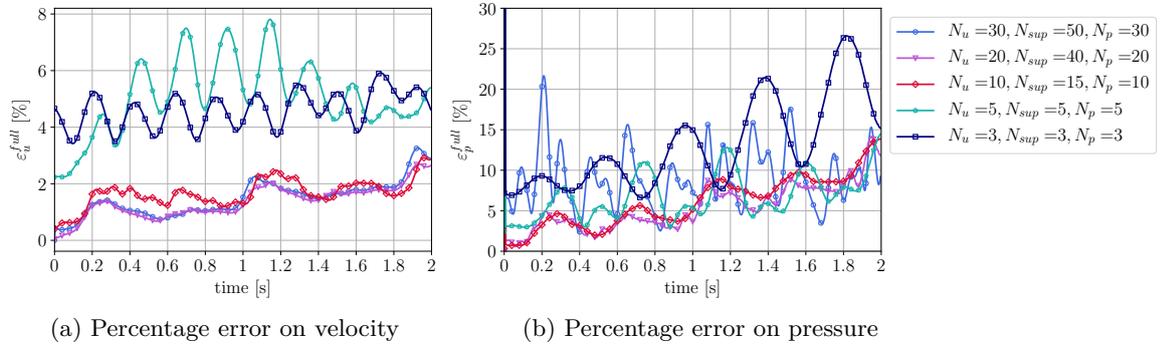


Figure 4.5: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$ and $N_{sup} > N_p$. In simulations time integration is carried out by means of a first order time scheme.

Figures 4.6 and 4.7 also display the results of simulations computed with a second order time scheme. Here, the same stability issue can be observed when $N_u \geq 10$ and the number of supremizer modes is equal to the number of pressure and velocity modes.

From the comparison between Figures 4.5 and 4.7 it can be seen that when a first

order time scheme is considered the results for the velocity field are similar to the one obtained with a second order time scheme, except for an explosion of the error just at the end of simulation for the second order, for the case $N_u = N_p = 30$, $N_{sup} = 50$. This fact is likely due to the reduced numerical dissipation associated with the second order numerical scheme used for time integration. Such lower dissipation might in fact expose the instabilities associated with the supremizer pressure treatment in a greater fashion.

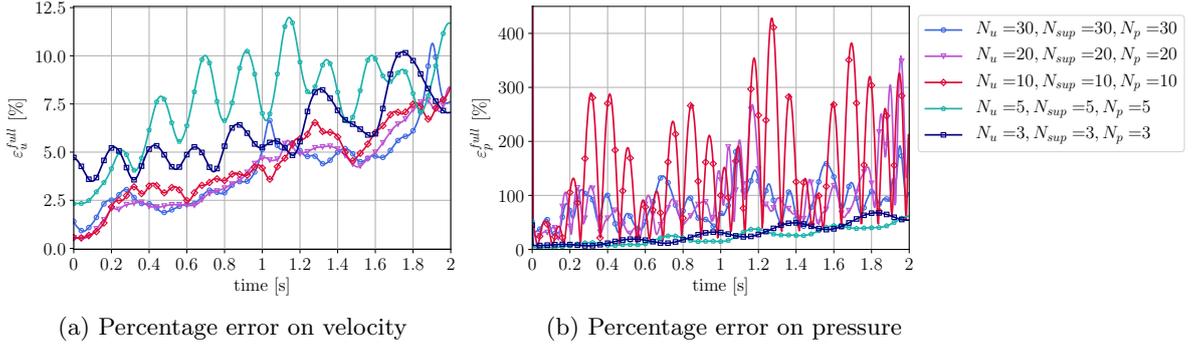


Figure 4.6: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p = N_{sup}$. In simulations time is evolved with a second order time scheme.

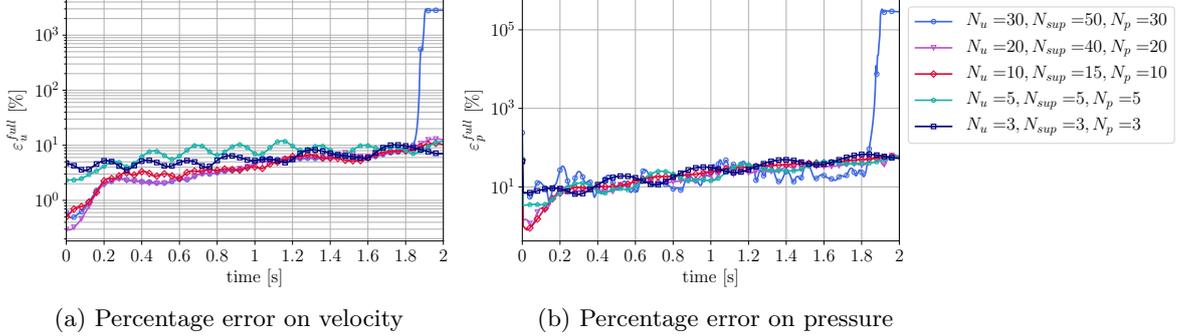


Figure 4.7: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$ and $N_{sup} > N_p$. In simulations time is evolved with a second order time scheme.

The present Section leads to the following conclusions:

- in most cases, in the standard SUP-ROM the number of supremizer modes should be higher than the number of pressure modes in order to avoid stability issues (Figures 4.4 and 4.6);
- the standard SUP-ROM performs better when a first order time scheme is considered,

because of the greater dissipation associated to it, as can be seen from the comparison between Figures 4.5 and 4.7;

- the standard SUP-ROM leads to more accurate results for the velocity field than for the pressure field; the main goal of the following Sections will be that of improving the pressure accuracy.

4.4 Effect of velocity correction in the SUP-ROM approach

This Section is dedicated to the analysis of the effect of the velocity correction terms on the dynamical system obtained with the supremizer enrichment and treated in Section 3.1.1. In particular the following studies are included:

- the problem of choosing R and R_c , the number of singular values in the truncated SVD of least squares problems (3.11) and (3.16), respectively;
- a comparison between the standard and the constrained velocity correction terms, considering different numbers of modes;
- study of the extrapolation efficiency of the method, building the correction terms from a reduced time interval;
- study of the $3S-DD-VMS-ROM$ approach developed in [46] for the construction of the velocity correction term.

4.4.1 Velocity correction: the influence of R

In this Section, the number of modes considered is fixed at $N_u = N_p = N_{sup} = 5$. This combination for the number of modes is chosen because it is not affected by stability issues, as can be appreciated from Figures 4.4 and 4.6. It also belongs to the *marginally-resolved* regime, which is the region on which our analysis is focused.

The solution for system (4.1) with $c_u = 1$ and $c_{p(1)} = c_{p(2)} = 0$ is computed. Figures 4.8 and 4.9 present the relative errors of velocity and pressure. In particular, Figure 4.8 refers to the solution of the system in which the standard velocity correction is added (Section 3.1.1), whereas Figure 4.9 refers to the constrained correction described in Section 3.1.2. Different curves correspond to different numbers of singular values R retained in the truncated singular value decomposition used to solve the least squares problem. From Figures 4.8 and 4.9, it is clear that there exists an *optimal* R , i.e. a value of R that minimizes the error with respect to the projection of the high fidelity solution. The solution for $R = 0$ corresponds to the solution of the dynamical reduced system without any correction term and it is used in Figures as a reference point for the curves corresponding to the correction effect: for small values of R the solution of the system in terms of velocity and pressure improves with respect to the solution of the standard system. For larger values of R , the reduced solution is not converging anymore to the full order one. For this reason, when a correction term is added to the reduced system the

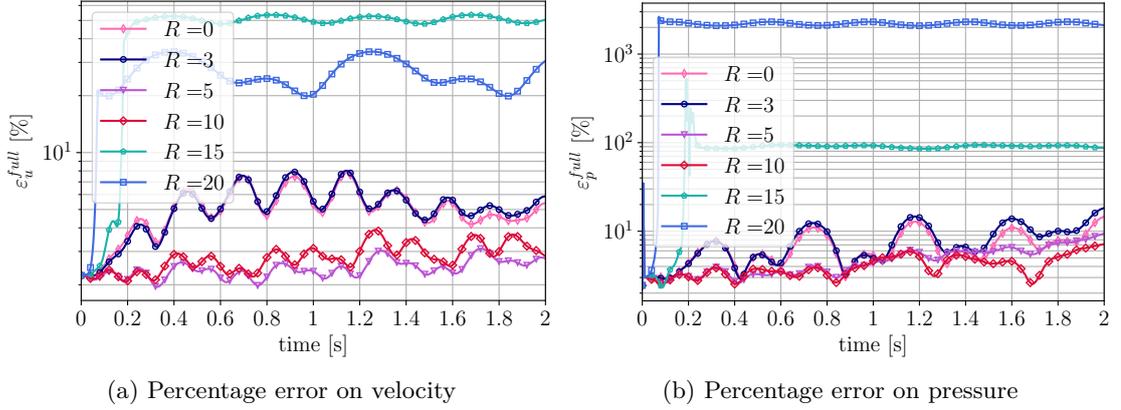


Figure 4.8: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. The correction term included is the unconstrained velocity correction.

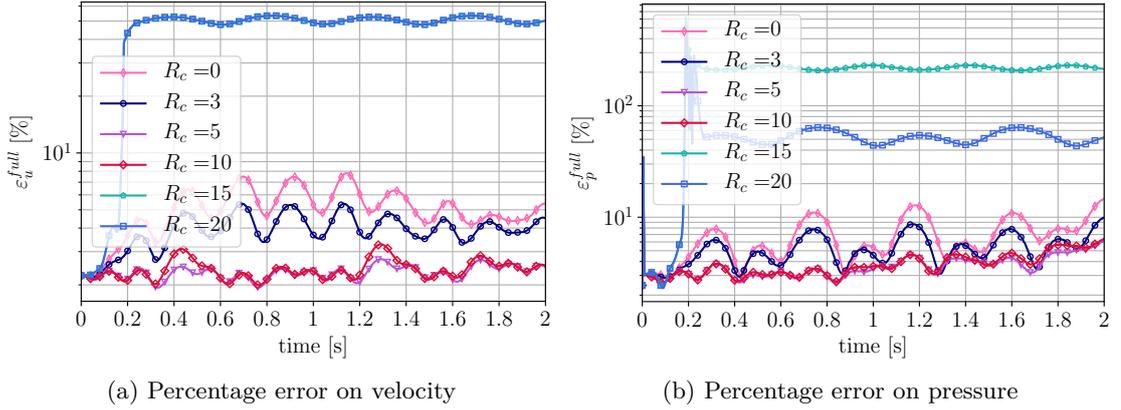


Figure 4.9: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. The correction term included is the constrained velocity correction.

number of singular values retained in the truncated SVD is optimized with respect to the error metric expressed in (3.15). The trend of the error metric for different values of R and R_c is represented in Figure 4.10.

It is worth reporting that at each value of R or R_c the optimization problems are aimed to minimize the difference between the approximated correction term τ_u^{ansatz} and the exact closure term τ_u^{exact} . However, the optimal values of R and R_c are the values for which the error metric (3.15) is minimized and they are not selected to minimize the distance from the exact correction term. For this reason, it can happen that the approximated correction performs better than the exact one in terms of accuracy with respect to the

full order fields, as can be seen from Figure 4.11.

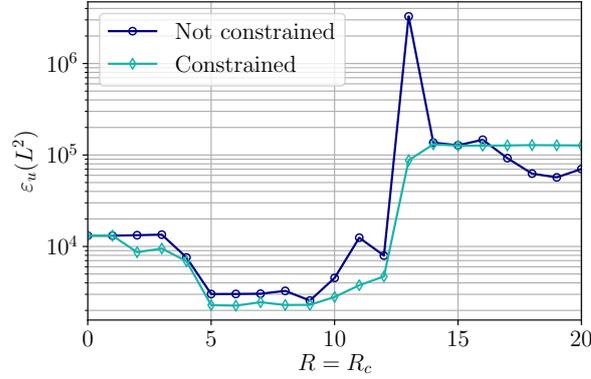
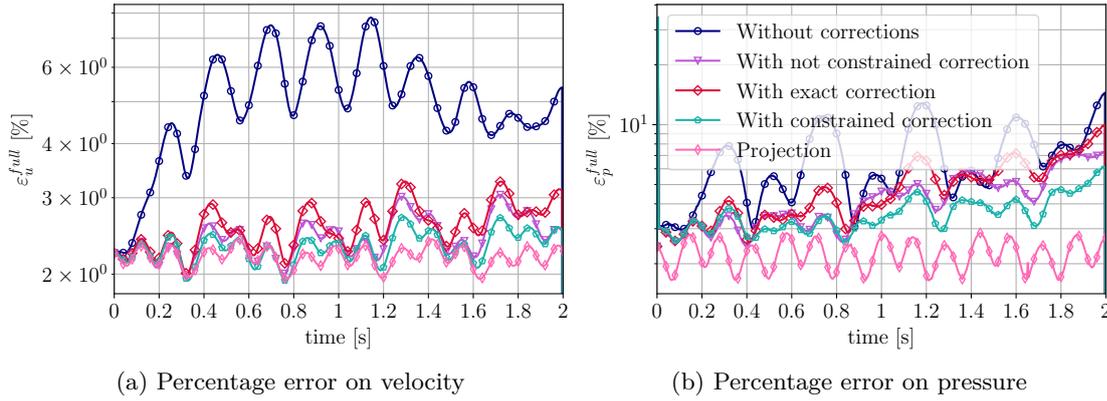


Figure 4.10: Variation of the metric $\varepsilon_u(L^2)$ varying the number of singular values retained in the singular value decomposition.



(a) Percentage error on velocity

(b) Percentage error on pressure

Figure 4.11: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$, including the simulation with the exact correction. The optimal R and R_c are considered.

In Table 4.1 the optimal numbers of singular values retained in the truncated SVD for the optimization problem in both the unconstrained and the constrained cases are reported. The optimal R or R_c depends not only on the degrees of freedom of the dynamical systems, but also on the scheme used for the approximation of the time derivative. In Table 4.1 the results for the first and the second order are displayed. From the Table, it can be noticed that in the *marginally-resolved* regime the optimal R and R_c are small values. This is due to the fact that the correction term in the reduced system corresponds to additional degrees of freedom, representing the interaction among the resolved and the unresolved modes.

Increasing values of R or R_c lead to larger degrees of freedom and to a higher complexity

Optimal number of singular values retained						
N_u	N_{sup}	N_p	optimal R (1 st or- der time scheme)	optimal R_c (1 st order time scheme)	optimal R (2 nd or- der time scheme)	optimal R_c (2 nd order time scheme)
1	1	1	1	1	1	1
2	2	2	4	2	4	2
3	3	3	3	4	2	4
4	4	4	8	8	5	5
5	5	5	9	6	9	6
6	6	6	18	15	9	15
7	7	7	21	7	12	13
8	9	8	10	14	14	14
9	10	9	9	21	9	20
10	15	10	16	22	16	22

Table 4.1: Optimal number of singular values retained in the truncated SVD, varying the number of modes considered in online simulations. Both optimization problems for the standard and constrained case are taken into account. The dynamical system is evolved according to an implicit Euler time scheme or to a second order implicit time scheme.

of the system, which can cause an ill-conditioning of the system itself. Moreover, larger values of R and R_c can affect the accuracy of the optimization problem, leading to an *overfitting* effect.

From the present Section, the following conclusions can be drawn:

- in the least squares problem which has to be solved to find the corrections, there exists an optimal value for the number of singular values in the truncated SVD. The optimal value will be considered in all the following simulations;
- the selected value optimizes the error on the velocity field, but not the difference with respect to the exact correction. Thus, it can provide better results than the exact correction.

4.4.2 Velocity correction: constrained and not constrained

This Section has the aim of making a comparison between the constrained and unconstrained corrections for the term τ_u . Parameters appearing in system (4.1) are set as $c_u = 1$, $c_{p(1)} = c_{p(2)} = 0$. Thus, in this Section the dynamical system only involving correction τ^u is considered. Both constrained and normal least squares problems are solved. As suggested in [11], the constrained method is more efficient than the unconstrained method when the number of modes is smaller, as can be seen from the comparison between Figures 4.12, 4.13 and 4.14.

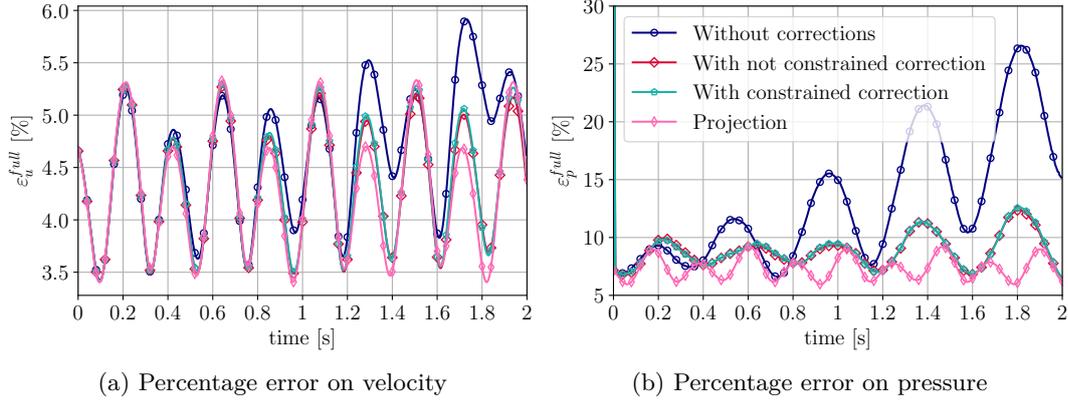


Figure 4.12: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity constrained and not.

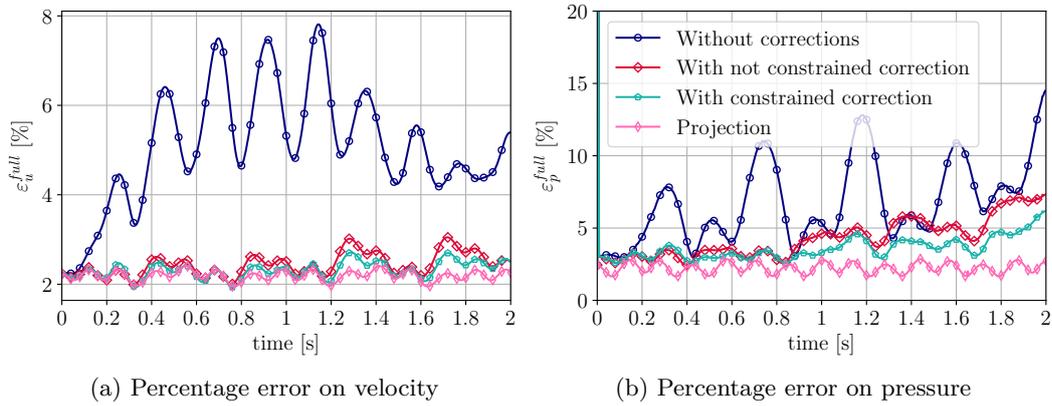


Figure 4.13: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not.

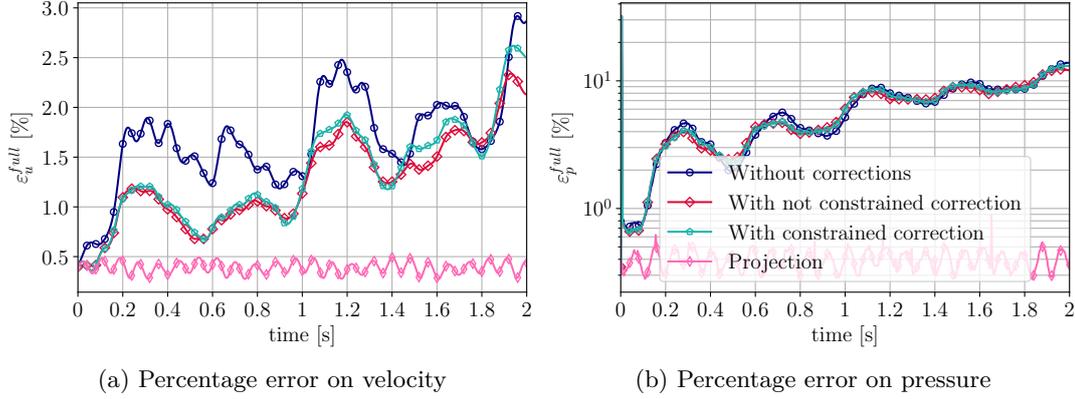


Figure 4.14: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 10$ and $N_{sup} = 15$. Results without any correction term, with the correction term for velocity constrained and not.

In Figure 4.12 it can be seen that when the number of modes $N_p = 3$ there is a larger improvement in terms of pressure accuracy than in the cases of larger N_p .

The constrained correction is derived including physical conditions in the optimization problem, which are supposed to positively influence the velocity and pressure fields. However, the gain in accuracy for the constrained method is typically marginal, except for lower modes, and the addition of constraints to the method is not sufficient to obtain an accurate pressure field.

In order to reach this goal, new correction terms are introduced and evaluated in the numerical simulations of the following Sections.

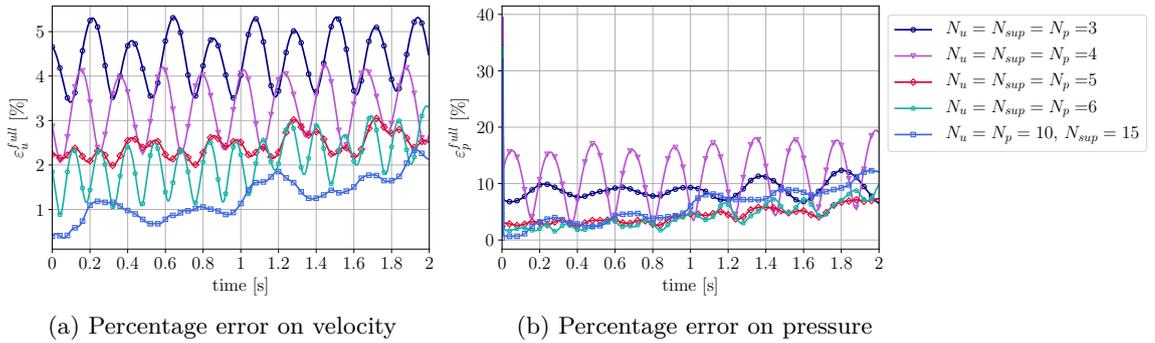


Figure 4.15: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the standard velocity correction is added to the dynamical system and the time is evolved according to a first order implicit scheme.

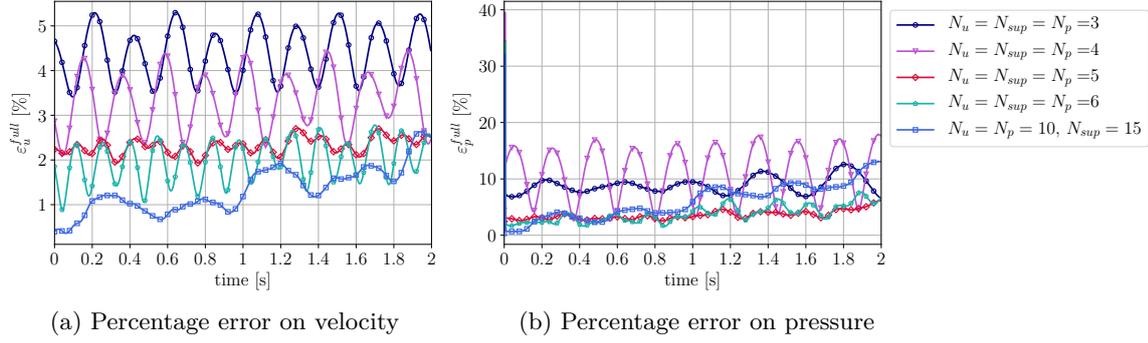


Figure 4.16: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the constrained velocity correction is added to the dynamical system and the time is evolved according to an Euler implicit scheme.

Figures 4.15 and 4.16 are also added in this Section in order to evaluate how the two types of velocity correction behave for different degrees of freedom, when the time derivative is computed according to a first order time scheme. In simulations the value set for R and R_c is the optimal one, so the results in the Figures are the best possible we can obtain by adding the velocity correction terms.

It is interesting to compare the plots obtained in this Section with the plots in Figure 4.3, which are the best possible results referred to the projection of the full order solution. The results obtained for the velocity field are very close to those obtained for the projection, especially for the *marginally-resolved* regime, i.e. $N_u = 3, 4, 5$ or 6 , whereas the correction is less effective as the number of modes grows. However, the reduced pressure field is not captured with as high accuracy as the reduced velocity.

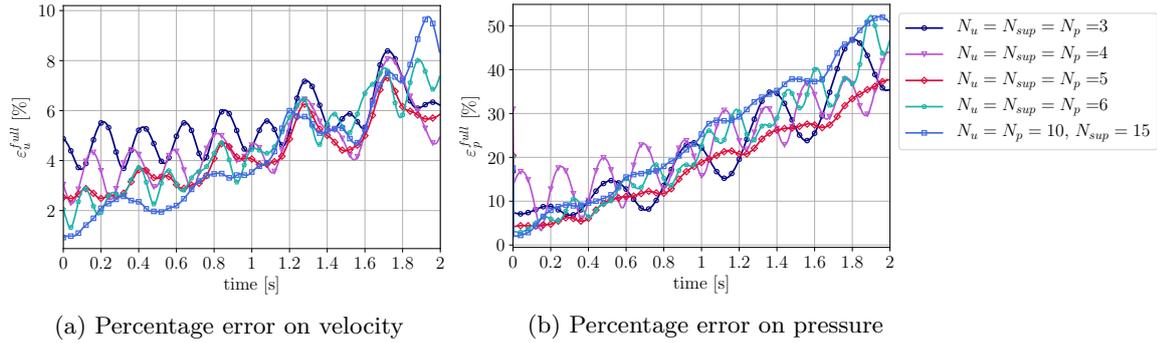


Figure 4.17: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the standard velocity correction is added to the dynamical system and the time is evolved according to a second order implicit scheme.

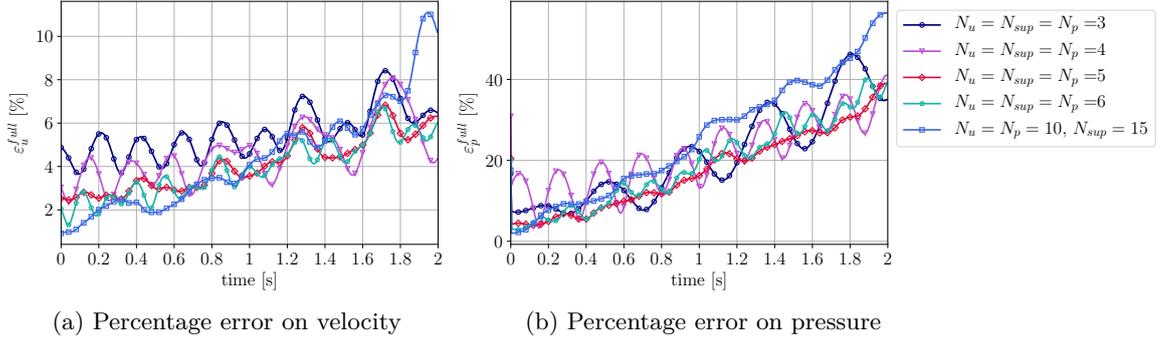


Figure 4.18: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering different numbers of modes. Results refer to the case in which the constrained velocity correction is added to the dynamical system and the time is evolved according to a second order implicit scheme.

In Figures 4.17 and 4.18 results for the standard and constrained velocity correction are presented in the case of a second order time evolution. From the comparison with Figures 4.15 and 4.16, it is clear that the corrections at the second order produce less accurate results than in the first order and show instability as time evolves. The fact that all the relative errors represented in Figures 4.17 and 4.18 increase with time might suggest that the second order time scheme is not able to dump possibly unstable higher modes, since it is a less dissipative integration scheme than the Euler scheme.

This Section leads to some important observations:

- the velocity correction, both constrained and not, improves the velocity and pressure fields with respect to the standard SUP-ROM;
- for the second order time scheme, results are more unstable in time since the scheme is less dissipative than the first order one;
- the constrained correction produces slightly better results for the *marginally-resolved* regime, but the effect is not so much evident in terms of pressure accuracy. For this reason, in the following part additional corrections are taken into consideration for the SUP-ROM approach.

4.4.3 Velocity correction: extrapolation efficiency

The present Section evaluates the performance of the method proposed in Sections 3.1.1 and 3.1.2 when larger time windows are considered and time extrapolation is carried out. In particular, the matrices \tilde{A} and \tilde{B} are built using the snapshots taken from a smaller interval that does not cover all the simulated time, in order to test the capability of the method to well approximate the exact correction term in larger time intervals. Two cases are considered:

- correction built starting from 100 time instances, simulation lasting 500 time instances, which correspond to 0,4s and 2s, respectively;
- correction built starting from 500 time instances, simulation lasting 2000 time instances, corresponding to 2s and 8s, respectively.

In both cases the combinations of modes considered are $N_u = N_p = N_{sup} = 5$ and $N_u = N_p = N_{sup} = 3$; the time scheme used for the time derivative is of first order.

Correction built starting from 100 time instances

In this first case, the term τ^u is built just using data of the first 100 steps of simulations; the results are displayed in Figures 4.19 and 4.20. The numbers of singular values R and R_c retained in the correction terms are the optimal values and they are different from the values of R found in Table 4.1.

In the case $N_u = N_p = N_{sup} = 5$ for instance, for the unconstrained case the optimal R is 6, whereas it is 16 when the entire time window is considered for the construction of the correction term.

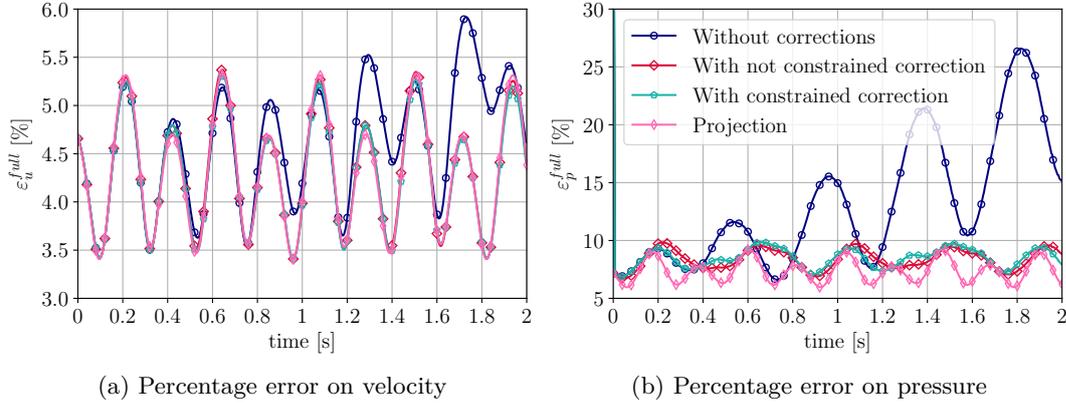


Figure 4.19: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the standard and constrained correction term for velocity. The optimal R and R_c are both 5.

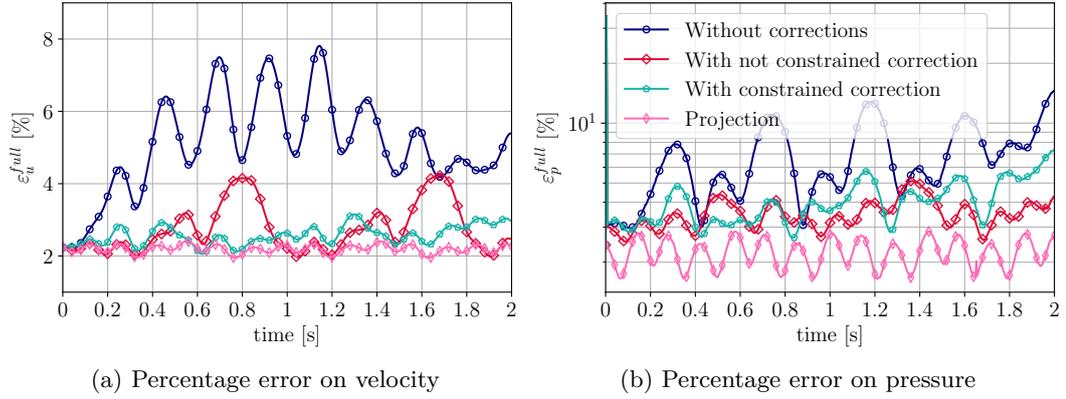


Figure 4.20: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 9 and 4 respectively.

The reduced velocity field in Figure 4.20 (in the case $N_u = N_{sup} = N_p = 5$) seems to suggest that the constrained method might have a more efficient extrapolation capability after the time step 100 when the number of modes for all fields is 5; however, the two techniques lead to similar results in both cases $N_u = N_p = N_{sup} = 3$ (Figure 4.19) and $N_u = N_p = N_{sup} = 5$ (Figure 4.20) and are able to improve the results of simulations without corrections.

Correction built starting from 500 time instances

In the second case a simulation lasting 2000 steps, i.e. 8 seconds, is computed using a correction term built from the first 500 steps. The results are displayed in Figures 4.21 and 4.22 for $N_u = N_p = N_{sup} = 3$ and $N_u = N_p = N_{sup} = 5$, respectively.

The correction improves a lot both velocity and pressure accuracy especially in the interval [2,8] seconds: 500 time steps are sufficient to construct matrices \tilde{A} and \tilde{B} appearing in the correction and to have a good improvement of results w.r.t. the case without corrections.

In addition, the error curves associated with all the test cases in which corrections have been applied appear to be significantly more stable than the test case without correction. This might suggest that if the training set contains a sufficient number of solution cycles, the correction will increase its ability to lead to stable solutions over time, and extended time integration and extrapolation will be possible. However, in such a stable scenario, constraining the minimization for the calculation of \tilde{A} and \tilde{B} is not leading to significant improvements, as can be seen from Figures 4.21 and 4.22.

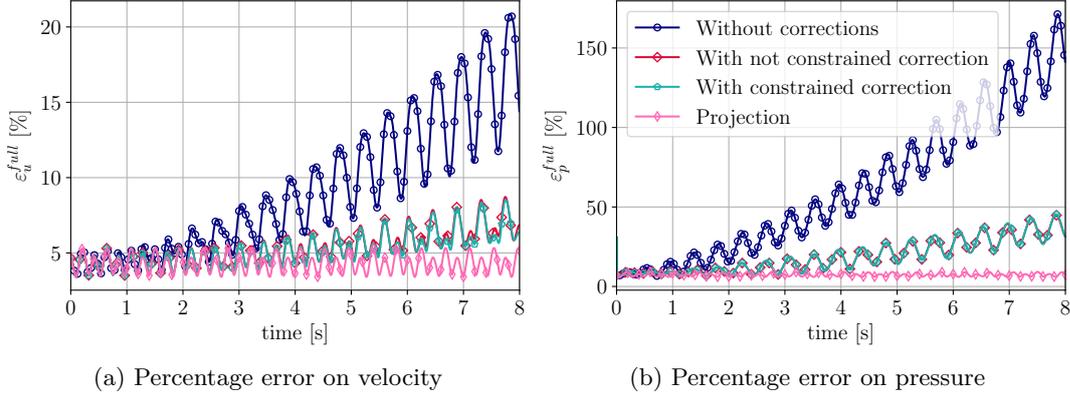


Figure 4.21: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 3 and 4, respectively.

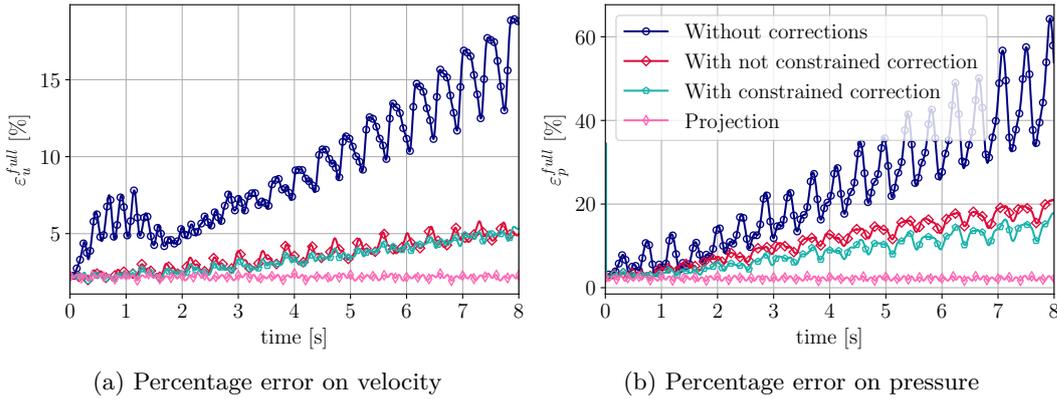


Figure 4.22: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$. Results without any correction term, with the correction term for velocity constrained and not. The optimal R and R_c are 9 and 6, respectively.

- The conclusion that can be drawn regarding the present Section is that the data-driven methods presented so far show an excellent extrapolation efficiency if a sufficient amount of data is considered to build the correction term.

4.4.4 Velocity correction: the 3S-DD-VMS-ROM approach

In this Section, the results of simulations carried out with the *three-scales* approach presented in Section 3.1.3 are displayed. The number of total resolved modes considered is $N_u = N_p = 10$ and $N_{sup} = 15$, but in the construction of the correction term the supremizer modes are discarded. In Figure 4.23 the percentage errors are displayed for velocity and pressure in order to compare what obtained in Section 4.4.2 with the results obtained with the *3S-DD-VMS-ROM* approach. Figure 4.23 (b) displays better pressure field results with respect to the classical approach; as for velocity, comparable percentage errors can be observed. In this simulation the value of r_1 (number of large resolved ROM modes) is fixed *a priori* and equal to 5. The optimal number of singular values is considered: $R_L = 13$ and $R_S = 1$.

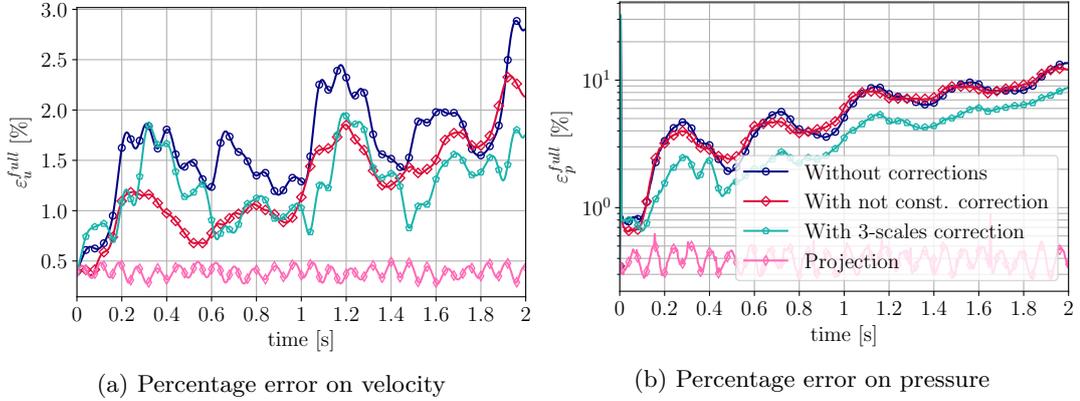


Figure 4.23: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 10$, $N_{sup} = 15$. Results without any correction term, with the correction term for velocity not constrained and with the correction term using 2 scales for resolved ROM are showed.

The following considerations can be drawn:

- the 3-scales correction produces slightly better results than the standard and the constrained correction, but it requires the resolution of an additional least squares problem with respect to the other techniques previously developed.
- in general, in all the results obtained considering the velocity correction, the pressure field shows poorer results than the velocity field. Such a fact has brought us to add in the SUP-ROM further corrections related to the terms which include the pressure reduced coefficients.

4.5 Effect of pressure corrections in the SUP-ROM approach

Results in Section 4.4 have showed that the velocity correction significantly improves the results in terms of velocity field but the same cannot be said for pressure field.

The results shown in Section 4.4 characterize in fact the performance of data-driven correction terms tested so far for LES simulation [10, 11], when applied to RANS models. To the best of our knowledge no attempt has been made to devise similar data-driven corrections for the improved reconstruction of the pressure field.

Since the pressure field is extremely important in applications to compute other relevant properties of the flow or output values such as forces, the new methods developed in Section 3.1.4 for increased pressure field accuracy are here investigated. In Figure 4.24 the following cases for system (3.9) are taken into account:

- $c_u = c_{p(1)} = c_{p(2)} = 0$, i.e. absence of any correction term;
- $c_u = 0, c_{p(1)} = 1, c_{p(2)} = 0$, in which the correction for term $-\mathbf{Hb}$ is added in the momentum equation;
- $c_u = 0, c_{p(1)} = 0, c_{p(2)} = 1$, in which the correction for term \mathbf{Pa} is added in the continuity equation;
- $c_u = 0, c_{p(1)} = c_{p(2)} = 1$, where the pressure corrections are added to the original system.

Figure 4.24 represents the percentage errors on the absolute value of velocity and for pressure when the number of modes considered is $N_u = N_p = N_{sup} = 3$.

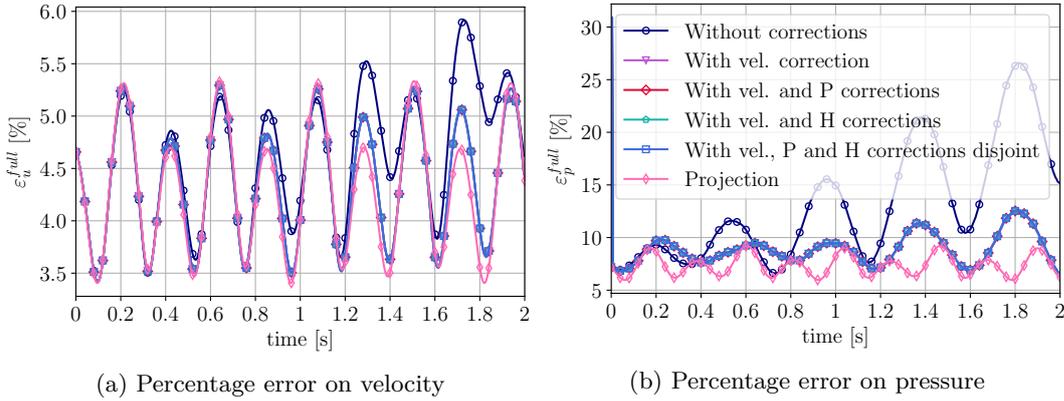


Figure 4.24: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results without any correction term, with the correction term for velocity only, with both the correction terms for velocity and pressure are displayed.

For simulations involving the velocity correction, the constrained data-driven correction is applied, since it provides better results, as pointed out in Section 4.4.2.

The number of singular values retained for matrices in the least squares problems is the optimal one. In particular, $R_c = 6$ is the value that minimizes the error metric for velocity (3.15); $R_{p1} = 6$, $R_{p2} = 4$, $R_{p_{tot}} = 6$ are the optimal numbers of singular values for the least squares problems in (3.20) and (3.21), obtained minimizing the error metric for pressure (3.22).

The images show that the effect of the velocity data-driven correction is much more evident than the effect of pressure corrections. The ineffectiveness of the pressure related corrections on the pressure field is further investigated with some specific tests. We must first understand if the poor pressure reconstruction is a product of inaccuracies of minimizations (3.20) and (3.21) or instead if it is due to inherent inabilities of the correction terms devised, in affecting the pressure field. To this end, we evaluate the performance of the exact correction terms. Removing the minimization error, this will suggest if the pressure correction terms can be effective in the present form.

In Figure 4.25 the results with $c_u = 0$ (no velocity correction) related to the following cases are displayed:

- $c_{p(1)} = c_{p(2)} = 0$, i.e. absence of any correction term;
- $c_{p(1)} = 1$, $c_{p(2)} = 0$, in which the exact correction for term $-\mathbf{Hb}$ is added to the momentum equation;
- $c_{p(1)} = 0$, $c_{p(2)} = 1$, in which the exact correction for term \mathbf{Pa} is added in the continuity equation;
- $c_{p(1)} = c_{p(2)} = 1$, in which the exact pressure corrections are added to the original system.

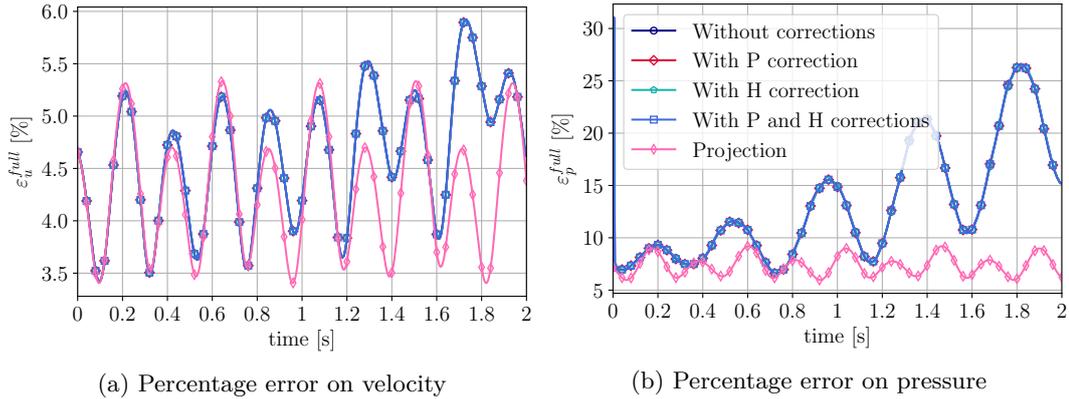


Figure 4.25: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 3$. Results with and without the exact pressure corrections are presented.

As can be seen in Figure 4.25, the pressure corrections have no effect on the dynamical system obtained with the supremizer enrichment. The approximations $\tilde{H}\mathbf{b}$ and $\tilde{P}\mathbf{a}$ are ineffective because the exact terms $\tau_{p(1)}^{\text{exact}}$ and $\tau_{p(2)}^{\text{exact}}$ are too, as can be seen in Figure 4.25. If there is no improvement with the exact term, we cannot hope to have an improvement with the approximated corrections. This clearly suggests that the structure of the corrections, rather than the accuracy of the minimization, must be improved.

Different considerations can be done:

- The terms $\mathbf{H}\mathbf{b}$ and $\mathbf{P}\mathbf{a}$ are linear terms, whereas the velocity term $\mathbf{a}^T\mathbf{C}\mathbf{a}$ is a non-linear term and it has a more evident effect on pressure and velocity accuracy; the same holds true for the correspondent correction terms. However, there can be cases at lower Reynolds number where the linear correction terms have a visible effect on results; for instance in [56] a linear correction for the viscous term in the momentum equation is introduced, producing an improvement for values of ν larger than 10^{-3} ;
- terms only involving the pressure modes are not provided in the system adopted for the supremizer formulation. Given the construction of the model, there is no way to build a correction which is directly acting upon the pressure reduced coefficients. Instead, the only pressure correction terms that can be included in the supremizer formulation either involve the pressure coefficients within the momentum equation, or involve the velocity and supremizer coefficients in the continuity equation.

For the reasons already pointed out, a different formulation is taken into account in the following Section. The PPE model, which includes a pressure equation, naturally offers the opportunity to include different possible specific pressure corrections.

4.6 Analysis of the POD Galerkin PPE-ROM without corrections

In this Section the solution of the dynamical system (4.2) are displayed, considering the parameters $c_u = c_D = c_G = 0$, i.e. without any correction term. The results obtained in terms of percentage errors on the pressure and velocity fields are displayed in Figures 4.26 and 4.27, where a first and a second order time scheme are used, respectively.

From Figure 4.26 one can note that the accuracy increases as the number of modes increases, as desirable. The stability issue exhibited by the supremizer approach solution and discussed in Section 4.3 is not observed in this case. However, when a second order time scheme is used, results appear more unstable, especially when a high number of pressure and velocity modes are considered in the reduced simulations. The reason for this fact might be the same already pointed out in Section 4.3, related to the numerical damping of the integration time schemes.

As was the case with the supremizer approach, also when a PPE approach is considered the reduced order regime we are interested in is the *marginally-resolved* regime. Thus, most of the numerical tests here presented are carried out considering $N_u = N_p = 3$ or $N_u = N_p = 5$.

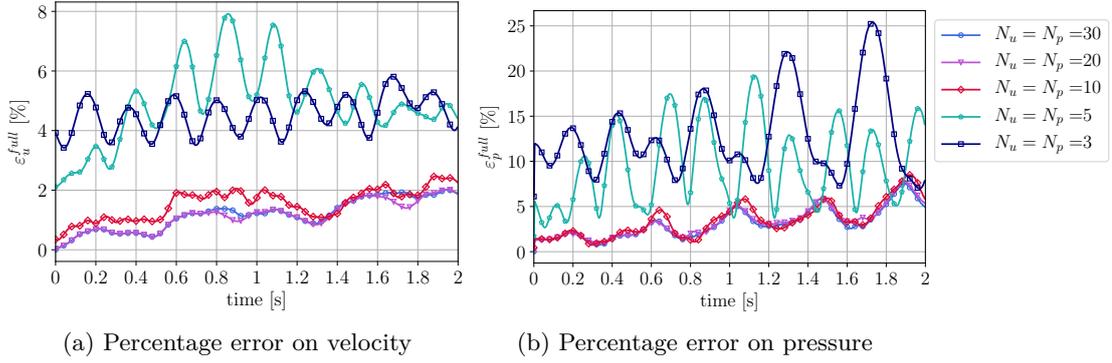


Figure 4.26: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$. In simulations time is evolved with a first order time scheme.

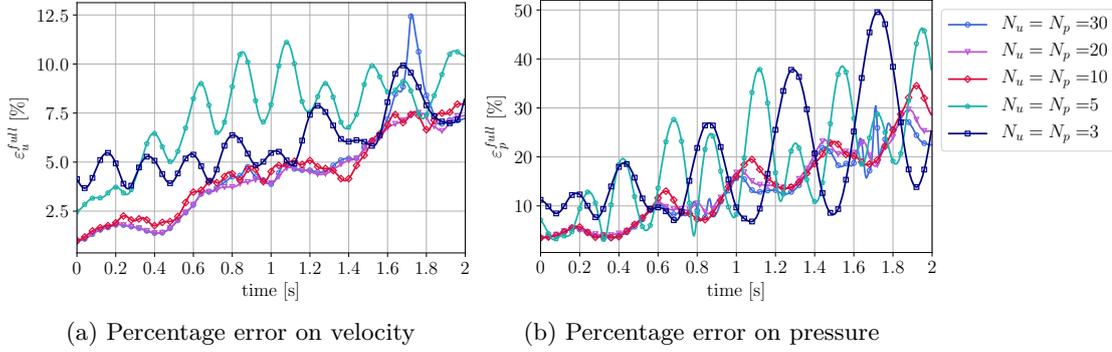


Figure 4.27: Percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, considering different combinations of number of modes, with $N_u = N_p$. In simulations time is evolved with a second order time scheme.

- The conclusions of this part are quite similar to the facts already pointed out in Section 4.3, such as the instability of the second order time scheme. The motivation of the next Sections will be the improvement of the pressure field.

4.7 Effect of corrections in the PPE-ROM approach

The present Section discusses the results obtained with the PPE approach enhanced with different data-driven corrections. The following studies are considered:

- evaluation of the different ansatzes and of the effect of the correction related to term \mathbf{Db} ;
- study of the effects of different pressure corrections in the Poisson pressure equation and of their combined use;
- evaluation of the combined influence of velocity correction and pressure corrections. In this Section, the results of different possibilities presented in Section 3.2.3 are also studied and compared.

4.7.1 Evaluation of the correction term τ_D

In this Section, the different ansatzes for the correction term τ_D are compared in terms of pressure error reduction. Thus, dynamical system (4.2) with $c_u = c_G = c_t = 0$ and $c_D = 1$ is computed and solved. In Figure 4.28 the linear correction term follows the ansatz presented in (3.28), whereas the quadratic correction refers to the ansatz presented in (3.31).

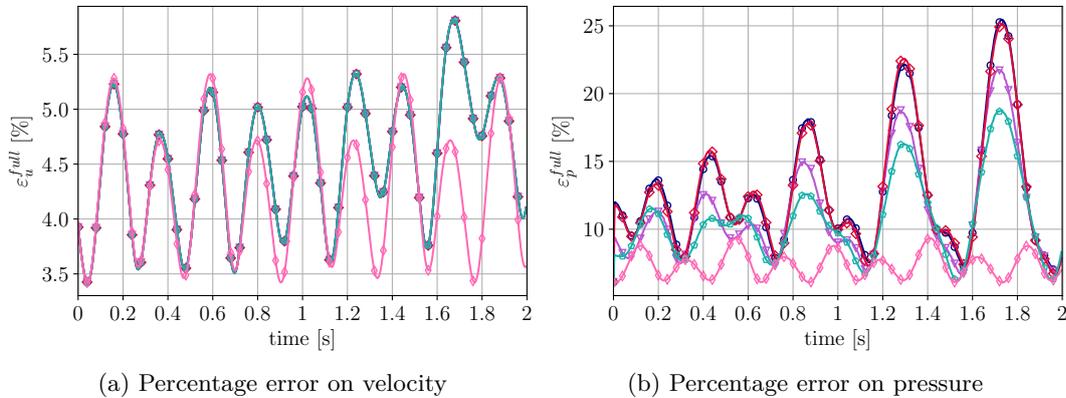


Figure 4.28: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results are displayed for the cases: without any correction ($\text{---}\circ\text{---}$), with the exact correction term τ_D ($\text{---}\triangle\text{---}$), with the linear ansatz ($\text{---}\diamond\text{---}$) and the quadratic ansatz ($\text{---}\square\text{---}$) for the correction τ_D . The previous cases are compared with the reconstruction error ($\text{---}\diamond\text{---}$).

First of all, Figure 4.28 suggests that the effect of the correction term influences only the pressure field and not the velocity field. The pressure error plots on the right also indicate

that the linear correction term does not produce a significant improvement of the results with respect to the solution obtained with the standard dynamical system. On the other hand, adding the pressure quadratic correction results in improved pressure accuracy, approaching that obtained with the exact correction. Moreover, the accuracy obtained by including the pressure quadratic correction is even higher than the one obtained with the exact correction term. This phenomenon was already pointed out in Section 4.4.1 and it is due to the fact that the number of singular values retained for the optimization problem does not correspond to the value that minimizes the error with respect to the exact correction term. In fact, for the velocity approximated correction, the value selected for R and R_c is the one that minimizes the metric (3.15); for the pressure approximated corrections here introduced the metric to be minimized is (3.22). The reason for this choice is that the pressure corrections are introduced in order to improve the accuracy of the pressure field.

From now on, the quadratic ansatz is always computed for the correction term τ_D , since it produces a better effect on the pressure field with respect to the linear ansatz.

4.7.2 Effect of pressure corrections in the PPE-ROM

In this subsection the correction term for τ_G , which was presented in Section 3.2.2, is introduced. We recall that such a correction, located in the Poisson pressure equation, is based on the reduced velocity vector. Thus, its single effect and its combined effect with the other pressure correction in the Poisson equation are here evaluated. In particular, dynamical system (4.2) is solved considering $c_u = c_t = 0$. Figure 4.29 displays the results for the following cases:

- $c_D = 0, c_G = 1$;
- $c_D = 1, c_G = 0$, where the quadratic ansatz for τ is computed;
- $c_D = c_G = 1$, where the corrections are obtained by solving two disjoint optimization problems for the pressure corrections presented in Sections 3.2.1 and 3.2.2;
- $c_D = c_G = 1$, where a unique optimization problem is solved, using an ansatz with a linear dependence on \mathbf{a} and a quadratic dependence on \mathbf{b} (Section 3.2.3 Case 1);
- $c_D = c_G = 1$, in which a unique optimization problem is solved, considering an ansatz with a quadratic dependence on vector \mathbf{ab} (Section 3.2.3 Case 2).

The plots in Figure 4.29 show that when only one correction is added in the system, the pressure approximated fields improves and the velocity approximation is unchanged in terms of accuracy with respect to the high fidelity solution. When the system (4.2) is solved including both the pressure corrections, the best result is achieved using the method presented in Case 2 of Section 3.2.3, in which the correction is a function of vector $\mathbf{ab} = (\mathbf{a}, \mathbf{b})$, as can be seen from Figure 4.29(b).

An important consideration is that the pressure corrections in the PPE-ROM approach improve the pressure result, whereas in the SUP-ROM approach results are unchanged adding the pressure corrections (Section 4.5).

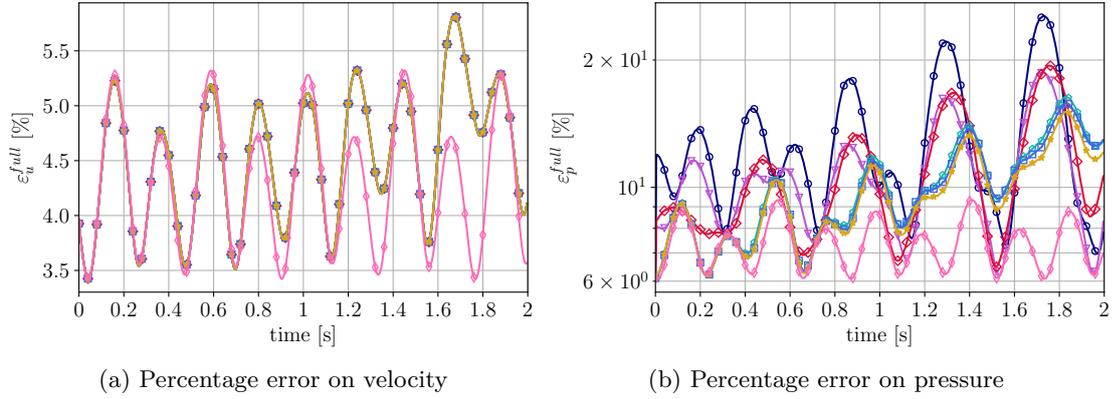


Figure 4.29: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections ($\text{---}\circ\text{---}$); with the correction τ_D ($\text{---}\nabla\text{---}$); with the correction τ_G ($\text{---}\diamond\text{---}$); with both τ_D and τ_G found from two disjoint least squares problems ($\text{---}\square\text{---}$), in Case 1 and 2 of section 3.2.3 ($\text{---}\square\text{---}$ and $\text{---}\star\text{---}$, respectively). Results are compared with the reconstruction errors, referred to the projected fields ($\text{---}\diamond\text{---}$).

- The conclusion of the present part is that both pressure corrections added in the PPE-ROM produce a significant improvement only in the pressure field.

4.7.3 Combined effect of velocity and pressure corrections in the PPE-ROM

The previous subsection results confirmed that adding data-driven corrections in the Poisson equation for pressure leads to improvements in the ROM pressure field accuracy.

The aim is now to understand if combining the Poisson equation and momentum equation corrections, additional gains can be obtained. Thus, in this Section the velocity correction term τ_u is introduced in system (4.2). Figure 4.30 considers the following cases:

- $c_u = c_D = 1, c_G = 0$;
- $c_u = c_G = 1, c_D = 0$;
- $c_u = c_G = c_D = 1$, where the method presented in (3.2.3) (Case 3) is computed and a unique least squares problem is solved to find all correction terms.

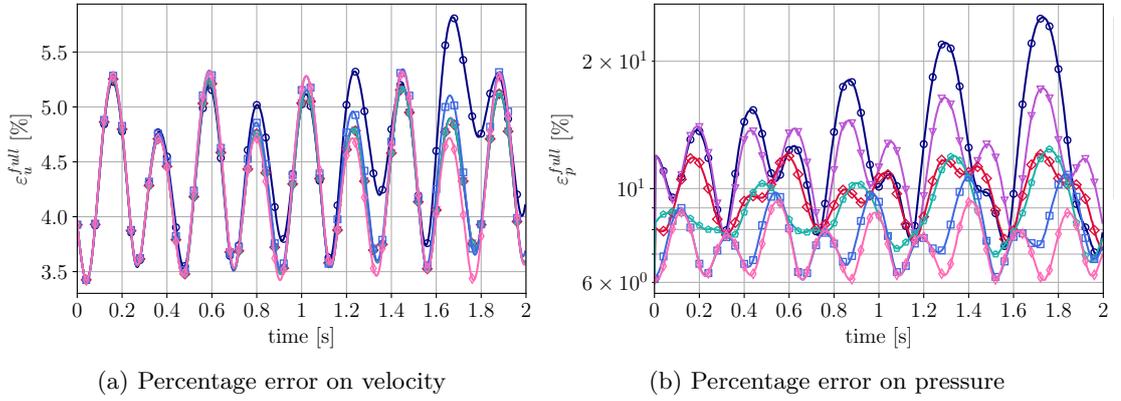


Figure 4.30: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections ($\text{---}\circ\text{---}$); with the velocity correction τ_u ($\text{---}\blacktriangledown\text{---}$); with the corrections τ_u and τ_D ($\text{---}\blacklozenge\text{---}$); with τ_u and τ_G ($\text{---}\blacklozenge\text{---}$), with τ_u , τ_D and τ_G in Case 3 of section 3.2.3 ($\text{---}\blacksquare\text{---}$). Results are compared with the reconstruction errors, referred to the projected fields ($\text{---}\blacklozenge\text{---}$).

Figure 4.30 displays the results obtained when the velocity correction is combined with one of the two pressure correction terms. The right plot clearly shows further gains in pressure accuracy with respect to the previous cases. The result obtained with the both velocity and pressure corrections is also very close to the reconstruction error for the velocity field, which is the best result we can obtain, as can be seen on the left diagram in Figure 4.30.

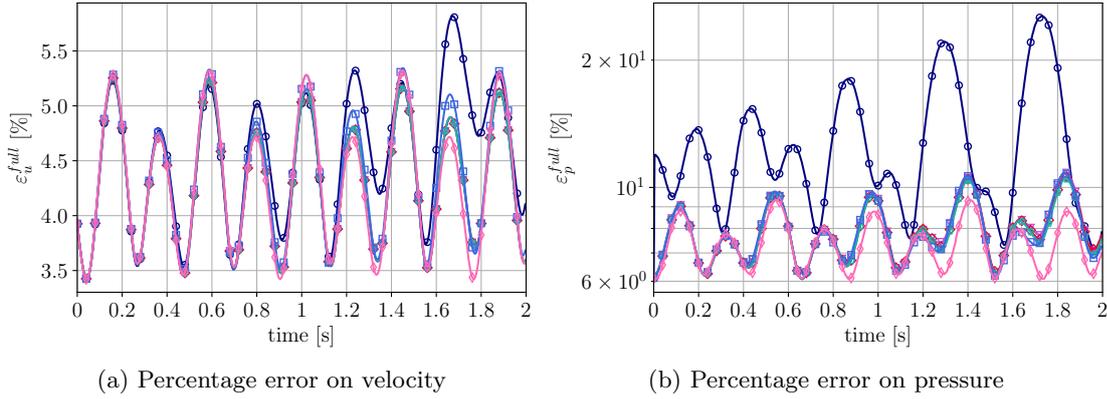


Figure 4.31: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections ($\text{---}\circ\text{---}$); with τ_u , τ_D and τ_G disjoint ($\text{---}\triangle\text{---}$); with the velocity and pressure corrections in Cases 1, 2 and 3 of section 3.2.3 ($\text{---}\diamond\text{---}$, $\text{---}\square\text{---}$ and $\text{---}\square\text{---}$, respectively). Results are compared with the reconstruction errors, referred to the projected fields ($\text{---}\diamond\text{---}$).

In Figure 4.31 the results obtained combining the momentum equation correction with all the Poisson corrections forms developed are showed. In particular, the methods proposed in Section 3.2.3 are compared. All the methods produce a pressure reduced solution very close to the projected pressure one and the results of all methods look similar. However, the most efficient method is the one presented in 3.2.3, as can be also seen in Figure 4.32, which displays the results of longer simulations lasting 8 seconds, with corrections built from the snapshots of the first 2 seconds.

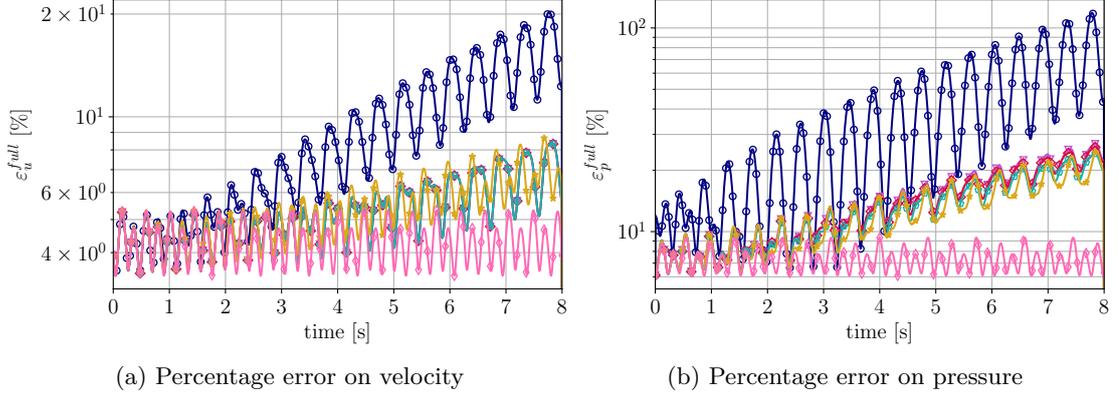


Figure 4.32: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 3$. Results in the following cases are displayed: without corrections ($\text{---}\circ\text{---}$); with τ_u , τ_D and τ_G disjoint ($\text{---}\nabla\text{---}$); with the velocity and pressure corrections in Cases 1, 2 and 3 of section 3.2.3 ($\text{---}\diamond\text{---}$, $\text{---}\square\text{---}$ and $\text{---}\star\text{---}$, respectively). Results are compared with the reconstruction errors, referred to the projected fields ($\text{---}\diamond\text{---}$).

The conclusions of the present Section are here listed:

- the velocity correction reduces the error for both velocity and pressure fields, whereas the pressure corrections added in the Poisson equation only improve the pressure field;
- the most significant improvement in the accuracy of the reduced pressure field is reached when all data-driven corrections are added to the reduced system.

4.8 Comparison: the turbulence modelling, the data-driven corrections and the combined effect

In this Section, the results obtained with the addition of the correction terms in systems (4.1) and (4.2) are compared with the solution of reduced systems where the turbulence modelling is included [1], i.e. $c_t = 1$. The results obtained using the turbulence inclusion are graphically represented on the test case grid making use of the open-source application *Paraview* and the results are compared to those obtained in previous Sections of the Thesis, in which no turbulence modelling is included.

4.8.1 Turbulence inclusion in the SUP-ROM approach

In the supremizer approach framework, the following cases are solved and compared, for both the first and the second order time evolution (Figures 4.33 and 4.34). The dynamical system considered is (4.1) and the following cases are analysed:

- $c_u = c_{p(1)} = c_{p(2)} = c_t = 0$, i.e. the SUP-ROM without any correction is performed;
- $c_u = 1$, $c_{p(1)} = c_{p(2)} = c_t = 0$, and $\tau_u(\mathbf{a}^i) = \tilde{A}\mathbf{a} + \mathbf{a}^T \tilde{B}\mathbf{a}$, i.e. just the velocity correction is added to the dynamical system;
- $c_u = c_{p(1)} = c_{p(2)} = 0$ and $c_t = 1$, i.e. just the turbulence modelling is considered;
- $c_u = c_t = 1$ and $c_{p(1)} = c_{p(2)} = 0$, i.e. both the velocity correction and the turbulence modelling are considered.

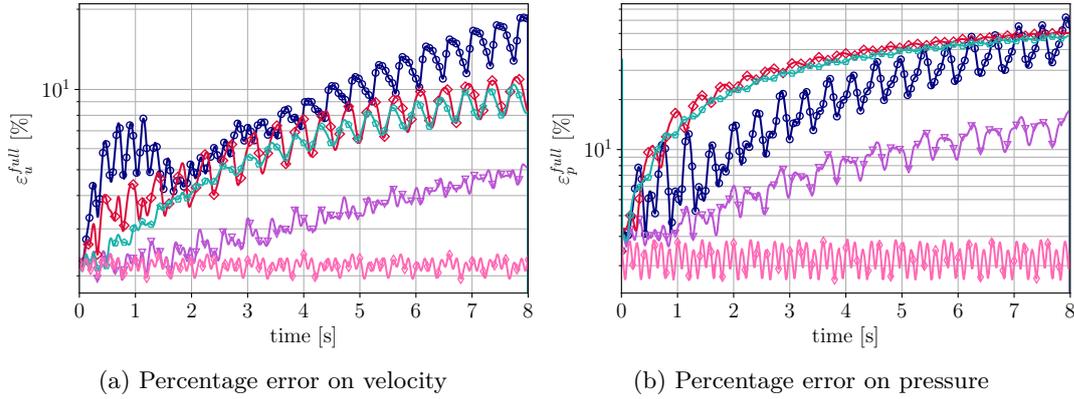


Figure 4.33: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$ and a first order time scheme. Results include the following cases: without any data-driven term ($\text{---}\circ\text{---}$); with velocity constrained correction ($\text{---}\triangle\text{---}$); with turbulence term ($\text{---}\diamond\text{---}$); with both velocity correction and turbulence term ($\text{---}\square\text{---}$); projection ($\text{---}\diamond\text{---}$).

The coefficients of the reduced eddy viscosity field $(g_i)_{i=1}^{N_{vt}}$ are computed making use of a fully-connected neural network, starting from the velocity coefficients $(a_i)_{i=1}^{N_u}$. The network is composed by 2 hidden layers, the ReLU function is used as activation function in the network and the learning rate is set as 10^{-5} .

The momentum equation correction term here considered is only the one referred to the nonlinear term and this velocity correction is obtained with the constrained method, since it provides the best performance on velocity accuracy for a low number of modes, as pointed out in Section 4.4.2. No pressure correction is considered in the momentum equation since both corrections simulated in Section 4.5 have not significant effects on velocity and pressure fields. Figure 4.33 displays the results obtained using a first order time scheme. In such a case, the inclusion of a turbulence modelling does not appear to have a completely positive effect on accuracy.

In addition, coupling the turbulence and correction effect does not lead to higher accuracy, especially when the pressure field is considered. In such a case in fact, the combined method leads to worse results than in the no-corrections case.

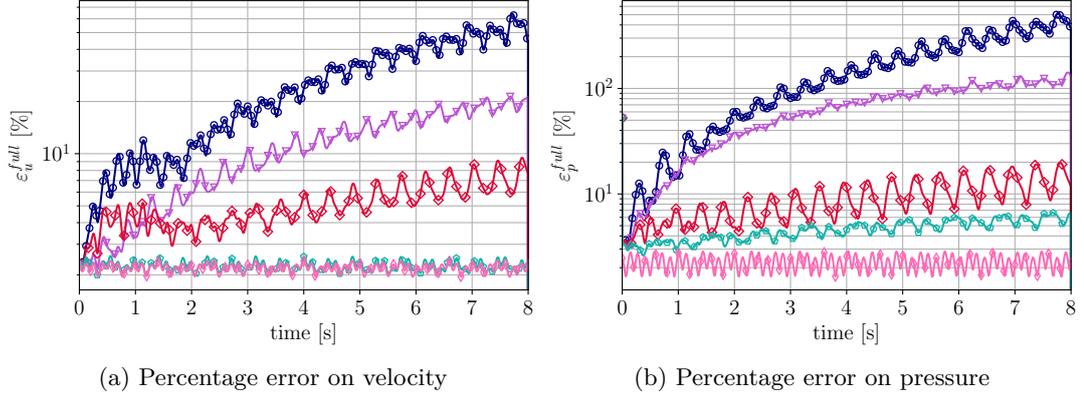


Figure 4.34: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = N_{sup} = 5$ and a second order time scheme. Results include the following cases: without any data-driven term ($\text{---}\circ\text{---}$); with velocity constrained correction ($\text{---}\blacktriangledown\text{---}$); with turbulence term ($\text{---}\blacklozenge\text{---}$); with both velocity correction and turbulence term ($\text{---}\blacklozenge\text{---}$); projection ($\text{---}\blacklozenge\text{---}$).

When considering a second order time evolution scheme (Figure 4.34) results obtained with turbulence modelling or with both correction closure terms and turbulence terms are improved with respect to the results of standard ROM and approach reconstruction errors. In particular, the results in terms of accuracy of the velocity reduced field are very similar to the projected field.

4.8.2 Turbulence inclusion in the PPE-ROM approach

The combined effect of data-driven terms and turbulence modelling is here evaluated for the PPE approach for a simulation lasting 8 seconds in Figures 4.35 and 4.36.

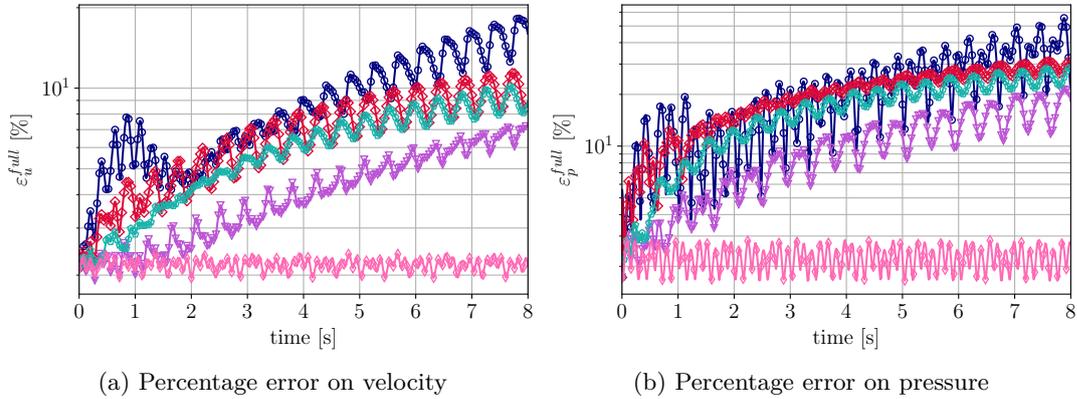


Figure 4.35: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 5$ and a first order time scheme. Results include the following cases: without any data-driven term ($\text{---}\circ\text{---}$); with all corrections ($\text{---}\triangle\text{---}$); with turbulence term ($\text{---}\diamond\text{---}$); with corrections and turbulence term ($\text{---}\square\text{---}$); projection ($\text{---}\star\text{---}$).

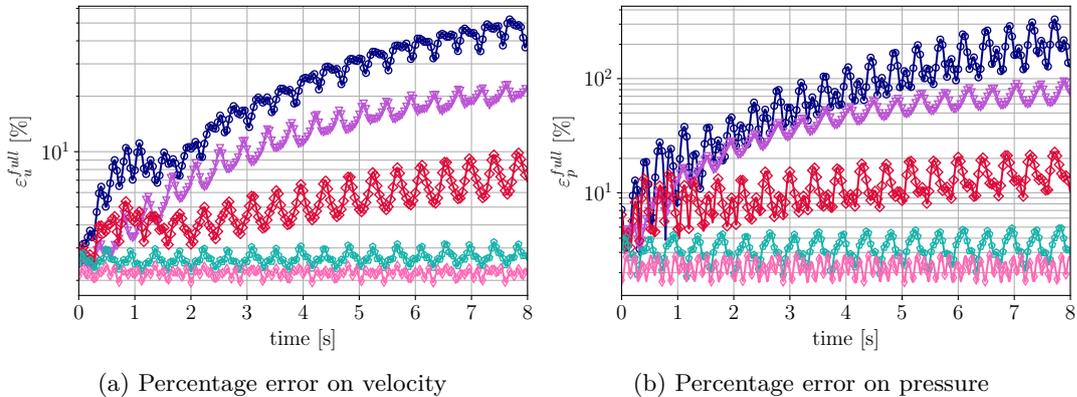


Figure 4.36: Percentage errors of the absolute value of velocity (a) and of pressure (b) with respect to the full order simulations, considering $N_u = N_p = 5$ and a second order time scheme. Results include the following cases: without any data-driven term ($\text{---}\circ\text{---}$); with all corrections ($\text{---}\triangle\text{---}$); with turbulence term ($\text{---}\diamond\text{---}$); with corrections and turbulence term ($\text{---}\square\text{---}$); projection ($\text{---}\star\text{---}$).

In simulations, the correction terms are built starting from the first 2 seconds, just as in 4.4.3 and all corrections are included using the approach detailed in Section 3.2.3 Case 3.

When the first order time evolution is considered in the dynamical system, the combination of turbulence and corrections leads to the results depicted in Figure 4.35. Figure 4.36 refers instead to the results obtained when a second order scheme is used.

From Figures 4.35 and 4.36 there is an evident difference, which confirms what observed with the SUP-ROM results. When a first order scheme is used (Figure 4.35) the combined turbulent-data-driven approach does not significantly improve with respect to the only turbulent or only data-driven approaches.

On the other hand, in the case in which a second order time scheme is computed, the effect of the combination of turbulence and corrections leads to an effective improvement of results. However, a difference also in the standard ROM results is observed. The blue line (in Figures 4.35 and 4.36) shows in fact better results in the case of the first order time scheme. In any case, the results achieved just including the turbulence model improve with respect to both the standard ROM, but also the opposite is true. Adding data-driven corrections to turbulent ROMs leads to improvements, too. In this way, the methods developed in this Thesis lead to more accurate results than in [1], where the turbulent models are deeply analysed.

Moreover, the model has an excellent extrapolation efficiency. The corrections have been in fact constructed just with the first 2 seconds data and they have a good effect also in the time interval [2,8] seconds.

The results obtained in Figures 4.34 and 4.36 show that the instability of the second order integration scheme is damped by the addition of the turbulence modelling.

The data-driven approach shows all its potential in improving the accuracy when the second order scheme is considered. It also acts as a stabilizer for the error as it does not increase as much as in the standard ROMs cases.

4.8.3 Combined effect of turbulence modelling and correction terms for different mode regimes

The integrals of the percentage errors for the absolute value of velocity and for pressure on a 500 time steps window are evaluated. The expressions for the overall time window L^2 errors are the following:

$$\int_0^T \varepsilon_u^{full}(t)dt \approx \sum_{j=1}^M \varepsilon_u^{full}(t_j) \Delta t, \quad \int_0^T \varepsilon_p^{full}(t)dt \approx \sum_{j=1}^M \varepsilon_p^{full}(t_j) \Delta t,$$

where T corresponds to 2s and to the time step $M = 500$ from the beginning of the simulation; the expressions of the percentage errors are expressed in (4.5), and $\Delta t = 0,004$ s. The integrals of the percentage errors are analysed for all the cited cases mentioned in the previous Sections, for a first order and for a second order time scheme.

The graphical results are showed in semi-logarithmic plots in Figures 4.37 and 4.38 for the supremizer approach, in Figures 4.39 and 4.40 for the Poisson approach.

When a first order time scheme is considered, the addition of a turbulence modelling does not work well both in the supremizer and in the Poisson models; the combination of a turbulence modelling and the data-driven correction has a negative effect on the performance of the solution, too. For the velocity correction, from Figure 4.37(b) a slight improvement can be seen with a constrained approach, instead of a classical one, especially when the number of modes belongs to the *marginally-resolved* regime.

The results in terms of integral of the error in Figures 4.37 and 4.39 show that a good improvement can be achieved with the introduction of the correction terms, but not with the usage of a turbulence modelling.

Given the results shown in the previous Sections, an implicit second order time scheme (described by (4.4)) is also taken into account in order to improve the results obtained with a turbulence modelling.

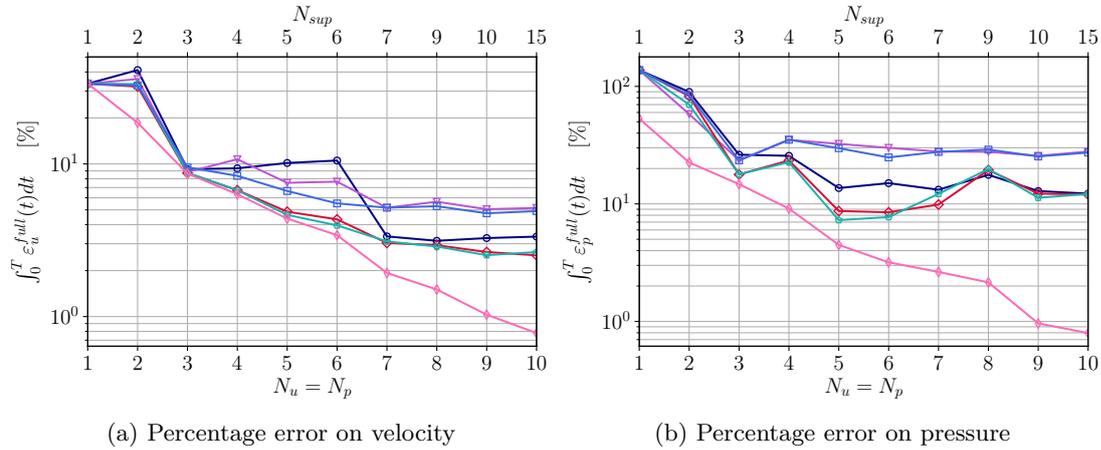


Figure 4.37: Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the SUP-ROM with a first order time scheme. The cases represented are the following: without any data-driven term ($\text{---}\circ\text{---}$); with turbulence modelling, without corrections ($\text{---}\triangle\text{---}$); without turbulence modelling, with standard and constrained correction ($\text{---}\diamond\text{---}$ and $\text{---}\square\text{---}$, respectively); with both turbulence modelling and constrained correction ($\text{---}\square\text{---}$); projection ($\text{---}\diamond\text{---}$).

In Figures 4.37 and 4.38 the number of modes considered for supremizer modes $N_{sup} > N_p$ when $N_u = N_p$ is equal to 8, 9 and 10, in order to avoid stability issues. From the Figures one can note that the data-driven corrections have a more evident positive effect when a small number of modes is considered; the improvement with respect to the standard Galerkin-ROM, especially for pressure field, is not visible when the number of modes is larger than 8.

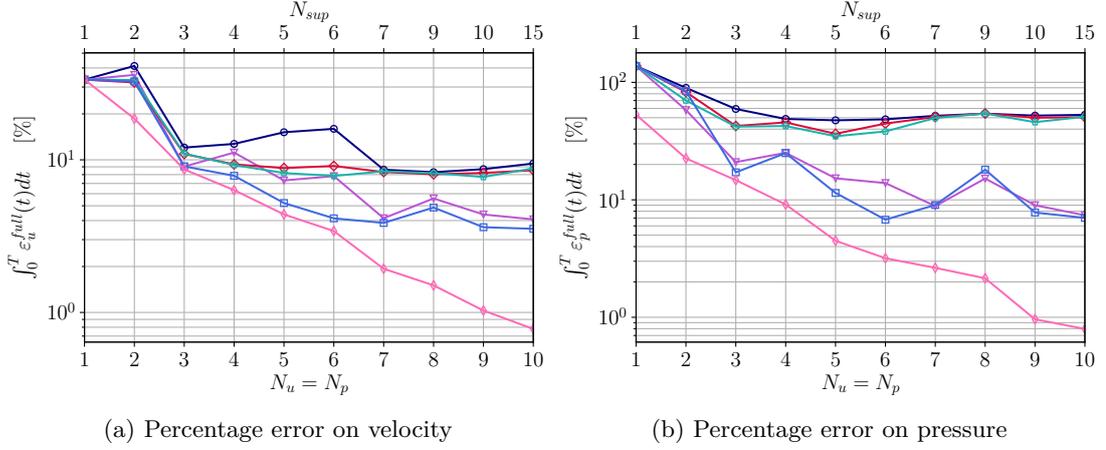


Figure 4.38: Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the SUP-ROM with a second order time scheme. The cases represented are the following: without any data-driven term (\circ); with turbulence modelling, without corrections (∇); without turbulence modelling, with standard and constrained correction (\diamond and \circ , respectively); with both turbulence modelling and constrained correction (\square); projection (\diamond).

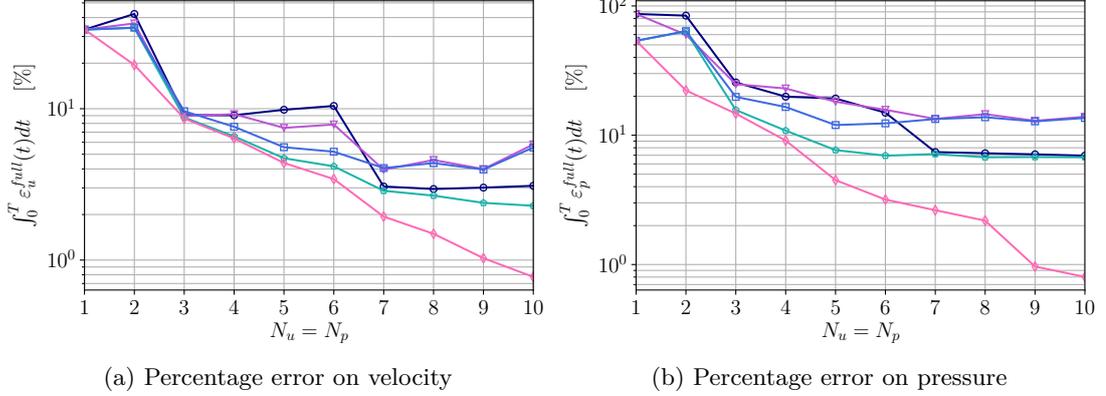


Figure 4.39: Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the PPE-ROM with a first order time scheme. The cases represented are the following: without any data-driven term (\circ); with turbulence modelling, without corrections (∇); without turbulence modelling, with corrections (\circ); with both turbulence modelling and corrections (\square); projection (\diamond).

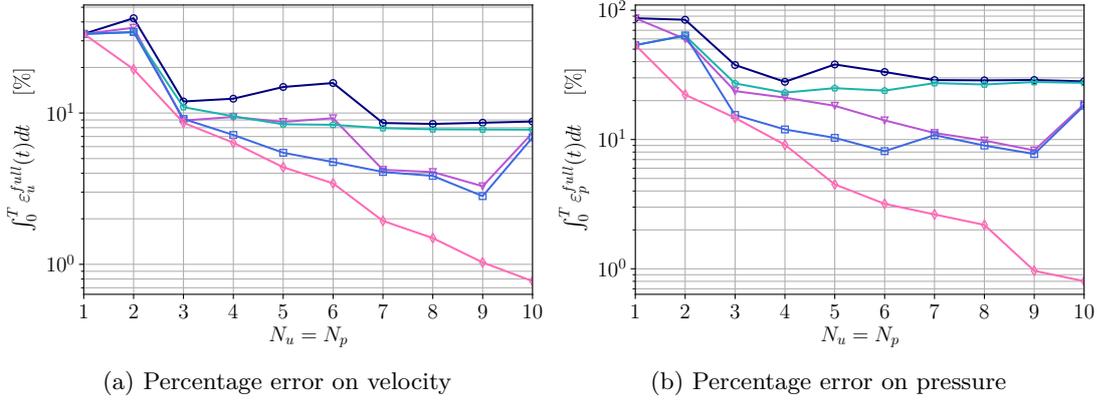


Figure 4.40: Integral of the percentage errors of the absolute value of velocity and of pressure with respect to the full order simulations, varying the number of modes. The model is the PPE-ROM with a second order time scheme. The cases represented are the following: without any data-driven term ($\text{---}\circ\text{---}$); with turbulence modelling, without corrections ($\text{---}\blacktriangledown\text{---}$); without turbulence modelling, with corrections ($\text{---}\square\text{---}$); with both turbulence modelling and corrections ($\text{---}\blacksquare\text{---}$); projection ($\text{---}\diamond\text{---}$).

4.8.4 Graphical results

The inclusion of the correction terms and of turbulence modelling in the reduced formulations is also examined from a graphical point of view for the SUP-ROM and PPE-ROM approaches on the test case grid.

The pressure and the velocity magnitude fields are represented in Figures 4.41 and 4.42, respectively, for different SUP-ROM and PPE-ROM simulation and the correspondent absolute errors of both fields with respect to the full order fields are displayed in Figures 4.43 and 4.44. The corrections taken into account are the constrained velocity correction examined in Section 3.1.2 for the SUP-ROM, and the joint velocity and pressure correction presented in Section 3.2.3 (Case 3). The time integration scheme to which the graphical fields refer is the second order one, since it provides the best results in Sections 4.8.1 and 4.8.2.

The POD is performed on the time interval $[79.992, 99.992]$ seconds and the reduced order systems (4.1) and (4.2) are solved in the interval $[79.992, 87.992]$ seconds, since the maximum length of the online simulations carried out is 8 seconds. For this reason, all the fields are captured at the final time step of online simulations, which is second 87.992.

In order to obtain the reduced fields, the reduced order systems with $N_u = N_p = 5$, and $N_{sup} = 5$ for the supremizer approach, are solved. The reduced fields are computed from the coefficients' vectors \mathbf{a} and \mathbf{b} and the POD modes $(\phi_i)_{i=1}^{N_u+N_{sup}}$ and $(\chi_i)_{i=1}^{N_p}$ as in (2.2).

It is observable a difference between the fields computed with the standard ROMs and those coming from the systems including only the correction terms. In fact, the fields in Figures 4.41 and 4.42(c) and (d) are closer to the full order fields especially in the region

around the cylinder. The improvement of the error nearby the circular cylinder is an important gain as it would lead to a better reconstruction of the ROM lift coefficient.

As expected, results get even better by including the turbulence terms in the reduced systems, as in Figures 4.41 and 4.42(e) and (f). The absolute errors represented in Figures 4.43 and 4.44 confirm the presence of a higher error nearby the cylinder in the fields resulting from standard ROMs simulations. From the range of the scales used to represent errors of the velocity field, it is worth remarking that there is a difference of one order of magnitude between the standard ROMs (Figure 4.44(a) and (b)) and the DD-ROMs (4.44(c) and (d)). These differences are observable also for the pressure errors and are even more evident when the turbulence terms are added in the ROM formulations (Figures 4.43 and 4.44 (e) and (f)).

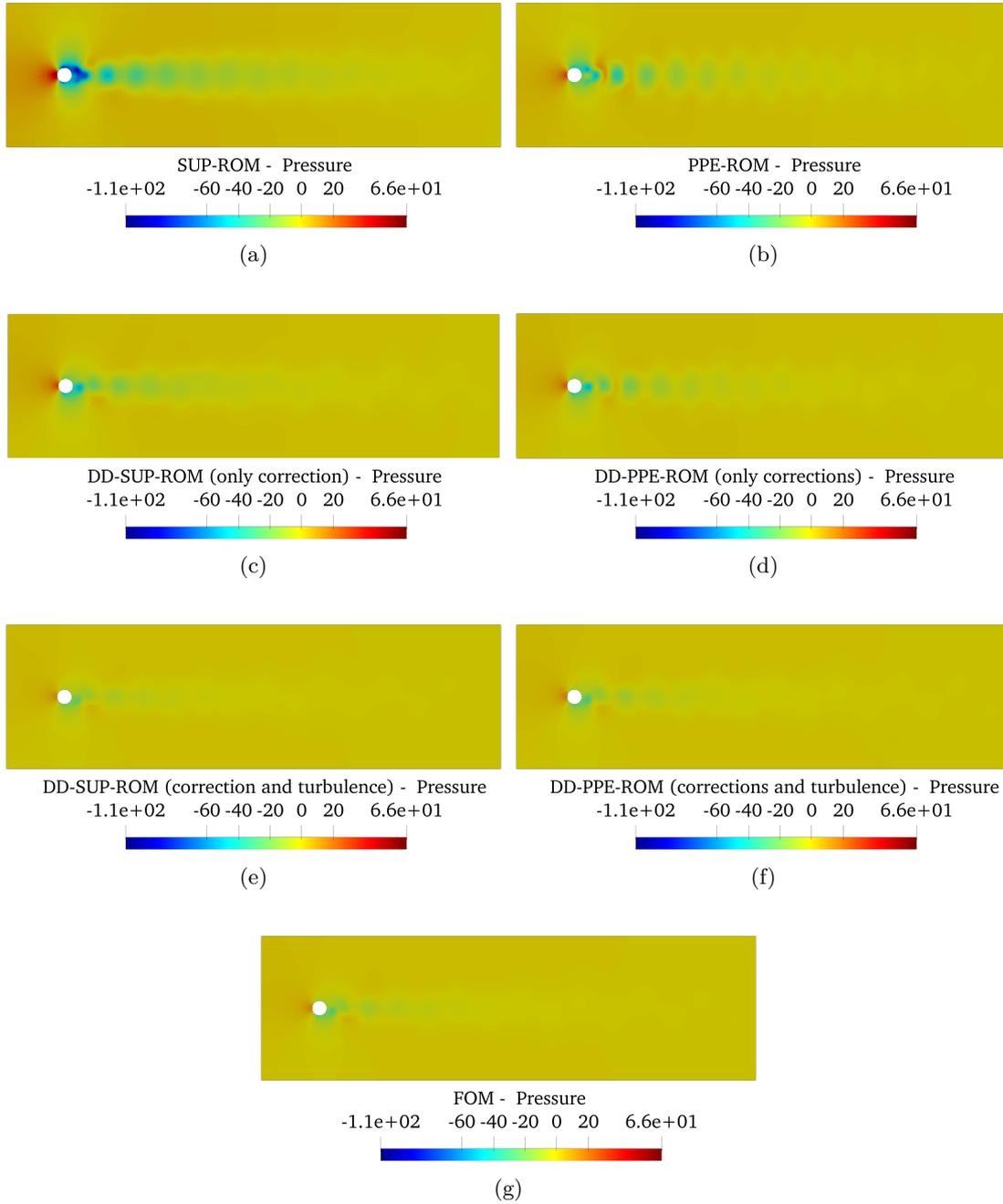


Figure 4.41: Representation of the pressure field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.

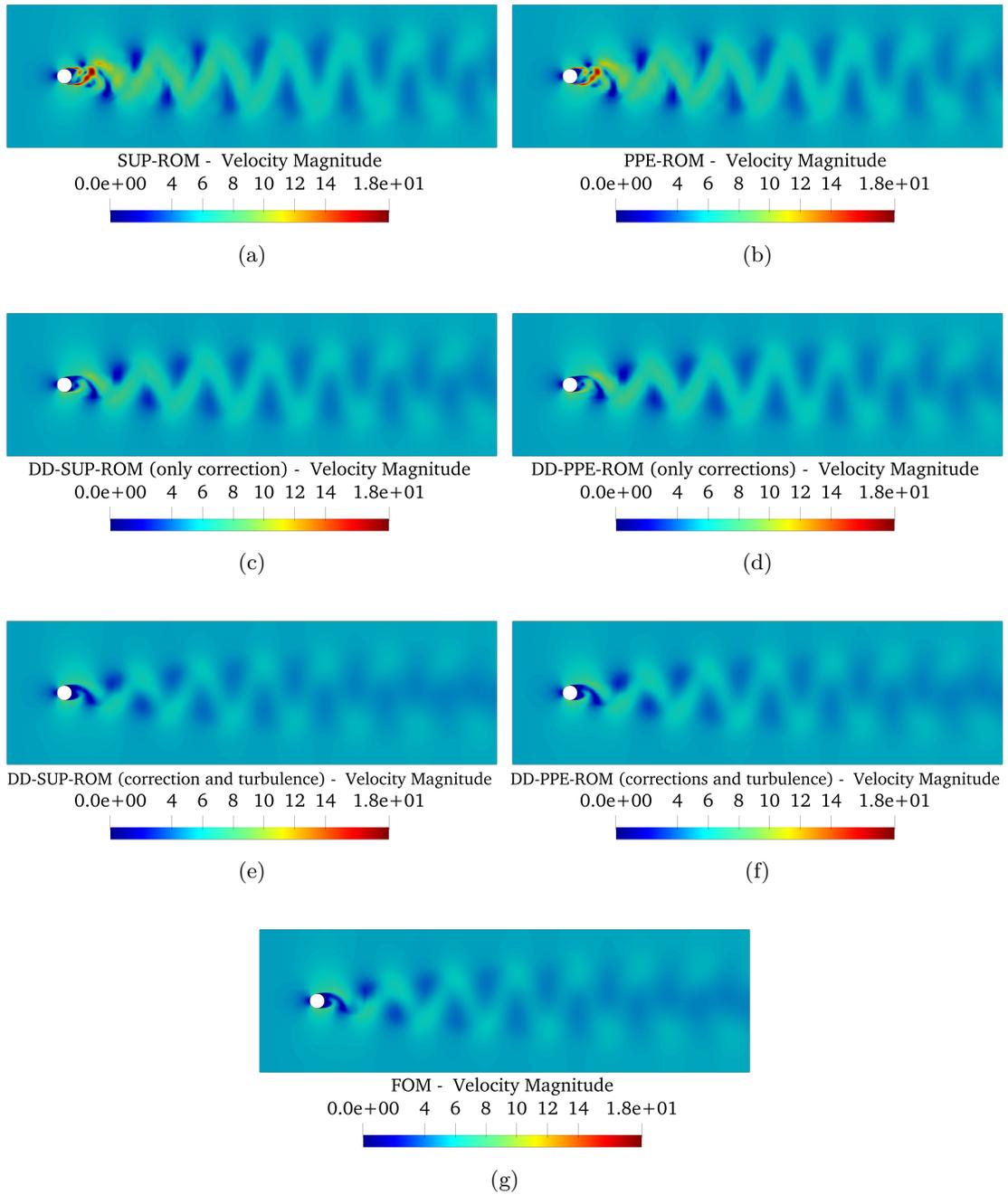


Figure 4.42: Representation of the velocity magnitude field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.

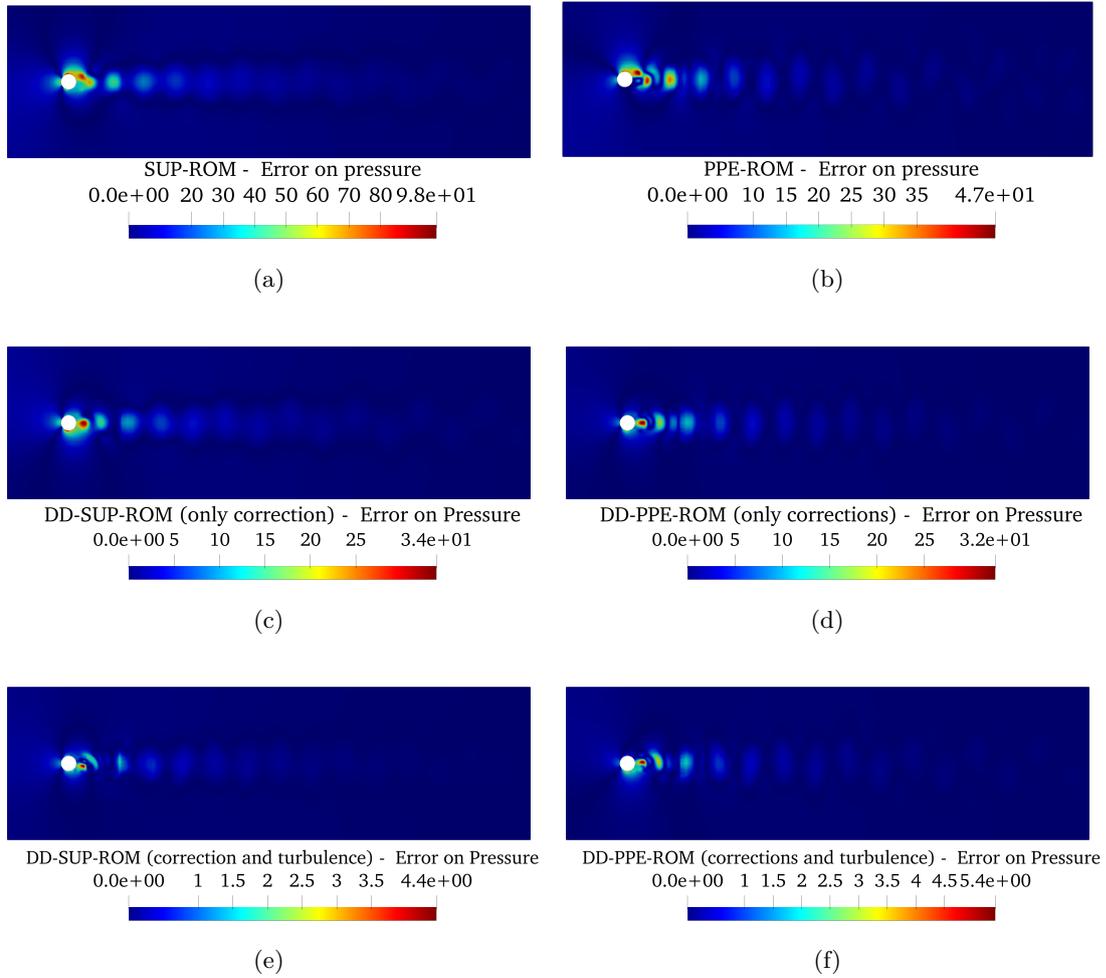


Figure 4.43: Representation of the absolute errors on the pressure field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.

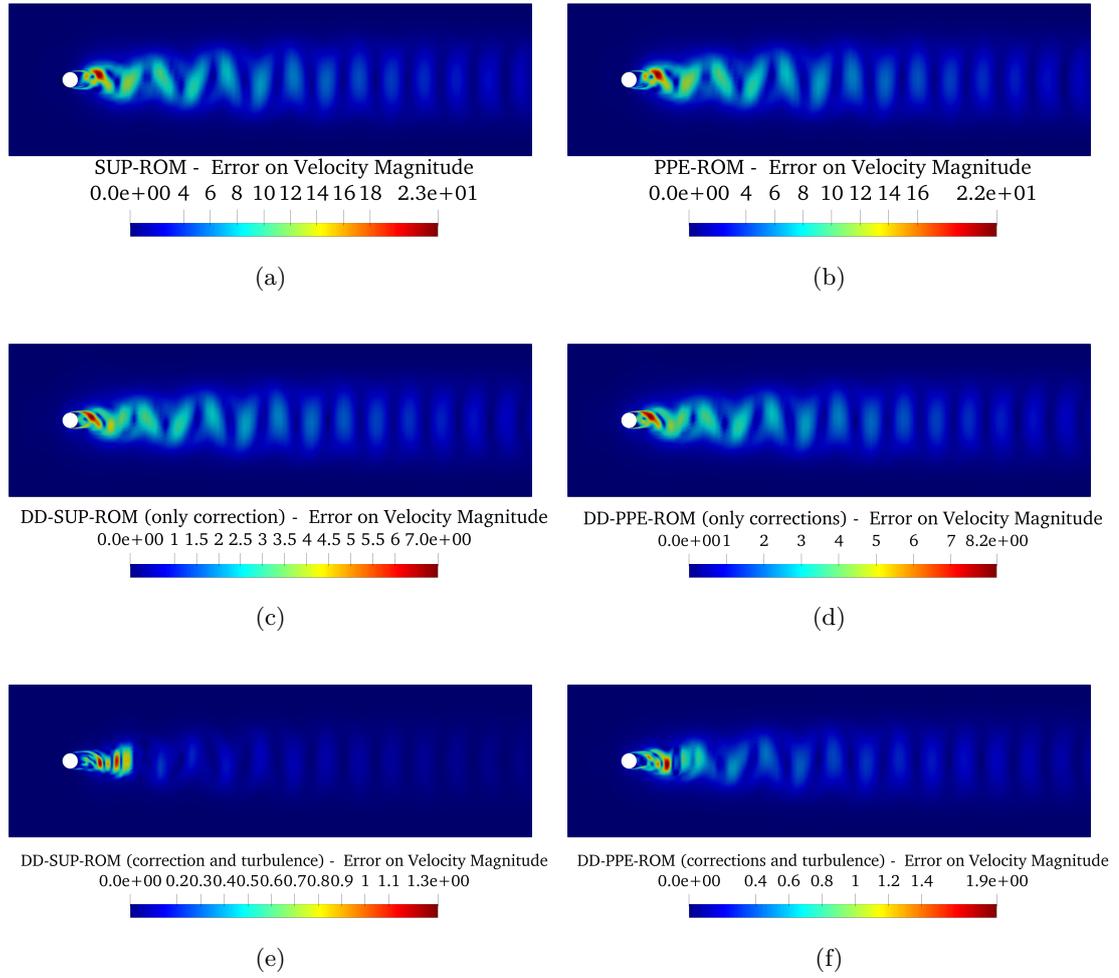


Figure 4.44: Representation of the absolute errors on the velocity magnitude field for the FOM, the SUP-ROM and the PPE-ROM simulations with and without the correction terms and the turbulent terms.

4.8.5 Analysis of the computational cost of simulations

This Section is dedicated to the evaluation of the computational time for the simulations in the online phase. The aim of this Section is to make a comparison among the costs of computing the high fidelity solution and the approximated solutions; moreover, the reduced simulations with and without the data-driven terms are compared in terms of computational cost.

The Full Order simulation last 1 h 31 min from second 80 to second 100, which is the time interval used to compute the POD and corresponding to 5000 time snapshots. The Proper Orthogonal Decomposition, performed with the *ITHACA-FV* library, last 1 h 34 min.

The time for the resolution of the reduced order systems is computed and a second order time scheme is used to compute the time derivative since it provides better results when the turbulence terms are included in the reduced order formulation, as can be seen from Figures 4.38 and 4.40. In Tables 4.2 and 4.3 the results of this analysis are displayed for the supremizer and Poisson approaches, respectively. The time intervals are found by taking the mean among 100 values.

Computational time with the SUP-ROM approach				
N_u	N_{sup}	N_p	Model	Computational time
10	15	10	SUP-ROM	1.4260
3	3	3	SUP-ROM	0.4304
3	3	3	DD-SUP-ROM (only correction)	0.8467
3	3	3	DD-SUP-ROM (correction and turbulence)	4.6779

Table 4.2: Computational time evaluated for some simulations carried out making use of the supremizer approach.

Computational time with the PPE-ROM approach			
N_u	N_p	Model	Computational time
10	10	PPE-ROM	1.0735
3	3	PPE-ROM	0.4812
3	3	DD-PPE-ROM (only correction)	1.1097
3	3	DD-PPE-ROM (correction and turbulence)	3.9641

Table 4.3: Computational time evaluated for some simulations carried out making use of a Poisson approach.

From the Tables, it is evident that the reduced systems without any data-driven term with 10 modes for velocity and pressure are solved in a time interval which is very similar to the one necessary to solve the reduced systems with the correction terms but with a smaller number of degrees of freedom (3 for velocity and for pressure). It is important to remark that the solution of the reduced system with an higher number of degrees of freedom but without corrections is less accurate than the result of the reduced system with a smaller number of degrees of freedom and with correction terms, as can be seen in Figures 4.38 and 4.40. Basically the correction terms allow to compute a solution more accurate than the one obtained from a standard ROM approach, with less degrees of freedom and in a similar amount of time. The inclusion of a turbulence modelling into the system leads to an increased computational time but also to more accurate results.

From the analysis of the effect of turbulence modelling at the reduced level, the following conclusions are drawn:

- the combination of corrections and turbulence modelling provides the most accurate reconstruction of the full order fields, as can be seen from the graphical representations displayed in Section 4.8.4;
- the addition of the turbulence terms improves the results obtained with the standard ROM approach only if a second order time integration scheme is considered. This fact can be the consequence of additional dissipation effects which characterize the turbulence modelling.

Chapter 5

Conclusions and Outlooks

This Thesis presents an investigation based on a data-driven approach applied in the Reduced Order Methods framework to the study of a turbulent flow around a circular cylinder. The technique developed in this work is based on introducing data-driven correction/closure terms to the online equations of the reduced order model. The main role of such terms is that of including the contribution of the unresolved modes. In the offline phase, the POD-Galerkin approach developed, named DD-ROM, requires the resolution of an optimization problem and calibrates the coefficients of the data-driven correction terms so as to minimize the distance between the effect of the correction and that of the unresolved modes on the solution.

Concluding remarks

In the online part of our analysis, different approaches and data-driven techniques are taken into account at the reduced order level. The following logical passages are considered.

- The online resolution approach initially considered in the Thesis is the SUP-ROM, based on the approximated supremizer enrichment of the POD velocity space [13, 14]. The first data-driven term introduced in the model is the velocity correction, based on the method developed in [10, 11, 12]. The velocity correction improves the results in terms of velocity and pressure field with respect to the standard SUP-ROM, as can be seen in Section 4.4. However, the improvement on the reduced pressure field is not as significant as that observed on the reduced velocity field.
- In order to further improve the pressure accuracy, correction terms including the pressure coefficients are proposed and added to the reduced system (Section 4.5). These terms do not appear to change the solution in a significant way and their effect is negligible with respect to the one of the velocity correction term. One possible reason for this fact is that the SUP-ROM formulation does not include a dedicated pressure equation, in which a correction term can directly affect the pressure field.
- Therefore, the different formulation of PPE-ROM is taken into account, in which

the Poisson pressure equation takes the place of the continuity equation. This formulation allows for the introduction of data-driven pressure correction terms, which finally lead to a significant improvement of the reduced pressure field, as can be seen in Section 4.7.

- In the final part of the Thesis (Section 4.8), the data-driven eddy viscosity turbulence modelling proposed in [1] is added to the reduced model alongside with the correction terms here proposed for the first time. The combination of both data-driven terms provides the best performance, leading to an increased accuracy of the reduced pressure and velocity field with respect to the high-fidelity solution, in both supremizer and Poisson approaches.

Outlooks

Finally, we propose some suggestions about future extensions of the work carried out in the present Thesis.

- The pressure data-driven corrections developed in this work, when introduced in the SUP-ROM formulation, have not significantly improved the results of the standard formulation in terms of the accuracy of the pressure field. However, the supremizer approach, first introduced in [13] and explored in [14], has been a successful technique for the stabilization of the POD-Galerkin ROMs. Therefore, further data-driven terms including the reduced pressure coefficients should be explored and tested to identify an effective pressure data-driven correction for the SUP-ROM formulation.
- In this Thesis, the only parameter considered in the reduced order simulations is time. The correction terms introduced in the reduced formulation are computed by solving an optimization problem, in which only data from a limited amount of time instants is taken into account. Thus, the matrices appearing in the data-driven corrections are assumed to be time and parameter independent. An interesting task for the future would be the introduction of a parameter in the reduced formulation — for instance the velocity at the inlet of the domain — as in [1]. In that case, the goal would be that of expressing the data-driven terms as a function of both the parameter considered and of time.
- In the present work, data-driven correction techniques have proven to be successful in the *marginally-resolved* regime for the case study of a turbulent flow around a cylinder. In future works, the data-driven techniques can be tested on case studies with a more complex setting and in different modal regimes, in order to analyse how the effect of the corrections varies case-by-case.

Bibliography

- [1] Saddam Hijazi, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. Data-driven POD-Galerkin reduced order model for turbulent flows. *Journal of Computational Physics*, 416:109513, 2020.
- [2] Mahmoud Gadalla, Marta Cianferra, Marco Tezzele, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. On the comparison of LES data-driven reduced order approaches for hydroacoustic analysis. *Computers & Fluids*, page 104819, 2021.
- [3] Matteo Zancanaro, Markus Mrosek, Giovanni Stabile, Carsten Othmer, and Gianluigi Rozza. Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. *Submitted*, 2021.
- [4] Willem Cazemier. *Proper orthogonal decomposition and low dimensional models for turbulent flows*. Rijksuniversiteit Groningen, 1997.
- [5] Jeanne A Atwell and Belinda B King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and computer modelling*, 33(1-3):1–19, 2001.
- [6] Peter Benner, Volker Mehrmann, and Danny C Sorensen. *Dimension reduction of large-scale systems*, volume 45. Springer, 2005.
- [7] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515, 2002.
- [8] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.
- [9] Pierre Sagaut. *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media, 2006.
- [10] Xuping Xie, Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40(3):B834–B857, 2018.
- [11] Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. Physically constrained data-driven correction for reduced-order modeling of fluid flows. *International Journal for Numerical Methods in Fluids*, 89(3):103–122, 2019.

- [12] Changhong Mou, Birgul Koc, Omer San, Leo G Rebholz, and Traian Iliescu. Data-driven variational multiscale reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 373:113470, 2021.
- [13] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5):1136–1161, 2015.
- [14] Giovanni Stabile and Gianluigi Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids*, 173:273–284, 2018.
- [15] J-L Guermond and Luigi Quartapelle. On the approximation of the unsteady Navier–Stokes equations by finite element projection methods. *Numerische mathematik*, 80(2):207–238, 1998.
- [16] *OpenFOAM* website. <https://openfoam.org/>.
- [17] Giovanni Stabile and Gianluigi Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations. *Computers and Fluids*, 2018.
- [18] Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, and Gianluigi Rozza. POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder. *Communications in Applied and Industrial Mathematics*, 8(1):210–236, (2017).
- [19] *ITHACA-FV|mathLab innovating with mathematics* website. <https://mathlab.sissa.it/ithaca-fv>.
- [20] Fadl Moukalled, L Mangani, Marwan Darwish, et al. *The finite volume method in computational fluid dynamics*, volume 113. Springer, 2016.
- [21] Hrvoje Jasak. Error analysis and estimation for the finite volume method with applications to fluid flows. 1996.
- [22] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 54–73. Elsevier, 1983.
- [23] Raad I Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1):40–65, 1986.
- [24] Giovanni Stabile, Matteo Zancanaro, and Gianluigi Rozza. Efficient Geometrical parametrization for finite-volume based reduced order methods. *International Journal for Numerical Methods in Engineering*, 121(12):2655–2682, 2020.

- [25] Osborne Reynolds. IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical transactions of the royal society of London*, (186):123–164, 1895.
- [26] D Brian Spalding. The numerical computation of turbulent flow. *Comp. Methods Appl. Mech. Eng.*, 3:269, 1974.
- [27] Andrej Nikolaevich Kolmogorov. Equations of turbulent motion in an incompressible fluid. In *Dokl. Akad. Nauk SSSR*, volume 30, pages 299–303, 1941.
- [28] Florian R Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32(8):1598–1605, 1994.
- [29] Stefano Lorenzi, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Computer Methods in Applied Mechanics and Engineering*, 311:151–179, 2016.
- [30] Gianluigi Rozza and Karen Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer methods in applied mechanics and engineering*, 196(7):1244–1260, 2007.
- [31] Anna-Lena Gerner and Karen Veroy. Certified reduced basis methods for parametrized saddle point problems. *SIAM Journal on Scientific Computing*, 34(5):A2812–A2836, 2012.
- [32] Imran Akhtar, Ali H Nayfeh, and Calvin J Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theoretical and Computational Fluid Dynamics*, 23(3):213–237, 2009.
- [33] Katarzyna Bizon and Gaetano Continillo. Reduced order modelling of chemical reactors with recycle by means of POD-penalty method. *Computers & chemical engineering*, 39:22–32, 2012.
- [34] Ivo Babuška. The finite element method with penalty. *Mathematics of computation*, 27(122):221–228, 1973.
- [35] John W Barrett and Charles M Elliott. Finite element approximation of the Dirichlet problem using the boundary penalty method. *Numerische Mathematik*, 49(4):343–366, 1986.
- [36] Irina Kalashnikova and Matthew F Barone. Efficient non-linear proper orthogonal decomposition/Galerkin reduced order models with stable penalty enforcement of boundary conditions. *International Journal for Numerical Methods in Engineering*, 90(11):1337–1362, 2012.
- [37] Sirod Sirisup and George Em Karniadakis. Stability and accuracy of periodic flow solutions obtained by a POD-penalty method. *Physica D: Nonlinear Phenomena*, 202(3-4):218–237, 2005.

-
- [38] S Kelbij Star, Giovanni Stabile, Francesco Belloni, Gianluigi Rozza, and Joris Degroote. A novel iterative penalty method to enforce boundary conditions in Finite Volume POD-Galerkin reduced order models for fluid dynamics problems. *arXiv preprint arXiv:1912.00825*, 2019.
- [39] Damiana Lazzaro and Laura B Montefusco. Radial basis functions for the multivariate interpolation of large scattered data sets. *Journal of Computational and Applied Mathematics*, 140(1-2):521–536, 2002.
- [40] Charles A Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive approximation*, 2(1):11–22, 1986.
- [41] Yi Wang, Bo Yu, Zhizhu Cao, Weizhong Zou, and Guojun Yu. A comparative study of POD interpolation and POD projection methods for fast and accurate prediction of heat transfer problems. *International Journal of Heat and Mass Transfer*, 55(17-18):4827–4836, 2012.
- [42] S Walton, O Hassan, and K Morgan. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Applied Mathematical Modelling*, 37(20-21):8930–8945, 2013.
- [43] Filippo Salmoiraghi, Angela Scardigli, Haysam Telib, and Gianluigi Rozza. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4-5):233–247, 2018.
- [44] Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for noninvasive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.
- [45] Nikolaus Adams. Mathematics of Large Eddy Simulation of Turbulent Flows. By LC Berselli, T. Iliescu and WJ Layton. Springer, 2006. 348 pp. ISBN 987 3 540 26316 6. *Journal of Fluid Mechanics*, 582:473, 2007.
- [46] Changhong Mou, Honghu Liu, David R Wells, and Traian Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *International Journal of Computational Fluid Dynamics*, 34(2):147–159, 2020.
- [47] Xuping Xie, David Wells, Zhu Wang, and Traian Iliescu. Approximate deconvolution reduced order modeling. *Computer Methods in Applied Mechanics and Engineering*, 313:512–534, 2017.
- [48] JH Gerrard. Flow around Circular Cylinders; Volume 1. Fundamentals. By MM Zdravkovich. Oxford Science Publications, 1997. 672 pp.£ 120. *Journal of Fluid Mechanics*, 350:375–378, 1997.
- [49] Volker John and Gunar Matthies. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods in Fluids*, 37(8):885–903, 2001.

- [50] Momchilo M Zdravkovich. *Flow around circular cylinders: Volume 2: Applications*, volume 2. Oxford university press, 1997.
- [51] Momchilo M Zdravkovich and Peter W Bearman. *Flow Around Circular Cylinders—Volume 1: Fundamentals*. 1998.
- [52] Muhammad Mohebujjaman, Leo G Rebholz, Xuping Xie, and Traian Iliescu. Energy balance and mass conservation in reduced order models of fluid flows. *Journal of Computational Physics*, 346:262–277, 2017.
- [53] Alfonso Caiazzo, Traian Iliescu, Volker John, and Swetlana Schyschlowa. A numerical investigation of velocity–pressure reduced order models for incompressible flows. *Journal of Computational Physics*, 259:598–616, 2014.
- [54] Mengwu Guo and Jan S Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer methods in applied mechanics and engineering*, 345:75–99, 2019.
- [55] S Kelbij Star, Giovanni Stabile, Francesco Belloni, Gianluigi Rozza, Joris Degroote, et al. Extension and comparison of techniques to enforce boundary conditions in finite volume POD-Galerkin reduced order models for fluid dynamic problems. *arXiv preprint arXiv:1912.00825*, 2019.
- [56] Birgul Koc, Muhammad Mohebujjaman, Changhong Mou, and Traian Iliescu. Computation error in reduced order modeling of fluid flows. *Advances in Computational Mathematics*, 45(5):2587–2621, 2019.