

# POLYTECHNIC UNIVERSITY OF TURIN

Department of Mathematical Sciences "Giuseppe Luigi Lagrange"  
Master course in Mathematical Engineering  
Pathway of Mathematical Models and Numerical Simulations

Master Thesis

## Reduced Order Methods for Inverse Problems in CFD



### **Advisors**

Prof. Claudio Canuto  
Prof. Gianluigi Rozza

### **Co-Advisors**

Dr. Giovanni Stabile  
Dr. Matteo Zancanaro  
Ing. Umberto Emil Morelli

### **Candidate**

Angelo Cetrangolo

Academic Year 2020-2021



*Alla mia famiglia e alla  
mia fidanzata Roberta  
sempre vicini in ogni  
traguardo della mia vita.*

## Abstract

Given a physical phenomenon, we mathematically represent this through equations, which define the so-called "direct problem". By the term "inverse problem", we mean the search for an input of the direct model that provides a certain output, of which, we have total or partial observations. Mathematically, these inverse problems can be treated as "optimal control problems", in which the control is the input of the direct problem which we want to find minimizing the gap between the observed data and those obtained by varying the control.

We focus on the inverse problems in fluid dynamics, because there are many applications in science and engineering. The amount of resources needed to solve them by a computer, such as time and storage, also called "computational cost", is great. Moreover, the wanted input can depend on one or more parameters. Then, the computational cost increases with the number of parameters. For this reason, we want to find a solution methods that allow us to reduce the computational cost.

To solve inverse problems, we employ a "optimize-then-discretize" approach. At first, an optimization procedure is defined. Then, it is discretized to solve it by a computer. To reduce the computational cost of high-fidelity solutions, we investigate the "Reduced Basis" (RB) method.

Quantifying fluid properties in the governing equations, which are essential for predictive modelling, remains a challenging problem. For this reason, we test the study case of the viscosity estimation from the data observed in lid-driven cavity flow. We study three set-ups for our test case, which differ in observed data used: only velocity, only pressure, or both.

At first, we implement an iterative procedure for the optimization step starting from an initial guess, also called the "Conjugate Gradient" (CG) method. Then, we use "Finite Volume" (FV) discretization to approximate the solution of the equations involving in the CG method. We call the previously described procedures as "Full Order" (FO) method.

We employ the RB method called the "Proper Orthogonal Decomposition" (POD) which exploits the information obtained from a certain number of simulations with the FOM, calculated in a typically expensive "offline phase", to construct the reduced approximations of the solutions. With reduced order approximations, for optimization step, we emulate full-order CG solver, where we replace the FV discretization of involving equations with a reduced approximation of them, which can be solved in a limited amount of time, with a lower computational effort, in a so-called "online phase".

From full-order results on the test case, we observe different performances and accuracy changing the type of observed data. Moreover, increasing the viscosity wanted, its estimation with only velocity data is less accurate and perform in larger amount of time than with other observation set-ups. Furthermore, the numerical methods employed are expensive. Finally, we test the RB method in the case with only velocity observations given, obtaining an accurate estimation of viscosity with low computational cost. Then, the RB techniques could be a good and viable way to enhance the computational performances of inverse problem solvers.

# Acknowledgements

First of all, I would like to thank you my thesis advisor Prof. Claudio Canuto of Politecnico di Torino for giving me the opportunity to work on this Master Thesis and the internship in collaboration with the Scuola Internazionale Superiore di Studi Avanzati (SISSA) in *mathLab* group.

A very special thank you goes to Prof. Gianluigi Rozza of SISSA for his availability, his patient and his academic and psychological support in last year of the pandemic. Thanks you for for introducing me to the wonderful world of PDE's optimal control problem, in particular inverse problem in CFD, unknown to me. Thank you for believing in my abilities and for teaching me that there is always a remedy for mistakes, the important thing is to maintain energy and enthusiasm in everything I do. I wish you the best of luck for the *mathLab* research group which, thanks to its organization, dedication and motivation, is able to improve.

A huge thank you goes to Dr. Giovanni Stabile for his patience and professionalism in solving my problems and listening all my doubts. I really want to thank my young supervisors in SISSA, Umberto Emil Morelli, for his support not only by theoretical point of view but also for computational procedure on inverse problem, Dr. Matteo Zancanaro, for his support in computational fluid-dynamics. I really thank them also for having encouraged and supported me during last year of my internship with SISSA.

I would like to acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD and ARIA "Accurate ROMs for Industrial Applications" - GA 872442, H2020-MSCA-2019.

Finally, but most importantly, I want to express my very profound gratitude to my parents, for giving me the opportunity to pursue this goal, and to my girlfriend, for providing me with support and continuous encouragement throughout my years of study.

*Angelo Cetrangolo.*

# Contents

<b>List of Tables</b>	5
<b>List of Figures</b>	6
<b>Introduction and Motivation</b>	9
<b>1 Inverse problems like PDEs-constrained optimal control problems</b>	14
1.1 Optimal control problems	15
1.2 Derivative-based optimization	16
1.2.1 Sensitivity approach	17
1.2.2 Adjoint approach	17
1.3 Optimality conditions and Karush-Kuhn-Tucker system	18
1.3.1 Optimality equation	18
1.3.2 Karush-Kuhn-Tucker optimality system	19
1.4 Ill-posedness and Regularization of inverse problems	19
1.4.1 Definition of ill-posed inverse problem	19
1.4.2 Solution strategy	20
1.5 Case study: Learning viscosity in lid-driven cavity flow	22
1.5.1 Direct problem	22
1.5.2 Inverse problem	25
<b>2 Full Order Model Formulation</b>	29
2.1 Non-linear Projected Conjugate Gradient Method	29
2.2 Finite Volume Discretization	32
2.3 Segregate Pressure-Based Solver in FVM	36
2.3.1 Momentum Equation	37
2.3.2 Pressure Equation	37
<b>3 Reduced Order Model Formulation</b>	41
3.1 Generation of the reduced basis space with a Proper Orthogonal Decomposition approach	44
3.1.1 Proof on construction of modes	46
3.2 Galerkin projection onto the reduced spaces	50
3.2.1 Reduced State Equations	51

3.2.2	Reduced Adjoint Equations . . . . .	53
3.2.3	Reduced Optimality Equation . . . . .	54
3.3	Stability issues . . . . .	54
3.3.1	Supremizer Enrichment . . . . .	56
3.4	Non-homogeneous Diriclet boundary conditions treatment . . . . .	59
3.4.1	Lifting Function Method . . . . .	60
3.5	Reduced Order Non-linear Projected Conjugate Gradient Method . . . . .	63
3.6	Offline-Online Decomposition . . . . .	66
<b>4</b>	<b>Numerical Experiments</b> . . . . .	<b>69</b>
4.1	Full Order Approximations . . . . .	70
4.1.1	Description of the full order implementation . . . . .	70
4.1.2	Comparison between different observed data . . . . .	73
4.1.3	Increasing wanted viscosity . . . . .	74
4.1.4	Final remarks . . . . .	79
4.2	Reduced Order Approximations . . . . .	80
4.2.1	Description of the reduced order implementation . . . . .	80
4.2.2	Results . . . . .	81
4.2.3	Final remarks . . . . .	83
	<b>Conclusion and perspectives</b> . . . . .	<b>91</b>
	<b>Bibliography</b> . . . . .	<b>101</b>

# List of Tables

- 4.1 Mesh properties with FV discretization. . . . . 70
- 4.2 Parameters used in the SIMPLE algorithm. . . . . 72
- 4.3 GAMG parameter. . . . . 72
- 4.4 Accuracy for  $\nu_d = 10^{-2}$  with difference observed data. . . . . 74
- 4.5 FOM vs ROM, Error for  $\mu = 1.55 \times 10^{-2}$  . . . . . 82

# List of Figures

1.1	Geometry of the lid-driven cavity. . . . .	23
4.1	Computational mesh . . . . .	71
4.2	FOM results on $v_x$ , $v_y$ and $p$ with velocity data input . . . . .	74
4.3	FOM results on $v_x$ , $v_y$ and $p$ with pressure data input . . . . .	75
4.4	FOM results on $v_x$ , $v_y$ and $p$ with pressure data input . . . . .	76
4.5	FOM results from $\nu_0 = 10^{-3}$ to $\nu_d = 1$ . . . . .	77
4.6	FOM results from $\nu_0 = 10^{-3}$ to $\nu_d = 10$ . . . . .	78
4.7	FOM results from $\nu_0 = 10^{-3}$ to $\nu_d = 100$ . . . . .	85
4.8	Energy of the eigenvalues reduction for the case study. . . . .	86
4.9	Performances and accuracy of the reduced CG method for the parameter $\mu = 1.55 \times 10^{-2}$ , from the initial guess $\nu_0 = 10^{-3}$ . . . . .	86
4.10	ROM results on $v_x$ , $v_y$ and $p$ and comparison with FOM . . . . .	87
4.11	Averaged log-error, ROM vs FOM solutions. . . . .	88
4.12	Comparison FOM vs ROM cost function and computational performances. . . . .	89

# List of Algorithms

- 1 Non-linear Projected Conjugate Gradient Method. . . . . 31
- 2 The SIMPLE Method. . . . . 39
- 3 Proper Orthogonal Decomposition Method. . . . . 51
- 4 Exact Supremizer Enrichment Method. . . . . 59
- 5 Reduced Non-linear Projected Conjugate Gradient Method. . . . . 67
- 6 The offline procedure. . . . . 68
- 7 The online procedure. . . . . 68

*If God has created the world,  
his primary worry was certainly not  
to make its understanding easy for us.*

[FEBRUARY 1954, Quoting Albert  
Einstein.]

# Introduction and Motivation

The present Master Thesis investigates the numerical solution of *inverse problems in fluid dynamics* framework.

Inverse problems in fluid dynamics are widespread in science and engineering. In [33], the authors determine unknown wall heat flux values using free stream measurements for laminar forced flow through a parallel duct. Another inverse problem studied is identification of flow obstructions, as in [39]. Moreover, application in identification of heat and contaminant sources, there is in [48]. A possible application sector is aerodynamic design, for example in [35] which investigates the diffuser with minimal axial deviation at the outlet and the stage with maximum thrust. In [9] we can see an inverse problem in biomedical science. In this work, the authors show results obtained on 2D curved domains, retrieving from the assimilated velocities a flow-related variable of medical relevance, namely the Wall Shear Stress (WSS). In industrial application, there are some inverse problem in optimal shape design, such as in [36] which search shape of nozzle with a desirable lower wall pressure distribution. In fluid-structure interaction setting, a possible inverse problem is to identify inflow velocity fields from the knowledge of fluid loads time history on a fixed solid represented by the 2D section of a bridge deck, as studied in [16]. Finally, but most importantly, a sector of application is marine science. For example in [6], the authors take into consideration an inverse problem capable to guess what are the physical conditions that can represent a desired velocity-height profile with shallow water equations.

For those applications it is very easy to observe a high computational cost deriving from PDEs solver employed. Moreover, the wanted input of direct problem can be depend on one or more parameter. Thus, the numerical solution of the corresponding inverse problem is expensive because the computational cost increases with the dimension of space parameter. For this reason, we research a way to reduce this cost. For this goal, we investigate the application of projection - based Reduced Order Methods for inverse problems in fluid dynamics.

The inverse problem in fluid dynamics, from a mathematical point of view, can be analyzed as an *optimal control problem*, with constrained given by *Partial Differential Equations (PDEs)*. This approach permits us to study the ill-posedness and the stability of the solutions. In this constrained optimization framework, there are many works that investigate the regularization of ill-posed inverse problems, the existence and uniqueness of the solutions and numerical full-order method, such as in [59, 58, 43, 70, 23, 27]. In particular, optimal control problems can be seen, from the theoretical point of view, as a minimization of a functional cost representing the difference between the desired data and

solutions of the problem governed by the control parameter variable. For this reason, at full-order level, we want to solve these problems with derivative-based methods. By the use of this approach, we can obtain the so called *Karush-Kuhn-Tucker (KKT) optimality system*. This system consists of three blocks of equations:

- *Direct Equations*, which are given by the equations of state together with boundary conditions;
- *Optimality Equations*, which are the sufficient conditions of first order optimality, dependent on state and adjoint variables;
- *Adjoint Equations*, that close the system.

This system can be obtained by the application of two different derivative-based methods, also known as the *sensitivity* and *adjoint* approaches, which will be recalled in the next chapter for the sake of completeness. The second procedure can be obtained by rewriting the constrained optimal control problem as an unconstrained problem via a *Lagrangian* approach. Then, we can derive the Lagrangian with respect to the state variables, under appropriate hypotheses, to obtain the adjoint equations and with respect to the control to obtain the optimality condition.

In this Thesis, the *Full Order Model (FOM)* for direct problem is the incompressible laminar steady *Navier–Stokes Equations (NSE)*. As well-known, the numerical solution of the incompressible NSE at Full Order level is particularly expensive in terms of computational cost and CPU time. Instead, *Reduced Order Models* exploit the information obtained from a set of full order simulations by a very expensive *offline* phase, to solve a lower dimensional problem with a lower computational cost, during the *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. In particular, these methods are employed with both physical and geometrical parametrization and/or turbulent unsteady dynamics, such as in the recent works [65, 31, 29, 30] with finite volume discretization.

Instead, in this work, we want to solve inverse problems with state equations given by Navier-Stokes equations. The resolution of the KKT system can be carried out through both coupled or decoupled approaches. To avoid stability issues due to the saddle-point structure of this system, we adopted a segregated approach to find the minimizer through a *Conjugate Gradient (CG) method*. This procedure is widely used in optimization as a basic method and also as a regularization technique. For more detail on this, we refer to [49, 4, 14, 46, 11, 53, 15, 27, 70].

At reduced order level, the *parametric optimal control problems ( $\mu$ -OCP)* have been studied in many different works to obtain real-time solutions, with low computational cost, varying the parameter. There are application in biomedical engineering [75], environmental sciences [69], heat inverse problems for industrial requirements [50], in non-linear and time dependent optimal flow [67, 68]. Usually, the application of ROM for optimal flow control or inverse problem, is in a finite element framework. In this work, instead, we want to extend reduced optimal flow techniques in finite volume framework. We adopt this approximation procedure because finite volume became the standard choice for real

world applications in several engineering fields (Aeronautics, Industrial flows, Automotive, Naval Engineering) providing high fidelity solutions. Moreover, we have an easier implementation and the possibility to work on generic polyhedral cells while being locally conservative. Furthermore, in flow simulations, increasing Reynolds numbers, there are less issues concerning stability and several turbulence models already available.

In particular, this Thesis focuses on *Snapshots-based ROMs* constructed with a *POD-Galerkin method (POD-Galerkin ROMs)* [62, 28]. These ROMs are built by following the four steps reported here:

1. Compute a set of solutions (or snapshots) of the Full Order model and store it;
2. Construct a set of basis functions (modes), identified applying the *Proper Orthogonal Decomposition* (POD) on snapshots (see for detail [20, 41, 72]); thus, we can reconstruct the unknown solution lives as linear span of this modes;
3. Project the FOM optimality equations onto the space spanned by these modes. The resulting *Galerkin ROM (G-ROM)* is a system of equations where the unknowns are the modal coefficients;
4. Solve the reduced order system with a particular reduced optimization algorithm.

Since the ROM exploits the information of FOM stored solutions, it is capable of providing good accuracy while significantly reducing the degrees of freedom. Thus, the solution of reduced system require lower computational effort and CPU time with respect to a high fidelity resolution.

The case study considered to test the efficiency of the approach developed is that of *learning a constant viscosity in a steady laminar lid-driven cavity flow*. The reduced order model for adjoint and state problems is treated with a pressure/velocity coupling technique, namely the *exact supremizer enrichment approach (SUP-ROM)* which consists in the addition of velocity modes — namely supremizer modes — in the velocity POD space in order to fulfill the inf-sup condition. The inf-sup condition is typical in *saddle-point structure* problems, and the treatment at reduced order level is introduced in [61, 8, 60, 17, 64].

For non-homogeneous Diriclet boundary conditions (BCs) treatment we use a *lifting function* to homogenize the snapshots, before the application of the POD for the evaluation of the modal basis, as it has been done in [66, 22].

Furthermore, to solve the reduced optimization problem we adopt a reduced version of FOM conjugate gradient method. This idea was born to avoid the stability issue related with the saddle point structure of reduced KKT system and to emulate the full order solution. We evaluate his effect in terms of accuracy in reduced fields and control, with respect to their FOM counterparts.

In this work, for the offline phase, Full Order solution are computed making use of the open source software *OpenFOAM*<sup>1</sup> for finite volume (FV) approximation of PDEs [51]. We implemented a non-linear version of the conjugate gradient method to find the minimizer

---

<sup>1</sup><https://www.openfoam.com/>

viscosity constant from observed data. Remarking that adjoint equations have the same pressure-velocity coupling of state equations, we adopted a Segregated Pressure-Based Solver, the *SIMPLE algorithm (Semi-Implicit Method for Pressure-Linked Equation)* [51], for both problems. The used POD algorithms are coded into the *ITHACA-FV*<sup>2</sup> (*In real Time Highly Advanced Computational Applications for Finite Volumes*) package, an open source C++ library containing several reduced order modelling techniques in a finite volume framework, based on OpenFOAM and developed at the *mathLab*<sup>3</sup> group in SISSA. Most of the theoretical aspects behind ITHACA-FV are deeply explained in [64, 66]. Linear and non-linear algebra operations, useful to solve the reduced systems in the online phase, which are not already implemented in OpenFOAM, are performed with the external library *Eigen* [21]. The source code of this library is provided together with ITHACA-FV. Data visualizations and plots are performed using *Paraview*, an open-source, multi-platform data analysis and visualization application [1, 5], and packages *numpy* [26], *matplotlib* [34] in *python* language [71].

Summarizing, this Thesis has the following structure:

- Chapter 1, Inverse problems as PDEs-constrained optimal control problems. In this chapter we will recall the general formulation of optimal control problems constrained by partial differential equations (PDEs). A brief overview on ill-posedness of inverse problems and regularization theory will also be included. Finally we will introduce our test case where the direct problem describes an isothermal, incompressible steady-state flow in a cavity; while, we introduce the inverse problem of find viscosity from observed data;
- Chapter 2, Full Order Model (FOM). In this chapter, we develop the Full-Order Model for the inverse problem using a conjugate gradient method to solve the KKT system in a decoupled way and we recall the finite volume discretization for state and adjoint equations, which are numerically solved employing a segregate pressure-based solver;
- Chapter 3, Reduced Order Model (ROM). In this chapter, we study the corresponding Reduced Order Model. A review on the construction of POD modes for state and adjoint fields is given. The Galerkin projection onto the POD spaces for state equations, adjoint equations and optimality condition, is separately performed. In this chapter we also investigate the stability issue of the reduced equations, fixed by the employment of the supremizer enrichment, and describe non-homogeneous Diriclet BC treatment with a lifting function method. Finally we develop a reduced version of the CG method for this particular test case;
- Chapter 4, Numerical Results. In this chapter we show numerical results of the full order model for the test case inverse problem. A comparison between the results for velocity data only, pressure data only or both of them is provided while varying

---

<sup>2</sup><https://mathlab.sissa.it/ITHACA-FV>

<sup>3</sup><https://mathlab.sissa.it/>

the constant viscosity in a fixed range. Finally we test the reduced order method previously introduced in both accuracy and velocity terms; the approximated fields for velocity and pressure are displayed and compared first of all with the Full Order Model solutions;

- Conclusion and perspectives. In this final chapter we make a comparison between desired and obtained goals, we critically comment on the results obtained and we underline the parts requiring a deeper analysis, introducing possible further developments for this research.

# Chapter 1

## Inverse problems like PDEs-constrained optimal control problems

The term "inverse problem" does not have a well-defined mathematical definition, since its meaning is based on notions of physics, but let's go back to the textbook definition given in [59]. Suppose there is a known map from the input to the output of our system

$$t : U \rightarrow T,$$

that models a mathematical-physical law or a tech device. Here,  $U$  is a set of "causes", the input to our model, and  $T$  is a set of "effects", the consequent output.

The computation of an effect  $t(u)$  from a known cause  $u$  is called a *direct problem*. Finding a cause  $u \in U$ , which implies a given effect  $t_d \in T$ , is called as *inverse problem*. Solving an inverse problem therefore means to invert this map  $t$  finding a input  $u$  such that equation  $t(u) = t_d$  is satisfied.

Sometimes, we want a certain desired effect  $t_d \in T$  and are looking for a certain  $u \in T$  which produce it. An inverse problem of this type is called a *optimal control problem*, where the control is our unknown cause and the cost functional is given by the difference between desired data and output of direct problem varying the control. A particular optimization problem is to identify a parameter of direct model knowing partially or completely the output. An inverse problem of this type is called an *identification parameter problem*.

As it will be shown, inverse problems typically are not well-posed in the sense of Hadamard, i.e., to *ill-posed problems*. For this reason regularization techniques are required to lead ill-posed problems in a well-posed formulation. Thus, we recall the definition of ill-possdness and iterative regularization method used to solve this type of problems.

We assume that inverse problems, treated in this work, can be seen, from the mathematical point of view, as a constrained optimization problem, whose constraints are given by the particular model used to represent the physical phenomenon. In most cases one has to deal with constraints given by by Partial Differential Equations (PDEs). For this

reason, we will introduce the general formulation of PDEs-constrained optimal control problems.

This chapter is organized as follows: in the first and second sections, we will discuss the continuous formulation of optimal control problems and derivative-based optimization strategies, respectively. In the third section, we will discuss the optimality condition. Subsequently, we will recall the mathematical theory of ill-posed problems and regularization methods. Finally, we propose a simple test case in which we derive the inverse formulation from direct problems given by steady state Navier-Stokes equations in laminar regime.

## 1.1 Optimal control problems

In many application contexts, as already mentioned, the direct model is given by equations. Usually, these inverse problems can be treated as constrained optimal control problems, in which the control is input of direct problem that we want to find and the cost functional to minimize is a gap between the observed data and those obtained by varying the control. Some aspects on theory of optimal control are presented below. For this section we refer to [43, 58, 74, 70]. We will limit ourselves to the case of systems represented by Partial Differential Equations (PDEs). For this goal, we start by introducing the mathematical entities that come into play in the theory:

- the *control function*  $\mathbf{u} \in U_{ad}$ , where  $U_{ad}$  is the space of *admissible controls*, appropriately chosen according to constraints imposed on  $\mathbf{u}$ . Observe that  $U_{ad} \subseteq U$ , where  $U$  is the most adequate functional space to describe the role that the control function  $\mathbf{u}$  assumes in the equations. If  $U_{ad} = U$  the control will be said to be *unconstrained*; if instead  $U_{ad} \subset U$  we are dealing with a *constrained* control;
- the *state* of the system  $\mathbf{s}(\mathbf{u}) \in S$  (an appropriate functional space), depending on the value assumed by the control  $\mathbf{u}$ , which satisfies the linear/non-linear state governing equations given by

$$F(\mathbf{s}, \mathbf{u}) = 0,$$

where  $F : S \times U \rightarrow S^*$  is the residual of the state equations;

- the *observation function*, indicated by  $\mathbf{t}(\mathbf{u})$  which depends on  $\mathbf{s}(\mathbf{u})$  and an appropriate operator  $C$ :

$$\mathbf{t}(\mathbf{u}) = C(\mathbf{s}(\mathbf{u})),$$

where  $\mathbf{t} \in T$  ( $T$  is the space of the *observed functions*);  $\mathbf{t}(\mathbf{u})$  must be compared with the *desired observation function*  $\mathbf{t}_d$ ;

- a *cost functional*  $J : S \times U \rightarrow \mathbb{R}$  is a quadratic objective functional to be minimized. In general,  $J$  will depend on  $\mathbf{t}_d$  and  $\mathbf{u}$  (also) through  $\mathbf{t}$ .

Let us consider Hilbert spaces  $S$  and  $U$ , the optimal control problem can be summarized in:

**Problem 1.1** Find optimal solution  $(\mathbf{s}, \mathbf{u}) \in S_{ad} \times U_{ad}$  such that

$$\min_{(\mathbf{s}, \mathbf{u}) \in S \times U} J(\mathbf{s}, \mathbf{u}) \quad \text{subject to} \quad F(\mathbf{s}, \mathbf{u}) = 0, \quad \mathbf{s} \in S_{ad}, \mathbf{u} \in U_{ad}.$$

In particular cases it is possible to prove the existence and uniqueness of the optimal solution for a general non-linear quadratic problem. For example, in [70], existence and uniqueness results is established by the following theorem.

**Theorem 1.1** *The Problem 1.1 has an optimal solution  $(\mathbf{s}, \mathbf{u}) \in S \times U$  under the following assumptions:*

1.  $S_{ad} \subseteq S$  and  $U_{ad} \subseteq U$  are convex and closed;
2. The feasible set  $G = \{ (\mathbf{s}, \mathbf{u}) \in S \times U : F(\mathbf{s}, \mathbf{u}) = 0 \} \neq \emptyset$  ;
3. State governing equations are continuous and well-posed;
4.  $J$  is weakly lower semi-continuous.

## 1.2 Derivative-based optimization

In this section we will explore optimization techniques to solve PDEs-constrained optimal control problems, as in [70, 74]. Essentially, there are two methods to do this:

- i. the *sensitivity approach*,
- ii. the *adjoint approach*.

We consider the optimization problem unconstrained with respect to state and control variables, that is,  $S_{ad} = S$  and  $U_{ad} = U$  and  $F(\mathbf{s}, \mathbf{u})$  to be the residual of state governing equations. Also, we assume that:

- a.  $J : S \times U \rightarrow \mathbb{R}$  and  $F : S \times U \rightarrow S^*$  are continuously Fréchet differentiable;
- b.  $\forall \mathbf{u} \in U, \exists! \mathbf{s}(\mathbf{u}) \in S : F(\mathbf{s}, \mathbf{u}) = 0$ ;
- c.  $F_s(\mathbf{s}(\mathbf{u}), \mathbf{u}) : S \rightarrow S^*$  is bijective.

Then we have an operator  $\mathbf{u} \mapsto \mathbf{s}(\mathbf{u})$  which is continuously differentiable thanks to the Implicit Function Theorem. Also we can rewrite Problem 1.1, with *reduced objective functional*  $\tilde{J}(\mathbf{u}) := J(\mathbf{s}(\mathbf{u}), \mathbf{u})$ , and we obtain the following *reduced problem*:

**Problem 1.2** *Find optimal solution  $\mathbf{u} \in U$  such that*

$$\min_{\mathbf{u} \in U} \tilde{J}(\mathbf{u}) \quad \text{subject to } \mathbf{u} \in \tilde{U}_{ad} := \{ \mathbf{u} \in U \mid (\mathbf{s}(\mathbf{u}), \mathbf{u}) \in S \times U \}.$$

After having justified this reduced formulation we want to calculate the derivative of this new objective functional  $\tilde{J}$  through a sensitivity analysis or an adjoint approach. Below we give an outline of both methods for completeness.

### 1.2.1 Sensitivity approach

To compute the derivative of the reduced objective function  $\tilde{J}$  with the sensitivity approach, we make the following steps:

Step 1. Calculate the *sensitivity*  $\delta_{\mathbf{v}}\mathbf{s} = d\mathbf{s}(\mathbf{u}, \mathbf{v}) = \mathbf{s}'(\mathbf{u})\mathbf{v}$  (N.B. sensitivities are directional derivatives) differentiating the state equation  $F(\mathbf{s}(\mathbf{u}), \mathbf{u}) = 0$ , for  $\mathbf{u} \in U$  and a direction  $\mathbf{v} \in U$

$$F_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u})\delta_{\mathbf{v}}\mathbf{s} = -F_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u})\mathbf{v}. \quad (1.1)$$

Step 2. Calculate the directional derivative  $d\tilde{J}(\mathbf{u}, \mathbf{v}) = \langle \tilde{J}'(\mathbf{u}), \mathbf{v} \rangle_{U^*, U}$  thanks to the chain rule,

$$d\tilde{J}(\mathbf{u}, \mathbf{v}) = \langle J_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u}), \delta_{\mathbf{v}}\mathbf{s} \rangle_{S^*, S} + \langle J_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u}), \mathbf{v} \rangle_{U^*, U}. \quad (1.2)$$

With this method the computational cost to calculate the derivative of this objective functional increases with the dimension of  $U$ , since in Step 1 the sensitivity must be calculated for a basis  $\mathcal{B}$  of  $U$ .

### 1.2.2 Adjoint approach

A more efficient way of calculating the derivative of  $\tilde{J}$  is via the adjoint approach with the following three steps:

Step 1. Calculate the *adjoint state*  $\mathbf{z} = \mathbf{z}(\mathbf{u}) \in S^{**} = S$  which solves the following *adjoint equation*

$$F_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u})^* \mathbf{z}(\mathbf{u}) = -J_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u}). \quad (1.3)$$

Step 2. Calculate the term  $\mathbf{s}'(\mathbf{u})^* J_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u})$  from the equation (1.1) which can be re-written

$$\mathbf{s}'(\mathbf{u})^* J_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u}) = F_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u})^* \mathbf{z}(\mathbf{u}),$$

thanks the Step 1.

Step 3. Calculate  $\tilde{J}'(\mathbf{u})$  reformulating the relation (1.2)

$$\langle \tilde{J}'(\mathbf{u}), \mathbf{v} \rangle_{U^*, U} = \langle \mathbf{s}'(\mathbf{u})^* J_{\mathbf{s}}(\mathbf{s}(\mathbf{u}), \mathbf{u}) + J_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u}), \mathbf{v} \rangle_{U^*, U},$$

$\Updownarrow$

$$\tilde{J}'(\mathbf{u}) = F_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u})^* \mathbf{z}(\mathbf{u}) + J_{\mathbf{u}}(\mathbf{s}(\mathbf{u}), \mathbf{u}).$$

The adjoint gradient representation can also be derived by a Lagrangian formulation of Problem 1.1, as shown in [70]. Define the *Lagrange function*  $L : S \times U \times S^{**} \rightarrow \mathbb{R}$ ,

$$L(\mathbf{s}, \mathbf{u}, \mathbf{z}) = J(\mathbf{s}, \mathbf{u}) + \langle \mathbf{z}, F(\mathbf{s}, \mathbf{u}) \rangle_{S^{**}, S^*}, \quad (1.4)$$

where  $\mathbf{z} \in S^{**} = S$  is the *Lagrangian multiplier*. Thus, the Lagrangian formulation casts the PDEs-constrained problem into an unconstrained one by introducing additional variables.

## 1.3 Optimality conditions and Karush-Kuhn-Tucker system

In this section, at first we make a review of the theoretical results on the *optimality equation* for Problem 1.2. Next we will deal with the *sufficient optimality conditions of first order*. For the theoretical results contained in this section, we refer to [70, 74].

### 1.3.1 Optimality equation

Considering Problem 1.2, we have the following general result regarding the first order optimality condition. Its proof can be found in [70].

**Theorem 1.2** *If  $\tilde{\mathbf{u}} \in \tilde{U}$  is a local minimum of reduced objective functional  $\tilde{J}$ , then  $\tilde{\mathbf{u}}$  satisfies the following variational inequality also called optimality condition,*

$$\langle \tilde{J}'(\tilde{\mathbf{u}}), \mathbf{v} - \tilde{\mathbf{u}} \rangle_{U^*, U} \geq 0, \quad \forall \mathbf{v} \in \tilde{U}. \quad (1.5)$$

**Remark 1.1** *If  $\tilde{J}$  is convex, then the optimality condition is not only necessary but also sufficient for optimality. Moreover, the local minimum is also global in this case.*

*If  $\tilde{U} \equiv U$ , then the inequality condition becomes equality one.*

The optimality constraint (1.5) can be rewritten in an equivalent form using the Lagrangian (1.4). Inserting state of the system  $\mathbf{s}(\mathbf{u})$ , let an arbitrary  $\mathbf{z} \in Z = S^{**}$ , we define the Lagrangian as,

$$L(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}) = J(\mathbf{s}, \mathbf{u}) + \langle \mathbf{z}, F(\mathbf{s}(\mathbf{u}), \mathbf{u}) \rangle_{Z, S^*}. \quad (1.6)$$

Differentiating the Lagrangian (1.6), we have

$$\langle L'(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}), \mathbf{v} \rangle_{U^*, U} = \langle L_s(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}), \mathbf{s}(\mathbf{u})\mathbf{v} \rangle_{S^*, S} + \langle L_u(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}), \mathbf{v} \rangle_{U^*, U}.$$

In particular, we consider  $\mathbf{z} = \mathbf{z}(\mathbf{u}) \in Z$  solution of the following constraint,

$$\langle L_s(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}), \mathbf{s}(\mathbf{u})\mathbf{v} \rangle_{S^*, S} = 0,$$

which is the adjoint equation (1.3). Thus, the adjoint gradient representation can be re-written in the following way,

$$\begin{aligned} \langle \tilde{J}'(\mathbf{u}), \mathbf{v} \rangle_{U^*, U} &= \langle L_u(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}(\mathbf{u})), \mathbf{v} \rangle_{U^*, U} = \langle J_u(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}(\mathbf{u})) + F_u(\mathbf{s}(\mathbf{u}), \mathbf{u})^* \mathbf{z}(\mathbf{u}), \mathbf{v} \rangle_{U^*, U} \\ &\quad \Updownarrow \\ \tilde{J}'(\mathbf{u}) &= J_u(\mathbf{s}(\mathbf{u}), \mathbf{u}, \mathbf{z}(\mathbf{u})) + F_u(\mathbf{s}(\mathbf{u}), \mathbf{u})^* \mathbf{z}(\mathbf{u}). \end{aligned}$$

Then, the Theorem 1.2 can be reformulate in terms of Lagrangian, as developed in [74].

**Corollary 1.1** *If  $\tilde{\mathbf{u}} \in \tilde{U}$  is a local solution of reduced Problem 1.1, then the following optimality constraint is satisfied,*

$$\begin{aligned} \langle L_u(\mathbf{s}, \tilde{\mathbf{u}}, \mathbf{z}), \mathbf{v} - \tilde{\mathbf{u}} \rangle_{U^*, U} &\geq 0 \quad \forall \mathbf{v} \in \tilde{U}, \\ &\quad \Updownarrow \\ \langle J_u(\mathbf{s}, \tilde{\mathbf{u}}, \mathbf{z}) + F_u(\mathbf{s}, \tilde{\mathbf{u}})^* \mathbf{z}, \mathbf{v} - \tilde{\mathbf{u}} \rangle_{U^*, U} &\geq 0 \quad \forall \mathbf{v} \in \tilde{U}. \end{aligned}$$

We observe that, also in this case, if  $\tilde{U} \equiv U$ , the optimality inequality becomes equality one.

### 1.3.2 Karush-Kuhn-Tucker optimality system

Now, we show the coupled optimality system for PDEs-constrained optimal flow control problems composed by state, adjoint and optimality equations, recalled in previous subsection. The following Corollary of Theorem 1.2 summarize the teoretical results on this system, as in [70].

**Corollary 1.2** *Let us  $(\mathbf{s}, \mathbf{u}) \in S \times U$  a solution of the Problem 1.1, then  $\exists \mathbf{z} \in Z$  adjoint variable (or Lagrange multiplier) that it satisfies the following optimality system, also known as Karush-Kuhn-Tucker (KKT) optimality conditions,*

$$\begin{cases} F(\mathbf{s}, \mathbf{u}) = 0, \\ J_s(\mathbf{s}, \mathbf{u}) + F_s(\mathbf{s}, \mathbf{u})^* \mathbf{z} = 0, \\ \langle J_u(\mathbf{s}, \mathbf{u}, \mathbf{z}) + F_u(\mathbf{s}, \mathbf{u})^* \mathbf{z}, \mathbf{v} - \mathbf{u} \rangle_{U^*, U} \geq 0 \quad \forall \mathbf{v} \in \tilde{U}. \end{cases} \quad (1.7)$$

Otherwise, with the Lagrangian formulation, we can re-write this system in compact form, as follows

$$\begin{cases} \langle L_z(\mathbf{s}, \mathbf{u}, \mathbf{z}), \mathbf{p} \rangle_{Z^*, Z} = 0 \quad \forall \mathbf{p} \in Z, \\ \langle L_s(\mathbf{s}, \mathbf{u}, \mathbf{z}), \mathbf{y} \rangle_{S^*, S} = 0 \quad \forall \mathbf{y} \in S, \\ \langle L_u(\mathbf{s}, \mathbf{u}, \mathbf{z}), \mathbf{v} \rangle_{U^*, U} \geq 0 \quad \forall \mathbf{v} \in \tilde{U}, \end{cases}$$

or in equivalent way,

$$\begin{cases} \langle F(\mathbf{s}, \mathbf{u}), \mathbf{p} \rangle_{Z^*, Z} = 0 & \forall \mathbf{p} \in Z, \\ \langle J_s(\mathbf{s}, \mathbf{u}) + F_s(\mathbf{s}, \mathbf{u})^* \mathbf{z}, \mathbf{y} \rangle_{S^*, S} = 0 & \forall \mathbf{y} \in S, \\ \langle J_u(\mathbf{s}, \mathbf{u}, \mathbf{z}) + F_u(\mathbf{s}, \mathbf{u})^* \mathbf{z}, \mathbf{v} \rangle_{U^*, U} \geq 0 & \forall \mathbf{v} \in \tilde{U}. \end{cases}$$

We can use the KKT conditions to find a solution of optimal flow Problem 1.1. For this goal, we employ Full Order numerical approximations or Reduced Order.

## 1.4 Ill-posedness and Regularization of inverse problems

In this Section we exploit the theory of ill-posed inverse problems and regularization methods.

We define *ill-posedness*, which is the main issue when we treat inverse problems. We recall *Hadamard's definition* of ill-posed inverse problems. The ill-posedness can be solved by using appropriate numerical methods which stabilizes the problem, called *regularization techniques*. For more theoretical detail on ill-posedness we refer to [59, 37], while, for regularization theory we refer to [38, 40].

### 1.4.1 Definition of ill-posed inverse problem

Now, we recall the definition of inverse problem. Given two normed spaces  $U$  and  $T$ , an operator  $t : X \rightarrow Y$  and a measurement  $t_d$ , we can formulate the inverse problem as the

solution of the following equation

$$t(u) = t_d \in T, \tag{1.8}$$

for  $u \in U$ , when  $t_d$  is given. In order to formulate correctly an inverse problem by a mathematical model point of view, the definition of the operator  $t$ , including its domain and image space, has to be defined. In general, solving inverse problem  $u = t^{-1}(t_d)$  means inverting direct problem, often, given by a differential equation. Usually, inverse problems are ill-posed in the sense of Hadamard, but this is not linked to properties of state equations. In fact, also if the corresponding direct problem is well-posed, the corresponding inverse problem can be ill-posed.

The inverse problem 1.8 is called *well-posed (in the sense of Hadamard)*, or equivalently *properly posed*, if the solution  $u \in U$  have the following properties:

- $\exists t^{-1} : T \rightarrow U$  the inverse operator of direct one  $t$  (*existence*);
- $\forall t_d \in T, \exists! u \in U$  such that  $u = t^{-1}(t_d)$  (*uniqueness*);
- the inverse function  $t^{-1}$  is continuous (*stability*).

Otherwise, the problem 1.8 is called *ill-posed*, or equivalently *improperly posed* [24].

We remark that the existence of a solution for the inverse problem, by a mathematical point of view, can be enforced by enriching the solution space  $T$ . The requirement of stability condition is more restrictive. In fact, if an inverse problem lacks stability, then computing its solution is a difficult task, because any measurement or numerical approximation is polluted by unavoidable errors. The direct inversion is not a good strategy to reconstruct  $u$ , due to the amplification of data's noise caused by ill-posedness. Thus, we must search a numerical solution. Moreover, we can employ appropriate numerical methods which stabilize the problem, called *regularization methods*.

In this Thesis we are interested in inverse problems for fluid flows, because there are many application.

## 1.4.2 Solution strategy

In general, to solve optimal flow control problems in CFD, there are two different strategies [23]:

- *optimize-then-discretize*, first an optimization procedure is defined, and then it is discretized;
- *discretize-then-optimize*, first inverse problem is discretized and then a discrete optimization problem is solved.

In this Thesis, we employ the optimize-then-discretize approach to solve inverse problems.

The most used regularization method for ill-posed problems is *Tikhonov regularization*. This technique is firstly introduced in a general framework in [73]. The Tikhonov approach consists in approximating a solution of (1.8) with the following optimization problem

$$u_\alpha = \arg \min_{u \in U} \|t(u) - t_d\|_T^2 + \alpha \|u - u_0\|_U^2,$$

where  $u_0 \in U$  is given by the prior information on the solution and  $\alpha > 0$  is the *regularization parameter*. As treated in [38], under some assumptions on the operator  $t$ , for a fixed value of  $\alpha$ , the solution  $u_\alpha$  is stable. In general, the convergence rate can be very small.

A detailed theoretical description of regularization techniques for linear problems could be found in [38]. Instead, for general nonlinear case, we refer to [40]. Usually, nonlinear ill-posed problems are numerically solved via iterative regularization methods using an optimize-then-discretize approach. Suppose that a certain iterative algorithm produce the following approximate solution's sequence

$$u^{k+1} = G_k(u^k, t_d), \quad k \in \mathbb{N}, \quad (1.9)$$

starting from given trial solution  $u^0$ . In general, for iterative methods the regularization parameter is the number of iterations  $k$ .

The previous algorithm belongs to the *regularizing family of operators* [3], with parameter of regularization  $k$ , if for every  $u^0 \in U$  the sequence (1.9) converges to an unknown solution  $u_T$  and the map  $G_k(u^k, t_d)$  is continuous at  $u \in U \setminus \{u_T\}$ . In other words, with a choice of stable approximations, there exists an index of iteration  $k = k(\sigma)$  such that  $\|u^{k(\sigma)} - u_T\|_U \rightarrow 0$  for  $\sigma \rightarrow 0$ . Under assumption on inverse problem, in [40, 3, 37], the authors prove that the above statements are satisfied, in particular, for *steepest descent algorithm*

$$u^{k+1} = u^k + \tau_k J'(u^k), \quad d^k = -J'(u^k)$$

and the *conjugate gradient* one

$$u^{k+1} = u^k + \tau_k d^k, \quad d^k = -J'(u^k) + \beta_k d^{k-1}, \quad \beta_0 = 0, \quad \beta_k = \frac{\|J'(u^k)\|_U^2}{\|J'(u^{k-1})\|_U^2},$$

where  $\tau_k$  is the *acceleration parameter*,  $\beta_k$  the *conjugate parameter* and  $d_k$  the *descendent direction*. Here  $J'(u^k)$  is the gradient of functional  $J(u^k) = \|t(u^k) - t_d\|_T^2$ . We remark that the gradient of functional cost can depend on the adjoint variable. For this reason, at each iteration we have to solve, not only direct problem but also, adjoint problem to compute  $J'(u^k)$  for descendent direction  $d^k$  and update the solution. It is possible to construct the *regularization algorithm* from these iterative procedures.

For more detail about the choice of index iteration  $k(\sigma)$  and proof of the convergence property for the steepest descent method, we refer to [40]. The theoretical results for nonlinear problem employing the steepest descent method are proof under assumption on the coefficients  $\tau_k$  and  $\beta_k$  d. Instead, employing conjugate gradient-type methods for nonlinear ill-posed problems, unfortunately, they don't satisfy these particular conditions. Then convergence results don't exist for these methods when applied to non-linear ill-posed problems. For more details on these iterative methods to solve ill-posed problems, we refer to [25, 18].

A very large set of classical regularization methods are based on *Landweber iteration*, which usually converges slowly (details could be found in [38]). In literature, there are developed regularization methods based on faster *Newton-type algorithms*, such as the *Levenberg-Marquardt regularization procedure* [40]. As explained in [40], discretization schemes are also regularization solution strategies for ill-posed problems.

## 1.5 Case study: Learning viscosity in lid-driven cavity flow

Inverse problems in fluid dynamics are widespread with application in aeronautics, geoscience, meteorology and mechanical engineering, as already mentioned.

One example of an inverse problem is to estimate a spatially-varying viscosity field from observed data. This example has already been covered in [12] where the authors developed an approach for solving inverse problems in the steady-state Navier-Stokes equations by combining deep neural networks and numerical PDE schemes.

In this section, we construct the KKT equations for a case where the direct problem is given by the Navier-Stokes equations. We develop the adjoint equations thanks Lagrangian formulation method with different observed data. Moreover, we compute the derivative of cost functional respect to the control not only for optimality condition but also for numerical simulation with gradient-type methods. Finally, we introduce sensitivity problem useful for following chapters to solve numerically the inverse problem.

### 1.5.1 Direct problem

In this subsection, we describe the direct problem involving incompressible steady-state flow in a two-dimensional square domain  $\Omega \subset \mathbb{R}^2$ . The geometry is shown in Figure 1.1, in which all the boundaries of the domain are walls. The top wall moves in the  $x$ -direction with constant velocity, while, the other boundaries are fixed.

The governing equations, considered in this thesis, are the steady-state Navier-Stokes equations for incompressible flows in  $\Omega$ . We write these state-constraint in strong form as below:

$$\begin{cases} -\nabla \cdot (\nu \nabla \mathbf{v}) + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p = \mathbf{g}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{v} = 0, & \text{in } \Omega, \end{cases} \quad (1.10)$$

where  $\mathbf{v}$  is velocity of the fluid,  $p$  the normalized pressure,  $\nu(\mathbf{x})$  the spatially-varying kinematic viscosity field, and  $\mathbf{g}$  the vector of body accelerations.

We impose a known velocity  $\mathbf{v}_N$  in Dirichlet sense at the top wall  $\Gamma$ . Instead, in the other walls  $\partial\Omega \setminus \Gamma$ , we assume no-slip conditions. Then, the boundary conditions can be summarized as:

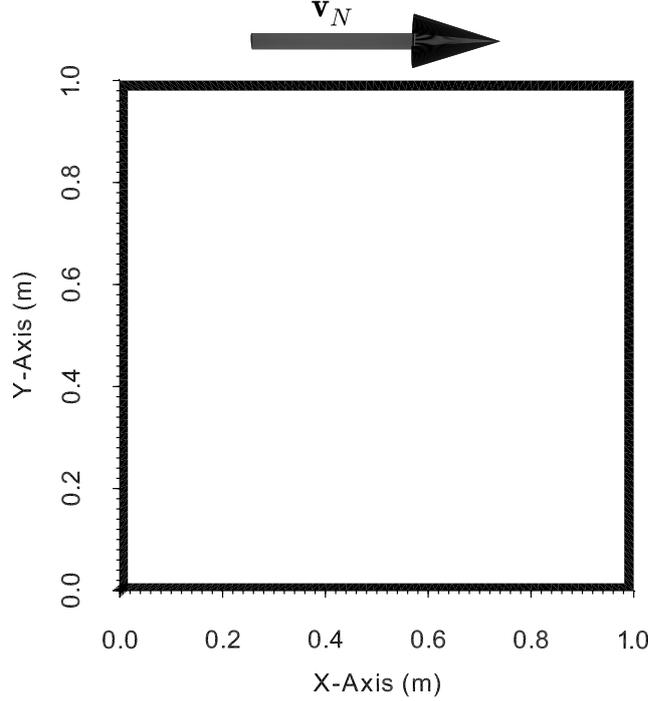
$$\begin{cases} \mathbf{v} = \mathbf{v}_N, & \text{on } \Gamma, \\ \mathbf{v} = \mathbf{0}, & \text{on } \partial\Omega \setminus \Gamma. \end{cases} \quad (1.11)$$

We can observe from the boundary conditions (1.11), that

$$\mathbf{v} \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega,$$

where  $\mathbf{n}$  is the outward unit normal vector to  $\partial\Omega$ . Then, the prescribed Dirichlet data is compatible with the incompressibility constraint. Indeed,

$$\int_{\Omega} \nabla \cdot \mathbf{v} \, d\Omega = \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n} \, d\gamma = 0 \quad (\text{compatibility constraint}).$$



**Figure 1.1:** Geometry of the lid-driven cavity.

From the "Functional Analysis" point of view (see [58]), the direct problem is a semi-linear elliptic problem. In particular,  $\Omega \in \mathbb{R}^2$  is a bounded Lipschitz domain. Moreover, for each term of equations, we can suppose:

- $\mathbf{v} \in [H^1(\Omega)]^2 \Rightarrow \Delta \mathbf{v}, (\mathbf{v} \cdot \nabla) \mathbf{v} \in [H^{-1}(\Omega)]^2$ ;
- $p \in L_0^2(\Omega) = \{ p \in L^2(\Omega) : \int_{\Omega} p \, d\Omega = 0 \}$ ; in fact, when only boundary conditions of Dirichlet type are imposed, the pressure appears merely in terms of its gradient; if  $(\mathbf{v}, p)$  is a solution, for any possible constant  $c$ , the couple  $(\mathbf{v}, p + c)$  is a solution too, since  $\nabla(p + c) = \nabla p$ ; then, to avoid such indeterminacy, one can require the pressure with null average, i.e.,  $\int_{\Omega} p \, d\Omega = 0$ .
- $\mathbf{g} \in [H^{-1}(\Omega)]^2$ ;
- $\mathbf{v}_N \chi_{\Gamma} \in [H^{1/2}(\partial\Omega)]^2$ , where  $\chi_{\Gamma}$  is the indicator function on boundary with non-homogeneous Dirichlet condition, and  $\int_{\partial\Omega} \mathbf{v}_N \chi_{\Gamma} \cdot \mathbf{n} \, d\gamma = 0$  (compatibility constraint)  $\Rightarrow \exists \mathbf{V} \in [H^1(\Omega)]^2 : \mathbf{V}|_{\partial\Omega} = \mathbf{v}_N \chi_{\Gamma}$  with  $\nabla \cdot \mathbf{V} = 0$  in  $\Omega$ .

From the assumption on the boundary condition, we can write  $\mathbf{v} = \hat{\mathbf{v}} + \mathbf{V}$  with  $\hat{\mathbf{v}}|_{\partial\Omega} = \mathbf{0}$ . Then, we can reformulate the direct problem 1.3.

**Problem 1.3** Find  $(\hat{\mathbf{v}}, p) \in [H_0^1(\Omega)]^2 \times L_0^2(\Omega)$  such that

$$\begin{cases} -\nabla \cdot (\nu \nabla \hat{\mathbf{v}}) + (\hat{\mathbf{v}} \cdot \nabla) \hat{\mathbf{v}} + (\mathbf{V} \cdot \nabla) \hat{\mathbf{v}} + (\hat{\mathbf{v}} \cdot \nabla) \mathbf{V} + \nabla p = \hat{\mathbf{f}}, & \text{in } \Omega, \\ \nabla \cdot \hat{\mathbf{v}} = 0, & \text{in } \Omega, \end{cases} \quad (1.12)$$

where,

$$\hat{\mathbf{f}} = -\nabla \cdot (\nu \nabla \mathbf{V}) + (\mathbf{V} \cdot \nabla) \mathbf{V} + \mathbf{g} \in [H^{-1}(\Omega)]^2.$$

Now, for computation of adjoint equations and gradient of cost functional with respect to the viscosity, in the next subsection, we recall the weakly form of direct problem. First of all, we summarize state and control space. Velocity  $\hat{\mathbf{v}}$ , pressure  $p$  and viscosity  $\nu$  are considered in the following solution spaces

$$V = [H_0^1(\Omega)]^2, \quad P = L_0^2(\Omega), \quad U = L^\infty(\Omega), \quad (1.13)$$

respectively. Then, we define the tri-linear form  $a : U \times V \times V \rightarrow \mathbb{R}$  and the bilinear form  $b : P \times V \rightarrow \mathbb{R}$  are defined as:

$$a(\nu, \mathbf{v}, \mathbf{w}) = \int_{\Omega} \nu \nabla \mathbf{v} : \nabla \mathbf{w} \, d\Omega, \quad b(q, \mathbf{v}) = - \int_{\Omega} q (\nabla \cdot \mathbf{v}) \, d\Omega,$$

and the non-linear convection term  $c : V \times V \times V \rightarrow \mathbb{R}$  is defined as follow:

$$c(\mathbf{v}, \mathbf{v}, \mathbf{w}) = \int_{\Omega} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, d\Omega.$$

Finally, we can write the weak formulation of state constraints (1.12) given by Problem 1.4.

**Problem 1.4** Find  $(\hat{\mathbf{v}}, p) \in V \times P$  such that

$$\begin{cases} a(\nu, \hat{\mathbf{v}}, \mathbf{w}) + b(p, \mathbf{w}) + c(\hat{\mathbf{v}}, \hat{\mathbf{v}}, \mathbf{w}) + c(\mathbf{v}, \hat{\mathbf{v}}, \mathbf{w}) + c(\hat{\mathbf{v}}, \mathbf{v}, \mathbf{w}) = \langle \mathbf{g}, \mathbf{w} \rangle, & \forall \mathbf{w} \in V, \\ b(q, \hat{\mathbf{v}}) = 0, & \forall q \in P. \end{cases} \quad (1.14)$$

The weak formulation 1.4 is equivalent to the following more compact one.

**Problem 1.5** Find  $(\mathbf{v}, p) \in [H^1(\Omega)]^2 \times P$

$$\begin{cases} a(\nu, \mathbf{v}, \mathbf{w}) + b(p, \mathbf{w}) + c(\mathbf{v}, \mathbf{v}, \mathbf{w}) = \langle \mathbf{g}, \mathbf{w} \rangle, & \forall \mathbf{w} \in V, \\ b(q, \mathbf{v}) = 0, & \forall q \in P, \\ \langle \mathbf{v} - \mathbf{v}_{N\chi\Gamma}, \mathbf{h} \rangle_{\partial\Omega} = 0, & \forall \mathbf{h} \in [H^{-1/2}(\partial\Omega)]^2. \end{cases} \quad (1.15)$$

In our case of study, we want to find the viscosity parameter in direct problem from different settings of observed data.

## 1.5.2 Inverse problem

In this subsection, we summarize the our inverse problem test case. We want to estimate a spatially-varying viscosity field  $\nu(\mathbf{x})$  from observed data in  $\Omega$ . In optimal control framework, we consider the following constrained optimization problem 1.6.

**Problem 1.6** Find optimal solution  $(\mathbf{v}, p, \nu) \in V \times P \times U$  such that

$$\min_{\mathbf{v}, p, \nu} J(\mathbf{v}, p, \nu),$$

subject to the state equations (1.10) and the boundary conditions (1.11).

We introduce the quadratic objective functional  $J$ . In our problem formulation, this cost functional depends on the observed data. If we observe only the velocity field, the functional is defined as below:

$$J_1(\mathbf{v}, p, \nu) = \frac{1}{2} \|\mathbf{v} - \mathbf{v}_d\|_Q^2, \quad (1.16)$$

which describes the aim of the problem, which is to match  $\mathbf{v}$  with the observation velocity  $\mathbf{v}_d \in Q \supset V$ . Instead, if we observe only the pressure field  $p_d \in P$ , we have the following cost function,

$$J_2(\mathbf{v}, p, \nu) = \frac{1}{2} \|p - p_d\|_P^2. \quad (1.17)$$

We can also observe both the velocity and the pressure fields. Then, the cost function becomes,

$$J_3(\mathbf{v}, p, \nu) = \frac{1}{2} \|\mathbf{v} - \mathbf{v}_d\|_Q^2 + \frac{1}{2} \|p - p_d\|_P^2. \quad (1.18)$$

We assume equivalence between state and adjoint spaces, that is,  $S = (V \times P) \equiv Z = (Z_{\mathbf{v}}, Z_p)$ . We define the Lagrangian by definition (1.4), where,  $\langle \mathbf{z}, F(\mathbf{s}, \mathbf{u}) \rangle_{Z, S^*}$  denotes the residual of state constraints in the weak formulation (1.15). As already mentioned, the control  $\nu$  is considered in the following functional space  $U = L^\infty(\Omega)$ . Moreover, we consider the Hilbert space  $Q = [L^2(\Omega)]^2$  for the desired velocity.

The objective function  $J(\mathbf{s}, \mathbf{u}) = J((\mathbf{v}, p), \nu)$  can be re-written, in compact form, as:

$$J_1(\mathbf{s}, \mathbf{u}) = \frac{1}{2} m(\mathbf{v} - \mathbf{v}_d, \mathbf{v} - \mathbf{v}_d), \quad J_2(\mathbf{s}, \mathbf{u}) = \frac{1}{2} n(p - p_d, p - p_d), \quad J_3(\mathbf{s}, \mathbf{u}) = J_1(\mathbf{s}, \mathbf{u}) + J_2(\mathbf{s}, \mathbf{u}),$$

where,  $m : V \times V \rightarrow \mathbb{R}$  and  $n : P \times P \rightarrow \mathbb{R}$ , defined as below:

$$m(\mathbf{v}, \mathbf{w}) = (\mathbf{v}, \mathbf{w})_Q, \quad n(p, q) = (p, q)_P.$$

Then, we define the Lagrangian,  $L(\mathbf{s}, \mathbf{u}, \mathbf{z}) = L((\mathbf{v}, p), \nu, (\mathbf{w}, q))$ , as below:

$$L(\mathbf{s}, \mathbf{u}, \mathbf{z}) = J(\mathbf{v}, p, \nu) + a(\nu, \mathbf{v}, \mathbf{w}) + b(p, \mathbf{w}) + c(\mathbf{v}, \mathbf{v}, \mathbf{w}) - \langle \mathbf{g}, \mathbf{w} \rangle + b(q, \mathbf{v}).$$

To compute the adjoint problem, we calculate the Fréchet derivative with respect to the state variables of the Lagrangian, denoted by  $\nabla_{\mathbf{s}} L(\mathbf{s}, \mathbf{u}, \mathbf{z})[\boldsymbol{\xi}]$ . At first, we consider the variation of the Lagrangian perturbing the state variable  $\mathbf{s} \rightarrow \mathbf{s} + \delta \mathbf{s}$ ,

$$\begin{aligned} L((\mathbf{v} + \delta \mathbf{v}, p + \delta p), \nu, (\mathbf{w}, q)) - L((\mathbf{v}, p), \nu, (\mathbf{w}, q)) &= J(\mathbf{v} + \delta \mathbf{v}, p + \delta p, \nu) - J(\mathbf{v}, p, \nu) + \\ &+ a(\nu, \mathbf{w}, \delta \mathbf{v}) + c(\delta \mathbf{v}, \mathbf{v}, \mathbf{w}) + c(\delta \mathbf{v}, \delta \mathbf{v}, \mathbf{w}) + c(\mathbf{v}, \delta \mathbf{v}, \mathbf{w}) + b(q, \delta \mathbf{v}) + b(\delta p, \mathbf{w}). \end{aligned}$$

Then, the Fréchet derivative of  $L$  is obtained by neglecting the second order terms. The derivative computed depends on the observed data. The different Fréchet derivative is given by

$$\begin{aligned}\nabla_{\mathbf{s}}L_1(\mathbf{s}, \mathbf{u}, \mathbf{z})[\boldsymbol{\xi}] &= \begin{pmatrix} m(\mathbf{v} - \mathbf{v}_d, \boldsymbol{\xi}_{\mathbf{v}}) + a(\nu, \mathbf{w}, \boldsymbol{\xi}_{\mathbf{v}}) + c(\boldsymbol{\xi}_{\mathbf{v}}, \mathbf{v}, \mathbf{w}) + c(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}, \mathbf{w}) + b(q, \boldsymbol{\xi}_{\mathbf{v}}) \\ b(\xi_p, \mathbf{w}) \end{pmatrix}, \\ \nabla_{\mathbf{s}}L_2(\mathbf{s}, \mathbf{u}, \mathbf{z})[\boldsymbol{\xi}] &= \begin{pmatrix} a(\nu, \mathbf{w}, \boldsymbol{\xi}_{\mathbf{v}}) + c(\boldsymbol{\xi}_{\mathbf{v}}, \mathbf{v}, \mathbf{w}) + c(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}, \mathbf{w}) + b(q, \boldsymbol{\xi}_{\mathbf{v}}) \\ b(\xi_p, \mathbf{w}) + n(p - p_d, \xi_p) \end{pmatrix}, \\ \nabla_{\mathbf{s}}L_3(\mathbf{s}, \mathbf{u}, \mathbf{z})[\boldsymbol{\xi}] &= \begin{pmatrix} m(\mathbf{v} - \mathbf{v}_d, \boldsymbol{\xi}_{\mathbf{v}}) + a(\nu, \mathbf{w}, \boldsymbol{\xi}_{\mathbf{v}}) + c(\boldsymbol{\xi}_{\mathbf{v}}, \mathbf{v}, \mathbf{w}) + c(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}, \mathbf{w}) + b(q, \boldsymbol{\xi}_{\mathbf{v}}) \\ b(\xi_p, \mathbf{w}) + n(p - p_d, \xi_p) \end{pmatrix},\end{aligned}$$

where  $\boldsymbol{\xi} = (\boldsymbol{\xi}_{\mathbf{v}}, \xi_p) \in S$ .

Now, we can state the *adjoint problem in weak form* as  $\nabla_{\mathbf{s}}L(\mathbf{s}, \mathbf{u}, \mathbf{z})[\boldsymbol{\xi}] = \mathbf{0}$ , for all  $\boldsymbol{\xi} \in S$ . More explicitly, with only velocity data, we obtain the following adjoint problem.

**Problem 1.7** Find  $\mathbf{z} = (\mathbf{w}, q) \in S$  such that

$$\begin{cases} m(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}) + a(\nu, \mathbf{w}, \boldsymbol{\xi}_{\mathbf{v}}) + c(\boldsymbol{\xi}_{\mathbf{v}}, \mathbf{v}, \mathbf{w}) + c(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}, \mathbf{w}) + b(q, \boldsymbol{\xi}_{\mathbf{v}}) = m(\mathbf{v}_d, \boldsymbol{\xi}_{\mathbf{v}}), & \forall \boldsymbol{\xi}_{\mathbf{v}} \in V, \\ b(\xi_p, \mathbf{w}) = 0, & \forall \xi_p \in P. \end{cases}$$

To compute the strong form of these equations, we integrate by parts each term applying the state boundary condition (1.11) and the state constraints (1.10). Moreover, the non-linear convective terms can be reformulated as:

$$\begin{aligned}c(\boldsymbol{\xi}_{\mathbf{v}}, \mathbf{v}, \mathbf{w}) &= \int_{\Omega} (\nabla \mathbf{v} \cdot \mathbf{w}) \cdot \boldsymbol{\xi}_{\mathbf{v}} d\Omega, \\ c(\mathbf{v}, \boldsymbol{\xi}_{\mathbf{v}}, \mathbf{w}) &= \int_{\partial\Omega} ((\mathbf{v} \cdot \mathbf{n})\mathbf{w}) \cdot \boldsymbol{\xi}_{\mathbf{v}} d\gamma - \int_{\Omega} (\nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla)\mathbf{w} \cdot \boldsymbol{\xi}_{\mathbf{v}} d\Omega.\end{aligned}$$

Then, we can rewritten the *adjoint problem in strong form*. For observed data only on the velocity, we have the following strong formulation.

**Problem 1.8** Find  $\mathbf{z} = (\mathbf{w}, q) \in S$  such that

$$\begin{cases} -\nabla \cdot (\nu \nabla \mathbf{w}) + \nabla \mathbf{v} \cdot \mathbf{w} - (\mathbf{v} \cdot \nabla)\mathbf{w} + \nabla q = \mathbf{v}_d - \mathbf{v}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{w} = 0, & \text{in } \Omega, \\ \mathbf{w} = \mathbf{0}, & \text{on } \partial\Omega. \end{cases} \quad (1.19)$$

Instead, in the case of observed data on the pressure, we obtain the following strong form.

**Problem 1.9** Find  $\mathbf{z} = (\mathbf{w}, q) \in S$  such that

$$\begin{cases} -\nabla \cdot (\nu \nabla \mathbf{w}) + \nabla \mathbf{v} \cdot \mathbf{w} - (\mathbf{v} \cdot \nabla)\mathbf{w} + \nabla q = \mathbf{0}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{w} = p - p_d, & \text{in } \Omega, \\ \mathbf{w} = \mathbf{0}, & \text{on } \partial\Omega. \end{cases} \quad (1.20)$$

Otherwise, if we observe both the velocity and the pressure fields, we have the following formulation.

**Problem 1.10** Find  $\mathbf{z} = (\mathbf{w}, q) \in S$  such that

$$\begin{cases} -\nabla \cdot (\nu \nabla \mathbf{w}) + \nabla \mathbf{v} \cdot \mathbf{w} - (\mathbf{v} \cdot \nabla) \mathbf{w} + \nabla q = \mathbf{v}_d - \mathbf{v}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{w} = p - p_d, & \text{in } \Omega, \\ \mathbf{w} = \mathbf{0}, & \text{on } \partial\Omega. \end{cases} \quad (1.21)$$

Now, we want to compute Fréchet derivative with respect to the control variable  $\mathbf{u} = \nu$  of Lagrangian  $L(\mathbf{s}, \mathbf{u}, \mathbf{z})$ , denoted by  $\nabla_{\mathbf{u}} L(\mathbf{s}, \mathbf{u}, \mathbf{z})[\kappa]$ . We derive it by perturbing the viscosity field  $\nu \rightarrow \nu + \delta\nu$ . Then, we have a variation on the Lagrangian

$$L((\mathbf{v}, p), \nu + \delta\nu, (\mathbf{w}, q)) - L((\mathbf{v}, p), \nu, (\mathbf{w}, q)) = a(\delta\nu, \mathbf{v}, \mathbf{w}),$$

Thus, the gradient of Lagrangian with respect to is

$$\nabla_{\mathbf{u}} L(\mathbf{s}, \mathbf{u}, \mathbf{z})[\kappa] = a(\kappa, \mathbf{v}, \mathbf{w}) = \int_{\Omega} \kappa \nabla \mathbf{v} : \nabla \mathbf{w} \, d\Omega, \quad \forall \kappa \in U.$$

It turns out to be very useful to express this Fréchet derivative of the Lagrangian in terms of the gradient of the cost functional  $J(\nu) = J(\mathbf{v}(\nu), p(\nu))$  with respect to.  $\mathbf{u} = \nu$ . Indeed, thanks the above representation, the derivative of the cost functional  $J(\nu)$  is

$$J'(\nu) = \nabla \mathbf{v}(\nu) : \nabla \mathbf{w}(\nu). \quad (1.22)$$

Now, we introduce the *sensitivity problem* related to the direct problem (1.10)-(1.11), because its is useful in gradient-type optimization method introduced in the next chapter. We derive it by perturbing in (1.10)-(1.11) the viscosity field  $\nu \rightarrow \nu + \delta\nu$ . This perturbation cause a variation in the velocity field,  $\mathbf{v} \rightarrow \mathbf{v} + \delta\mathbf{v}$ , and in the pressure field,  $p \rightarrow p + \delta p$ . Thus, subtracting direct problem from the obtained perturbed problem, we have

$$\begin{cases} -\nabla \cdot (\nu \nabla \delta\mathbf{v}) - \nabla \cdot (\delta\nu \nabla (\mathbf{v} + \delta\mathbf{v})) + (\mathbf{v} \cdot \nabla) \delta\mathbf{v} + (\delta\mathbf{v} \cdot \nabla) (\mathbf{v} + \delta\mathbf{v}) + \nabla \delta p = \mathbf{0}, & \text{in } \Omega, \\ \nabla \cdot \delta\mathbf{v} = 0, & \text{in } \Omega, \\ \delta\mathbf{v} = \mathbf{0}, & \text{on } \partial\Omega. \end{cases}$$

Then, the *sensitivity problem* is obtained by neglecting the second order terms.

**Problem 1.11** Find  $(\delta\mathbf{v}, \delta p)$  such that

$$\begin{cases} -\nabla \cdot (\nu \nabla \delta\mathbf{v}) - \nabla \cdot (\delta\nu \nabla \mathbf{v}) + (\mathbf{v} \cdot \nabla) \delta\mathbf{v} + (\delta\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla \delta p = \mathbf{0}, & \text{in } \Omega, \\ \nabla \cdot \delta\mathbf{v} = 0, & \text{in } \Omega, \end{cases} \quad (1.23)$$

with boundary conditions

$$\delta\mathbf{v} = \mathbf{0}, \quad \text{on } \partial\Omega. \quad (1.24)$$

Then, we can observe that  $\mathbf{v}(\nu + \delta\nu) = \mathbf{v}(\nu) + \delta\mathbf{v}(\nu, \mathbf{v}, \delta\nu)$ . Moreover,  $\delta\mathbf{v}$  is linear with respect to the control variation  $\delta\nu$ , therefore  $\delta\mathbf{v}(\nu, \mathbf{v}, \delta\nu_1 + \delta\nu_2) = \delta\mathbf{v}(\nu, \mathbf{v}, \delta\nu_1) + \delta\mathbf{v}(\nu, \mathbf{v}, \delta\nu_2)$ . With the same argument  $\delta p$  is linear with respect to the control variation  $\delta\nu$ .

However, the direct solution of such coupled systems is very difficult and require the implementation of high-fidelity numerical approximation schemes. Thus, in the upcoming

chapter, we solve the optimal flow control problem with *optimize-then-discretize* strategy, hence, dividing the solution process into two steps: *optimization*, that is, deriving coupled KKT optimality system, and *discretization*, that is, implementing the numerical methods to solve the KKT system. At full-order level, we adopt an iterative procedure that solve the KKT system in a decoupled way. In this work, we discuss its solution by *Non-linear Projected Conjugate Gradient Method*.

## Chapter 2

# Full Order Model Formulation

In this Chapter, we present the numerical method used to solve inverse problems, in fluid dynamics framework. In particular, we show the implementation for our test case, as introduced before in Section 1.5. We develop the Full Order Model with finite volume discretization and a gradient-type optimization solver. We recall the finite volume approximation scheme for Navier-Stokes Equations to simulate the direct problem. A modified version of algorithm used to solve the direct problem is employed also to solve the adjoint and sensitivity ones. In Numerical Analysis, the classical Conjugate Gradient Method (CGM) [49, 58] is employed to solve linear system. In this work, we use a non-linear version of this CG method to solve inverse problem. In our case, this method have also a *projection step*. This step is a posteriori correction due to constraints on control.

### 2.1 Non-linear Projected Conjugate Gradient Method

This Section addresses the solution of KKT system using the segregated approach. We employ a conjugate gradient-type method. With this method, we solve in a decoupled way the state and adjoint problems using the finite volume method. Then, we update the control variable at each iteration with descendent direction linked to the functional cost.

Now, thanks to the gradient computed in (1.22), the gradient-type method can be defined. In particular, we consider the following iterative procedure for the estimation of the function  $\nu$  that minimizes the functional  $J(\nu)$ . For a deeper analysis, see [53, 58]. Given an initial guess  $\nu^0 \in U$ , for  $n > 0$  a new viscosity field is computed as:

$$\nu^{n+1} = \nu^n + \tau^n d^n, \quad n = 0, 1, 2, \dots \quad (2.1)$$

where  $n$  is the iteration number,  $\tau^n$  the step length and  $d^n$  the descendent direction.

For conjugate gradient method, the descendent direction  $d^n$ , also called conjugate gradient direction, is given by

$$d^0 = -J'(\nu^0), \quad d^n = -J'(\nu^n) + \beta^n d^{n-1} \quad n \geq 1, \quad (2.2)$$

where  $\beta^n$  is the conjugate coefficient. It can be calculated, for example, using Fletcher-Reeves expression (see [15]) as follows:

$$\beta^n = \frac{\|J'(\nu^n)\|_U^2}{\|J'(\nu^{n-1})\|_U^2}. \quad (2.3)$$

Once  $d^n$  is computed, the parameter  $\tau^n$  should be chosen in such a way to solve the following scalar minimization sub-problem

$$\tau^n = \arg \min_{\tau \in \mathbb{R}} J(\nu^n + \tau d^n). \quad (2.4)$$

The optimal step length  $\tau^n$  depends on observed data. For example, with only observed velocity field, thanks to the *sensitivity problem* (1.23)-(1.24), we can re-write the cost function at iteration  $n$  as

$$\begin{aligned} J_1(\nu^n + \tau d^n) &= \frac{1}{2} \|\mathbf{v}(\nu^n + \tau d^n) - \mathbf{v}_d\|_Q^2 = \frac{1}{2} \|\mathbf{v}(\nu^n) + \tau \delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n) - \mathbf{v}_d\|_Q^2 = \\ &= \frac{1}{2} \|\mathbf{v}(\nu^n) - \mathbf{v}_d\|_Q^2 + \frac{\tau^2}{2} \|\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n)\|_Q^2 + \tau (\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n), \mathbf{v}(\nu^n) - \mathbf{v}_d)_Q. \end{aligned} \quad (2.5)$$

Then, differentiating with respect to  $\tau$ , we obtain the critical point equation

$$\frac{dJ(\nu^n + \tau d^n)}{d\tau} = 0.$$

From (2.5), we write critical point equation in explicit form as

$$\tau \|\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n)\|_Q^2 + (\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n), \mathbf{v}(\nu^n) - \mathbf{v}_d)_Q = 0. \quad (2.6)$$

Finally, we calculate the optimal step size for the observed data on velocity as below

$$\tau_1^n = \frac{(\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n), \mathbf{v}_d - \mathbf{v}(\nu^n))_Q}{\|\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n)\|_Q^2}. \quad (2.7)$$

In the other cases, with different observed data, we have the following expressions:

$$\tau_2^n = \frac{(\delta p(\nu^n, \mathbf{v}(\nu^n), d^n), p_d - p(\nu^n))_P}{\|\delta p(\nu^n, \mathbf{v}(\nu^n), d^n)\|_P^2}, \quad (2.8)$$

$$\tau_3^n = \frac{(\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n), \mathbf{v}_d - \mathbf{v}(\nu^n))_Q + (\delta p(\nu^n, \mathbf{v}(\nu^n), d^n), p_d - p(\nu^n))_P}{\|\delta \mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n)\|_Q^2 + \|\delta p(\nu^n, \mathbf{v}(\nu^n), d^n)\|_P^2}, \quad (2.9)$$

for pressure only and pressure-velocity combination respectively.

We recall that, to use this iterative procedure, we have to compute at each iteration the derivative  $J'(\nu)$  by (1.22) which depends on state and adjoint velocity. Thus, we must solve at each iteration the direct problem (1.10)-(1.11) and the adjoint problem to compute it.

To avoid turbulence effects on our system, we suppose that the viscosity has side constraint, i.e.  $\forall \mathbf{x} \in \Omega \nu(\mathbf{x}) \in [\varepsilon, +\infty]$ . Due to the presence of this inequality constraint  $\nu \geq \varepsilon$ , a *projection*  $\pi_{[\varepsilon, +\infty)}$  onto the set of admissible value is needed. At each iteration, the projection step is given by the formula below,

$$\nu^{n+1}(\mathbf{x}) = \pi_{[\varepsilon, +\infty)}(\nu^n + \tau^n d^n)(\mathbf{x}) = \begin{cases} \varepsilon, & \text{if } \nu^n(\mathbf{x}) + \tau^n d^n(\mathbf{x}) \leq \varepsilon, \\ \nu^n(\mathbf{x}) + \tau^n d^n(\mathbf{x}), & \text{otherwise,} \end{cases} \quad (2.10)$$

where  $\varepsilon > 0$  is a small constant value. For more details about the treatment of constraints on control variable using a *Projection Methods*, you can see [23, 27, 4].

For the particular case in which the velocity field is the only observed one, we can summarize the *Non-linear Projected Conjugate Gradient Method* in Algorithm 1.

---

**Algorithm 1:** Non-linear Projected Conjugate Gradient Method.

---

**Input:** Choose  $\nu^0 \in U$ ,  $n_{max}$  and  $J_{tol}$ , and set  $n = 0$ .

- 1 **while**  $n \leq n_{max}$  **do**
- 2     Solve the *direct problem* (1.10)-(1.11) with  $\nu = \nu^n$  to store  $\mathbf{v}(\nu^n)$ ;
- 3     Compute the cost function  $J(\nu^n)$  by (1.16);
- 4     **if**  $J(\nu^n) < J_{tol}$  **then**
- 5         | Stop;
- 6     **end**
- 7     Solve the *adjoint problem* (1.19) with  $\nu = \nu^n$  and  $\mathbf{v} = \mathbf{v}(\nu^n)$  to store  $\mathbf{w}(\nu^n)$ ;
- 8     Compute the derivative  $J'(\nu^n)$  by (1.22);
- 9     **if**  $n > 0$  **then**
- 10         | Calculate the conjugate coefficient,  $\beta^n$ , by the use of (2.3);
- 11     **end**
- 12     Compute the search direction,  $d^n$ , by (2.2);
- 13     Solve the *sensitivity problem* (1.23)-(1.24) with  $\nu = \nu^n$ ,  $\delta\nu = d^n$  and  $\mathbf{v} = \mathbf{v}(\nu^n)$  to store  $\delta\mathbf{v}(\nu^n, \mathbf{v}(\nu^n), d^n)$ ;
- 14     Calculate the optimal step size,  $\tau^n$ , by evaluating (2.7);
- 15     Update the viscosity field  $\nu^n$  by (2.10);
- 16      $n = n + 1$ ;
- 17 **end**
- 18 **return**  $\nu^n$ .

---

With the other observed data, we have the same algorithm except for different adjoint equations and formula of optimal step length, as previous described.

For simplicity, we suppose that the viscosity is constant. Then, the control space becomes  $U = [\varepsilon, +\infty)$ . For this assumption, the gradient (1.22) changes in the following way

$$J'(\nu) = \int_{\Omega} \nabla \mathbf{v}(\nu) : \nabla \mathbf{w}(\nu) d\Omega, \quad (2.11)$$

consequentially the conjugate coefficient (2.3) gets modified in the following way:

$$\beta^n = \frac{|J'(\nu^n)|^2}{|J'(\nu^{n-1})|^2}. \quad (2.12)$$

We can solve the KKT-system with a segregate method using the Algorithm 1 except for the previous changes on the gradient (2.11) and the expression for beta (2.12), caused by the assumptions on the control space  $U$ .

## 2.2 Finite Volume Discretization

In this section, we will introduce the *discretization step*, or in other words, *Finite Volume (FV)* approximations of the state, adjoint and sensitivity equations. We solve the inverse problem in fluid-dynamics by applying the iterative optimization procedure, discussed in the previous Section, which involves these equations.

At first, we study the discretization of state and adjoint equations, together with their boundary conditions. At full order level, we use a *Finite Volume Method (FVM)*. Here, we recall, briefly, the finite volume approximation. For more details on FVM, we refer to [51, 13, 64].

The computational domain  $\Omega$  is tessellated to obtain a mesh grid  $\mathcal{T}(\Omega) = \{\Omega_e\}_{e=1}^{N_{FV}}$ . We suppose that, this mesh is composed by a set of convex and non overlapping polygons, called *finite volume (FV)*, such that  $\Omega = \cup_{e=1}^{N_{FV}} \Omega_e$  and  $\Omega_i \cap \Omega_j = \emptyset$  for  $i \neq j$ . Then, the solution is restricted to the finite dimensional space given by the space of functions, that are piecewise constant functions over each finite volume cell  $\Omega_e$ . The finite volume approximation of equations is derived starting directly from the integrated form of the equations.

We first discretize the state equations given by incompressible steady-state Navier–Stokes equations in (1.10). For sake of simplicity, we assume that no body forces are present, i.e.  $\mathbf{g} = \mathbf{0}$ . We write the state momentum balance equation and the state continuity equation in integral form, for each volume  $\Omega_e \in \mathcal{T}$ , as

$$\begin{cases} \int_{\Omega_e} (\mathbf{v} \cdot \nabla) \mathbf{v} dV - \int_{\Omega_e} \nabla \cdot \nu \nabla \mathbf{v} dV + \int_{\Omega_e} \nabla p dV & = 0, \\ \int_{\Omega_e} \nabla \cdot \mathbf{v} dV & = 0. \end{cases} \quad (2.13)$$

The discretization procedure of all terms in the momentum and continuity equations is explained in what follows.

The gradient terms and, in particular, the gradient of pressure, making use of the Gauss's gradient theorem, are discretised as:

$$\int_{\Omega_e} \nabla p dV = \int_{\partial\Omega_e} p d\mathbf{S} \approx \sum_f^{N_e^f} p_f \mathbf{S}_f,$$

where  $p_f$  is the pressure value at center of the face,  $\mathbf{S}_f$  is the area vector of the face  $f$  of the control volume  $\Omega_e$  and  $N_e^f$  is the number of faces related to the  $e$ -th cell  $\Omega_e$ .

The convective term is discretized, exploiting the incompressibility constraint and the divergence theorem, as follows:

$$\begin{aligned} \int_{\Omega_e} (\mathbf{v} \cdot \nabla) \mathbf{v} dV &= \int_{\Omega_e} \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) dV = \int_{\partial\Omega_e} (d\mathbf{S} \cdot (\mathbf{v} \otimes \mathbf{v})) \approx \\ &\approx \sum_f^{N_e^f} \mathbf{S}_f \cdot (\mathbf{v}_f \otimes \mathbf{v}_f) = \sum_f^{N_e^f} (\mathbf{S}_f \cdot \mathbf{v}_f) \mathbf{v}_f = \sum_f^{N_e^f} F_f \mathbf{v}_f, \end{aligned}$$

where  $\mathbf{v}_f$  indicates the velocity at the centre of the faces and  $F_f = \mathbf{S}_f \cdot \mathbf{v}_f$  represents the mass flux through each face of the control volume. Two considerations have to be underlined for this procedure. The first one is that velocity values are initially computed at the cell centres. Therefore, the values at the center of the faces have to be deduced from the ones calculated at the cell centres. Many different techniques are available to obtain it. The basic idea behind them all is that the face value is approximated by interpolating the values at the center of the cells. The second clarification is about fluxes. During an iterative process to solve the state equations, the mass flux  $F_f$  is computed using the previous converged velocity in the first iteration, that satisfies the continuity equation, to remove the non-linearity.

Making use of the Gauss's divergence theorem, the diffusion term is discretised as:

$$\int_{\Omega_e} \nabla \cdot \nu \nabla \mathbf{v} dV = \int_{\partial\Omega_e} (d\mathbf{S} \cdot \nu \nabla \mathbf{v}) \approx \nu \sum_f^{N_e^f} \mathbf{S}_f \cdot (\nabla \mathbf{v})_f,$$

where the term  $(\nabla \mathbf{v})_f$  indicates the gradient of the velocity field at the centre of each face. This is computed, starting from the cell's centre values of the neighbouring cells, using a finite difference scheme that includes a correction in the case of non-orthogonal tessellations. In case of orthogonal meshes (i.e. the face dividing two cells is orthogonal with respect to. the line connecting the two cell centers) the above laplacian term could be calculated employing the following approximation:

$$\mathbf{S}_f \cdot (\nabla \mathbf{v})_f \approx |\mathbf{S}_f| \frac{\mathbf{v}_e - \mathbf{v}_n^f}{|\mathbf{d}_{en}^f|},$$

where  $\mathbf{d}_{en}^f$  is the vector connecting the centers of cells  $e$  and  $n$ ,  $\mathbf{v}_e$  is the velocity at the center of the  $e$ -th cell  $\Omega_e$  and  $\mathbf{v}_n^f$  is the velocity evaluated at the center of the cell  $n$  divided from the cell  $e$  by the face  $f$ . When the mesh is not orthogonal, the last expression has to be corrected as follows:

$$\mathbf{S}_f \cdot (\nabla \mathbf{v})_f \approx |\Lambda_f| \frac{\mathbf{v}_e - \mathbf{v}_n^f}{|\mathbf{d}_{en}^f|} + \mathbf{k}_f \cdot (\nabla \mathbf{v})_f,$$

where  $\mathbf{S}_f$  has been decomposed into a parallel and an orthogonal component with respect to  $\mathbf{d}_{en}^f$ , namely  $\Lambda_f$  and  $\mathbf{k}_f$ , respectively. Then, the term  $(\nabla \mathbf{v})_f$  can be approximated by the use of an interpolation between  $(\nabla \mathbf{v})_e$  and  $(\nabla \mathbf{v})_n^f$  at the centers of the surrounding cells. There are different techniques to compute the vectors  $\Lambda_f$  and  $\mathbf{k}_f$ , such as *minimum*

correction approach, *orthogonal correction* approach and *over-relaxed* approach. More details these correction approaches can be found in [51].

Finally, the term originated from the divergence of velocity is discretized as:

$$\int_{\Omega_e} \nabla \cdot \mathbf{v} dV = \int_{\partial\Omega_e} (d\mathbf{S} \cdot \mathbf{v}) \approx \sum_f^{N_e^f} \mathbf{S}_f \cdot \mathbf{v}_f = \sum_f^{N_e^f} F_f.$$

The complete discretized state equations then reads

$$\begin{cases} \sum_f^{N_e^f} F_f \mathbf{v}_f + \sum_f^{N_e^f} p_f \mathbf{S}_f - \nu \sum_f^{N_e^f} \left( |\boldsymbol{\Lambda}_f| \frac{\mathbf{v}_e - \mathbf{v}_n^f}{|\mathbf{d}_{en}^f|} + \mathbf{k}_f \cdot (\nabla \mathbf{v})_f \right) = \mathbf{0}, & \forall e = 1, \dots, N_{FV}, \\ \sum_f^{N_e^f} F_f = 0, & \forall e = 1, \dots, N_{FV}. \end{cases} \quad (2.14)$$

By the use of a similar approach, we discretize the adjoint equations. Since the inverse problem with both pressure and velocity data is the combination of the two single problems (pressure only and velocity only), we report here just the discretization process for the combined case.

Starting from (1.21), we can obtain the integral form of these equations, as:

$$\begin{cases} \int_{\Omega_e} \nabla \mathbf{v} \cdot \mathbf{w} dV - \int_{\Omega_e} (\mathbf{v} \cdot \nabla) \mathbf{w} dV - \int_{\Omega_e} \nabla \cdot \nu \nabla \mathbf{w} dV + \int_{\Omega_e} \nabla q dV = \int_{\Omega_e} (\mathbf{v}_d - \mathbf{v}) dV, \\ \int_{\Omega_e} \nabla \cdot \mathbf{w} dV = \int_{\Omega_e} (p - p_d) dV. \end{cases} \quad (2.15)$$

Since these equations are similar to state equations (2.13), then, most of the terms in (2.15) have the same structure. Summarizing, the diffusive, convective, pressure gradient and velocity divergence terms are given by

$$\begin{aligned} \int_{\Omega_e} \nabla \cdot \nu \nabla \mathbf{w} dV &\approx \nu \sum_f^{N_e^f} \left( |\boldsymbol{\Lambda}_f| \frac{\mathbf{w}_e - \mathbf{w}_n^f}{|\mathbf{d}_{en}^f|} + \mathbf{k}_f \cdot (\nabla \mathbf{w})_f \right), \\ \int_{\Omega_e} (\mathbf{v} \cdot \nabla) \mathbf{w} dV &\approx \sum_f^{N_e^f} F_f \mathbf{w}_f, \\ \int_{\Omega_e} \nabla q dV &\approx \sum_f^{N_e^f} q_f \mathbf{S}_f, \\ \int_{\Omega_e} \nabla \cdot \mathbf{w} dV &\approx \sum_f^{N_e^f} \mathcal{F}_f, \end{aligned}$$

respectively. Where, the new unknowns have the same meaning of the corresponding terms into the discretized state equations. While, using the mean value theorem for the

integrals, the implicit and explicit source terms in (2.15) are approximated as follows:

$$\begin{aligned}\int_{\Omega_e} \nabla \mathbf{v} \cdot \mathbf{w} dV &\approx |\Omega_e| (\nabla \mathbf{v})_e \cdot \mathbf{w}_e, \\ \int_{\Omega_e} (\mathbf{v}_d - \mathbf{v}) dV &\approx |\Omega_e| ((\mathbf{v}_d)_e - \mathbf{v}_e), \\ \int_{\Omega_e} (p - p_d) dV &\approx |\Omega_e| (p_e - (p_d)_e),\end{aligned}$$

where the velocity gradient  $(\nabla \mathbf{v})_e$ , at the center of  $e$ -th cell  $\Omega_e$ , is pre-computed as the mean value on the volume. Thanks to the Gauss's theorem, it is approximated as:

$$(\nabla \mathbf{v})_e \approx \frac{1}{|\Omega_e|} \int_{\Omega_e} \nabla \mathbf{v} dV = \frac{1}{|\Omega_e|} \int_{\partial \Omega_e} (d\mathbf{S} \otimes \mathbf{v}) \approx \frac{1}{|\Omega_e|} \sum_f^{N_e^f} \mathbf{S}_f \otimes \mathbf{v}_f.$$

Then, the discretized adjoint equations for velocity and pressure observed data, is given by

$$\begin{cases} |\Omega_e| (\nabla \mathbf{v})_e \cdot \mathbf{w}_e - \sum_f^{N_e^f} F_f \mathbf{w}_f - \nu \sum_f^{N_e^f} \left( |\Lambda_f| \frac{\mathbf{w}_e - \mathbf{w}_f}{|\mathbf{d}_{en}^f|} + \mathbf{k}_f \cdot (\nabla \mathbf{w})_f \right) + \\ + \sum_f^{N_e^f} q_f \mathbf{S}_f = |\Omega_e| ((\mathbf{v}_d)_e - \mathbf{v}_e), \quad \forall e = 1, \dots, N_{FV}, \\ \sum_f^{N_e^f} \mathcal{F}_f = |\Omega_e| (p_e - (p_d)_e), \quad \forall e = 1, \dots, N_{FV}. \end{cases} \quad (2.16)$$

With the Non-linear Projected Conjugate Gradient Method for KKT system, given by Algorithm 1, we must solve the Sensitivity problem (1.23). This problem has the same integral operators with respect to the first discretized equations. We discretize this problem with the same procedure, as follows:

$$\begin{cases} \sum_f^{N_e^f} \delta F_f \mathbf{v}_f + \sum_f^{N_e^f} F_f \delta \mathbf{v}_f - (\nu + \delta \nu) \sum_f^{N_e^f} \left( |\Lambda_f| \frac{\delta \mathbf{v}_e - \delta \mathbf{v}_f}{|\mathbf{d}_{en}^f|} + \mathbf{k}_f \cdot (\nabla \delta \mathbf{v})_f \right) + \\ + \sum_f^{N_e^f} \delta p_f \mathbf{S}_f = \mathbf{0}, \quad \forall e = 1, \dots, N_{FV}, \\ \sum_f^{N_e^f} \delta F_f = 0, \quad \forall e = 1, \dots, N_{FV}, \end{cases} \quad (2.17)$$

where the unknowns terms are the same of the aforementioned discretized equations.

**Remark 2.1** *All the above discretized problems, i.e. (2.14)-(2.16)-(2.17), have the same saddle point structure written in matrix form as*

$$\begin{bmatrix} \mathbf{A}_v & \mathbf{B}_p \\ \mathbf{B}_v & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{S}_v \\ S_p \end{bmatrix}, \quad (2.18)$$

indicating by  $\mathbf{A}_v$  the coefficient matrix coming from momentum equation, depending on the particular problem, by  $\mathbf{B}_p$  the matrix containing the terms related to the gradient of pressure, by  $\mathbf{B}_v$  the matrix representing the incompressibility constraint operator, by  $\mathbf{S}_v$  and  $S_p$  the vectors of source terms of this equations.

The coupling between velocity and pressure, with a saddle point structure, is usually not easy to be solved using a coupled approach. For this reason we rely on a segregated approach. In this approach, the momentum equation is solved with a tentative pressure and later corrected exploiting the divergence free constraint. In this Thesis we use a *segregate pressure-based* strategy, to solve these saddle point problems. This solver recalled for state, adjoint and sensitivity equations, in the following Section. For more detail on this solver, we refer to [51].

## 2.3 Segregate Pressure-Based Solver in FVM

In this Thesis, a segregated pressure-based approach has been selected to solve the velocity-pressure coupling. With this strategy, we try to overtake the stability issue resulting from the saddle point structure of these equations. In particular, the *Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)* algorithm is here employed in its steady-state laminar form. To solve numerically the equations, we employ the open source CFD software, developed primarily by *OpenCFD Ltd* 2004, called *OpenFOAM*[45]. In the following paragraphs, we will recall the SIMPLE solver developed for the Navier-Stokes Equations, employed in our case for the resolution of the direct problem. Moreover, we extended its application to the adjoint and to the sensitivity equations, in a slightly modified version. For the implementation of the SIMPLE algorithm we refer to the OpenFOAM documentation<sup>1</sup> and to [51].

The solver employ a segregated solution procedure. This means that, the equations for each variable, characterizing the velocity-pressure coupled system, are solved one at a time in an iterative way, by substituting the last obtained solution as a known variable into the following equation to be solved. The non-linearity appearing in the momentum equation is approximated computing it from the velocity and pressure values of the previous iteration. The link with pressure is introduced to avoid a decoupling between the velocity and the pressure equations and hence to have not high frequency oscillations in the solution (*check board effect*). The first equation to be solved is the momentum one. It returns a velocity field which, in general, doesn't satisfy the continuity equation. After that, the momentum and the continuity equations are employed to construct a Laplacian equation for the pressure, taking advantage of the divergence of the momentum equation. The aim is to obtain a correction for the pressure field, which, if inserted in the momentum equation, satisfy the continuity equation. After, correcting the velocity field, the above iterative algorithm is repeated until convergence is reached. As already mentioned, the state, adjoint and sensitive equations share the same velocity-pressure coupling structure. For this reason, here, we report only a step by step SIMPLE algorithm construction for the state equations, while, we will remark only the differences needed by the other problems.

---

<sup>1</sup><https://openfoamwiki.net/index.php/SimpleFoam>

### 2.3.1 Momentum Equation

The Navier-Stokes equations into the SIMPLE algorithm are not written in a saddle-point matrix structure, as in (2.18). In the following, the numerical procedure used to solve the momentum equation is briefly exploited. The first step, to be performed, is the assembly of the matrices  $\mathbf{A}_v$  and  $\mathbf{B}_p$ . Then, we solve this equation for velocity unknown  $\mathbf{v}$  with pressure values of the previous iteration, to obtain the velocity estimation  $\mathbf{v}^*$ . In the usual semi-discrete matrix form, it can be written as:

$$\mathbf{A}_P \mathbf{v}_P + \sum_N \mathbf{A}_N \mathbf{v}_N = \mathbf{b}_P,$$

where  $\mathbf{A}_P$  is the matrix coefficient associated with the centre point of the cell  $P$ ,  $\mathbf{A}_N$  contains the matrix coefficients associated with all neighbours around the point  $P$  and  $\mathbf{b}_P$  is the source vector term. The sum  $\sum_N$  is taken over all neighbours influencing the computation around the point  $P$ .

The next step is the under relaxation of momentum equation. The under relaxation is required in order to prevent divergent solutions. Even though, the discrete version of the momentum equation is non-linear, the non-linearity is resolved using the solution of the previous iteration. This causes a large change of the new velocity leading often to divergence, see for more details [51]. Employing implicit under-relaxation, the new semi-discrete momentum equation can be written as follows:

$$\frac{1}{\alpha_v} \mathbf{A}_P \mathbf{v}_P + \sum_N \mathbf{A}_N \mathbf{v}_N = \mathbf{b}_P + \frac{1 - \alpha_v}{\alpha_v} \mathbf{A}_P \mathbf{v}_P^{n-1}.$$

Here  $\alpha_v$  denotes the under relaxation factor and  $\mathbf{v}_P^{n-1}$  is the solution at the previous iteration step.

After the under relaxation, the contribution of the pressure gradient is added to the right hand-side of the matrix and the system gets solved. The equation in semi discrete form is written as follows:

$$\frac{1}{\alpha_v} \mathbf{A}_P \mathbf{v}_P + \sum_N \mathbf{A}_N \mathbf{v}_N = \mathbf{b}_P + \frac{1 - \alpha_v}{\alpha_v} \mathbf{A}_P \mathbf{v}_P^{n-1} - \nabla p_P,$$

where  $\nabla p_P$  denotes the contribution of the pressure gradient to the equation of the cell centre  $P$ . The equation is finally solved in order to obtain the estimate  $\mathbf{u}^*$  by also considering the contribution of the pressure equation.

### 2.3.2 Pressure Equation

In this subsection, we construct the pressure equation solved into the SIMPLE algorithm, in order to ensure the incompressibility constraint. A good explanation of the derivation of pressure equation, involved in the SIMPLE method, can be found in [55].

For the derivation of the pressure equation, we start from the momentum equation in semi-discrete form after the solution of the momentum equation:

$$\mathbf{A}_P \mathbf{v}_P^* = - \sum_N \mathbf{A}_N \mathbf{v}_N^* + \mathbf{b}_P + \frac{1 - \alpha_v}{\alpha_v} \mathbf{A}_P \mathbf{v}_P^{n-1} - \nabla p_P^{n-1} = \mathbf{H}[\mathbf{v}^*] - \nabla p_P^{n-1}$$

After, dividing the above equation by  $a_P^*$ , we obtain:

$$\mathbf{v}_P^* = \frac{\mathbf{H}[\mathbf{v}^*]}{\mathbf{A}_P^*} - \frac{\nabla p_P^{n-1}}{\mathbf{A}_P^*},$$

where  $\mathbf{A}_P^* = \frac{1}{\alpha_v} \mathbf{A}_P$  are the modified diagonal coefficients of the matrix after the under-relaxation.

The goal of the next step is to find a correction for the velocity  $\mathbf{v}'_P$  and for the pressure  $p'$  fields in order to find a new velocity  $\mathbf{v}_P = \mathbf{v}_P^* + \mathbf{v}'_P$  which satisfies the continuity equation:

$$\mathbf{v}_P = \frac{\mathbf{H}[\mathbf{v}^*]}{\mathbf{A}_P^*} + \frac{\mathbf{H}[\mathbf{v}']}{\mathbf{A}_P^*} - \frac{\nabla p_P^{n-1}}{\mathbf{A}_P^*} - \frac{\nabla p'_P}{\mathbf{A}_P^*}.$$

The equation for the velocity correction can be written as:

$$\mathbf{v}'_P = -\frac{\sum_N \mathbf{A}_N \mathbf{v}'_N}{\mathbf{A}_P^*} - \frac{\nabla p'_P}{\mathbf{A}_P^*} = \frac{\mathbf{H}[\mathbf{u}']}{\mathbf{A}_P^*} - \frac{\nabla p'_P}{\mathbf{A}_P^*}.$$

Notice that, here, it is assumed that the diagonal coefficients of the corrector equation are the same as in the momentum equation previously solved. Taking the divergence of the above equation (we want to obtain a velocity field  $\mathbf{v}_P$  which satisfies the continuity equation) we get an equation for the pressure  $p_P = p_P^{n-1} + p'_P$ . For state, sensitivity and also adjoint equations without pressure data we set this divergence to zero and obtain the following pressure equation

$$\nabla \cdot \left( \frac{\nabla p_P}{\mathbf{A}_P^*} \right) = \nabla \cdot \left( \frac{\mathbf{H}[\mathbf{v}^*]}{\mathbf{A}_P^*} + \frac{\mathbf{H}[\mathbf{v}']}{\mathbf{A}_P^*} \right).$$

The changes for adjoint equations with pressure data only are written at the end of this Subsection, for completeness.

If we neglect the contribution of the neighbours in the velocity correction (i.e. we set  $\mathbf{H}[\mathbf{u}'] = 0$ ), we obtain the pressure equation solved in the SIMPLE algorithm:

$$\nabla \cdot \left( \frac{\nabla p_P}{\mathbf{A}_P^*} \right) = \nabla \cdot \left( \frac{\mathbf{H}[\mathbf{v}^*]}{\mathbf{A}_P^*} \right).$$

To stabilize the iterative method, we under-relax the pressure field used for the subsequent steps, as follows:

$$p^{n+1} = p^n + \alpha_p (p^{n+1} - p^n),$$

where  $\alpha_p$  is smaller than 1. For the SIMPLE method the velocity at point  $P$  can be written:

$$\mathbf{v}_P = \frac{1}{\mathbf{A}_P^*} \mathbf{H}[\mathbf{v}^*] - \frac{1}{\mathbf{A}_P^*} \nabla p_P.$$

Moreover, this relation is used together with the discretized version of continuity equation to update the mass fluxes  $F_f$ , as follows:

$$F_f = \mathbf{v}_f \cdot \mathbf{S}_f = \frac{1}{\mathbf{A}_P^*} \mathbf{H}[\mathbf{v}^*] \cdot \mathbf{S}_f - \frac{1}{\mathbf{A}_P^*} \nabla p_P \cdot \mathbf{S}_f.$$

**Algorithm 2:** The SIMPLE Method.

**Input:** An initial guess for the pressure field  $p^*$  and the velocity field  $\mathbf{v}^*$ ;  
Under-relaxation factors  $\alpha_v$  and  $\alpha_p$ ; A convergence criterion.

**1 repeat**

**2** Momentum predictor step: solve the discretized momentum equation for the guessed pressure field  $p^*$  as

$$\mathbf{A}_{\mathbf{P}}^* \mathbf{v}_{\mathbf{P}}^* = \mathbf{H}[\mathbf{v}^*] - \nabla p^*,$$

relaxed with the factor  $\alpha_v$ ;

**3** Correct pressure: the new pressure field is computed based on the obtained velocity field from the last step as

$$\nabla \cdot \left( \frac{\nabla \tilde{p}}{\mathbf{A}_{\mathbf{P}}^*} \right) = \nabla \cdot \left( \frac{\mathbf{H}[\mathbf{v}^*]}{\mathbf{A}_{\mathbf{P}}^*} \right),$$

then under-relax pressure field with factor  $\alpha_p$ . The under-relaxed pressure field is called  $p^{**}$ ;

**4** Correct velocity: using the corrected under-relaxed pressure field from previous step correct the velocity explicitly by

$$\mathbf{v}_{\mathbf{P}}^{**} = \frac{1}{\mathbf{A}_{\mathbf{P}}^*} \mathbf{H}[\mathbf{v}^*] - \frac{1}{\mathbf{A}_{\mathbf{P}}^*} \nabla p^{**},$$

where  $\mathbf{v}^{**}$  is the new corrected velocity field;

**5** Update the mass flux as

$$F_f = \frac{1}{\mathbf{A}_{\mathbf{P}}^*} \mathbf{H}[\mathbf{v}^{**}] \cdot \mathbf{S}_f - \frac{1}{\mathbf{A}_{\mathbf{P}}^*} \nabla p^{**} \cdot \mathbf{S}_f,$$

where  $\nabla p^{**} \cdot \mathbf{S}_f$  is computed with Gauss-linear approximation and non-orthogonal correction;

**6** Set  $\mathbf{v}^* = \mathbf{v}^{**}$  and  $p^* = p^{**}$ ;

**7 until** achieving convergence;

**Output:** The converged solution field  $(\mathbf{v}, p)$ .

We can summarize this strategy with the following SIMPLE Algorithm 2.

Instead, for what regards the adjoint problem with observed pressure data, the adjoint velocity isn't divergence free. Then we obtain a different pressure equation:

$$\nabla \cdot \left( \frac{\nabla q_P}{\mathbf{A}_{\mathbf{P}}^*} \right) = \nabla \cdot \left( \frac{\mathbf{H}[\mathbf{w}^*]}{\mathbf{A}_{\mathbf{P}}^*} + \frac{\mathbf{H}[\mathbf{w}']}{\mathbf{A}_{\mathbf{P}}^*} \right) + (pd)_P - p_P,$$

which changes the Algorithm 2 accordingly.



## Chapter 3

# Reduced Order Model Formulation

In this chapter, we present the numerical solution for inverse problems in *Computational Fluid Dynamics (CFD)* with reduced order methods in parametrized settings. The problem, we want to deal with, is a lid-driven cavity flow, where the viscosity field has to be estimated by exploiting pressure and velocity data.

Numerical methods are capable of approximating the solutions for such a problem with a reliable accuracy, as we have shown in the previous Chapter. However, as well-known, the numerical solution, associated to inverse problems in CFD, at the Full Order level, is particularly expensive in terms of computational cost and CPU time. Moreover, a input of our direct problem, which we want estimate in inverse problem, can depends on one or more parameter. Then, in this Chapter, we refer, in this framework, to *parametric optimal flow control problems*, or more in particular, in this work, to *parametric inverse problem*, with a slight abuse of notation. In this parametrized settings, the computations are performed in a repetitive way, for many different values of the parameter. Thus, the computational cost increases rapidly with respect to the dimensions of parameter space. On the contrary, Reduced Order Models exploit information obtained from a certain number of full order simulations, calculated in a typically expensive *offline phase*, to set up a lower dimensional problem, that can be solved in a limited amount of time, with a lower computational effort, in an *online phase*. Based on such an offline/online paradigm, the *Reduced Order Methods (ROMs)* have been widely applied to optimal control simulations to reduce the computational cost of the resulting parametric numerical simulations with finite element formulations, how it is proven in [42, 69, 75, 67, 68, 7]. In this Thesis we applied such a technique to finite volume approximation of parametric inverse problems.

Our ROM implementation employs the ITHACA-FV library (In real Time Highly Advanced Computational Applications for Finite Volumes) <sup>1</sup>, an open access C++ library

---

<sup>1</sup><https://github.com/mathLab/ITHACA-FV>

based on OpenFOAM<sup>2</sup>.

In the context of this Thesis, we aim to develop ROMs which are able to approximate the solution of *Parametrized Optimal Flow Control Problems* (OFCP( $\mu$ )) used, in particular, for parametric inverse problems. As in [28], we denote with  $M$  the solution manifold composed by all solutions of the parametric problem varying the parameters in the parameter space  $\mathbb{P}$ , i.e.,

$$M = \{ u(\mu) \mid \mu \in \mathbb{P} \} \subset \mathbb{V},$$

where each  $u(\mu) \in \mathbb{V}$  corresponds to the solution of the exact problem. In general, the exact solution is not available in analytic way. Thus, what we look for is a numerical approximated of the solution  $u_\delta(\mu) \in \mathbb{V}_\delta$ . Following the definition for the exact continuous problem, we also define the discrete version of the solution manifold

$$M_\delta = \{ u_\delta(\mu) \mid \mu \in \mathbb{P} \} \subset \mathbb{V}_\delta,$$

where each  $u_\delta(\mu) \in \mathbb{V}_\delta$  corresponds to the *truth solution* of the discrete problem. The computational cost may be very high since it depends directly on  $N_\delta = \dim(\mathbb{V}_\delta)$ . Reduced order modelling for parametric problem is based on the assumption that the solution fields lives in a low dimensional manifold, i.e., that the span of a low number of appropriately chosen basis functions approximate the solution manifold with a small error between truth and reduced solutions. We shall call these basis functions the *Reduced Basis (RB)*. It will allow us to approximate the truth solution  $u_\delta(\mu)$ , by the use of an  $N$ -dimensional subspace  $\mathbb{V}_{rb}$  of  $\mathbb{V}_\delta$ . The assumption of the low dimensionality for the solution manifold implies that  $N \ll N_\delta$ . For more details, from the theoretical point of view, see [28].

Relying on this hypothesis, any element of the solution manifold can be properly approximated by the linear span of a low number of global reduced basis functions. In particular, for OFCPs( $\mu$ ), the state and adjoint fields can be approximated as a linear combination of the dominant *modes* (basis functions) multiplied by scalar coefficients. The modes are assumed to be dependent on space variables only, while, the coefficients only depend on the parameter value. The last assumption leads to the following approximation

---

<sup>2</sup><https://www.openfoam.com/>

of the fields:

$$\mathbf{v}(\mathbf{x}; \mu) \approx \mathbf{v}_r(\mathbf{x}; \mu) = \sum_{i=1}^{N_v} a_i(\mu) \phi_i(\mathbf{x}), \quad (3.1)$$

$$(3.2)$$

$$p(\mathbf{x}; \mu) \approx p_r(\mathbf{x}; \mu) = \sum_{i=1}^{N_p} b_i(\mu) \psi_i(\mathbf{x}), \quad (3.3)$$

$$(3.4)$$

$$\mathbf{w}(\mathbf{x}; \mu) \approx \mathbf{w}_r(\mathbf{x}; \mu) = \sum_{i=1}^{N_w} c_i(\mu) \zeta_i(\mathbf{x}), \quad (3.5)$$

$$(3.6)$$

$$q(\mathbf{x}; \mu) \approx q_r(\mathbf{x}; \mu) = \sum_{i=1}^{N_q} d_i(\mu) \eta_i(\mathbf{x}), \quad (3.7)$$

$$(3.8)$$

where, the  $(\mathbf{v}_r, p_r)$ ,  $(\mathbf{w}_r, q_r)$  are the reduced state and adjoint fields, respectively, while  $\phi_i$ ,  $\psi_i$ ,  $\zeta_i$  and  $\eta_i$  are the modes for velocity, pressure and their respective adjoint fields. The reduced order degrees of freedom are denoted by  $a_i$ ,  $b_i$ ,  $c_i$  and  $d_i$ , while,  $N_v$ ,  $N_p$ ,  $N_w$  and  $N_q$  define the dimension of reduced basis spaces for the corresponding variables.

The *reduced basis spaces* for state velocity field  $V_{rb} = \text{span} \left\{ [\phi_i]_{i=1}^{N_u} \right\}$ , state pressure one  $P_{rb} = \text{span} \left\{ [\psi_i]_{i=1}^{N_p} \right\}$ , adjoint velocity one  $W_{rb} = \text{span} \left\{ [\zeta_i]_{i=1}^{N_w} \right\}$  and adjoint pressure one  $Q_{rb} = \text{span} \left\{ [\eta_i]_{i=1}^{N_q} \right\}$  can be generated following several strategies, as reported in the literature. In this work, we utilize the *Proper Orthogonal Decomposition (POD)* approach. We define the *snapshots* as the high-order solutions of the truth problem for different values of the parameters. With POD method, we construct the reduced order solution spaces from the information captured by these snapshots. For an overview on *snapshots-based method* you can see to [62].

In this Chapter, we provide a review of POD modes construction for both state and adjoint fields. Then, a Galerkin projection is performed onto the POD spaces for state equations, adjoint equations and optimality condition, one at a time. After that, the stability issue for reduced equations will be analyzed, providing a possible solution given by a supremizer enrichment. Non-homogeneous Diriclet boundary conditions (BCs) get treated by employing a lifting function method. Moreover, we will develop a reduced version of CG method as an optimization solver for our problem. Finally, we will exploit the offline-online decomposition to perform some reduced solutions.

### 3.1 Generation of the reduced basis space with a Proper Orthogonal Decomposition approach

In this section, we exploit a particular snapshots-based method to construct modal spaces of reduced solutions: the *Proper Orthogonal Decomposition (POD)*. POD is a method to compress the information contained into a set of numerical realizations into a reduced number of orthogonal basis functions (modes). These realizations, or snapshots, are obtained by solving a high fidelity problem for many different values of a parameter, chosen into the parameter space. The modes obtained with POD are capable of retaining the most relevant dynamics content when suitably combined. For a general overview on POD methods for Parametrized Partial Differential Equations (PPDE), see [28, 62, 56]. For a theoretical point of view about the generation of POD-spaces we refer to [20, 41, 72]. In particular, for finite volume POD-Galerkin applied to fluid dynamics, see [66, 64, 32]. Applications to OFCP( $\mu$ ) in a finite element setting can be found in [7, 74, 2].

The POD modes are "optimal", in the sense that, for every number of chosen modes, the error between the  $L_2$  projection of the snapshots onto the modes and the snapshots themselves is minimized. It has to be remarked that, the FOM, presented in Chapter 2, is solved for each value of the parameter  $\mu \in \mathbb{P}_m = \{ \mu_1, \dots, \mu_m \} \subset \mathbb{P}$  where  $\mathbb{P}_m$  is a finite set of samples inside the parameter space  $\mathbb{P}$ . The *snapshots matrices*  $\mathbf{S}^v$ ,  $\mathbf{S}^p$ ,  $\mathbf{S}^w$  and  $\mathbf{S}^q$ , for velocity, pressure, adjoint velocity and adjoint pressure respectively, are then given by  $m$  full-order snapshots:

$$\mathbf{S}^v = \{ \mathbf{v}(\mathbf{x}, \mu_1), \dots, \mathbf{v}(\mathbf{x}, \mu_m) \} \in \mathbb{R}^{N_v^h \times m}, \quad (3.9)$$

$$(3.10)$$

$$\mathbf{S}^p = \{ p(\mathbf{x}, \mu_1), \dots, p(\mathbf{x}, \mu_m) \} \in \mathbb{R}^{N_p^h \times m}, \quad (3.11)$$

$$\mathbf{S}^w = \{ \mathbf{w}(\mathbf{x}, \mu_1), \dots, \mathbf{w}(\mathbf{x}, \mu_m) \} \in \mathbb{R}^{N_w^h \times m}, \quad (3.12)$$

$$\mathbf{S}^q = \{ q(\mathbf{x}, \mu_1), \dots, q(\mathbf{x}, \mu_m) \} \in \mathbb{R}^{N_q^h \times m}, \quad (3.13)$$

where  $N_v^h$ ,  $N_p^h$ ,  $N_w^h$  and  $N_q^h$  are the degrees of freedom for the solution fields, respectively. Let  $X = [L^2(\Omega)]^2$  and  $Y = L^2(\Omega)$  be separable Hilbert space endowed with the inner product  $(\cdot, \cdot)_X$  and  $(\cdot, \cdot)_Y$ , respectively, and the corresponding induced norm. We assume that the snapshots are linearly independent (otherwise we decrease  $m$ ). We set the *snapshot spaces* as:

$$V_m = \text{span} \{ \mathbf{v}_1, \dots, \mathbf{v}_m \} \subset X, \quad (3.14)$$

$$P_m = \text{span} \{ p_1, \dots, p_m \} \subset Y, \quad (3.15)$$

$$W_m = \text{span} \{ \mathbf{w}_1, \dots, \mathbf{w}_m \} \subset X, \quad (3.16)$$

$$Q_m = \text{span} \{ q_1, \dots, q_m \} \subset Y. \quad (3.17)$$

The *POD-method* consists of properly choosing an orthonormal basis for the snapshot spaces such that  $\forall N_v, N_p, N_w, N_q = 1, \dots, m$  the mean square error between the snapshots and the corresponding orthogonal projection (3.2)-(3.8) onto the reduced bases is minimized. Then the POD-spaces  $V_{POD} = \text{span} \{ [\phi_i]_{i=1}^{N_u} \} \subseteq V_m$ ,  $P_{POD} = \text{span} \{ [q_i]_{i=1}^{N_p} \} \subseteq$

$P_m, W_{POD} = \text{span} \left\{ [\zeta_i]_{i=1}^{N_w} \right\} \subseteq W_m$  and  $Q_{POD} = \text{span} \left\{ [\eta_i]_{i=1}^{N_q} \right\} \subseteq Q_m$  are the ones fulfilling the following minimization problems:

$$\begin{aligned} \min_{V_{POD}} \sqrt{\frac{1}{m} \sum_{n=1}^m \left\| \mathbf{v}_n - \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X \phi_k \right\|_X^2} \quad \forall N_v = 1, \dots, m, \\ \text{subject to } (\phi_i, \phi_j)_X = \delta_{ij} \quad \text{for } 1 \leq i \leq N_v, 1 \leq j \leq i, \end{aligned} \quad (3.18)$$

$$\begin{aligned} \min_{P_{POD}} \sqrt{\frac{1}{m} \sum_{n=1}^m \left\| p_n - \sum_{k=1}^{N_p} (p_n, \psi_k)_Y \psi_k \right\|_Y^2} \quad \forall N_p = 1, \dots, m, \\ \text{subject to } (\psi_i, \psi_j)_Y = \delta_{ij} \quad \text{for } 1 \leq i \leq N_p, 1 \leq j \leq i, \end{aligned} \quad (3.19)$$

$$\begin{aligned} \min_{W_{POD}} \sqrt{\frac{1}{m} \sum_{n=1}^m \left\| \mathbf{w}_n - \sum_{k=1}^{N_w} (\mathbf{w}_n, \zeta_k)_X \zeta_k \right\|_X^2} \quad \forall N_w = 1, \dots, m, \\ \text{subject to } (\zeta_i, \zeta_j)_X = \delta_{ij} \quad \text{for } 1 \leq i \leq N_w, 1 \leq j \leq i, \end{aligned} \quad (3.20)$$

$$\begin{aligned} \min_{Q_{POD}} \sqrt{\frac{1}{m} \sum_{n=1}^m \left\| q_n - \sum_{k=1}^{N_q} (q_n, \eta_k)_Y \eta_k \right\|_Y^2} \quad \forall N_q = 1, \dots, m, \\ \text{subject to } (\eta_i, \eta_j)_Y = \delta_{ij} \quad \text{for } 1 \leq i \leq N_q, 1 \leq j \leq i, \end{aligned} \quad (3.21)$$

where  $\mathbf{v}_n, p_n, \mathbf{w}_n$  and  $q_n$  are the  $n$ -th snapshots which have been computed for a value of the parameter  $\mu \in \mathbb{P}_m$ . Therefore, the construction of the POD-spaces requires the computation of the *POD-basis*  $[\phi_i]_{i=1}^{N_v}, [\psi_i]_{i=1}^{N_p}, [\zeta_i]_{i=1}^{N_w}$  and  $[\eta_i]_{i=1}^{N_q}$  of ranks  $N_v \ll N_v^h, N_p \ll N_p^h, N_w \ll N_w^h$  and  $N_q \ll N_q^h$ , (for the assumption of the low dimensionality of the solution manifold).

**Remark 3.1** Usually, in a standard finite element framework, since the velocity field belongs to the natural functional space  $[H^1(\Omega)]^2$ , to compute its reduced basis functions minimizing the mean square error, the  $H^1$  norm is preferred and the  $L^2$  norm is employed for pressure (the same happens for the adjoint fields). Here, for both velocity and pressure, the  $L^2$  norm is chosen, because using a finite volume method, both the velocity and the pressure belong to discontinuous finite volume spaces and, the computation of the gradient of the velocity, necessary for the  $H^1$  semi-norm computation, add further computational cost.

**Remark 3.2** The minimization problem can be solved by performing a Singular Value Decomposition (SVD). For example, for what concerns the velocity snapshots, we have:

$$\mathbf{S}^v = \mathbf{\Xi}^v \mathbf{\Sigma}^v \mathbf{\Upsilon}^v,$$

where  $\mathbf{\Xi}^v \in \mathbb{R}^{N_v^h \times N_v^h}$  is a square matrix of the left-singular vectors,  $\mathbf{\Upsilon}^v \in \mathbb{R}^{m \times m}$  is a square matrix of the right-singular vectors and  $\mathbf{\Sigma}^v \in \mathbb{R}^{N_v^h \times m}$  is a rectangular diagonal matrix of the singular values. The resulting POD modes  $\phi_i$  are then given by the columns of the matrix  $\mathbf{\Xi}^v$ . This procedure might be however computationally expensive, especially when a refinement of the mesh turns out to be necessary to properly discretize the domain

$\Omega$ . An equivalent and more efficient way to construct this modes, based on the method of snapshots, consists in solving the eigenvalue problems related to the correlation matrix of the fields. This modes construction is exploited in the next Subsection.

### 3.1.1 Proof on construction of modes

For more details on how the minimization problems is solved, we refer to [41, 72, 20]. We report here the calculation for the velocity field.

Since the elements  $\phi_i$  are assumed to be orthonormal in  $X$ , we obtain:

$$\|\mathbf{v}_n - \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X \phi_k\|_X^2 = \|\mathbf{v}_n\|_X^2 - \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X^2. \quad (3.22)$$

Then, the above minimization problem for velocity field is equivalent to:

$$\begin{aligned} \min J(\phi_1, \dots, \phi_{N_v}) = \min_{[\phi_i]_{i=1}^{N_v}} & - \sum_{n=1}^m \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X^2 \quad \forall N_v = 1, \dots, m, \\ \text{subject to} & \quad \epsilon_k(\phi_1, \dots, \phi_{N_v}) = \mathbf{0} \quad \text{for } 1 \leq k \leq N_v, \end{aligned} \quad (3.23)$$

where

$$\epsilon_k(\phi_1, \dots, \phi_{N_v}) = \begin{pmatrix} (\phi_k, \phi_k)_X - 1 \\ (\phi_{k+1}, \phi_k)_X \\ \vdots \\ (\phi_{N_v}, \phi_k)_X \end{pmatrix}.$$

Introducing the Lagrange functional  $\mathcal{L} : X^{N_v} \times \mathbb{R}^{N_v} \times \dots \times \mathbb{R} \rightarrow \mathbb{R}$  associated with problem (3.23)

$$\mathcal{L}(\phi_1, \dots, \phi_{N_v}, \omega_1, \dots, \omega_{N_v}) = J(\phi_1, \dots, \phi_{N_v}) + \sum_{k=1}^{N_v} (\epsilon_k(\phi_1, \dots, \phi_{N_v}), \omega_k)_{\mathbb{R}^{N_v-k+1}},$$

then, the first-order necessary optimality conditions ( $\nabla_{\phi} \mathcal{L}(\phi_1, \dots, \phi_{N_v}, \omega_1, \dots, \omega_{N_v}) = \mathbf{0}$ ) for the optimization problem (3.23) is given by the following operator equation, as introduced in [72]:

$$R\phi_i = \lambda_i \phi_i, \quad \text{for } 1 \leq i \leq N_v \quad (3.24)$$

where, we denote the first components of the vector  $\omega_i$  by  $\lambda_i = \omega_i^1$  and the introduced operator  $R \in B(X)$  reads:

$$R = \sum_{j=1}^m (\cdot, \mathbf{v}_j)_X \mathbf{v}_j.$$

Here  $B(X)$  denotes the Banach space of all continuous linear operator on  $X$ .

We can prove (see, e.g., [72, 20]) the following properties on the operator  $R : X \rightarrow X$ :

- bounded,  $\|Rw\|_X^2 \leq \sum_{i=1}^m \|\mathbf{v}_i\|_X^2 \|w\|_X^2 < \infty \quad \forall w \in X$ ,
- self-adjoint,  $(Rw, v)_X = (w, Rv)_X \quad \forall v, w \in X$ ,

- non-negative,  $(Rw, w)_X = \sum_{i=1}^m (\mathbf{v}_i, w)_X^2 \geq 0 \quad \forall w \in X$ ,
- compact, since the image of  $R$ , i.e.  $Im(R) = \text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ , is finite dimensional and bounded.

Thanks to the eigenfunctions expansion theorem (see, e.g., [63]), with the previous properties, there exist a complete orthonormal basis  $\{\phi_i\}_{i=1}^\infty$  for  $X$  and a decreasing sequence  $\{\lambda_i\}_{i=1}^\infty$  so that

$$R\phi_i = \lambda_i \phi_i, \quad \lambda_1 \geq \lambda_2 \geq \dots \geq 0 \quad \text{and} \quad \lambda_i \rightarrow 0 \quad \text{as} \quad i \rightarrow \infty. \quad (3.25)$$

Moreover, the functions  $\phi_i$  form an orthonormal basis for the range of  $R$ , i.e.  $Im(R) = V_m$ , with the following spectral decomposition for this operator

$$Rw = \sum_{n=1}^m \lambda_n (w, \phi_n)_X \phi_n = \sum_{n=1}^m (Rw, \phi_n)_X \phi_n \quad \forall w \in X.$$

This shows us, that exists a solution for the first-order necessary condition (3.24) but this is not a sufficient optimality condition for (3.23). We can prove, anyway, that  $\{\phi_i\}_{i=1}^{N_v}$  calculated by solving the eigenvalue problem (3.24), is an optimal solution. Let  $\{\bar{\phi}_i\}_{i=1}^{N_v}$  be an orthonormal set in  $V_m$ . Since  $\{\phi_i\}_{i=1}^m$  is an orthonormal basis for  $V_m$ , we obtain:

$$\bar{\phi}_k = \sum_{i=1}^m (\bar{\phi}_k, \phi_i)_X \phi_i \quad \text{for} \quad k = 1, \dots, N_v,$$

which implies (see, e.g., [72])

$$J(\bar{\phi}_1, \dots, \bar{\phi}_{N_v}) \geq - \sum_{j=1}^{N_v} \lambda_j.$$

Moreover, we note that condition (3.25) leads to

$$\sum_{n=1}^m (\mathbf{v}_n, \phi_k)_X^2 = (R\phi_k, \phi_k) = \lambda_k \quad \forall k \in \mathbb{N}. \quad (3.26)$$

Then  $\{\phi_i\}_{i=1}^{N_v}$  is optimal because  $\forall \{\bar{\phi}_i\}_{i=1}^{N_v}$  orthonormal set of  $V_m$  we get:

$$J(\bar{\phi}_1, \dots, \bar{\phi}_{N_v}) \geq J(\phi_1, \dots, \phi_{N_v}). \quad (3.27)$$

Therefore, we show that, denoting by  $\{\phi_i\}_{i=1}^\infty$  and  $\{\lambda_i\}_{i=1}^\infty$  the eigenfunctions and corresponding eigenvalues for the operator  $R$ , satisfying  $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$  and being  $N_v \in \mathbb{N}$  with  $N_v \leq m$ , then,  $\{\phi_i\}_{i=1}^{N_v}$  solves (3.23) with minimum

$$J(\phi_1, \dots, \phi_{N_v}) = - \sum_{k=1}^{N_v} \lambda_k.$$

Moreover,  $\{\phi_i\}_{i=1}^m$  is also a solution to (3.18).

The resolution of the eigenvalue problem (3.24) is more difficult from a practical point of view. Thus, one can compute the POD-basis of rank  $N_v$  solving a finite dimensional  $m \times m$  eigenvalue problem (see, e.g., [20, 72, 41]). To find the first  $N_v$  strictly positive eigenvalues  $\{\lambda_k\}_{k=1}^{N_v}$  and the corresponding eigenfunctions  $\{\phi_k\}_{k=1}^{N_v} \subset V_m$ , we can write  $\phi_k$  exploiting the following linear combination of snapshots:

$$\phi_k = \pi_k \sum_{j=1}^m y_j^k \mathbf{v}_j \quad \text{for } k = 1, \dots, N_v,$$

where the constants  $\pi_k$  will be calculated in such a way that  $\|\phi_k\|_X = 1$  holds. Our goal is to determine the coefficients of this expression. Inserting this linear combination in the eigenvalues problem (3.24) we obtain:

$$\sum_{i=1}^m \left( \sum_{j=1}^m (\mathbf{v}_i, \mathbf{v}_j)_X y_j^k \right) \mathbf{v}_i = \lambda_k \sum_{i=1}^m y_i^k \mathbf{v}_i \quad \text{for } k = 1, \dots, N_v.$$

Since the snapshots are assumed to be linearly independent in  $X$ , we determine the coefficients  $y_i^k$  from the eigenvector of the following  $m \times m$  eigenvalue problem:

$$\sum_{j=1}^m (\mathbf{v}_i, \mathbf{v}_j)_X y_j^k = \lambda_k y_i^k \quad \text{for } k = 1, \dots, N_v \quad \text{and } i = 1, \dots, m. \quad (3.28)$$

Utilizing the *correlation matrix* for the velocity solutions  $\mathbf{C}^v \in \mathbb{R}^{m \times m}$ , being

$$\mathbf{C}_{ij}^v = (\mathbf{v}_i, \mathbf{v}_j)_X, \quad (3.29)$$

the eigenvalue problem (3.28) can be expressed as

$$\mathbf{C}^v \mathbf{V} = \boldsymbol{\lambda}^v \mathbf{V}, \quad (3.30)$$

where  $\mathbf{V} \in \mathbb{R}^{m \times m}$  is a square matrix, whose columns are the eigenvectors  $v_k \in \mathbb{R}^m$  with  $(v_k)_i = y_i^k$  for  $k, i = 1, \dots, m$ , and  $\boldsymbol{\lambda}^v \in \mathbb{R}^{m \times m}$  is a diagonal matrix, whose diagonal entries are the eigenvalues  $\boldsymbol{\lambda}_{kk}^v = \lambda_k$  for  $k = 1, \dots, m$ .

We observe that  $\mathbf{C}^v$  is:

- symmetric,  $\mathbf{C}_{ij}^v = \mathbf{C}_{ji}^v$ ,
- positive definite,  $(\mathbf{C}^v v, v)_{\mathbb{R}^m} = \|\sum_{i=1}^m v_i \phi_i\|_X^2 \geq 0 \quad \forall v \in \mathbb{R}^m$  and  $(\mathbf{C}^v v, v)_{\mathbb{R}^m} = 0$  if and only if  $v = 0$ .

Then for this properties  $\mathbf{C}^v$  has strictly positive eigenvalue  $\{\lambda_k\}_{k=1}^m$  and eigenvectors  $\{v_k\}_{k=1}^m$  solving the eigenvalue problem (3.28). Moreover, normalizing the basis functions, using the properties of  $\mathbf{C}^v$  and (3.28), we can calculate the coefficient  $\pi_k$  as

$$1 = \|\phi_k\|_X^2 = \pi_k^2 \left\| \sum_{i=1}^m v_i \phi_i \right\|_X^2 = \pi_k^2 (\mathbf{C}^v v, v)_{\mathbb{R}^m} = \pi_k^2 \lambda_k \Rightarrow \pi_k = \frac{1}{\sqrt{\lambda_k}} \quad \forall k = 1, \dots, N_v.$$

□

Finally, we can summarize the computation of the POD-basis  $\{\phi_k\}_{k=1}^{N_v}$  of rank  $N_v$  for velocity field as follows:

1. Solve the eigenvalue problem (3.30) to compute the strictly positive eigenvalues  $\{\lambda_k\}_{k=1}^{N_v}$  and the eigenvectors  $\{v_k\}_{k=1}^{N_v}$  for  $k = 1, \dots, N_v$ ,
2. determine the POD-basis functions as

$$\phi_k = \frac{1}{\sqrt{\lambda_{\mathbf{k}\mathbf{k}}^v}} \mathbf{S}^v \mathbf{V}_{\mathbf{k}} \quad \forall k = 1, \dots, N_v. \quad (3.31)$$

**Remark 3.3** *The  $m \times m$  eigenvalue problem (3.30) can be solved by a numerical algorithm but, before that, the correlation matrix  $\mathbf{C}^v$  has to be computed by a numerical quadrature formula.*

**Remark 3.4** *From (3.22)-(3.26) it follows the error formula*

$$\begin{aligned} \sqrt{\bar{\epsilon}_{N_v}^{POD}} &\equiv \sqrt{\frac{1}{m} \sum_{n=1}^m \left\| \mathbf{v}_n - \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X \phi_k \right\|_X^2} = \\ &= \sqrt{\frac{1}{m} \sum_{n=1}^m \left( \|\mathbf{v}_n\|_X^2 - \sum_{k=1}^{N_v} (\mathbf{v}_n, \phi_k)_X^2 \right)} = \sqrt{\frac{1}{m} \sum_{k=N_v+1}^m \lambda_{\mathbf{k}\mathbf{k}}^v} \quad \forall N_v = 1, \dots, m. \end{aligned} \quad (3.32)$$

Hence, the decay rate of the positive eigenvalues  $\{\lambda_k\}_{k=1}^m$  plays a fundamental role for an accurate POD application. If one has to utilize a complete orthonormal basis  $\{\phi_k\}_{k=1}^m \subset X$  to represent elements of the snapshots space  $V_m$ , this leads to a high-dimensional approximation scheme. Instead, if the error estimation  $\sqrt{\bar{\epsilon}_{N_v}^{POD}}$  is sufficiently small for a not too large  $N_v \in \{1, \dots, m\}$ , then elements in  $V_m$  can be approximated by using their projection on the POD-space  $V_{POD} = \text{span} \left\{ [\phi_k]_{k=1}^{N_v} \right\}$  with the following projection

$$\begin{aligned} P_{N_v} : X &\rightarrow V_m \\ f &\mapsto P_{N_v}[f] = \sum_{i=1}^{N_v} (f, \phi_i)_X \phi_i. \end{aligned}$$

Therefore, it may be useful to define a relative information content of the POD-basis by the rate (see, e.g. [20])

$$\mathcal{E}(N_v) = \frac{\sum_{i=1}^{N_v} \lambda_i}{\sum_{i=1}^m \lambda_i} \in [0, 1]. \quad (3.33)$$

We can, for example, utilize this ratio to determine a basis of rank  $N_v$  such that it contains the 99% of the information contained in  $V_m$ , i.e.,  $\mathcal{E}(N_v) \approx 99\%$ . Thanks to (3.26), the

quantity  $\mathcal{E}(N_v)$  can be determine by the following formula, without knowing the eigenvalues  $\lambda_{N_v+1}, \dots, m$ ,

$$\mathcal{E}(N_v) = \frac{\sum_{i=1}^{N_v} \lambda_i}{\sum_{i=1}^m \|\mathbf{v}_i\|_X^2}.$$

The same procedure has to be repeated also for the pressure, the adjoint velocity and the adjoint pressure fields, considering the snapshots matrices  $\mathbf{S}^p$ ,  $\mathbf{S}^w$  and  $\mathbf{S}^q$ , respectively. In short, we can compute the corresponding correlation matrices  $\mathbf{C}^p$ ,  $\mathbf{C}^w$ ,  $\mathbf{C}^q \in \mathbb{R}^{m \times m}$  as

$$\mathbf{C}_{ij}^p = (p_i, p_j)_Y, \tag{3.34}$$

$$\mathbf{C}_{ij}^w = (\mathbf{w}_i, \mathbf{w}_j)_X, \tag{3.35}$$

$$\mathbf{C}_{ij}^q = (q_i, q_j)_Y, \tag{3.36}$$

$$\mathbf{C}_{ij}^q = (q_i, q_j)_Y, \tag{3.37}$$

$$\mathbf{C}_{ij}^q = (q_i, q_j)_Y, \tag{3.38}$$

and solve the eigenvalue problems

$$\mathbf{C}^p \mathbf{P} = \lambda^p \mathbf{P}, \tag{3.39}$$

$$\mathbf{C}^w \mathbf{W} = \lambda^w \mathbf{W}, \tag{3.40}$$

$$\mathbf{C}^w \mathbf{W} = \lambda^w \mathbf{W}, \tag{3.41}$$

$$\mathbf{C}^q \mathbf{Q} = \lambda^q \mathbf{Q}. \tag{3.42}$$

$$\mathbf{C}^q \mathbf{Q} = \lambda^q \mathbf{Q}. \tag{3.43}$$

Then, the POD modes for this fields can be computed as follows:

$$\psi_k = \frac{1}{\sqrt{\lambda_{kk}^p}} \mathbf{S}^p \mathbf{P}_k \quad \forall k = 1, \dots, N_p, \tag{3.44}$$

$$\zeta_k = \frac{1}{\sqrt{\lambda_{kk}^w}} \mathbf{S}^w \mathbf{W}_k \quad \forall k = 1, \dots, N_w, \tag{3.45}$$

$$\zeta_k = \frac{1}{\sqrt{\lambda_{kk}^w}} \mathbf{S}^w \mathbf{W}_k \quad \forall k = 1, \dots, N_w, \tag{3.46}$$

$$\zeta_k = \frac{1}{\sqrt{\lambda_{kk}^w}} \mathbf{S}^w \mathbf{W}_k \quad \forall k = 1, \dots, N_w, \tag{3.47}$$

$$\eta_k = \frac{1}{\sqrt{\lambda_{kk}^q}} \mathbf{S}^q \mathbf{Q}_k \quad \forall k = 1, \dots, N_q. \tag{3.48}$$

For more clarity, we recapitulate the Proper Orthogonal Decomposition procedure into Algorithm 3.

After introducing the POD method, one may perform the projection step in order to construct the reduced order system. The projection procedure is exploited in the next Section.

## 3.2 Galerkin projection onto the reduced spaces

Once the POD functional spaces are set, the reduced velocity, pressure, adjoint velocity and adjoint pressure fields can be approximated by (3.2)-(3.4)-(3.6)-(3.8). The unknown

---

**Algorithm 3:** Proper Orthogonal Decomposition Method.

---

**Input:** Snapshots matrices  $\mathbf{S}^v \in \mathbb{R}^{N_v^h \times m}$ ,  $\mathbf{S}^p \in \mathbb{R}^{N_p^h \times m}$ ,  $\mathbf{S}^w \in \mathbb{R}^{N_w^h \times m}$  and  $\mathbf{S}^q \in \mathbb{R}^{N_q^h \times m}$  for state velocity, state pressure, adjoint velocity and adjoint pressure, respectively; Relative information content  $\mathcal{E}$ .

- 1 Construct the correlation matrices  $\mathbf{C}^v, \mathbf{C}^p, \mathbf{C}^w, \mathbf{C}^q \in \mathbb{R}^{m \times m}$  by employing (3.29)-(3.35)-(3.37)-(3.38);
- 2 Solve the corresponding eigenvalue problems (3.30)-(3.40)-(3.42)-(3.43) to store eigenvectors matrices  $\mathbf{V}, \mathbf{P}, \mathbf{W}$  and  $\mathbf{Q}$ , and the corresponding eigenvalues as the diagonal entries of matrices  $\boldsymbol{\lambda}^v, \boldsymbol{\lambda}^p, \boldsymbol{\lambda}^w$  and  $\boldsymbol{\lambda}^q$ ;
- 3 Compute the modal basis functions  $[\phi_i]_{i=1}^m, [\psi_i]_{i=1}^m, [\zeta_i]_{i=1}^m$  and  $[\eta_i]_{i=1}^m$  as in (3.31)-(3.45)-(3.47)-(3.48), respectively;
- 4 Collect the first  $N_v$  state velocity basis functions,  $N_p$  state pressure ones,  $N_w$  adjoint velocity ones and  $N_q$  adjoint pressure ones, which have relative information content at least equal to  $\mathcal{E}$ .

**Output:** Basis functions  $[\phi_i]_{i=1}^{N_v}, [\psi_i]_{i=1}^{N_p}, [\zeta_i]_{i=1}^{N_w}$  and  $[\eta_i]_{i=1}^{N_q}$ .

---

vectors of coefficients  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$  and viscosity constant  $\nu$ , can be then obtained through a Galerkin projection of the KKT system onto the POD reduced basis spaces. For application of finite volume POD-Galerkin projection in fluid dynamics we refer to [66, 64, 30, 29, 31].

In this Section, we report the building procedure for the reduced order model, i.e. the projection of the Navier-Stokes equations and adjoint equations onto the POD space spanned by the velocity-pressure and adjoint velocity-pressure POD modes, respectively. Moreover, we introduce the first-order optimality condition with a reduced order approach. For simplicity, we treat only the case of wanted viscosity which is uniform in the domain.

### 3.2.1 Reduced State Equations

In this subsection, the Galerkin projection of the state equations (1.10) onto the state POD space is investigated and discussed. The idea here is to employ both the momentum balance and continuity equation.

The reduced order model of the momentum equation is computed performing an  $L^2$ -projection onto the reduced bases space  $V_{POD}$  spanned by the POD velocity modes, by following what has been exposed previously into Section 3.1. In particular, we obtain the following projection:

$$(\phi_i, (\mathbf{v} \cdot \nabla) \mathbf{v} - \nabla \cdot (\nu \nabla \mathbf{v}) + \nabla p)_X = 0 \quad \forall i = 1, \dots, N_v. \quad (3.49)$$

Substituting the POD approximations (3.2)-(3.4) for  $\mathbf{v}$  and  $p$ , respectively, in (3.49), one obtains the following reduced form of the momentum balance:

$$\nu \mathbf{B} \mathbf{a} - \mathbf{a}^T \mathbf{C} \mathbf{a} - \mathbf{H} \mathbf{b} = \mathbf{0}, \quad (3.50)$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are the vectors of unknowns for reduced state velocity and pressure respectively, while  $\mathbf{B} \in \mathbb{R}^{N_v \times N_v}$ ,  $\mathbf{C} \in \mathbb{R}^{N_v \times N_v \times N_v}$  and  $\mathbf{H} \in \mathbb{R}^{N_v \times N_p}$  are the reduced discretized differential operators which are computed as follows:

$$\begin{aligned} B_{ij} &= (\phi_i, \Delta \phi_j)_X & i, j &= 1, \dots, N_v, \\ C_{ijk} &= (\phi_i, (\phi_j \cdot \nabla) \phi_k)_X & i, j, k &= 1, \dots, N_v, \\ H_{ij} &= (\phi_i, \nabla \psi_j)_X & i &= 1, \dots, N_v, \quad j = 1, \dots, N_p. \end{aligned}$$

**Remark 3.5** *It is important to mention that the convective non-linear term in the Navier–Stokes equations is approximated at the reduced order level through a third order tensor  $\mathbf{C}$ . The dimension of the  $\mathbf{C}$  tensor is increasing with the cube of the number of basis functions. This last approach in treating the non-linear term in the momentum equation could potentially increase the computational cost and may lead to high-storage costs when the number of reduced velocity modes grows.*

**Remark 3.6** *An important remark is that the system (3.50) has  $N_v + N_p$  unknowns given by the velocity coefficient  $\mathbf{a}$  and the pressure ones  $\mathbf{b}$  but just  $N_v$  equations. Therefore one must search  $N_p$  additional equations in order to close the reduced system. In the reduced framework, the continuity equation can't be directly exploited because the velocity modes, which are generated with divergence free snapshots, are in turn divergence free up to numerical precision. The available approaches to tackle this problem are either the use of the Poisson equation for pressure [64, 66] or the use of the supremizer stabilization method [8, 61], which consists into the enrichment of the velocity modal space by employing the supremizer modes. These modes are computed, such that the reduced counterpart of the inf-sup condition is fulfilled. The latter approach, usually employed in finite element framework, has been also extended to a FV formulations [64].*

In this work, the supremizer stabilization method has been chosen for this goal. Later on we will comment the stability issue created by a non-fulfilled inf-sup condition for a saddle-point configuration, which requires a supremizer enrichment method. This approach will ensure that velocity POD modes are not all divergence free so one can project the continuity equation onto the space spanned by the POD pressure modes

$$(\psi_i, \nabla \cdot \mathbf{v})_Y = 0 \quad i = 1, \dots, N_p.$$

This will give the following *state reduced system*:

$$\begin{cases} \nu \mathbf{B} \mathbf{a} - \mathbf{a}^T \mathbf{C} \mathbf{a} - \mathbf{H} \mathbf{b} &= \mathbf{0}, \\ \mathbf{P} \mathbf{a} &= \mathbf{0}, \end{cases} \quad (3.51)$$

where the reduced matrix  $\mathbf{P} \in \mathbb{R}^{N_p \times N_v}$  is the matrix associated with the continuity equation. The entries of this matrix are given by:

$$P_{ij} = (\psi_i, \nabla \cdot \phi_j) \quad i = 1, \dots, N_p, \quad j = 1, \dots, N_v.$$

### 3.2.2 Reduced Adjoint Equations

In this subsection, the Galerkin projection of the adjoint equations (1.19)-(1.20)-(1.21) onto the adjoint POD space is obtained with the same procedure employed in the previous Subsection. Also in this case, we employ a supremizer enrichment technique which ensures that adjoint velocity POD modes are not all divergence free. We report the calculation for the inverse problem only with respect to the case where just velocity data  $\mathbf{v}_d$  are observed.

The projection of the adjoint momentum and continuity equations (1.19) reads as follows:

$$\begin{aligned} (\zeta_i, -\nabla \cdot (\nu \nabla \mathbf{w}) + \nabla \mathbf{v} \cdot \mathbf{w} - (\mathbf{v} \cdot \nabla) \mathbf{w} + \nabla q + \mathbf{v})_X &= (\zeta_i, \mathbf{v}_d)_X & i = 1, \dots, N_w, \\ (\eta_i, \nabla \cdot \mathbf{w})_Y &= 0 & i = 1, \dots, N_q. \end{aligned} \quad (3.52)$$

The *adjoint reduced order dynamical system*, resulting from inserting the reduced approximations (3.6)-(3.8) into (3.52), is the following:

$$\begin{cases} \nu \mathbf{D} \mathbf{c} + \mathbf{a}^T \mathbf{E} \mathbf{c} - \mathbf{K} \mathbf{d} - \mathbf{F} \mathbf{a} &= -\mathbf{g}, \\ \mathbf{Q} \mathbf{c} &= \mathbf{0}, \end{cases} \quad (3.53)$$

where  $\mathbf{c}$  and  $\mathbf{d}$  are the vectors of reduced order degrees of freedom for adjoint velocity and pressure, respectively, while,  $\mathbf{D} \in \mathbb{R}^{N_w \times N_w}$ ,  $\mathbf{E} \in \mathbb{R}^{N_w \times N_v \times N_w}$ ,  $\mathbf{F} \in \mathbb{R}^{N_w \times N_v}$ ,  $\mathbf{g} \in \mathbb{R}^{N_w}$ ,  $\mathbf{K} \in \mathbb{R}^{N_w \times N_q}$  and  $\mathbf{Q} \in \mathbb{R}^{N_q \times N_w}$  are either a reduced order vector, matrix or tensor which are computed as follows:

$$\begin{aligned} D_{ij} &= (\zeta_i, \Delta \zeta_j)_X & i, j = 1, \dots, N_w, \\ E_{ijk} &= (\zeta_i, (\phi_j \cdot \nabla) \zeta_k - \nabla \phi_j \cdot \zeta_k)_X & i, k = 1, \dots, N_w, j = 1, \dots, N_v, \\ F_{ij} &= (\zeta_i, \phi_j)_X & i = 1, \dots, N_w, j = 1, \dots, N_v, \\ g_i &= (\zeta_i, \mathbf{v}_d)_X & i = 1, \dots, N_w, \\ K_{ij} &= (\zeta_i, \nabla \eta_j)_X & i = 1, \dots, N_w, j = 1, \dots, N_q, \\ Q_{ij} &= (\eta_i, \nabla \cdot \zeta_j)_Y & i = 1, \dots, N_q, j = 1, \dots, N_w. \end{aligned}$$

For what concerns the case of pressure observed data, instead, by the same procedure, we obtain the following reduced order system for the adjoint equations (1.20):

$$\begin{cases} \nu \mathbf{D} \mathbf{c} + \mathbf{a}^T \mathbf{E} \mathbf{c} - \mathbf{K} \mathbf{d} &= \mathbf{0}, \\ \mathbf{Q} \mathbf{c} - \mathbf{L} \mathbf{b} &= -\mathbf{m}, \end{cases} \quad (3.54)$$

where we introduced the new reduced matrix  $\mathbf{L} \in \mathbb{R}^{N_q \times N_p}$  and reduced vector  $\mathbf{m} \in \mathbb{R}^{N_q}$ . The entries of the two additional matrices are given by:

$$\begin{aligned} L_{ij} &= (\eta_i, \psi_j)_Y & i = 1, \dots, N_q, j = 1, \dots, N_p, \\ m_i &= (\eta_i, p_d)_Y & i = 1, \dots, N_q. \end{aligned}$$

Finally, when both velocity and pressure fields are observed, the reduced form for the adjoint equations (1.21) is the one following:

$$\begin{cases} \nu \mathbf{D} \mathbf{c} + \mathbf{a}^T \mathbf{E} \mathbf{c} - \mathbf{K} \mathbf{d} - \mathbf{F} \mathbf{a} &= -\mathbf{g}, \\ \mathbf{Q} \mathbf{c} - \mathbf{L} \mathbf{b} &= -\mathbf{m}. \end{cases} \quad (3.55)$$

### 3.2.3 Reduced Optimality Equation

We can observe that, we have  $N_v + N_p + N_w + N_q + 1$  unknowns given by the vectors of degrees of freedom  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$  for the state velocity, state pressure, adjoint velocity and adjoint pressure fields, respectively, and the viscosity constant  $\nu$ . Therefore, one more equation is needed in order to close the system. This equation is given by the optimality condition of KKT system.

In our case, when we solve the KKT-system with a segregate method using the Algorithm 1 in FOM, we update the viscosity field  $\nu$  at each iteration, thanks to the solution fields stored from state and adjoint equations. The converged solution  $(\mathbf{v}, p, \nu, \mathbf{w}, q)$ , anyway, satisfies also the optimality condition for the inverse problem, which tries to capture the viscosity constant from observed data, given by

$$\nabla_\nu L(\mathbf{v}, p, \nu, \mathbf{w}, q) = J'(\nu) = 0$$

where the derivative of the functional cost  $J(\nu)$  is computed by (2.11). Then, the *Optimality Condition* for this problem is given by

$$\int_{\Omega} \nabla \mathbf{v} : \nabla \mathbf{w} \, d\Omega = 0. \quad (3.56)$$

In this subsection, we want to compute the reduced order equation for the optimality condition (3.56). From inserting the reduced approximations (3.2)-(3.6) into (3.56), we obtain the *reduced form of optimality condition* as follows:

$$\mathbf{a}^T \mathbf{N} \mathbf{c} = 0, \quad (3.57)$$

where the new reduced matrix  $\mathbf{N} \in \mathbb{R}^{N_v \times N_w}$  is given by

$$N_{ij} = (\nabla \phi_i, \nabla \zeta_j)_X \quad i = 1, \dots, N_v, j = 1, \dots, N_w.$$

## 3.3 Stability issues

The state and adjoint reduced problem, as formulated in (3.51)-(3.53), present stability issues, as analyzed in [8]. In this section, we discuss the case of state problem given by the incompressible Navier-Stokes equations in steady-state. Instead, for the adjoint equations, we have the same issues of direct ones because they share the same structure, also called *saddle-point structure*. For more detail on stability issues for problems with this structure, we refer to [8, 61, 64].

When we use a mixed formulation to solve the Navier-Stokes equations, in order to have a well-posed problem (i.e. no spurious pressure modes) the approximation spaces  $V_h \subset V$  and  $P_h \subset P$  for state velocity and pressure fields, respectively, must fulfill the discrete *Ladyzhenskaya-Brezzi-Babuska (LLB) Inf-Sup condition*

$$\exists \beta > 0 \quad \text{such that} \quad \inf_{p_h \in P_h \setminus \{0\}} \sup_{\mathbf{v}_h \in V_h \setminus \{0\}} \frac{b(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_V \|p_h\|_P} \geq \beta > 0, \quad (3.58)$$

where  $\beta$  is independent from the discretization parameter  $h$ .

The solution spaces  $V$  and  $P$ , defined in (1.13), are equipped with the norm  $\|\cdot\|_V = (\cdot, \cdot)_V^{1/2}$  and  $\|\cdot\|_P = (\cdot, \cdot)_V^{1/2}$ , where  $(\mathbf{v}, \mathbf{w})_V = (\nabla \mathbf{v}, \nabla \mathbf{w})_X$  for  $\mathbf{v}, \mathbf{w} \in \mathbf{V}$  and  $(p, q)_P = (p, q)_Y$ , respectively. At full-order level with a FV method, no attention is paid to this condition because we employ a segregate procedure. Instead, at reduced order level with a Galerkin projection method the LBB condition has to be verified for  $V_{POD}$  and  $P_{POD}$  for a coupled-type solver. The original POD modes (3.31) are a linear combination of the velocity snapshots which are divergence free. Then, the inf-sup condition isn't satisfied. Moreover, the projection of the continuity equation onto the pressure modes is not useful, since the velocity modes have zero divergence up to numerical precision.

To overcome this issue there are many approach in literature. When, we solve in coupled way the saddle-point problems, we want that the inf-sup condition (3.58) is satisfied. We can use a reduced version of segregate solver, already used in full-order formulation. Then, we solve reduced system in decoupled way, in order to avoid saddle point structure, see e.g. [65]. With this approach, we must not have the fulfillment of inf-sup condition because we have no stability issues of this kind.

A simpler method is using only a POD base for velocity. Thus, we neglect the contribution of the gradient pressure in the momentum balance at reduced order level. The pressure field is then a posteriori reconstructed employing the Poisson equation for pressure, as presented in [66]. This choice is justified for some set-up as enclosed flows or in the case of inlet-outlet problems with outlet far from the obstacle, but in many applications the pressure term is needed and can not be neglected.

In ROMs exploiting also the pressure modes two different coupled approaches can be found whether they use only the momentum equation or also the continuity/Poisson equation. For simplicity, in the first approach, it is assumed that velocity and pressure at reduced order level share the same coefficient, i.e.  $\mathbf{a} = \mathbf{b}$ , and we make a Galerkin projection only of the momentum equation (see [44, 32]). This approach shows lack of accuracy for what regards the reconstruction of the pressure field at reduced order level. But, an accurate pressure approximation is imperative as many outputs of interest depend highly on the pressure field. Consequently, stabilization methods have been considered in order to reproduce the pressure field, in an accurate approximation, given by the second approach.

There are two different strategies for pressure stabilization during the resolution of the reduced problem in coupled way. With both procedures, it is assumed that, velocity and pressure are approximated at reduced order level with different coefficient, i.e.  $\mathbf{a} \neq \mathbf{b}$ . These strategies differ in the equation employed for pressure, i.e. the Poisson equation or the continuity equation. The first one is based on a Leray-Helmholtz projection, by exploiting at reduced order level, a Poisson equation for pressure. This approach makes possible the separation of the pressure reduced degrees of freedom from the ones of the velocity. The Poisson equation for pressure can be obtained by taking the divergence of the momentum equation and, then, taking advantage from divergence-free property of velocity field. This method, denoted as PPE-ROM, is presented in [66, 64, 32].

As already mentioned, in this Thesis, we adopt the second strategy which enrich the velocity POD-space by velocity-like modes. We obtain these modes by solving a supremizer problem, associated with each of the pressure modes or snapshots, in order to satisfy

a reduced version of LBB inf-sup condition (3.58). Thus, we enforce the well-posedness of the reduced problem (3.51). By the same argument, we use a supremizer stabilization methods for adjoint reduced momentum and continuity equations. In the following subsection we discuss this method, denoted as SUP-ROM. For more details with finite volume formulation, we refer to [64, 32].

### 3.3.1 Supremizer Enrichment

The *supremizer enrichment approach* allows the usage of the continuity equation at the reduced order level for the state and adjoint equations (1.10)-(1.19). This method ensures the fulfilment of the inf-sup condition (3.58) at reduced order level. Within this approach, the velocity and adjoint velocity supremizer basis functions are computed and added to POD-spaces  $V_{POD}$  and  $W_{POD}$ , respectively. We observe that, the new *supremizer POD-spaces*,  $V_{POD}^{SUP}$  and  $W_{POD}^{SUP}$ , are not any-more composed by only orthogonal basis functions. In fact, the POD is applied separately onto the velocity snapshots and onto the supremizer snapshots. This stabilization method is introduced for finite element formulation in [61, 8, 60, 17, 58] and It has been extended for finite volume formulation in [64].

For the construction of *supremizer problem*, we refer in particular to [60, 64] and, for an extension to a optimal flow control, to [7]. In order to verify the LBB condition for both adjoint and state problems, we introduce the following *supremizer operators*  $\mathcal{T}_{\mathbf{v}} : P_h \rightarrow V_h$  and  $\mathcal{T}_{\mathbf{w}} : Q_h \rightarrow W_h$ , where,  $Q_h \subset P$  and  $W_h \subset V$  denote the approximation spaces of adjoint pressure and velocity fields. These operators are defined with the following Riesz representation:

$$\begin{aligned} (\mathcal{T}_{\mathbf{v}} p_h, \mathbf{v}_h)_V &= b(p_h, \mathbf{v}_h) & \forall \mathbf{v}_h \in V_h, p_h \in P_h, \\ (\mathcal{T}_{\mathbf{w}} q_h, \mathbf{w}_h)_V &= b(q_h, \mathbf{w}_h) & \forall \mathbf{w}_h \in W_h, q_h \in Q_h, \end{aligned} \quad (3.59)$$

respectively, called *state and adjoint supremizer equations*. We note the following property of supremizer operator

$$\begin{aligned} \mathcal{T}_{\mathbf{v}} p_h &= \arg \sup_{\mathbf{v}_h \in V_h \setminus \{\mathbf{0}\}} \frac{b(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_V} & \text{for } p_h \in P_h, \\ \mathcal{T}_{\mathbf{w}} q_h &= \arg \sup_{\mathbf{w}_h \in W_h \setminus \{\mathbf{0}\}} \frac{b(q_h, \mathbf{w}_h)}{\|\mathbf{w}_h\|_V} & \text{for } q_h \in Q_h, \end{aligned} \quad (3.60)$$

because, applying the Riesz representation theorem (see [63]) to  $b(q_h, \cdot) \in V'$  for  $q_h \in Q_h$  and  $b(p_h, \cdot) \in V'$  for  $p_h \in P_h$ , we obtain

$$\begin{aligned} \|\mathcal{T}_{\mathbf{v}} p_h\|_V &= \|b(p_h, \cdot)\|_{V'}, \\ \|\mathcal{T}_{\mathbf{w}} q_h\|_V &= \|b(q_h, \cdot)\|_{V'}, \end{aligned}$$

respectively, where  $V'$  is the dual space of  $V$ . Moreover, by definitions (3.59) it follows that

$$\begin{aligned} \|\mathcal{T}_{\mathbf{v}} p_h\|_V^2 &= b(p_h, \mathcal{T}_{\mathbf{v}} p_h) & \text{for } p_h \in P_h, \\ \|\mathcal{T}_{\mathbf{w}} q_h\|_V^2 &= b(q_h, \mathcal{T}_{\mathbf{w}} q_h) & \text{for } q_h \in Q_h. \end{aligned} \quad (3.61)$$

It is possible to demonstrate [60], thanks to property (3.60) and relation (3.61), that

$$\begin{aligned} \inf_{p_h \in P_h \setminus \{0\}} \sup_{\mathbf{v}_h \in V_h \setminus \{\mathbf{0}\}} \frac{b(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_V \|p_h\|_P} &= \inf_{p_h \in P_h \setminus \{0\}} \frac{b(p_h, \mathcal{T}_{\mathbf{v}} p_h)}{\|\mathcal{T}_{\mathbf{v}} p_h\|_V \|p_h\|_P} = \inf_{p_h \in P_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{v}} p_h\|_V}{\|p_h\|_P}, \\ \inf_{q_h \in Q_h \setminus \{0\}} \sup_{\mathbf{w}_h \in W_h \setminus \{\mathbf{0}\}} \frac{b(q_h, \mathbf{w}_h)}{\|\mathbf{w}_h\|_V \|q_h\|_P} &= \inf_{q_h \in Q_h \setminus \{0\}} \frac{b(q_h, \mathcal{T}_{\mathbf{w}} q_h)}{\|\mathcal{T}_{\mathbf{w}} q_h\|_V \|q_h\|_P} = \inf_{q_h \in Q_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{w}} q_h\|_V}{\|q_h\|_P}. \end{aligned} \quad (3.62)$$

With finite volume discretization, we have a stable truth approximation. Then, there exist  $\beta_v, \beta_w > 0$  such that

$$\begin{aligned} \inf_{p_h \in P_h \setminus \{0\}} \sup_{\mathbf{v}_h \in V_h \setminus \{\mathbf{0}\}} \frac{b(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_V \|p_h\|_P} &= \inf_{p_h \in P_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{v}} p_h\|_V}{\|p_h\|_P} \geq \beta_v > 0, \\ \inf_{q_h \in Q_h \setminus \{0\}} \sup_{\mathbf{w}_h \in W_h \setminus \{\mathbf{0}\}} \frac{b(q_h, \mathbf{w}_h)}{\|\mathbf{w}_h\|_V \|q_h\|_P} &= \inf_{q_h \in Q_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{w}} q_h\|_V}{\|q_h\|_P} \geq \beta_w > 0. \end{aligned} \quad (3.63)$$

Finally, we show that the supremizer operators  $\mathcal{T}_{\mathbf{v}}$  and  $\mathcal{T}_{\mathbf{w}}$  realize the supremum in LBB conditions (3.63) at full-order level.

In the cases of reduced order model, even if the snapshots  $\mathbf{S}^{\mathbf{v}}$  and  $\mathbf{S}^{\mathbf{w}}$  have been obtained by stable numerical methods, there is not guaranty that the original properties of the full order system are preserved after the Galerkin projection onto the reduced basis spaces  $V_{POD}$  and  $W_{POD}$ . To overcome this issue, we add *supremizer basis functions*

$$V^s = \text{span} \left\{ [\sigma_{\mathbf{i}}^{\mathbf{v}}]_{i=1}^{N_v^s} \right\} \quad \text{and} \quad W^s = \text{span} \left\{ [\sigma_{\mathbf{i}}^{\mathbf{w}}]_{i=1}^{N_w^s} \right\},$$

to the reduced basis spaces  $V_{POD}$  and  $W_{POD}$ , respectively, defining the velocity *SUP-POD spaces*

$$V_{POD}^{SUP} = V_{POD} \oplus V^s \quad \text{and} \quad W_{POD}^{SUP} = W_{POD} \oplus W^s.$$

These basis functions are chosen solving a supremizer problem which ensures stability of reduced solution. The supremizer solution  $\mathbf{s}_i^{\mathbf{v}} = \mathcal{T}_{\mathbf{v}} p_i$  and  $\mathbf{s}_i^{\mathbf{w}} = \mathcal{T}_{\mathbf{w}} q_i$ , for each pressure basis  $p_i \in P_h$  and  $q_i \in Q_h$ , can be found solving the *supremizer problems in weak formulation* (3.59). Set  $\mathbf{s}_i^{\mathbf{v}} = \mathcal{T}_{\mathbf{v}} p_i$  for  $p_i \in P_h$  and  $\mathbf{s}_i^{\mathbf{w}} = \mathcal{T}_{\mathbf{w}} q_i$  for  $q_i \in Q_h$ , the weak form of supremizer problems (3.59) are given in the following more explicit form:

Find  $\mathbf{s}_i^{\mathbf{v}} \in V_h$  and  $\mathbf{s}_i^{\mathbf{w}} \in W_h$  such that

$$(\nabla \mathbf{s}_i^{\mathbf{v}}, \nabla \mathbf{v}_h)_X = (\nabla p_i, \mathbf{v}_h)_X \quad \forall \mathbf{v}_h \in V_h \quad \text{and} \quad (\nabla \mathbf{s}_i^{\mathbf{w}}, \nabla \mathbf{w}_h)_X = (\nabla q_i, \mathbf{w}_h)_X \quad \forall \mathbf{w}_h \in W_h.$$

Using integration by part in  $V$ , we obtain the following *strong form of supremizer problems*

$$\begin{cases} \Delta \mathbf{s}_i^{\mathbf{v}} = -\nabla p_i & \text{in } \Omega, \\ \mathbf{s}_i^{\mathbf{v}} = \mathbf{0} & \text{on } \partial\Omega, \end{cases} \quad \begin{cases} \Delta \mathbf{s}_i^{\mathbf{w}} = -\nabla q_i & \text{in } \Omega, \\ \mathbf{s}_i^{\mathbf{w}} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \quad (3.64)$$

The enriched velocity spaces with supremizer solution  $V_{POD}^{SUP}$  and  $W_{POD}^{SUP}$  satisfy a reduced version of inf-sup condition (3.63). In fact, thanks to  $P_{POD} \subset P_h$  and  $Q_{POD} \subset Q_h$ , the

definition of supremizers (3.59), since the velocity space is enriched by supremizers, i.e.  $\mathcal{T}_{\mathbf{v}}p_r \in V_{POD}^{SUP}$  for  $p_r \in P_{POD}$  and  $\mathcal{T}_{\mathbf{w}}q_r \in W_{POD}^{SUP}$  for  $q_r \in Q_{POD}$ , we obtain

$$\inf_{p_r \in P_{POD} \setminus \{0\}} \sup_{\mathbf{v}_r \in V_{POD}^{SUP} \setminus \{0\}} \frac{b(p_r, \mathbf{v}_r)}{\|\mathbf{v}_r\|_V \|p_r\|_P} = \inf_{p_r \in P_{POD} \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{v}}p_r\|_V}{\|p_r\|_P} \geq \quad (3.65)$$

$$\geq \inf_{p_h \in P_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{v}}p_h\|_V}{\|p_h\|_P} = \inf_{p_h \in P_h \setminus \{0\}} \sup_{\mathbf{v}_h \in V_h \setminus \{0\}} \frac{b(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_V \|p_h\|_P} \geq \beta_v > 0, \quad (3.66)$$

$$\inf_{q_r \in Q_{POD} \setminus \{0\}} \sup_{\mathbf{w}_r \in W_{POD}^{SUP} \setminus \{0\}} \frac{b(q_r, \mathbf{w}_r)}{\|\mathbf{w}_r\|_V \|q_r\|_P} = \inf_{q_r \in Q_{POD} \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{w}}q_r\|_V}{\|q_r\|_P} \geq \quad (3.67)$$

$$\geq \inf_{q_h \in Q_h \setminus \{0\}} \frac{\|\mathcal{T}_{\mathbf{w}}q_h\|_V}{\|q_h\|_P} = \inf_{q_h \in Q_h \setminus \{0\}} \sup_{\mathbf{w}_h \in W_h \setminus \{0\}} \frac{b(q_h, \mathbf{w}_h)}{\|\mathbf{w}_h\|_V \|q_h\|_P} \geq \beta_w > 0.$$

We can summarize that the POD spaces enriched with the supremizer basis function,  $V_{POD}^{SUP}$  and  $W_{POD}^{SUP}$ , ensure that the reduced LBB inf-sup conditions (3.66)-(3.67) is fulfilled.

As introduced in [8, 64], there are two different strategies which can be employed to enrich the velocity spaces  $V_{POD}$ ,  $W_{POD}$  and select the supremizer spaces  $V_{POD}^{SUP}$  and  $W_{POD}^{SUP}$ , such that the inf-sup conditions (3.66)-(3.67) is met.

The first one, called the *exact supremizer enrichment*, in which one solves the supremizer problems (3.64) for each pressure mode  $\psi_i$  and  $\eta_j$  obtaining at the end the velocity-like mode  $\mathbf{s}^{\mathbf{v}}(\psi_i)$  and  $\mathbf{s}^{\mathbf{w}}(\eta_j)$  for  $i = 1, \dots, N_p$  and  $j = 1, \dots, N_q$ , respectively. Then, the supremizers  $[\sigma_i^v]_{i=1}^{N_s^v} = [\mathbf{s}^{\mathbf{v}}(\psi_i)]_{i=1}^{N_s^v}$  and  $[\sigma_j^w]_{j=1}^{N_s^w} = [\mathbf{s}^{\mathbf{w}}(\eta_j)]_{j=1}^{N_s^w}$  are added to the velocity basis  $[\phi_i]_{i=1}^{N_v}$  and  $[\zeta_j]_{j=1}^{N_w}$ , respectively, as presented above. Using such an approach, we prove that, the resulting ROM, obtained by the Galerkin projection onto the reduced spaces, is inf-sup stable, as demonstrated above.

In the second approach, called *approximate supremizer enrichment*, the supremizer problems (3.64) are solved for each pressure snapshots  $p_i$  and  $q_j$  for  $i, j = 1, \dots, m$ . Thus, given the supremizers  $\mathbf{s}^{\mathbf{v}}(p_i)$  and  $\mathbf{s}^{\mathbf{w}}(q_j)$  for  $i, j = 1, \dots, m$ , respectively, the snapshots matrices of supremizer are assembled in  $\mathbf{S}_{\mathbf{v}}^s = \{\mathbf{s}^{\mathbf{v}}(p_1), \dots, \mathbf{s}^{\mathbf{v}}(p_m)\} \in \mathbb{R}^{N_v^h \times m}$  and  $\mathbf{S}_{\mathbf{w}}^s = \{\mathbf{s}^{\mathbf{w}}(q_1), \dots, \mathbf{s}^{\mathbf{w}}(q_m)\} \in \mathbb{R}^{N_w^h \times m}$ . Then, a POD is applied on the supremizer snapshots matrices, that yields the supremizer POD modes  $[\sigma_i^v]_{i=1}^{N_s^v}$  and  $[\sigma_i^w]_{i=1}^{N_s^w}$ . These modes are then used to enrich the velocity POD space. This method permits to strongly reduce the online computational cost. Indeed, the supremizer basis functions do not depend on the particular pressure basis functions because they are computed starting directly from the pressure snapshots. However, with such an approach, it is not possible to rigorously show that the inf-sup condition is satisfied. Therefore, we only check it in post-processing.

In this Thesis, we use an exact supremizer enrichment to have a stable solution of state and adjoint reduced systems (3.51)-(3.53) because this satisfies the priori inf-sup condition ad reduced-order level (3.66)-(3.67). Moreover, we compute these modes only once in offline phase with low computational cost. We can summarize this method for our problems with the following Algorithm 4.

---

**Algorithm 4:** Exact Supremizer Enrichment Method.
 

---

**Input:** POD basis functions  $[\phi_i]_{i=1}^{N_v}$ ,  $[\psi_i]_{i=1}^{N_p}$ ,  $[\zeta_i]_{i=1}^{N_w}$  and  $[\eta_i]_{i=1}^{N_q}$  for state velocity, state pressure, adjoint velocity and adjoint pressure, respectively.

1 **for**  $i = 1, \dots, N_p$  **do**

2     Solve

$$\begin{cases} \Delta \mathbf{s}_i^v = -\nabla \psi_i & \text{in } \Omega, \\ \mathbf{s}_i^v = \mathbf{0} & \text{on } \partial\Omega, \end{cases}$$

to obtain the state supremizer  $\sigma_i^v = \mathbf{s}_i^v(\psi_i)$  corresponding to the  $i$ -th pressure basis function  $\psi_i$ ;

3 **end**

4 Add the state supremizer  $[\sigma_i^v]_{i=1}^{N_v^s}$  to the state velocity basis functions  $[\phi_i]_{i=1}^{N_v}$  and set  $N_v^s = N_p$ ;

5 **for**  $i = 1, \dots, N_q$  **do**

6     Solve

$$\begin{cases} \Delta \mathbf{s}_i^w = -\nabla \eta_i & \text{in } \Omega, \\ \mathbf{s}_i^w = \mathbf{0} & \text{on } \partial\Omega, \end{cases}$$

to obtain the adjoint supremizer  $\sigma_i^w = \mathbf{s}_i^w(\eta_i)$  corresponding to the  $i$ -th pressure basis function  $\eta_i$ ;

7 **end**

8 Add the adjoint supremizer  $[\sigma_i^w]_{i=1}^{N_w^s}$  to the adjoint velocity basis functions  $[\zeta_i]_{i=1}^{N_w}$  and set  $N_w^s = N_q$ .

**Output:** State and adjoint supremizer enriched basis functions  $[\phi_i]_{i=1}^{N_v} \cup [\sigma_i^v]_{i=1}^{N_v^s}$  and  $[\zeta_i]_{i=1}^{N_w} \cup [\sigma_i^w]_{i=1}^{N_w^s}$ , respectively.

---

### 3.4 Non-homogeneous Diriclet boundary conditions treatment

The our problem involves non-homogeneous Dirichlet boundary conditions (BCs) on  $\Gamma$  for the state velocity field in Navier-Stokes equations. The goal here is the treatment of the non-homogeneous Dirichlet boundary conditions at reduced order level. This goal is essential to build an accurate model. In literature different approaches to impose the boundary conditions in the ROM can be found. Here, it is explained how the Dirichlet boundary conditions are enforced, at the reduced order level. The methods employed, to enforce non-homogeneous Dirichlet condition, include the penalty method and the lifting function method.

The *penalty method* is used in [32, 29, 44]. This approach consists to weakly impose the boundary conditions, employing a penalty term in the reduced momentum equation. However, this method depends on a penalty parameter whose value is set by a sensitivity analysis. In general, a higher value of penalization factor leads to a stronger enforcement

of the boundary conditions. This might ill-condition the reduced dynamical system. With this method, the POD is done here directly on the non-homogeneous velocity snapshots, unlike with the lifting function method.

In this Thesis, in order to enforce the boundary conditions at the reduced order level, a *lifting function method* is used. This strategy is firstly introduced in [19, 22, 30, 66, 29]. Within this method, before applying the POD, the in-homogeneous boundary conditions are removed from the original snapshots. Using such an approach, it is possible to produce homogeneous basis functions. After, at ROM level, it is possible to deal with any Diricet boundary condition. Then, the reduced problem is solved and the lifting function is added again to the solution. In the following subsection, we present this methodology for the application in our problems.

### 3.4.1 Lifting Function Method

In this subsection, we discuss the enforcement of non-homogeneous Diricet BC thanks the *Lifting Function Method* for our state problem. Let  $\Gamma \subset \partial\Omega$  be the non-homogeneous Dirichlet boundary, the lifting, or control, function method involves the use of the so-called *lifting function*. This function handles the non-homogeneous value  $\mathbf{v}_N = (v_N, 0)$  on the boundary  $\Gamma$ . This is followed by homogenizing the velocity snapshots, subtracting from each snapshots a suitably scaled version of the lifting function.

We can write the velocity field composed by two term, the homogeneous part  $\tilde{\mathbf{v}}$  such that  $\tilde{\mathbf{v}}|_{\partial\Omega} = \mathbf{0}$  and the in-homogeneous one, as follows:

$$\mathbf{v} = \tilde{\mathbf{v}} + \mathbf{v}_N \phi_L,$$

where  $\phi_L \in \mathbb{R}^{N_v^h}$  is the lifting function, which has homogeneous Diricet boundary conditions in all parts of the Diricet boundary, except for the  $x$ -component at  $\Gamma$  where it has unitary value. We observe that, to retain the divergence free property of the snapshots, the lifting function has also to be divergence free.

Each snapshot of the velocity snapshots matrix  $\mathbf{v}_i \in \mathbf{S}^v$  is then modified as:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{v}_N \phi_L \quad \forall i = 1, \dots, m. \quad (3.68)$$

Thus, the new snapshots matrix for velocity is denoted by

$$\mathbf{S}^{\tilde{v}} = \{ \tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_m \} \in \mathbb{R}^{N_v^h \times m},$$

that contains only snapshots with homogeneous boundary conditions. This snapshots matrix will be used for computing the reduced order bases for the velocity POD spaces.

During the online stage, It is sought to approximate the reduced order velocity field as follows:

$$\mathbf{v}(\mathbf{x}; \mu) \approx \mathbf{v}_r(\mathbf{x}; \mu) = v_N \phi_L + \sum_{i=1}^{N_v} a_i(\mu) \phi_i(\mathbf{x}). \quad (3.69)$$

During the projection stage illustrated previously also the above modified approximation of the velocity has to be considered. The projection of KKT equations produces

then some additional terms inside the coupled dynamical system. Since the lifting function is divergence free, it change only the reduced momentum equations for state and adjoint problem and the optimality condition but there are not changes for the continuity equations.

In particular, the reduced state momentum equation (3.50) becomes

$$\nu \left( v_N \mathbf{B}^{BC} + \mathbf{B} \mathbf{a} \right) - \left( v_N \mathbf{C}^{BC} + \mathbf{a}^T \mathbf{C} \right) \mathbf{a} - \mathbf{H} \mathbf{b} = v_N^2 \mathbf{e}^{BC},$$

The additional lifting terms  $\mathbf{B}^{BC} \in \mathbb{R}^{N_v}$ ,  $\mathbf{C}^{BC} \in \mathbb{R}^{N_v \times N_v}$  and  $\mathbf{e} \in \mathbb{R}^{N_v}$ , involved in this equation, are given by

$$\begin{aligned} B_i^{BC} &= (\phi_i, \Delta \phi_L)_X & i &= 1, \dots, N_v, \\ C_{ij}^{BC} &= (\phi_i, (\phi_j \cdot \nabla) \phi_L)_X + (\phi_i, (\phi_L \cdot \nabla) \phi_j)_X & i, j &= 1, \dots, N_v, \\ e_i^{BC} &= (\phi_i, (\phi_L \cdot \nabla) \phi_L)_X & i &= 1, \dots, N_v. \end{aligned}$$

With regard to the adjoint equations, the new reduced approximation of velocity (3.69) changes the adjoint momentum equations for inverse problem with only pressure data (3.54) as follows:

$$\nu \mathbf{D} \mathbf{c} + \left( v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E} \right) \mathbf{c} - \mathbf{K} \mathbf{d} = \mathbf{0},$$

where the boundary matrix  $\mathbf{E}^{BC} \in \mathbb{R}^{N_w \times N_w}$  is computed by

$$E_{ij}^{BC} = (\zeta_i, (\phi_L \cdot \nabla) \zeta_j - \nabla \phi_L \cdot \zeta_j)_X \quad i, j = 1, \dots, N_w.$$

Instead, for the reduced adjoint problem with velocity data (3.53)-(3.55), the adjoint momentum equation is modified as

$$\nu \mathbf{D} \mathbf{c} + \left( v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E} \right) \mathbf{c} - \mathbf{K} \mathbf{d} - \mathbf{F} \mathbf{a} = v_N \mathbf{f}^{BC} - \mathbf{g},$$

where the additional vector  $\mathbf{f}^{BC} \in \mathbb{R}^{N_w}$  is calculated as follows

$$f_i^{BC} = (\zeta_i, \phi_L)_X \quad i = 1, \dots, N_w.$$

Finally, the reduced optimality condition (3.57) is modified as below

$$\mathbf{a}^T \mathbf{N} \mathbf{c} + v_N \mathbf{n}^{BC} \cdot \mathbf{c} = 0,$$

where, the boundary vector  $\mathbf{n}^{BC} \in \mathbb{R}^{N_w}$  is given by

$$n_i^{BC} = (\nabla \phi_L, \nabla \zeta_i)_X \quad i = 1, \dots, N_w.$$

**Remark 3.7** *The way of choosing a suitable lifting function is problem dependent. For example, in the case of the reduction of unsteady non-parametrized cases, where reduction aims to reproduce time snapshots and potentially extrapolate in time, a possible choice of the lifting function could be the average of the offline velocity snapshots (see i.e. [66]). A more general approach, for the generation of appropriate lifting functions, is to solve linear potential flow problems with a unitary boundary condition for each non-homogeneous*

boundary condition to be set (see i.e. [30]). These problems are steady ones, therefore, an iterative procedure with a tentative velocity field is carried out till convergence. While solving each of these potential flows, the value of the starting guess velocity field at the Dirichlet boundary has to be zero everywhere except for one scalar entity, where the lifting function is sought. The converged velocity field will be considered as the lifting function corresponding to the non-homogeneous boundary condition at the aforementioned entity. Besides the requirement of having unitary boundary condition, the lifting functions have to be divergence free fields.

□

Now, we can summarize the Galerkin projection of KKT system onto the velocity homogeneous SUP-POD spaces by writing the *reduced order KKT systems* for our test case. In particular, to reconstruct the viscosity constant with only velocity data, we obtain the following reduced KKT system

$$\begin{cases} \nu (v_N \mathbf{B}^{BC} + \mathbf{B}\mathbf{a}) - (v_N \mathbf{C}^{BC} + \mathbf{a}^T \mathbf{C}) \mathbf{a} - \mathbf{H}\mathbf{b} & = v_N^2 \mathbf{e}^{BC}, \\ \mathbf{P}\mathbf{a} & = \mathbf{0}, \\ \mathbf{a}^T \mathbf{N}\mathbf{c} + v_N \mathbf{n}^{BC} \cdot \mathbf{c} & = 0, \\ \nu \mathbf{D}\mathbf{c} + (v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E}) \mathbf{c} - \mathbf{K}\mathbf{d} - \mathbf{F}\mathbf{a} & = v_N \mathbf{f}^{BC} - \mathbf{g}, \\ \mathbf{Q}\mathbf{c} & = \mathbf{0}. \end{cases} \quad (3.70)$$

Instead, for the inverse problem with only pressure observed, we have the following system

$$\begin{cases} \nu (v_N \mathbf{B}^{BC} + \mathbf{B}\mathbf{a}) - (v_N \mathbf{C}^{BC} + \mathbf{a}^T \mathbf{C}) \mathbf{a} - \mathbf{H}\mathbf{b} & = v_N^2 \mathbf{e}^{BC}, \\ \mathbf{P}\mathbf{a} & = \mathbf{0}, \\ \mathbf{a}^T \mathbf{N}\mathbf{c} + v_N \mathbf{n}^{BC} \cdot \mathbf{c} & = 0, \\ \nu \mathbf{D}\mathbf{c} + (v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E}) \mathbf{c} - \mathbf{K}\mathbf{d} & = \mathbf{0}, \\ \mathbf{Q}\mathbf{c} - \mathbf{L}\mathbf{b} & = -\mathbf{m}. \end{cases} \quad (3.71)$$

Otherwise, if we observe both the velocity and the pressure fields, the reduced KKT system changes as follow:

$$\begin{cases} \nu (v_N \mathbf{B}^{BC} + \mathbf{B}\mathbf{a}) - (v_N \mathbf{C}^{BC} + \mathbf{a}^T \mathbf{C}) \mathbf{a} - \mathbf{H}\mathbf{b} & = v_N^2 \mathbf{e}^{BC}, \\ \mathbf{P}\mathbf{a} & = \mathbf{0}, \\ \mathbf{a}^T \mathbf{N}\mathbf{c} + v_N \mathbf{n}^{BC} \cdot \mathbf{c} & = 0, \\ \nu \mathbf{D}\mathbf{c} + (v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E}) \mathbf{c} - \mathbf{K}\mathbf{d} - \mathbf{F}\mathbf{a} & = v_N \mathbf{f}^{BC} - \mathbf{g}, \\ \mathbf{Q}\mathbf{c} - \mathbf{L}\mathbf{b} & = -\mathbf{m}. \end{cases} \quad (3.72)$$

There are many ways to solve the previous non-linear systems of equations (3.53)-(3.54)-(3.55) with unknowns  $\mathbf{a}$ ,  $\nu$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$ , in a coupled approach or in segregate one.

In the first method, we solve the non linear KKT system through *one-shot* approach, which aims at solving the discretized systems arising from the optimality conditions in a monolithic way, instead than pursuing a partitioned scheme, like in most common iterative procedures. We directly solve the system for all the unknown variables, for example using a Newton-Raphson procedure and its variants or other non linear solver for systems. For more detail on this methods, we refer to [11, 46, 49].

Instead, with a *segregate* procedure, we decouple the KKT system and update the control  $\nu$  thanks a iterative method. For example, with gradient descendent methods, we use the derivative of the cost functional  $J'(\nu)$  at each iteration, which depend on state and adjoint variables, to update the control. We remark that, the optimality equation, depending on the solution of state and adjoint equations, is satisfied only when the iterative procedure reaches convergence. For more detail on this procedure, we refer to [53, 58, 14].

Once the system is solved, it is possible to reconstruct the state and adjoint velocity and pressure fields using the values of the  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$  vectors and the fields approximation as presented in (3.69)-(3.4)-(3.6)-(3.8).

### 3.5 Reduced Order Non-linear Projected Conjugate Gradient Method

In this Section, we discuss a reduced version of optimization solver used in full-order approximation. For the numerical solution of optimization problems under PDEs constraints in decoupled way, two different paradigms can be adopted, both relying on iterative procedures [47]: *optimize-then-reduce* or *reduce-then-optimize*. With the optimize-then-reduce approach, we first apply the iterative method to the (continuous) system of optimality conditions, then we discretize the various steps of the algorithm at reduced order level. Instead, with the second approach, we first discretize our optimal control problem with reduced order model, then we apply an iterative algorithm to solve the discrete version of the system of optimality conditions. For our test case, we follow the first procedure.

As explained in Chapter 2, all high fidelity solutions are obtained by the employment of a segregated procedure given by non-linear projected gradient conjugate method in Algorithm 1. Summarizing, at full order level, we iterate state and adjoint system updating the viscosity parameter, with conjugate gradient direction, until convergence is reached. We rely on Galerkin projection for the construction of the reduced order model.

It is clear that the resulting reduced KKT systems (3.70)-(3.71)-(3.72) can be solved by the use of, e.g., a Newton method or any other kind of non-linear iterative solver. With this method, we solve the reduced KKT system in coupled way. But, we observe that the reduced coupled KKT system have a saddle-point structure also at reduced order level, which leads stability issues [52, 68].

For this reason, some efforts have been spent to develop the following algorithm that emulates the steps of the Full-Order Non-linear Projected Conjugate Gradient Method in order to avoid this saddle-point structure. Given an initial guess  $\nu^0$ , for step  $n > 0$  we

update the viscosity constant, as full order model, by

$$\nu^{n+1} = \pi_{[\varepsilon, +\infty)}(\nu^n + \tau^n d^n), \quad (3.73)$$

where  $d^n$  is the conjugate gradient descent direction given by (2.2) and  $\tau^n$  the step-length, that optimize the functional cost in the descent direction which depend on solution of sensitivity problem. Moreover, the function  $\pi_{[\varepsilon, +\infty)}$  projects the new viscosity value in the range  $[\varepsilon, +\infty)$ .

The derivative of cost functional with respect to the viscosity parameter (2.11), at iteration  $n > 0$ , can be computed using reduced coefficient, as follow

$$J'(\nu) = \mathbf{a}(\nu)^T \mathbf{N} \mathbf{c}(\nu) + v_N \mathbf{n}^{BC} \cdot \mathbf{c}(\nu). \quad (3.74)$$

We observe that, with this iterative procedure, the derivative  $J'(\nu)$  depends on the modal coefficient of direct and adjoint velocity fields,  $\mathbf{a}(\nu)$  and  $\mathbf{c}(\nu)$ , respectively. Thus, we must solve the reduced direct and adjoint systems.

At first, we compute the state velocity and pressure coefficients, i.e.  $\mathbf{a}(\nu^n)$  and  $\mathbf{b}(\nu^n)$ , solving for  $\nu = \nu^n$  the *reduced direct non-linear system* in a coupled way. As already introduced, this non-linear system is given by

$$\mathcal{F}(\mathbf{x}) = \mathbf{0},$$

with vector of unknown  $\mathbf{y} = [a_L; \mathbf{a}; \mathbf{b}] \in \mathbb{R}^{N_v + N_p + 1}$  ( $a_L$  is the known coefficient of lifting function, i.e.  $a_L = v_N$ ) and the multivariate function  $\mathcal{F}$  defined as follow

$$\mathcal{F}(\mathbf{y}) = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \\ (\nu \mathbf{B}^{BC} - a_L \mathbf{e}^{BC}) & (\nu \mathbf{B} - a_L \mathbf{C}^{BC} - \mathbf{a}^T \mathbf{C}) & -\mathbf{H} \\ \mathbf{0} & \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} a_L \\ \mathbf{a} \\ \mathbf{b} \end{bmatrix} - \begin{bmatrix} v_N \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \quad (3.75)$$

To solve this non-linear system, we use the *Powell's dogleg method*, also called *Powell's hybrid method*, which attempts to minimize the sum of the squares of the function values. It does this using a combination of Newton's method and the steepest descent method. This is a so-called *trust region method*. This means that every step moves the current point to within a finite region. This makes the method more stable than Newton's method. For more details, we refer to [57].

After, we solve the *reduced adjoint linear system* to store the adjoint velocity coefficient  $\mathbf{c}(\nu^n)$ . For example, for inverse problem with only velocity data, we solve the adjoint linear system with state velocity coefficient given by solving the previous direct problem for  $\nu = \nu^n$ , i.e.  $\mathbf{a} = \mathbf{a}(\nu^n)$ . In this case, the adjoint linear system is written as follow

$$\mathfrak{A} \mathbf{x} = \mathbf{b}$$

where, the unknown  $\mathbf{x} = [\mathbf{c}; \mathbf{d}] \in \mathbb{R}^{N_w + N_q}$ , the matrix  $\mathfrak{A} \in \mathbb{R}^{(N_w + N_q) \times (N_w + N_q)}$  and the right hand side  $\mathbf{b} \in \mathbb{R}^{N_w + N_q}$  is given by

$$\mathfrak{A} = \begin{bmatrix} (\nu \mathbf{D} + v_N \mathbf{E}^{BC} + \mathbf{a}^T \mathbf{E}) & -\mathbf{k} \\ \mathbf{Q} & \mathbf{0} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} v_N \mathbf{f}^{BC} + \mathbf{F} \mathbf{a} - \mathbf{g} \\ \mathbf{0} \end{bmatrix}. \quad (3.76)$$

To solve this linear system, we use a LU decomposition of matrix  $\mathfrak{A}$  with complete pivoting. For more detail on this solver, we refer to [49].

Now, stored  $\mathbf{a}(\nu^n)$  and  $\mathbf{c}(\nu^n)$ , we can compute the gradient  $J'(\nu^n)$  at iteration  $n > 0$  with (3.74) to update the conjugate gradient descent direction  $d^n$  with formula (2.2). After, we must compute the step-length  $\tau^n$  which optimize the cost functional in this direction  $d^n$ . Firstly, we re-write the cost functional  $J(\nu)$  with reduced coefficient in the following way

$$J_1(\nu) = \frac{1}{2} \left( \mathbf{a}(\nu)^T \left( \mathbf{M}\mathbf{a}(\nu) + 2v_N \mathbf{M}^{\mathbf{BC}} - 2\mathbf{a}^{\mathbf{d}} \right) + \|v_n \phi_L - \mathbf{v}_d\|_X^2 \right), \quad (3.77)$$

$$(3.78)$$

$$J_2(\nu) = \frac{1}{2} \left( \mathbf{b}(\nu)^T \left( \mathbf{R}\mathbf{b}(\nu) - 2\mathbf{b}^{\mathbf{d}} \right) + \|p_d\|_Y^2 \right), \quad (3.79)$$

$$J_3(\nu) = J_1(\nu) + J_2(\nu), \quad (3.80)$$

where the additional terms  $\mathbf{M} \in \mathbb{R}^{N_v \times N_v}$ ,  $\mathbf{M}^{\mathbf{BC}}$ ,  $\mathbf{a}^{\mathbf{d}} \in \mathbb{R}^{N_v}$ ,  $\mathbf{R} \in \mathbb{R}^{N_p \times N_p}$  and  $\mathbf{b}^{\mathbf{d}} \in \mathbb{R}^{N_p}$  are given by

$$\begin{aligned} M_{ij} &= (\phi_i, \phi_j)_X & i, j &= 1, \dots, N_v, \\ M_i^{\mathbf{BC}} &= (\phi_i, \phi_L)_X & i &= 1, \dots, N_v, \\ a_i^{\mathbf{d}} &= (\phi_i, \mathbf{v}_d)_X & i &= 1, \dots, N_v, \\ R_{ij} &= (\psi_i, \psi_j)_Y & i, j &= 1, \dots, N_p, \\ b_i^{\mathbf{d}} &= (\psi_i, p_d)_Y & i &= 1, \dots, N_p. \end{aligned}$$

For iteration  $n > 0$ , to compute the optimum step-length  $\tau^n$ , we first write the cost functional for  $\nu = \nu^n + \tau d^n$  and optimize it with respect to  $\tau$ . In particular, we analyse the case with only velocity data observed. In this case, using the reduced coefficients, we have

$$J_1(\nu^n + \tau d^n) = \frac{1}{2} \left( \mathbf{a}(\nu^n + \tau d^n)^T \left( \mathbf{M}\mathbf{a}(\nu^n + \tau d^n) + 2v_N \mathbf{M}^{\mathbf{BC}} - 2\mathbf{a}^{\mathbf{d}} \right) + \|v_n \phi_L - \mathbf{v}_d\|_X^2 \right). \quad (3.81)$$

To solve this sub-optimal problem, we must analyse how a perturbation on viscosity constant  $\nu \rightarrow \nu + \delta\nu$  cause a variation on state velocity coefficient  $\mathbf{a}(\nu) \rightarrow \mathbf{a}(\nu + \delta\nu)$ .

As already mentioned in Subsection 1.5.2, a perturbation on viscosity  $\nu \rightarrow \nu + \delta\nu$  generate a variation on velocity field,  $\mathbf{v} \rightarrow \mathbf{v} + \delta\mathbf{v}$ , and in the pressure field,  $p \rightarrow p + \delta p$ . At reduced order level, it means a perturbation on corresponding reduced coefficients,  $[v_N; \mathbf{a}] \rightarrow [v_N; \mathbf{a}] + [0; \delta\mathbf{a}]$  and  $\mathbf{b} \rightarrow \mathbf{b} + \delta\mathbf{b}$ . Then, subtracting the reduced direct problem for  $\nu$  from the same problem with viscosity  $\nu + \delta\nu$  and neglecting the second order terms, we obtain the following *reduced sensitivity problem*

$$\begin{cases} \nu \mathbf{B} \delta \mathbf{a} - (v_N \mathbf{C}^{\mathbf{BC}} + \mathbf{a}^T \mathbf{C}) \delta \mathbf{a} - \delta \mathbf{a}^T \mathbf{C} \mathbf{a} - \mathbf{H} \delta \mathbf{b} & = \mathbf{0}, \\ \mathbf{P} \delta \mathbf{a} & = \mathbf{0}. \end{cases} \quad (3.82)$$

We observe that this problem is linear. Then, it can be solved thanks to LU decomposition with complete pivoting as adjoint problem. Indeed, we can write the *reduced linear*

sensitivity system in the following matrix form

$$\begin{bmatrix} (\nu \mathbf{B} - v_N \mathbf{C}^{BC} - \mathbf{a}^T \mathbf{C} - (\mathbf{C}\mathbf{a})^T) & -\mathbf{H} \\ \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \mathbf{a} \\ \delta \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \quad (3.83)$$

Then, we can observe that  $\mathbf{a}(\nu + \delta\nu) = \mathbf{a}(\nu) + \delta\mathbf{a}(\nu, \mathbf{a}(\nu), \delta\nu)$  and  $\mathbf{b}(\nu + \delta\nu) = \mathbf{b}(\nu) + \delta\mathbf{b}(\nu, \mathbf{a}(\nu), \delta\nu)$ . Moreover,  $\delta\mathbf{a}$  and  $\delta\mathbf{b}$  are linear respect to the control variation  $\delta\nu$ .

Now, we rewrite the cost functional (3.81), as follow

$$\begin{aligned} J_1(\nu^n + \tau d^n) &= \frac{1}{2} \left( (\mathbf{a}(\nu^n) + \tau \delta\mathbf{a}(\nu^n))^T \left( \mathbf{M}(\mathbf{a}(\nu^n) + \tau \delta\mathbf{a}(\nu^n)) + 2v_N \mathbf{M}^{BC} - 2\mathbf{a}^d \right) + \right. \\ &+ \left. \|v_n \phi_L - \mathbf{v}_d\|_X^2 \right) = J_1(\nu^n) + \tau \delta\mathbf{a}(\nu^n)^T \left( \mathbf{M}\mathbf{a}(\nu^n) + v_N \mathbf{M}^{BC} - \mathbf{a}^d \right) + \frac{\tau^2}{2} \delta\mathbf{a}(\nu^n)^T \mathbf{M} \delta\mathbf{a}(\nu^n), \end{aligned} \quad (3.84)$$

where  $\delta\mathbf{a}(\nu^n)$  is pre-computed by the reduced linear sensitivity system for  $\nu = \nu^n$  and  $\delta\nu = d^n$ . Then the optimal step-length  $\tau_1^n$  is given by

$$\tau_1^n = - \frac{\delta\mathbf{a}(\nu^n)^T \left( \mathbf{M}\mathbf{a}(\nu^n) + v_N \mathbf{M}^{BC} - \mathbf{a}^d \right)}{\delta\mathbf{a}(\nu^n)^T \mathbf{M} \delta\mathbf{a}(\nu^n)}. \quad (3.85)$$

Instead, in the other cases with different observed data, we can obtain the following formulas, with the same arguments,

$$\tau_2^n = - \frac{\delta\mathbf{b}(\nu^n)^T \left( \mathbf{R}\mathbf{b}(\nu^n) - \mathbf{b}^d \right)}{\delta\mathbf{b}(\nu^n)^T \mathbf{R} \delta\mathbf{b}(\nu^n)}, \quad (3.86)$$

$$\tau_3^n = - \frac{\delta\mathbf{a}(\nu^n)^T \left( \mathbf{M}\mathbf{a}(\nu^n) + v_N \mathbf{M}^{BC} - \mathbf{a}^d \right) + \delta\mathbf{b}(\nu^n)^T \left( \mathbf{R}\mathbf{b}(\nu^n) - \mathbf{b}^d \right)}{\delta\mathbf{a}(\nu^n)^T \mathbf{M} \delta\mathbf{a}(\nu^n) + \delta\mathbf{b}(\nu^n)^T \mathbf{R} \delta\mathbf{b}(\nu^n)}. \quad (3.87)$$

We remark that, these formula is the same of FOM, but using the reduced coefficient vectors.

In the particular case of only velocity field observed, we can summarize the *Reduced Non-linear Projected Conjugate Gradient Method* in Algorithm 5, as in FOM.

With the other observed data, we have the same algorithm except for different adjoint equations and formula of the optimal step length.

### 3.6 Offline-Online Decomposition

Now, in this Section, we focus on some crucial assumptions one has to guarantee, in order to rely on an efficient application of the ROMs. Indeed, one of the main goals of reduced approaches is to achieve a rapid solution of the problem at hand. This aim can be reached assuming that the ROMs can be divided in two stages:

- an *offline stage*, where the snapshots are manipulated and the basis functions are built. Here, all the quantities that are parameter independent are pre-computed and stored. This stage is performed only once; this stage is summarized with the following Algorithm 6;

---

**Algorithm 5:** Reduced Non-linear Projected Conjugate Gradient Method.

---

**Input:** Choose  $\nu^0 \geq \varepsilon$ ,  $n_{max}$  and  $J_{tol}$ , and set  $n = 0$ .

```

1 while  $n \leq n_{max}$  do
2   Solve the reduced direct problem (3.75) with  $\nu = \nu^n$  to store  $\mathbf{a}(\nu^n)$ ;
3   Compute cost function  $J(\nu^n)$  by (3.78);
4   if  $J(\nu^n) < J_{tol}$  then
5     | Stop;
6   end
7   Solve the reduced adjoint problem (3.76) with  $\nu = \nu^n$  and  $\mathbf{a} = \mathbf{a}(\nu^n)$  to store
    $\mathbf{c}(\nu^n)$ ;
8   Compute the derivative  $J'(\nu^n)$  by (3.74);
9   if  $n > 0$  then
10    | Calculate the conjugate coefficient,  $\beta^n$ , with the formula (2.3);
11  end
12  Compute the search direction,  $d^n$ , by (2.2);
13  Solve the reduced sensitivity problem (3.83) with  $\nu = \nu^n$ ,  $\delta\nu = d^n$  and
    $\mathbf{a} = \mathbf{a}(\nu^n)$  to store  $\delta\mathbf{a}(\nu^n, \mathbf{a}(\nu^n), d^n)$ ;
14  Calculate the optimal step-size,  $\tau^n$ , with the formula (3.85);
15  Update the viscosity field  $\nu^n$  by (3.73);
16   $n = n + 1$ ;
17 end
18 return  $\nu^n$ .

```

---

- an *online stage*, where the parameter-dependent quantities are evaluated and the optimality system is assembled and solved. This stage does not depend on the high fidelity dimension and assures the solution of the system in a smaller amount of time with respect to the standard finite volume approach. This phase is given by the following Algorithm 7.

We remark that, the parameter in our test case is the wanted viscosity constant, i.e.  $\mu = \nu_d$ .

**Algorithm 6:** The offline procedure.

**Input:** An abstract truth model with segregate solver given by Algorithm 1 and finite volume approximation; A minimum relative information content  $\mathcal{E}$ ; A discrete training set  $\mathbb{P}_m = \{ \mu_1, \dots, \mu_m \} \subset \mathbb{P}$ .

- 1 **for**  $i = 1, \dots, m$  **do**
- 2     Compute  $(\mathbf{v}(\mu_i), p(\mu_i), \nu(\mu_i), \mathbf{w}(\mu_i), q(\mu_i))$ , as the full-order finite volume solution with Non-linear Projected Conjugate Gradient optimization solver 1;
- 3 **end**
- 4 Set the snapshots matrices  $\mathbf{S}^v$ ,  $\mathbf{S}^p$ ,  $\mathbf{S}^w$  and  $\mathbf{S}^q$  given by the corresponding full-order solutions;
- 5 Evaluate the lifting function  $\phi_L$ ;
- 6 Set the homogeneous snapshots matrix  $\mathbf{S}^{\tilde{v}}$  for the state velocity from the inhomogeneous one  $\mathbf{S}^v$  thanks to the formula (3.68);
- 7 Compute the basis functions  $[\phi_i]_{i=1}^{N_v}$ ,  $[\psi_i]_{i=1}^{N_p}$ ,  $[\zeta_i]_{i=1}^{N_w}$  and  $[\eta_i]_{i=1}^{N_q}$  thanks to POD Algorithm 3 with relative information content  $\mathcal{E}$ ;
- 8 Enlarge state and adjoint velocity POD spaces with exact supremizer enrichment methods given by Algorithm 4;
- 9 Evaluate the reduced matrices and vectors of the KKT reduced system except for the vector given by Galerkin projection of observed data onto the corresponding POD space (i.e.  $\mathbf{g}$ ,  $\mathbf{a}_d$  for (3.70),  $\mathbf{m}$ ,  $\mathbf{b}_d$  for (3.71) and  $\mathbf{g}$ ,  $\mathbf{m}$ ,  $\mathbf{a}_d$ ,  $\mathbf{b}_d$  for (3.72)) because these are parameter-dependent.

**Output:** A reduced basis model based on the POD basis space with supremizer enrichment and lifting function treatment of non-homogeneous Diriclet boundary condition.

**Algorithm 7:** The online procedure.

**Input:** A reduced order KKT system (i.e. (3.70),(3.71) or (3.72) depending on the different observed data); The pre-computed parameter-independent reduced matrices and vectors; The reduced approximation of state and adjoint solution fields  $(\mathbf{v}_r, p_r)$  and  $(\mathbf{w}_r, q_r)$  by (3.69)-(3.4)-(3.6)-(3.8); A parameter  $\mu \in \mathbb{P}$ .

- 1 Evaluate the parameter-dependent vectors which is different depending on observed data( i.e.  $\mathbf{g}(\mu)$ ,  $\mathbf{a}_d(\mu)$  for only velocity data,  $\mathbf{m}(\mu)$ ,  $\mathbf{b}_d(\mu)$  for only pressure data and  $\mathbf{g}(\mu)$ ,  $\mathbf{m}(\mu)$ ,  $\mathbf{a}_d(\mu)$ ,  $\mathbf{b}_d(\mu)$  for both);
- 2 Solve the corresponding reduced KKT system with Reduced Non-linear Projected Conjugate Gradient Algorithm 5 to store the reduced solution  $(\mathbf{a}(\mu), \mathbf{b}(\mu), \nu(\mu), \mathbf{c}(\mu), \mathbf{d}(\mu))$ ;
- 3 Use the reduced degree of freedom  $\mathbf{a}(\mu)$ ,  $\mathbf{b}(\mu)$ ,  $\mathbf{c}(\mu)$  and  $\mathbf{d}(\mu)$  to reconstruct the reduced approximations of state and adjoint fields with (3.69)-(3.4)-(3.6)-(3.8).

**Output:** The reduced solution approximations  $(\mathbf{v}_r(\mu), p_r(\mu), \nu(\mu), \mathbf{w}_r(\mu), q_r(\mu))$ .

## Chapter 4

# Numerical Experiments

This final chapter contains the results related to the numerical simulations obtained with full-order and reduced-order approximations for our test case, as already introduced in Subsection 1.5.2.

As already introduced in Subsection 1.5.1, the direct problem is a incompressible steady-state flow problem in a two-dimensional square domain  $\Omega$ . Let the square domain  $\Omega = (0,1) \times (0,1)$  as in Figure 1.1. Let  $\partial\Omega$  be the boundary of the domain  $\Omega$ . Then, the different boundaries of the domain to be considered are:

- the top wall  $\Gamma := \{ \mathbf{x} \in \partial\Omega \mid \mathbf{x} = (x,1) \}$ ;
- the other boundaries  $\partial\Omega \setminus \Gamma := \{ \mathbf{x} \in \partial\Omega \mid \mathbf{x} = (0, y), \text{ or, } \mathbf{x} = (x,0), \text{ or, } \mathbf{x} = (1, y) \}$ ;

We impose a known velocity  $\mathbf{v}_N = (1,0)$  in Dirichlet sense at the top wall  $\Gamma$ . Instead, in the other walls  $\partial\Omega \setminus \Gamma$ , we assume no-slip conditions. For simplicity, we suppose that the viscosity is uniform in the domain, i.e.  $\nu \in U = [\varepsilon, +\infty)$  where  $\varepsilon > 0$ , and no body forces are present, i.e.  $\mathbf{g} = \mathbf{0}$ . As already mentioned,  $\varepsilon$  is a small positive constant in order to avoid the turbulence effects in the direct problem. We set  $\varepsilon = 7.5 \times 10^{-4}$ .

Summarizing, we consider a slight modification of the state equations (1.10) and the boundaries conditions (BCs) (1.11) that does not change its essential aspects. Then, the mathematical formulation of the direct problem is given by the following Problem 4.1.

**Problem 4.1** Find  $(\mathbf{v}, p)$  such that

$$\begin{cases} -\nu\Delta\mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p = \mathbf{0}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{v} = 0, & \text{in } \Omega, \end{cases} \quad (4.1)$$

with BCs

$$\begin{cases} \mathbf{v} = \mathbf{v}_N, & \text{on } \Gamma, \\ \mathbf{v} = \mathbf{0}, & \text{on } \partial\Omega \setminus \Gamma. \end{cases} \quad (4.2)$$

The direct problem does not have an analytical solution. Then, we assume that the direct problem is well solved and focus our attention on the solution of the inverse problem.

The test case is the viscosity parameter identification from the observed data in a lid-driven cavity flow. By the observations on the solutions of direct problem in all the domain  $\Omega$ , we test the different inverse problem solution methodologies discussed in Chapters 2 - 3.

We begin this chapter by presenting and discussing the results obtained by using the full-order version of the inverse solver, as already introduced in Chapter 2.

In the second part of the present chapter, we test the POD-Galerkin method, as previously developed in Chapter 3, to improve computational cost and CPU time with a reasonable accuracy respect to the full order case.

For each model we will describe the implementation aspects.

## 4.1 Full Order Approximations

In this section, we will show the numerical results for the application of full-order model described in Chapter 2. Three settings are considered which differ on the basis of observed fields:

1. only velocity data;
2. only pressure data;
3. both velocity and pressure data.

At first, we present the results obtained having as input data to the inverse problem only velocity, only pressure or both. Then, we compare the results obtained with the three different settings also varying the wanted viscosity.

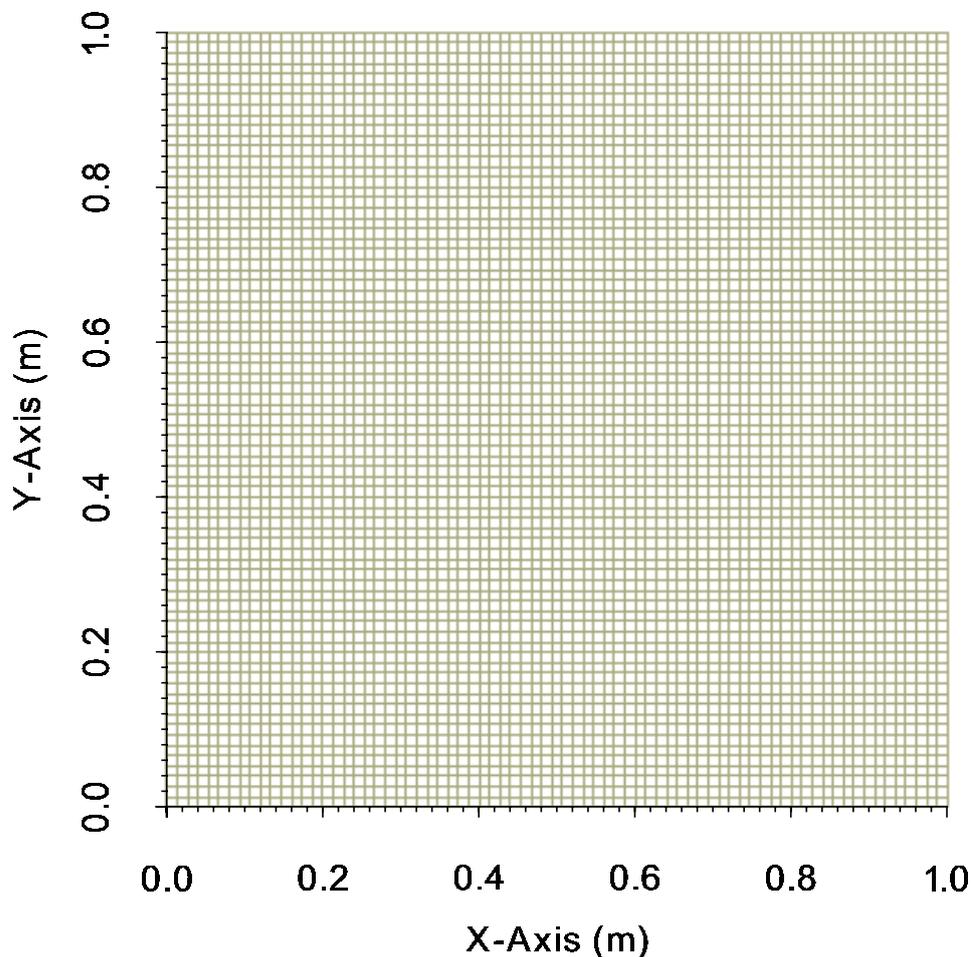
### 4.1.1 Description of the full order implementation

At first, we discuss the numerical solver of direct problem 4.1. Due to its simplicity, the domain  $\Omega$  is discretized by uniform, structured, orthogonal, hexahedral meshes. In all tests, we use the same number of edges for the two axes. In Table 4.1, we summarize the properties of the mesh used in the space discretization of the domain  $\Omega$ . While, the Figure 4.1) shows us the computational domain obtained.

**Table 4.1:** Mesh properties with FV discretization.

Property	Value
Type	Uniform, structured, orthogonal
Cell shapes	hexahedron
Dimension	$75 \times 75$
Degree of freedom	5625

With respect to the FV scheme, since we have a structured orthogonal grid, no correction is needed when computing the gradient normal to the cells faces. Then, we use linear method to interpolate the values from cell centers to face centers. The resulting scheme is second order in accuracy. As already introduced in Section 2.2, from the finite volume



**Figure 4.1:** Computational mesh.

discretization of direct problem, we obtain the discretized system (2.14). We find the FV approximation of the velocity and the pressure fields by using the SIMPLE Algorithm 2.

To solve the inverse problem, we employ the Non-linear Projected Conjugate Gradient Algorithm 1 developed in Section 2.1. Briefly summarizing, we find the viscosity constant minimizing the functional cost  $J$  given by the norm of the error between the observed data and the solution fields of the direct problem varying the viscosity. We investigate the three settings of observed data, as previous introduced. This three set-up distinguish on functional cost to be minimize  $J_1, J_2, J_3$ , as previous defined in (1.16), (1.17) and (1.18), respectively. With the CG method, we minimize the cost function  $J$  following an iterative procedure, in which, we update the viscosity parameter thanks the conjugate

gradient direction  $d$  depends in particular on derivative of cost function  $J'$ , as previous defined in (2.11). Then, to compute this direction  $d$  dependent on state and adjoint velocity, we must numerically cascade solve the state 4.1 and the adjoint problem, which is different depending on the chosen setting, i.e. Problem 1.8, Problem 1.9 and Problem 1.10, respectively. To minimize the cost function  $J$  in the conjugate gradient direction  $d$ , we compute the step length  $\tau$ , as analyzed in Section 2.1. The optimal step length depends also on the sensitivity problem 1.11. We repeat this iterative procedure until the cost function is minimized.

As previous analyzed, to use this method, we must solve the state, adjoint and sensitivity problem. The solution of the state problem 4.1 is previous discussed. The finite volume discretization of the adjoint and sensitivity problems are linear systems, as developed in Section 2.2. To solve the adjoint and sensitivity equations we employ a modified versions of SIMPLE solver, as discussed in Section 2.3. The parameter set involved in the SIMPLE solver for the state problem, are summarized in the Table 4.2, as previous defined in Section 2.3. More precisely, the residual  $\sigma$  is evaluated by substituting the current solution into the equation and taking the magnitude of the difference between the left and right hand sides; it is also normalised in to make it independent of the scale of problem being analysed. We adopt the same parameters in the modified SIMPLE solver for the adjoint and the sensitivity problems.

**Table 4.2:** Parameters used in the SIMPLE algorithm.

Parameter	Symbol	Value
Velocity relaxation factors	$\alpha_v$	0.7
Pressure relaxation factors	$\alpha_p$	0.3
Residual controls	$\sigma$	$10^{-6}$

The linear solver used is the preconditioned conjugate gradient solver called *Geometric-Algebraic Multi-Grid (GAMG)* [54]. It uses the principle of: generating a quick solution on a mesh with a small number of cells; mapping this solution onto a finer mesh; using it as an initial guess to obtain an accurate solution on the fine mesh. The Table 4.3 summarizes the parameters used for GAMG solver with Gauss-Seidel smoother.

**Table 4.3:** GAMG parameter.

Parameter	Value
n° Cells In Coarsest Level	10
Speed of coarsening or refinement levels mesh	1
Residual control	$10^{-10}$
Ratio of current to initial residual control (for velocity and pressure)	$10^{-2}, 10^{-4}$

All the computations are performed in *ITHACA-FV*<sup>1</sup> (*In real Time Highly Advanced*

---

<sup>1</sup><https://mathlab.sissa.it/ITHACA-FV>

*Computational Applications for Finite Volumes*) package based on OpenFOAM<sup>2</sup> and developed at the *mathLab*<sup>3</sup> group in SISSA. The open-source C++ library OpenFOAM is used for the discretization and numerical simulation of PDEs based on a FV approach, employed in the context of continuum mechanics and especially in computational fluid dynamics. We refer the interested reader to [51] or [54] for all the details about this library. While, ITHACA-FV is an open source C++ library containing several reduced order modelling techniques in a finite volume framework. Most of the theoretical aspects behind ITHACA-FV are deeply explained in [64, 66]. In particular, we made use of the steady solver *simpleFoam* [54], which implements the SIMPLE algorithm for NS equations simulation, and modified version are developed for adjoint and sensitivity resolution. The optimization step with Non-linear projected conjugate gradient Algorithm 1 are developed in C++ code<sup>4</sup> with the help of ITHACA-FV library.

### 4.1.2 Comparison between different observed data

Now, we numerically analyze the performances of the inverse solvers introduced in Section 2.1, with different fields observed from initial guess  $\nu_0 = 10^{-3}$  in 25 iterations.

At first, we use the velocity data and find the viscosity constant  $\nu_v$ . Then, we plug the estimated viscosity field into direct problem and solve for  $\mathbf{v}$  and  $p$ . The results are shown in Figure 4.2. We observe that the predictions are very close to the reference, with better velocity results than pressure one. This is due to the choice of the functional cost  $J_1$  given by (1.16) depending on the velocity field desired.

In the second set up, we use only pressure data to find the viscosity constant  $\nu_p$ . Then, we obtain the results for  $\mathbf{v}$  and  $p$ , as shown in Figure 4.3. We observe that, also in this case, the predictions are very close to the reference. Obviously, in this set-up, we have better pressure result than the previous case. Moreover, we observe a small improvement for the velocity field. For these reasons, we expect better accuracy in the viscosity constant  $\nu_p$  found than in the previous one  $\nu_v$ .

Finally, with both velocity-pressure data, we find the viscosity constant  $\nu_{vp}$ . For this framework, we obtain the difference between velocity-pressure predictions and reference ones, as shown in Figure 4.4. We observe that, as in the previous cases, the predictions are very accurate. Moreover, we note that the performances for this case, in term of the velocity and pressure fields, are between the results with only velocity data and with only pressure data. This suggests that we have accuracy on viscosity between the previous two cases.

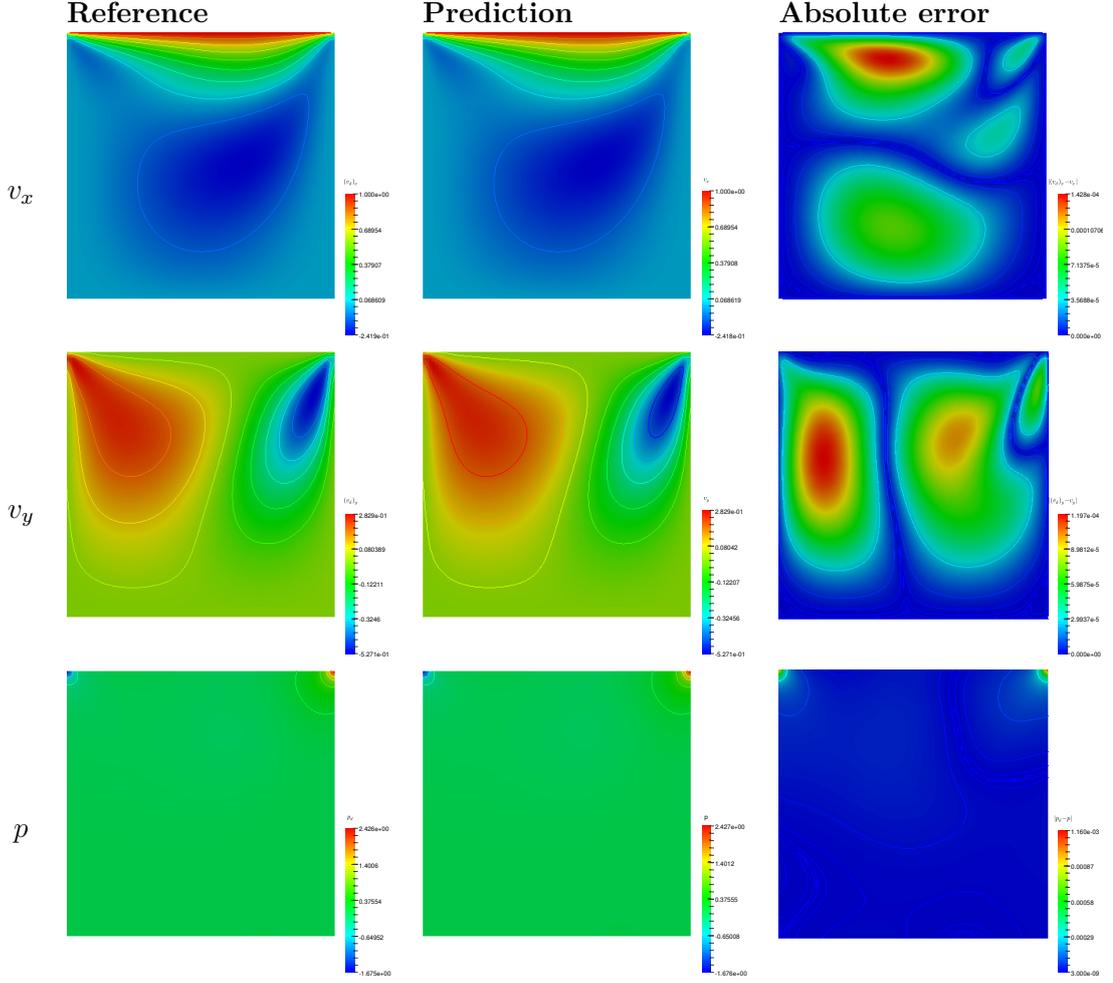
For a better comparison of these three approaches, we summarize the viscosity constant found  $\nu_v$ ,  $\nu_p$  and  $\nu_{vp}$  and the relative error with respect the viscosity searched  $\nu_d = 10^{-2}$  in Table 4.4. We can observe, that as we had predicted earlier, with pressure data we have the best accuracy for viscosity. The worst results are given with only velocity data.

---

<sup>2</sup><https://www.openfoam.com/>

<sup>3</sup><https://mathlab.sissa.it/>

<sup>4</sup><https://github.com/cetrangelo96/ITHACA-FV/tree/angelo/tutorials/CFD/CVinversecavityCG>



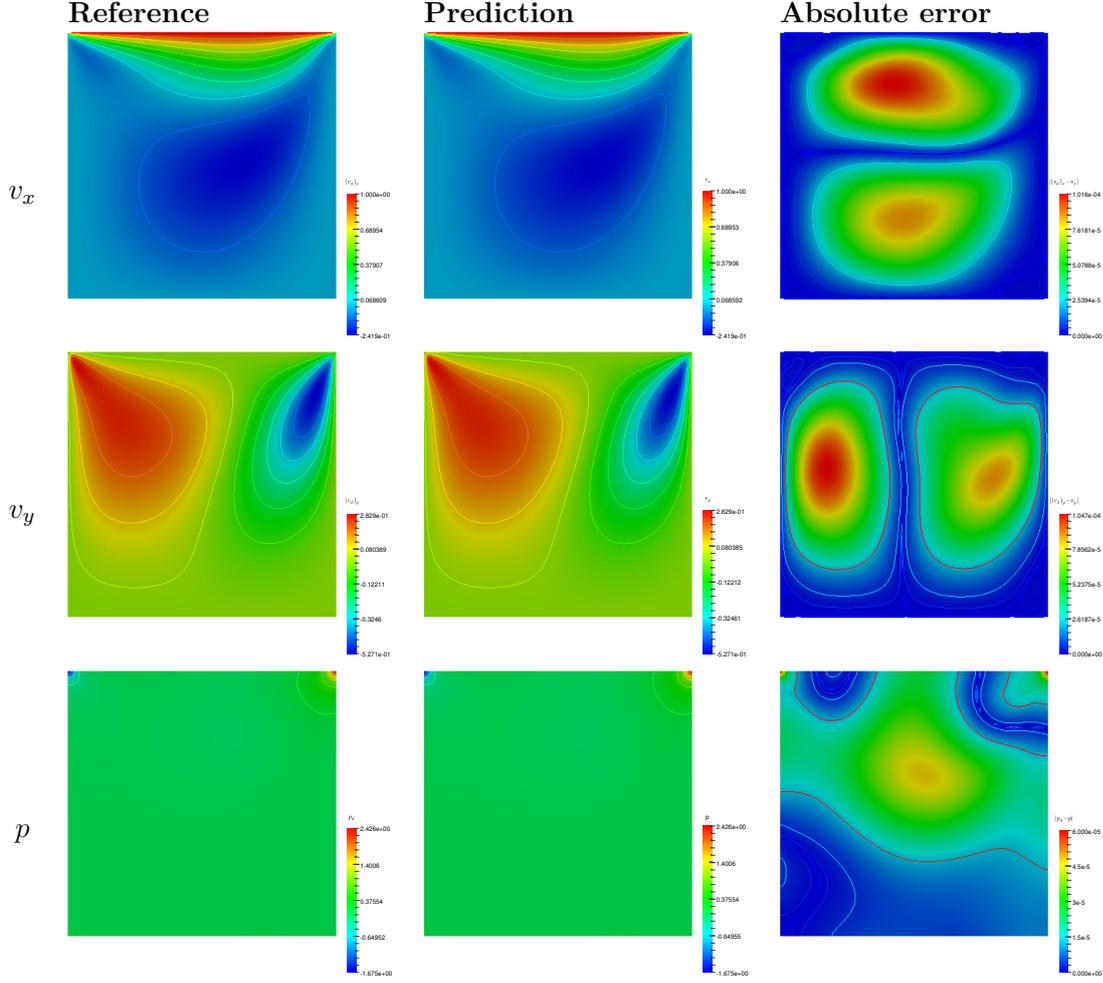
**Figure 4.2:** The reference (*left column*) velocity ( $x$ - component *First row*,  $y$ - component *Second row*) and pressure (*Third row*), the predictions (*central column*) from the CG method, and the corresponding error (*right column*). We note that the cost function only contains the velocity data.

**Table 4.4:** Accuracy for  $\nu_d = 10^{-2}$  with difference observed data.

Observed field	Viscosity estimation	Relative error $ \frac{\nu - \nu_d}{\nu_d} $
velocity	$\nu_v \simeq 1.0006 \times 10^{-2}$	0.06%
pressure	$\nu_p \simeq 1.00002 \times 10^{-2}$	0.002%
velocity-pressure	$\nu_{vp} \simeq 1.0003 \times 10^{-2}$	0.03%

### 4.1.3 Increasing wanted viscosity

In this subsection, we want to analyze the effects of the viscosity increase have on the results obtained by the CG method with different observed data.



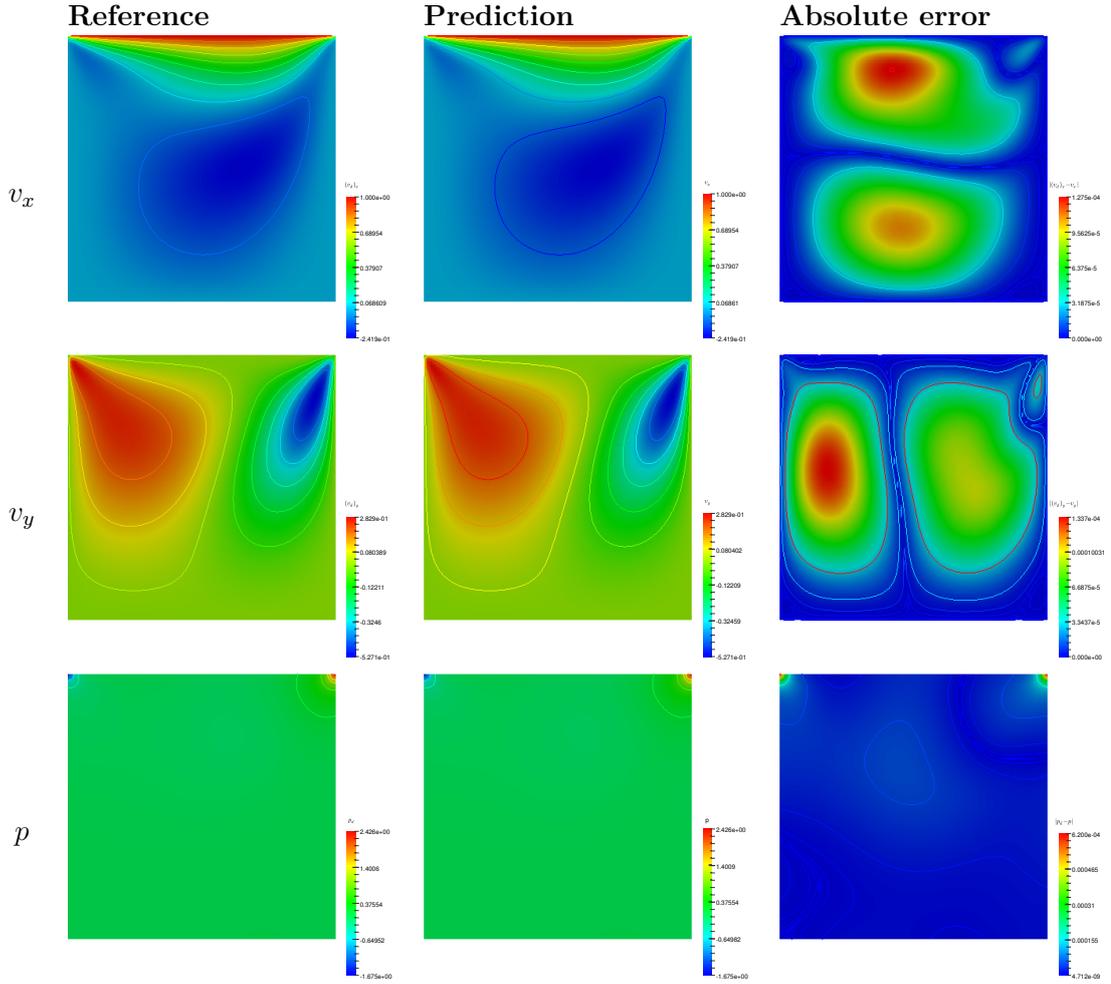
**Figure 4.3:** The reference (*left column*) velocity ( $x$ - component *First row*,  $y$ - component *Second row*) and pressure (*Third row*), the predictions (*central column*) from the CG method, and the corresponding error (*right column*). We note that the cost function only contains the pressure data.

In Figure 4.5, we compare the results at full order level of inverse problem for  $\nu_0 = 10^{-3}$  and  $\nu_d = 1$ . We can see the progress, with respect to iteration  $i = 1, \dots, 50$ , of the cost functional  $J$ , the absolute value of the optimal condition  $|J'|$ , the relative error on the viscosity constant, the relative error field respect to desired data, with different observed data. In particular, the relative error fields are given by:

$$\varepsilon_v^{rel} = \frac{|\mathbf{v} - \mathbf{v}_d|}{|\mathbf{v}_d|}, \quad \varepsilon_p^{rel} = \frac{|p - p_d|}{|p_d|}, \quad \varepsilon_\nu^{rel} = \frac{|\nu - \nu_d|}{|\nu_d|},$$

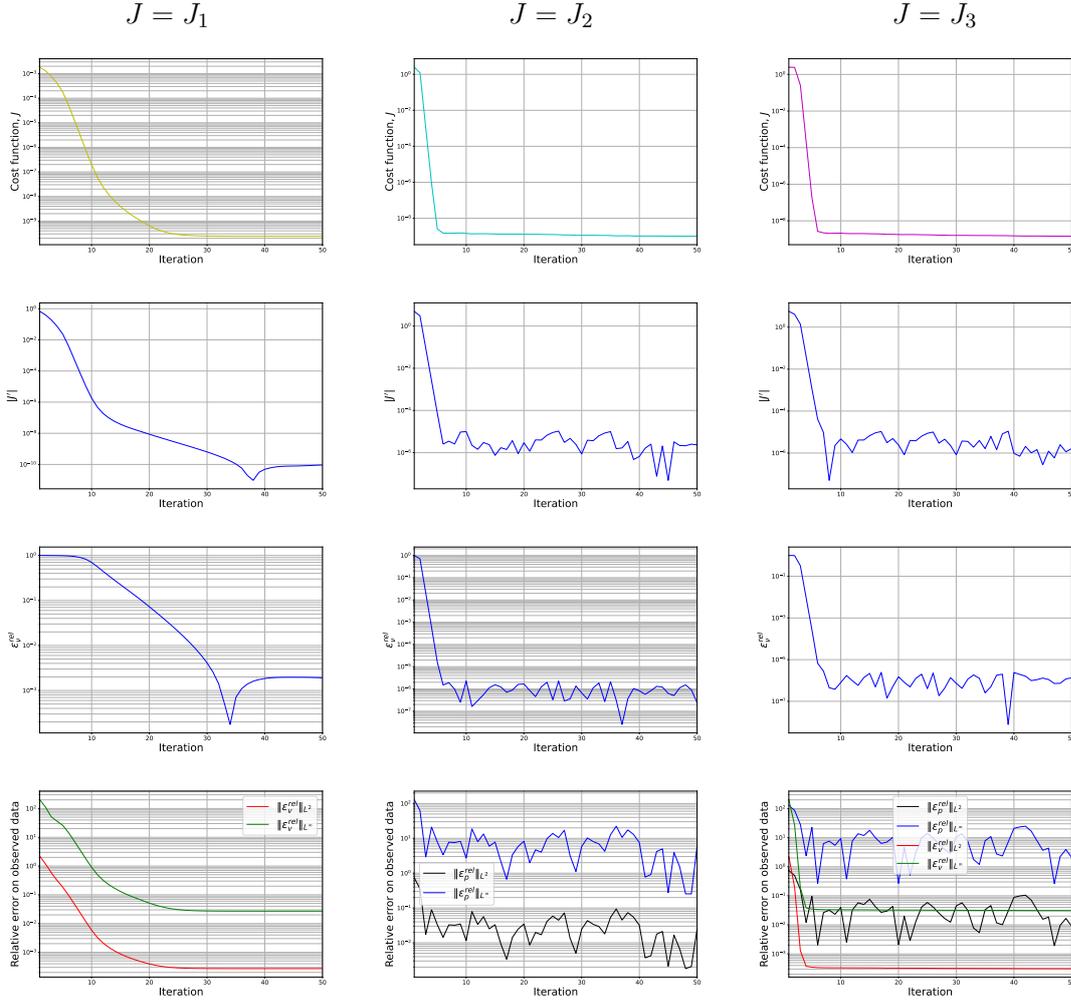
for velocity, pressure and viscosity, respectively.

With this first test, we show the accuracy and the performances with different observed data to find the direct solution for  $Re = 1$ , with good results. In particular, from the



**Figure 4.4:** The reference (*left column*) velocity ( $x$ - component *First row*,  $y$ - component *Second row*) and pressure (*Third row*), the predictions (*central column*) from the CG method, and the corresponding error (*right column*). We note that the cost function contains both velocity-pressure data.

behavior of the cost functions with pressure data  $J_2, J_3$ , as functions of the number of iterations of the algorithm, have a sharp decay in the first 10 iterations. Then, the convergence rate has a dramatic decrease reaching a plateau after 10 iterations. Instead, with only velocity data, we observe a decay more smooth and slow than the other cases. From the gradient plot, we see a rapidly decrease with only velocity data and, after 50 iterations, it stand around  $10^{-10}$ . Instead, with pressure data, we have also a fast decay but, after 10 iterations, a oscillating behavior around  $10^{-6}$ . For these reasons, we expect better viscosity accuracy for velocity data only. Instead, we have the opposite behavior. In fact, the graph of relative error on the viscosity has a plateau around  $10^{-3}$  for only velocity data, instead with pressure data, this oscillate around  $10^{-6}$ . We think that, this

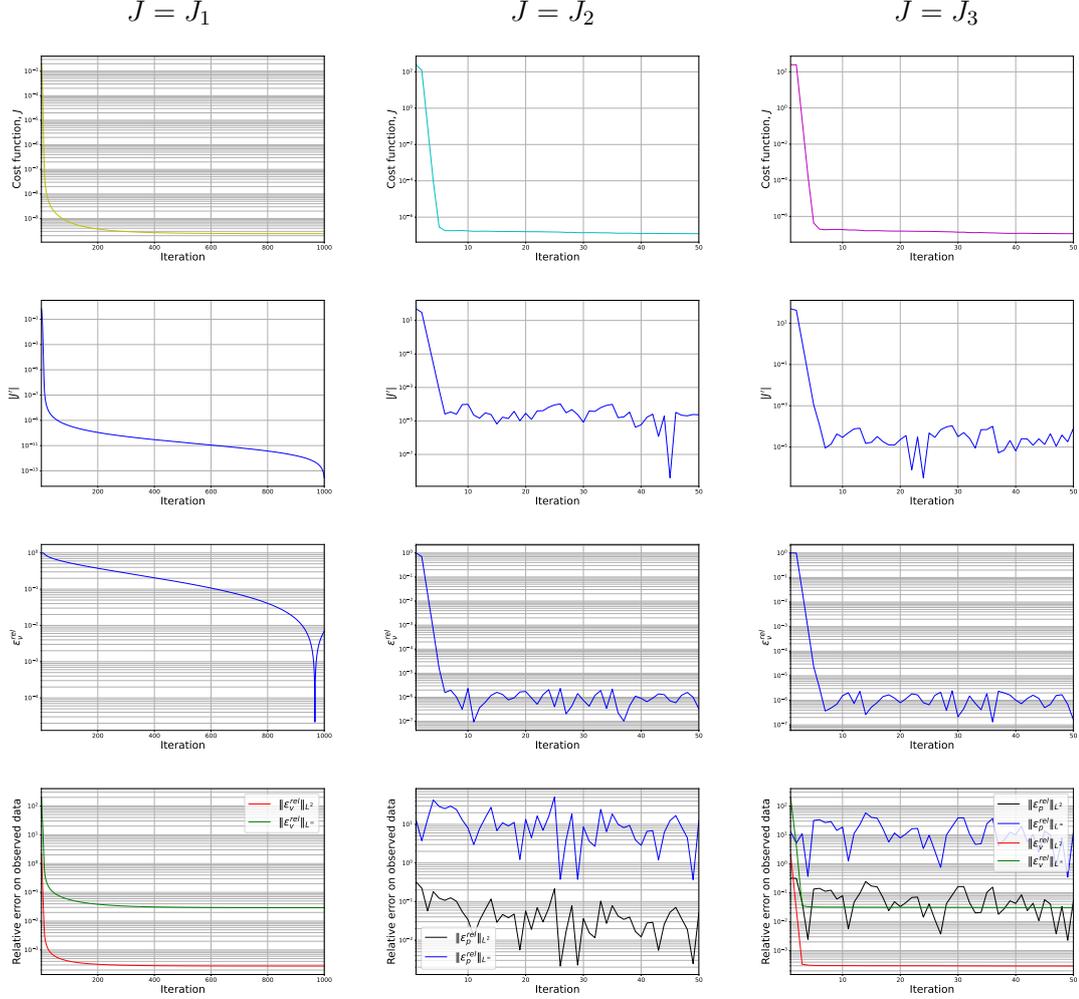


**Figure 4.5:** Comparison the FOM results with respect to the iteration with only velocity data (*left column*), with only pressure data (*central column*) and with both (*right column*), for  $\nu_0 = 10^{-3}$  and  $\nu_d = 1$ . *First row:* the cost function. *Second row:* the absolute value of the gradient. *Third row:* the relative error on  $\nu$ . *Fourth row:* the  $L^2$ - and  $L^\infty$ - norms of relative error on given data.

is due a better sensibility of the pressure field, with respect to the viscosity constant, than the velocity field. Analyzing the  $L^2$ - and the  $L^\infty$ - norms of the relative error on observed data, we note a smooth decay for the first set-up. In the second case, we have an oscillating behaviour for the pressure field, and this features is also with both observed data. Moreover, in the last case, we can see a sharper decay for the velocity field than the first case, with a plateau achieved before 10 iterations. Thus, it is faster than first set-up (25 iterations). This suggests to us that with pressure data, we optimize the velocity data better than without the pressure observations. This is due, not only to the link between pressure and velocity solutions in NS equations but also to the best sensibility of pressure

data to viscosity, in this case.

In Figure 4.6, we compare the results of the FOM with different observed data starting from the same initial guess  $\nu_0 = 10^{-3}$ . We increase the viscosity searched  $\nu_d = 10$ . In this figure, we will show the small convergence rate with only velocity data does not comparable with the other choices of the observed fields. We remark that, the Reynolds number searched is decreased with respect to the previous one, i.e.  $Re = 0.1$ . We test this



**Figure 4.6:** Comparison the FOM of test case with only velocity data (*left column*), with only pressure data (*central column*) and with both (*right column*), for  $\nu_0 = 10^{-3}$  and  $\nu_d = 10$ . *First row:* the cost functional. *Second row:* the absolute value of the gradient. *Third row:* the relative error on  $\nu$ . *Fourth row:* the  $L^2$ - and  $L^\infty$ - norms of relative error on given data.

to see the performance and the accuracy to estimate the viscosity constant corresponding to Reynolds number  $Re = 0.1$  employing the inverse problem solvers. We note with pressure data the same behaviour of previous case with smaller viscosity than now, in term of the cost functional, the absolute of his gradient, the accuracy on viscosity and

observed fields in 50 iterations. Moreover, the  $L^2$ - and  $L^\infty$ - norms of relative error on velocity behave the same. Instead, with only velocity data, the functional  $J_1$  has a very slow decay, instead, the gradient has a initial sharp decay in 100 iterations after we decrease slowly. The relative error on the control  $\nu$  also around 1000 iteration drops below the threshold of 1%. By the computational point of view, this set-up becomes really uncompetitive achieving the accurate results in a large amount of CPU time (in this test the solver employs takes approximately 86400s in terms of CPU time for 1000 iterations). For this reason, this is not a good solver at low Reynolds numbers.

Finally, we test only the solvers with pressure data for a higher viscosity than previous one. In Figure 4.7, we compare the results with full order solver of inverse problem for  $\nu_0 = 10^{-3}$  and  $\nu_d = 100$ . We test the performances and the accuracy for high viscosity wanted with pressure data. All the plots showed have the same behaviour of previous tests. We can see that, also a low Reynolds number  $Re = 0.001$ , the pressure field is sensitive to the changes in the viscosity. Therefore, the CG methods with pressure data are a good solvers at higher viscosity.

#### 4.1.4 Final remarks

Now, we can give the conclusion notes on the different full order inverse solvers also increasing the viscosity  $\nu_d \in \{0.01, 1, 10, 100\}$ .

Summarizing the accuracy and performance features of different inverse solvers, we have the best results to find viscosity with pressure data than only velocity data. The reason of these differences is the greater sensibility of the pressure field with respect to the viscosity variations than the velocity one.

Increasing the wanted viscosity, this gap grows until the solver with velocity data becomes useless for low Reynolds numbers. Instead, the CG methods with pressure data remain a very good solvers at low Reynolds numbers. The performances and the accuracy for  $\nu_d = 100$  are the same of these for  $\nu_d = 0.001$ . This show us that, the sensitivity of pressure field to viscosity is the same for different range of flows investigated.

To draw the final conclusions on the performances of the tested inverse solvers, we note their computational cost. This is of particular interest in our research because we want to achieve real-time estimations. The full order solvers developed with the CG methods take a long CPU time. In particular, the estimations of viscosity  $\nu_d = 1$  from initial guess  $\nu_0 = 10^{-3}$  given in 25 iterations (see Figure 4.5) are obtained in more or less 1200s for each type of observed data. Notice that, all the computations were performed in serial on a Intel<sup>TM</sup>Core<sup>i</sup>7-4510U CPU processor. For this reason, the employment of these solvers of inverse problem in a parametric framework, are unsustainable in term of computational cost. This requires to reduce the computational cost. The results with ROM for a real-time estimation of viscosity constant in lid-driven square cavity flow are showed in the following section.

## 4.2 Reduced Order Approximations

In this section, we will apply the reduced order methodologies, developed in previous Chapter 3, for inverse problem in CFD. We test our methodologies on the study case solved at full order level in the previous Section 4.1. We show the results for case study with only velocity data at reduced order level. Because the test case is a parameter identification problem, we choose as parameter for reduced order modelling the viscosity coefficient which want to find, i.e.  $\mu = \nu_d$ .

At beginning, we test the sensitivity of reduced solver in different reduced spaces. Moreover, we compare it with the full-order solver in term of accuracy and performances.

### 4.2.1 Description of the reduced order implementation

In our case we decided to apply snapshots-based reduced basis method to a real-time estimation of physical parameter  $\nu_d$ , as already discussed in Chapter 3.

Based on an offline/online paradigm, the ROM is employed for inverse problem simulations to reduce the computational cost. The offline and online phase for our test case is given by the Algorithms 6 and 7, respectively.

For the offline stage, we employ the full order solver to evaluate the full order fields, as previously developed in Chapter 2. The description of this full order solver for our test case is widely discussed in Section 4.1.

We only recall the parameters of full order solver. In particular, the computational mesh employed is shown in Figure 4.1 and his properties are summarized in Table 4.1. The coefficient for the SIMPLE algorithm and the GAMG solver of the equations, involved in the CG method 1, are listed in Tables 4.2 and 4.3, respectively.

In the offline phase,  $m = 50$  snapshots are computed, choosing  $m$  equispaced value for the parameter  $\mu$  in the range  $[10^{-3}, 10^{-2}]$  (corresponding to Reynolds number range  $[100, 1000]$ ). Then, we store the solution fields and the viscosity constant at the end of the full order solver with only velocity data, as previously set up.

To construct the reduced modal basis functions for the state and the adjoint pressure-velocity fields we employ the POD Algorithm 3 on full-order snapshots matrices, as exploited in Section 3.1. This step is performed through several reduced order modelling techniques implemented in ITHACA-FV library with Algorithm 3.

For the optimization step, in the online phase, we employ a reduced version of CG method used at full-order level, as developed in Section 3.5. This iterative procedure is implemented in C++ code<sup>5</sup> with the help of ITHACA-FV library. This online solver, summarized in the Algorithm 5, involves the state problem 4.1, the adjoint one 1.8 and the sensitivity one 1.11. Then, we must construct the reduced state, adjoint and sensitivity systems in the offline phase.

The reduced order dynamical system is computed, as developed in Section 3.2, with Galerkin projection of state, adjoint and sensitivity equations onto the POD spaces. In

---

<sup>5</sup><https://github.com/cetrangelo96/ITHACA-FV/tree/angelo/tutorials/CFD/ReducedCVinversecavityCG>

particular, because stability issue deriving by coupled solver of reduced system with saddle-point structure, we employ the exact supremizer enrichment method 4. This method is employed to full-fill inf-sup conditions, not only for state modal basis space but also for adjoint one, as previous discussed in Section 3.3. Instead, for inhomogeneous Diriclet BC treatment, we use lifting function method, as recalled in Section 3.4. After this projection step, with the particular considerations on the stability and the BC treatment, we obtain the reduced order direct, adjoint and reduced systems given by (3.75), (3.76) and (3.83), respectively.

In the online phase, the reduced systems solutions, involved in the Reduced Non-Linear Projected Conjugate Gradient Algorithm 5, are performed with the external library *Eigen* [21], which source code is provided together with ITHACA-FV. In particular, as previously mentioned in Section 3.5, we employ Powell’s dogleg method<sup>6</sup> for state non-linear system resolution and LU decomposition methods with complete pivoting<sup>7</sup> to solve adjoint and sensitivity linear system. These methods are pre-built in Eigen library.

## 4.2.2 Results

In this Section we propose some first numerical results on POD-Galerkin applied to the parametric test case. We will focus on the identification of the viscosity parameter from the observations of velocity field in lid-driven cavity flow setting.

Figure 4.8a shows the decay of the normalized eigenvalues for each fields. The eigenvalues are normalized by dividing them by the sum of all values, as follow

$$\Lambda^v = \frac{\lambda_i^v}{\sum_{k=1}^m \lambda_k^v}, \quad \Lambda^p = \frac{\lambda_i^p}{\sum_{k=1}^m \lambda_k^p}, \quad \Lambda^w = \frac{\lambda_i^w}{\sum_{k=1}^m \lambda_k^w}, \quad \Lambda^q = \frac{\lambda_i^q}{\sum_{k=1}^m \lambda_k^q},$$

where  $m$  is the total number of eigenvalues. We can see the eigenvalues in a decreasing order on a logarithmic scale for  $m$  modes. Figure 4.8b shows the cumulative energy of the eigenvalues for the state velocity  $\mathbf{v}$ , the state pressure  $p$ , the adjoint velocity  $\mathbf{w}$  and the adjoint pressure  $q$ . The  $i$ -th cumulative eigenvalue corresponds to the relative information content of POD-basis computed as in (3.33). In particular, we obtain the following cumulative eigenvalues formulas:

$$\mathcal{E}_v(i) = \sum_{k=1}^i \Lambda_k^v, \quad \mathcal{E}_p(i) = \sum_{k=1}^i \Lambda_k^p, \quad \mathcal{E}_w(i) = \sum_{k=1}^i \Lambda_k^w, \quad \mathcal{E}_q(i) = \sum_{k=1}^i \Lambda_k^q.$$

It can be seen that, for  $i = 10 \ll m$  POD modes, the energy captured by the arranged eigenvalues for the the state fields is decreased to approximately  $10^{-9}$ , for the adjoint fields to approximately  $10^{-4}$ . In order to retain the 99% of the system energy  $N_v = N_p = 9$  for state fields  $\mathbf{v}, p$  and  $N_w = N_q = 1$  for adjoint fields  $\mathbf{w}, q$  are selected. We choose this number of adjoint modes because the estimation is not particularly affected by this.

<sup>6</sup>[https://eigen.tuxfamily.org/dox/unsupported/classEigen\\_1\\_1HybridNonLinearSolver.html](https://eigen.tuxfamily.org/dox/unsupported/classEigen_1_1HybridNonLinearSolver.html)

<sup>7</sup>[https://eigen.tuxfamily.org/dox/classEigen\\_1\\_1FullPivLU.html](https://eigen.tuxfamily.org/dox/classEigen_1_1FullPivLU.html)

The reduced order spaces are generated from the corresponding eigenvectors. The supremizer modes corresponding to the pressure modes are added to the basis velocity POD-space. The state and the adjoint reduced spaces are enriched by  $N_v^s = 9$  and  $N_w^s = 1$  modes, respectively. Thus, the dimension of reduced state problem is  $N_S = N_v + N_p + N_v^s + 1 = 28$ , where an additional basis is for the lifting for non-homogeneous Dirichlet conditions. While, the adjoint problem has dimension  $N_A = N_w + N_q + N_w^s = 3$ . As previously analyzed in Section 3.5, the sensitivity variables are in the state POD-spaces without the lifting function  $\phi_L$ , i.e. it has dimension  $N_s = 27$ .

Figure 4.9 show the performances of the Reduced Non-Linear Projected Conjugate Gradient (RNLP CG) Algorithm 5 for the estimation of viscosity parameter  $\mu = 1.55 \times 10^{-2}$  from initial guess  $\nu_0 = 10^{-3}$ .

In Figure 4.9a, we can see the rapidly decay of cost function  $J_1$ , computed with the modal coefficients in (3.78), in 7 iteration, with a plateau in more or less  $4 \times 10^{-4}$ . At iteration 18, the CG method stop because the variation on control give zero variation on cost function. The online phase is performed in 3.32152 s CPU time.

In Figure 4.9b, we see the behaviour of the absolute value of the gradient, computed with the modal coefficients in (3.74), with the same trend of the cost function.

In Figure 4.9c, we show the progress of the relative error on the viscosity with a plateau around 5.8%.

Instead, in Figure 4.9d we can see the  $L^2$ - and  $L^\infty$ - norms of the relative error between the observed velocity and the reduced one computed for each iteration. Summarizing, from Figure 4.9, we conclude that the RNLP CG method is a excellent reduced solver for this set-up for a real-time (3.32152 s CPU time) estimation of the viscosity parameter with relative error on viscosity of 5.8%.

To have qualitatively insight on the results, we compare the full order solutions for the state velocity,  $\mathbf{v}_f = ((v_f)_x, (v_f)_y)$ , and the state pressure,  $p_f$ , variables with the reduced ones,  $\mathbf{v}_r = ((v_r)_x, (v_r)_y)$  and  $p_r$ , respectively, for the parameter  $\mu = 1.55 \times 10^{-2}$ . The reduced approach gives promising results, as can be noticed in Figure 4.10. Indeed, the POD-Galerkin projection is able to recover the high-fidelity fields. From the absolute error plots, we can see at most of order  $10^{-2}$  for each fields.

To compare FOM and ROM results for parameter  $\mu = 1.55 \times 10^{-2}$ , we summarize the absolute and the relative error on the state and the control variables between this two solver in Table 4.5. We observe that, for the velocity and the pressure fields, we report  $L^2$ - norm of error. We remark that, the high-fidelity solution with 5625 cells is computed in 2100 s CPU time, while, the reduced online solver perform in 3.32152 s with  $\mathcal{N} = N_S + N_A + 1 = 32$  degree of freedom (dof) for state, adjoint and control variables. From this Table, we can see the good features in terms of the accuracy of reduced solutions

**Table 4.5:** FOM vs ROM, Error ( $L^2$ - norm of error for fields) for  $\mu = 1.55 \times 10^{-2}$ .

Field	Absolute error	Relative error
velocity	0.002	0.03
pressure	0.001	1.7
viscosity	0.0009	5%

respect to FOM ones, take advantages from a really small CPU time employed.

The accuracy of the online solver is visible in the plots in Figure 4.11. This figure shows the absolute and the relative log-error referred to different number of the state pressure and the velocity modes  $n \in \{1, \dots, 9\}$  and for adjoint ones fixed to  $N_w = N_q = 1$ . We test the online solver over a training set  $\Xi$  of uniformly distributed parameters in the range  $\mathbb{P} = [0.001, 0.02]$ . The absolute error calculated for each solution variable using the following norms

$$\begin{aligned} \mathfrak{E}_v^{abs} &= \|\mathbf{v}_f - \mathbf{v}_r\|_X, & \mathfrak{E}_p^{abs} &= \|p_f - p_r\|_Y, & \mathfrak{E}_\nu^{abs} &= |\nu_f - \nu_r| \\ \mathfrak{E}_w^{abs} &= \|\mathbf{w}_f - \mathbf{w}_r\|_X, & \mathfrak{E}_q^{abs} &= \|q_f - q_r\|_Y, \end{aligned}$$

while, the relative error are computed by

$$\begin{aligned} \mathfrak{E}_v^{rel} &= \left\| \frac{|\mathbf{v}_f - \mathbf{v}_r|}{|\mathbf{v}_f|} \right\|_X, & \mathfrak{E}_p^{rel} &= \left\| \frac{p_f - p_r}{p_f} \right\|_Y, & \mathfrak{E}_\nu^{rel} &= \frac{|\nu_f - \nu_r|}{|\nu_f|} \\ \mathfrak{E}_w^{rel} &= \left\| \frac{|\mathbf{w}_f - \mathbf{w}_r|}{|\mathbf{w}_f|} \right\|_X, & \mathfrak{E}_q^{rel} &= \left\| \frac{q_f - q_r}{q_f} \right\|_Y. \end{aligned}$$

These errors are averaged over  $|\Xi| = 20$  uniformly distributed parameters. We notice that the POD-Galerkin approach is very effective for all the involved quantities reaching, after  $n = 9$  modes for state fields, absolute error values around  $2 \times 10^{-3}$  for the state variables, around  $6 \times 10^{-3}$  for the adjoint variables and around  $6 \times 10^{-4}$  for the viscosity estimation. As expected, from average relative error, we can see that the approximations of state fields and control variable become more accurate as the number of modes generating the POD space increases. In particular, using  $N_v = N_p = 9$  basis function for state velocity and pressure field, the estimation of viscosity at reduced order level becomes 5.8% accurate with respect to full-order approximation.

Furthermore, we calculate the difference between FOM and ROM approximations of objective functional  $J$  as below:

$$\mathfrak{E}_J = |J(\mathbf{v}_f, \mathbf{v}_d(\mu)) - J(\mathbf{v}_r, \mathbf{v}_d(\mu))|,$$

and report its average on 20 value of parameter in Figure 4.12b. For 9 POD modes, as expected, the objective functional approximated by both full order and reduced order methods differ by  $5 \times 10^{-6}$ . In Table 4.12a we summarize the computational parameter to compare FOM and ROM solver for the last test.

### 4.2.3 Final remarks

Now, we give the conclusion notes on the reconstruction of the reduced order framework based upon the POD-Galerkin for the parametrized version of test case.

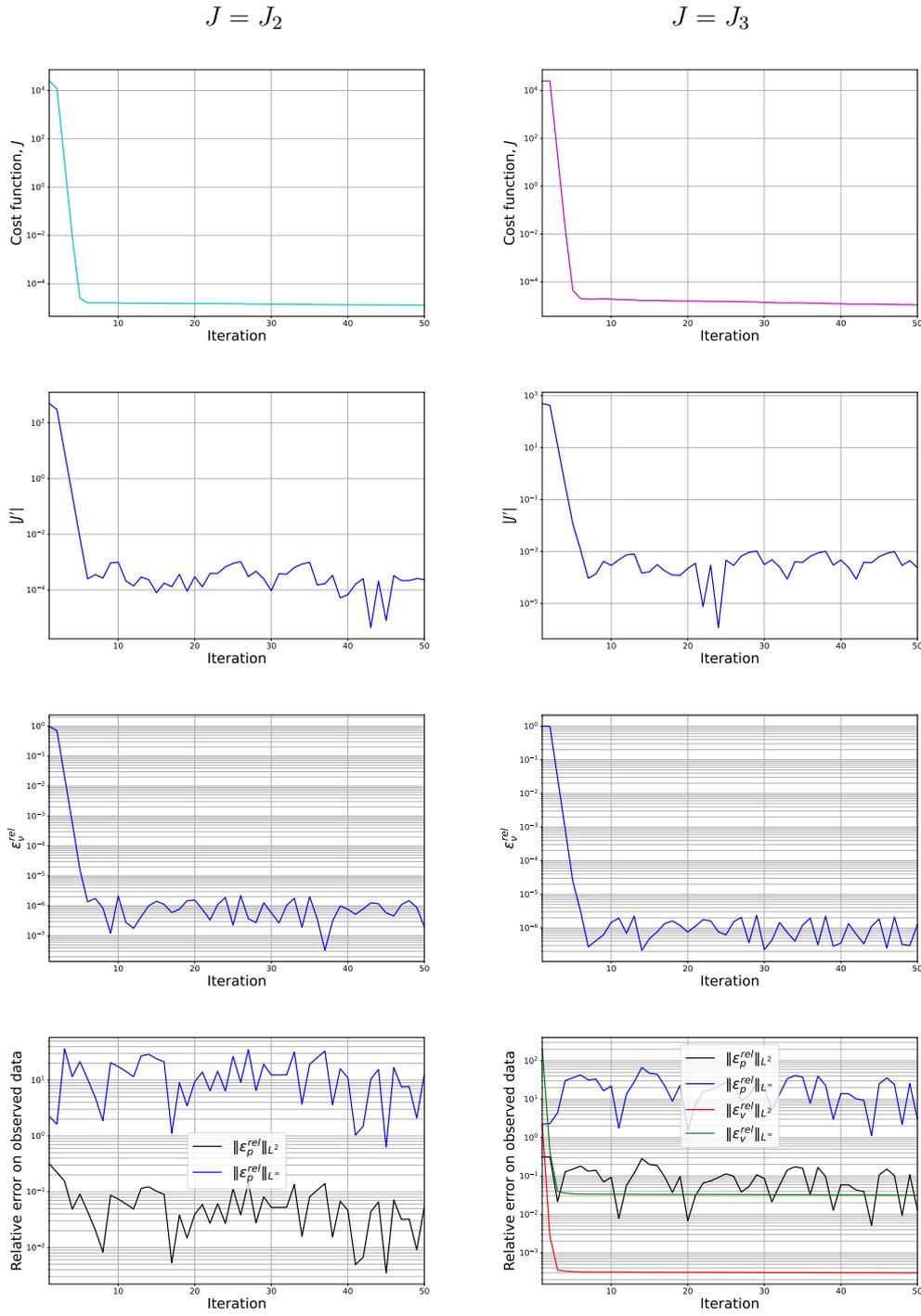
From the numerical results, it is shown reduction in the dimensions of the spaces from  $\mathcal{O}(10^4)$  to  $\mathcal{O}(10^1)$  and reduction in the computational time from  $\mathcal{O}(10^5)$  to  $\mathcal{O}(1)$  seconds.

We remark that, the numerical results show the offline phase to be computationally expensive. However, it is explainable since the computational cost of this phase depends

upon the dimension of the sampled set  $m$  and the cost of truth solutions linked to mesh size (in this case  $dof = 5625$ ).

We reiterate the offline phase, although being costly, needs to be carried out only once. Instead, thanks to the reduced order spaces, for tuning different parameters we only need to repeat the online phase with the computational cost lower than the high fidelity methods. For this reason, the online solver developed in Chapter 4, it is a very useful tool for a computational real-time simulations of the parametric test case.

In terms of accuracy, the identification parameter problem, tested with reduced order model, gives good results with the relative error on estimation of 6%. Then, we have analyzed a reduced order solver with not only better results in the amount of CPU time but also the good accuracy with respect to the high-fidelity solutions.



**Figure 4.7:** Comparison the FOM of test case with only pressure data (*left column*) and with pressure-velocity (*right column*), for  $\nu_0 = 10^{-3}$  and  $\nu_d = 100$ . *First row:* the cost function. *Second row:* the absolute value of the gradient. *Third row:* the relative error on  $v$ . *Fourth row:* the  $L^2$ – and  $L^\infty$ – norms of relative error on given data.

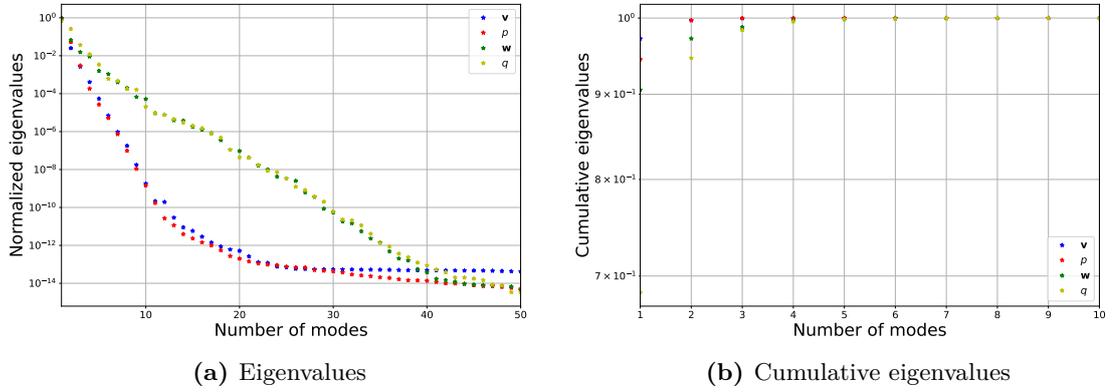


Figure 4.8: Energy of the eigenvalues reduction for the case study.

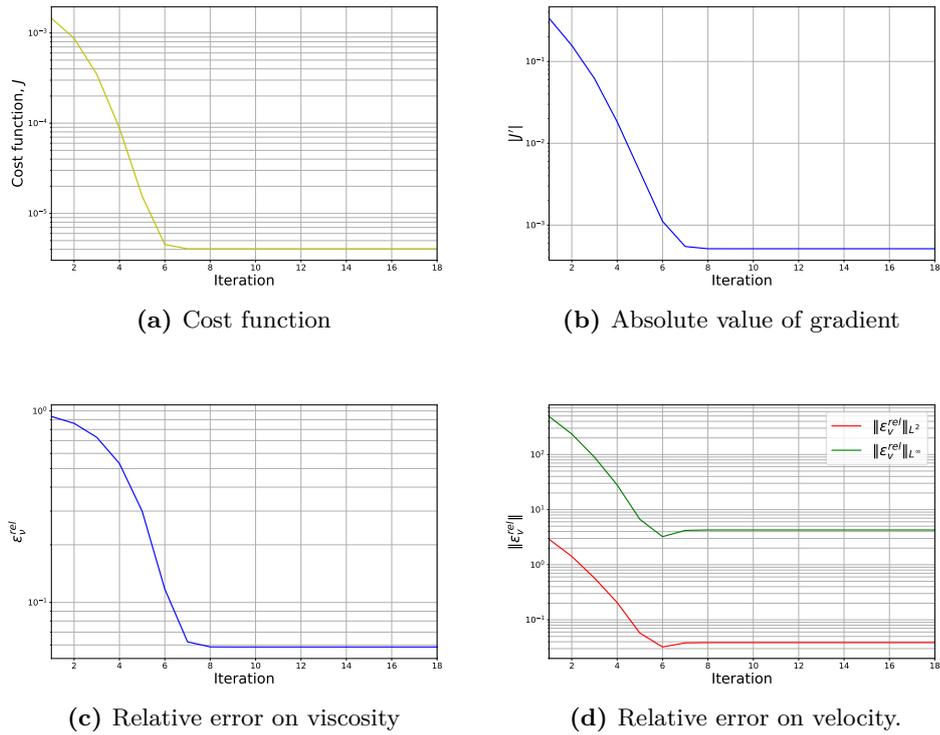
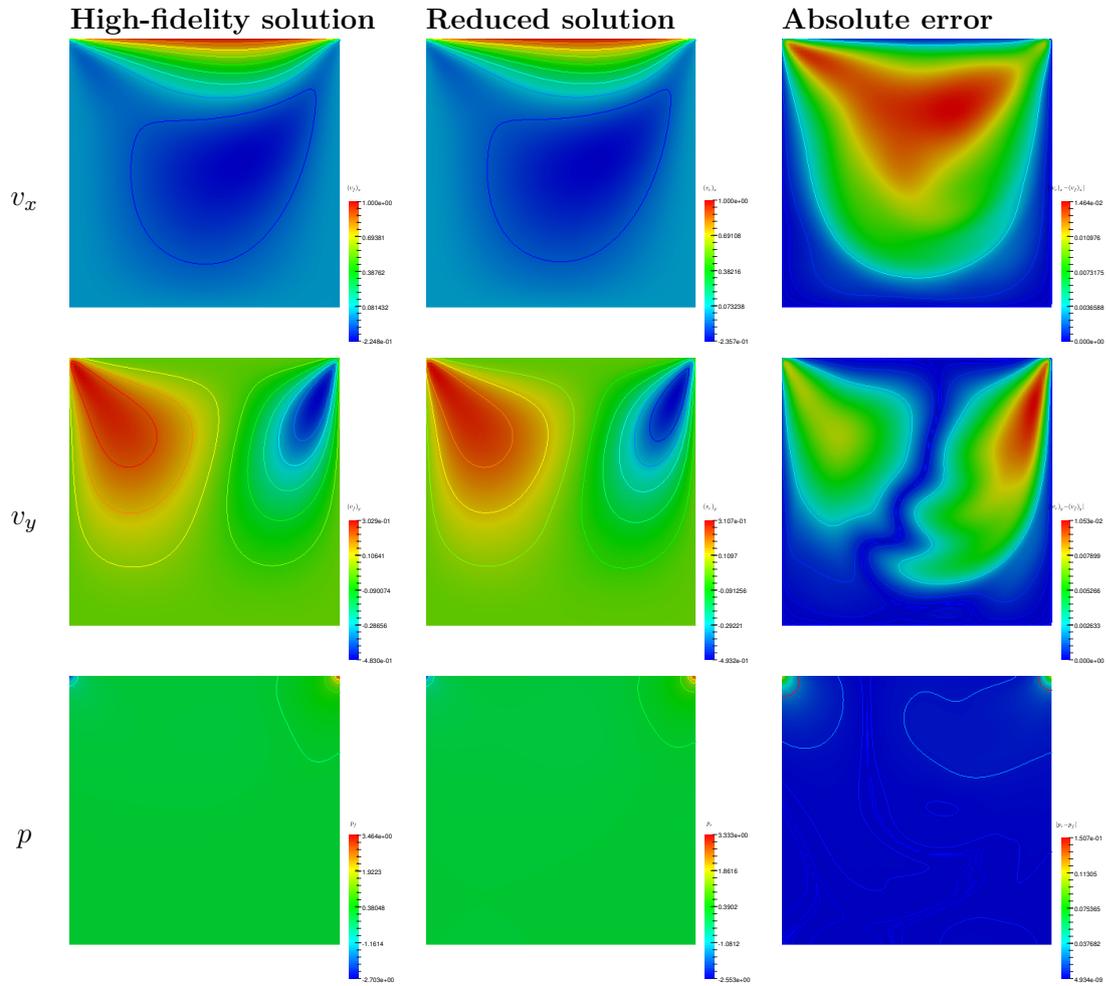
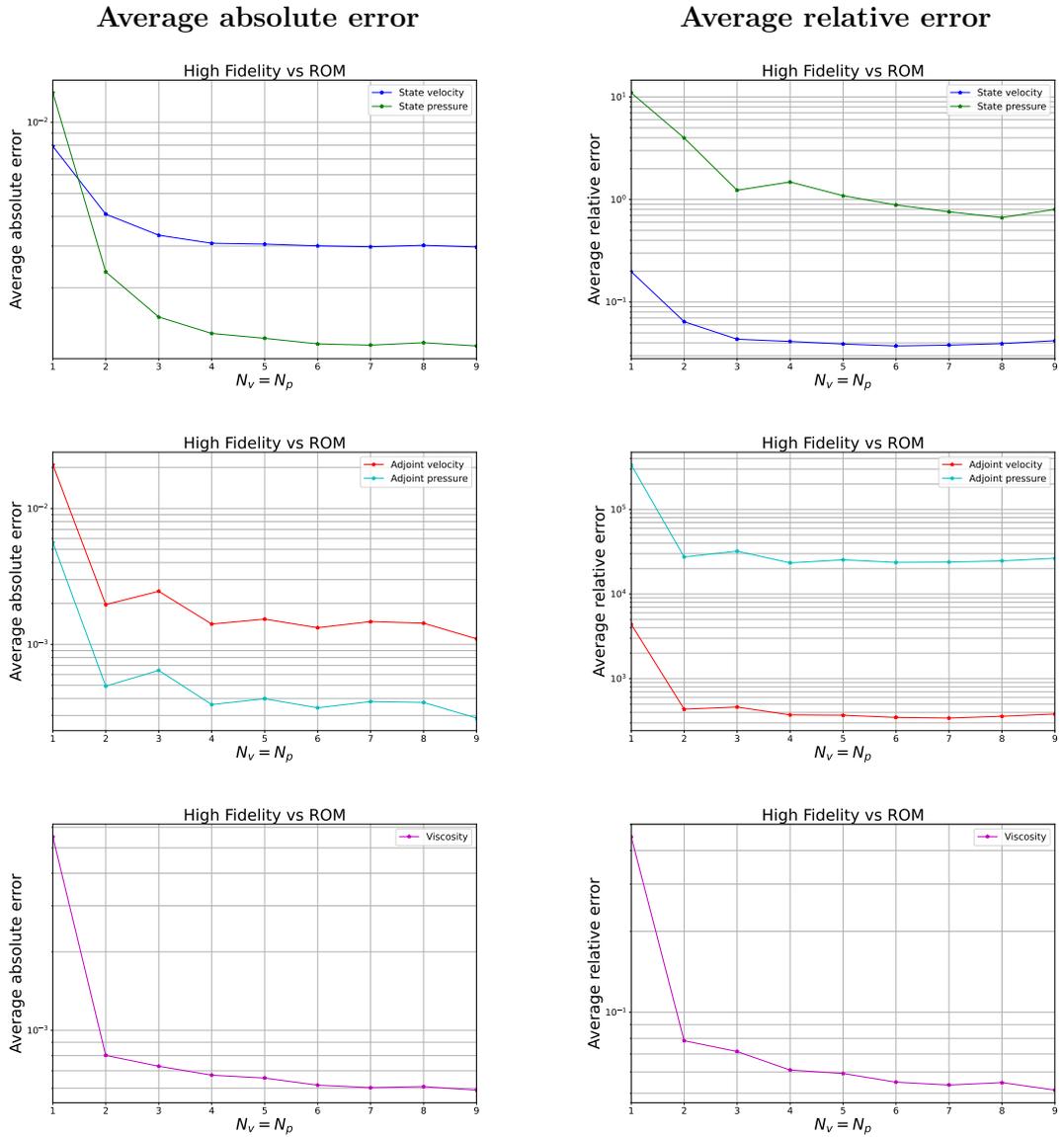


Figure 4.9: Performances and accuracy of the reduced CG method for the parameter  $\mu = 1.55 \times 10^{-2}$ , from the initial guess  $\nu_0 = 10^{-3}$ .



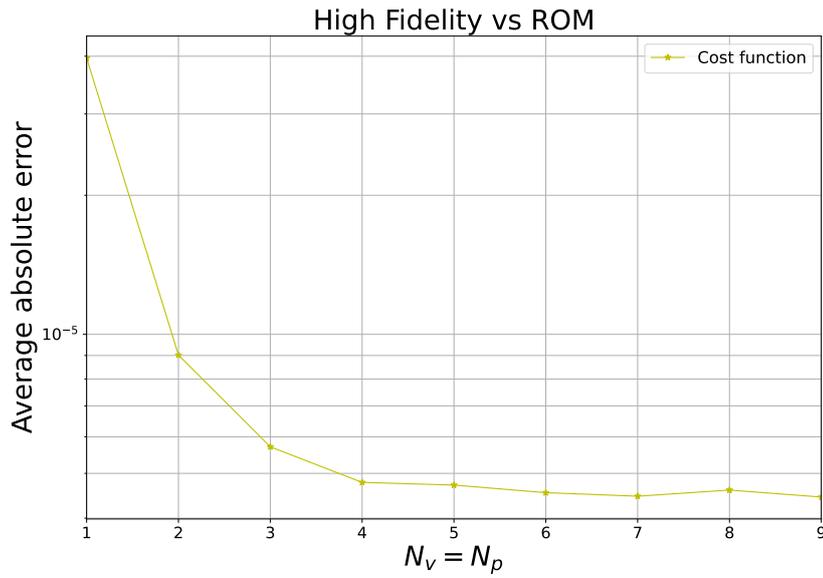
**Figure 4.10:** Comparison between the Full-order solutions (*left column*) and the Reduced ones (*central column*), with the corresponding absolute error (*right column*). *First row:* the  $x$ -component of velocity. *Second row:* the  $y$ -component of velocity; *Third row:* the pressure.



**Figure 4.11:** Averaged absolute (*left column*) and relative (*right column*) log-error for the state (*First row*), adjoint (*Second row*) and control (*Third row*) variables.

Mesh size (dof)	5625
Range of parameter $\mathbb{P}$	[0.001, 0.02]
Snapshots number $m$	50
Offline phase	65362 seconds
Training set dimension $ \Xi $	20
FOM results	2314 seconds
Dimension of reduced space $\mathcal{N}$	32
Online phase	3.5214 seconds

(a) Computational performance of ROM procedure

(b) Error on cost function  $J$ , ROM vs FOM.**Figure 4.12:** Comparison FOM vs ROM cost function and computational performances.



# Conclusion and perspectives

## Concluding remarks

In this Thesis, we wanted to investigate the numerical solution of a particular inverse problem in CFD with low computational cost. Our case of study is on a cavity flow where the viscosity has to be deduced from velocity and pressure data. We have constructed a projection-based reduced order framework for parametrized optimal flow control problems and we have tested its applicability on this case.

First of all, in Chapter 1, we have investigated the mathematical principles of PDEs constrained optimal control problems to deduce the optimization system given by state, adjoint and optimality equations. After that, we implemented a full-order solver by relying on an optimize-then-discretize approach, as shown in Chapter 2. For this scope, we employed an iterative procedure to also regularize the inverse problem. In particular, we exploited a CG method to optimize a cost function given by the norm of error between observations and data while varying the viscosity. To compute the conjugate gradient descent direction, a numerical resolution is needed not only for the direct problem, given by the Navier-Stokes equations, but also for the adjoint problem. At the same time, for what concerns the step length calculation, a sensitivity problem solution has to be carried out. To solve the equations involved in CG method, we recalled the finite volume discretization, then, we employed SIMPLE solver.

Being the CG method an iterative procedure, these problems have to be resolved at each iteration. Then, we observed a high computational cost. Taking into consideration the numerical results showed in Section 4.1, it can be noticed that the results appear to be reliable enough for our scopes but the necessary CPU time is very high. Then, this full-order solver is not suited for parametric frameworks.

To mitigate the computational effort, we have tried to mimic the reduced methods developed for optimal control problems in a finite elements framework (see [52, 6]) by modifying them to perform at their best for finite volume approximations. To achieve this goal, a data compression strategy has been employed: a POD-Galerkin method, for both the direct and the adjoint problem, has been exploited to obtain reduced systems. The optimization at reduced-order level has been performed through a reduced version of the CG full-order solver. With this online procedure, we were able to estimate parameter in state problem from observed data with low computational cost. By taking advantage of this architecture, a considerable reduction for what regards the required CPU time has been achieved, while, also obtaining good results in terms of accuracy.

Reduced basis methods can be very versatile in this context, where many parameters can be involved. In this particular field of application, simulations can be computationally demanding and costly: RB techniques could be a good and viable way to enhance the computational performances.

## Future perspective

In this Thesis, we introduced the ill-posed inverse problems and regularization theories which should be more deeply analyzed.

A possible development of this work is observing the fields of direct problem only in some points. This allows us to analyze the sensitivity of estimation to the number and position of these observation points. Considering noise in the measurements, we can analyze the uncertainty quantification on the viscosity estimation, using better regularization methods. These two experiments are already exploited for heat flux estimation, for example in [50].

In this work, at reduced order level, we employed a optimize-then-reduce approach. Thus, we can exploit the reduce-then-optimize approach and compare with the other procedure. This solution strategies allows us to firstly discretize the optimal control problem with reduced approximations and then apply a optimization procedure, as employed in heamodynamics applications [47].

The test case of spatial varying viscosity field identification has been studied from a theoretical point of view. This problem is very ill-posed, what makes is impossible to be treated by the use of a Conjugate Gradient method, as it has been done for the uniform case. Different methods should be investigated for the resolution of this problem. A strategy based on combining deep neural networks and numerical PDE schemes is used to solve this problem in [12].

In this work, at the reduced order level, we employed a coupled solver for the direct, adjoint and sensitivity reduced systems focusing on a gradient method. Anyway, a segregate solver would allow us to avoid the saddle point structure characterizing this reduced problem, without the necessity to rely on stabilization techniques. This approach is already used in [65] to solve geometrically parametrized incompressible Navier-Stokes problem.

We have performed a Galerkin-projection of KKT system in reduced spaces. We can solve the reduced KKT system in a coupled way, typically used for Finite Element methods, i.e. see [52, 6, 68]. Then, a comparison between the reduced version of the conjugate gradient and some other reduced coupled solvers could be very interesting.

The natural continuation of this work could be the extension of the developed methods to more general frameworks for other types of inverse problems, not only for parameter identification ones, but also, e.g., for parametrized boundary controls or parametrized source fields in fluid dynamics with finite volume approximation.

For our test case, only a physical parametrization has been taken into consideration, but another possibility is to test the developed architectures on geometrical parametrization problems, with applications, for example, in the shape optimization field [10].

Another possible step forward for the results obtained in this thesis, is to extend what has been done for steady-state flows to time dependent inverse problems.

One more possible future research topic, concerning the arguments analyzed in these chapters, is related to the effects of non-linearity due to an increasing Reynolds number, which leads to the presence of turbulence approximations in the state equations. In addition, for a fluid dynamics framework, we can take into account the effects of compressibility or even the fluid-structure interactions phenomena.

Finally, it would be useful to extend the numerical studies developed for finite element optimal flow control applications to a finite volume configuration, to better satisfy the increasing demand for handy tools from the industrial world.



# Bibliography

- [1] James P. Ahrens, Berk Geveci, and C. Charles Law. «ParaView: An End-User Tool for Large-Data Visualization». In: *The Visualization Handbook*. 2005.
- [2] Shafqat Ali. «Stabilized reduced basis methods for the approximation of parametrized viscous flows.» PhD thesis. SISSA, 2018. URL: <http://hdl.handle.net/20.500.11767/82794>.
- [3] Oleg M Alifanov. *Inverse heat transfer problems*. Springer Science & Business Media, 2012.
- [4] Jasbir Singh Arora. «Chapter 13 - More on Numerical Methods for Constrained Optimum Design». In: *Introduction to Optimum Design (Fourth Edition)*. Ed. by Jasbir Singh Arora. Fourth Edition. Boston: Academic Press, 2017, pp. 555–599. ISBN: 978-0-12-800806-5. DOI: <https://doi.org/10.1016/B978-0-12-800806-5.00013-5>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128008065000135>.
- [5] Utkarsh Ayachit. «The ParaView Guide: A Parallel Visualization Application». In: 2015.
- [6] Francesco Ballarin, Gianluigi Rozza, and Maria Strazzullo. «Reduced order methods for parametric flow control problems and applications». In: *arXiv preprint arXiv:2011.12101* (2020).
- [7] Francesco Ballarin, Gianluigi Rozza, and Maria Strazzullo. *Reduced order methods for parametric flow control problems and applications*. 2020. arXiv: [2011.12101](https://arxiv.org/abs/2011.12101) [[math.NA](https://arxiv.org/abs/2011.12101)].
- [8] Francesco Ballarin et al. «Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations». In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1136–1161. ISSN: 1097-0207. DOI: [10.1002/nme.4772](https://doi.org/10.1002/nme.4772).
- [9] Marta D’Elia, Mauro Perego, and Alessandro Veneziani. «A variational data assimilation procedure for the incompressible Navier-Stokes equations in hemodynamics». In: *Journal of Scientific Computing* 52.2 (2012), pp. 340–359.

- [10] Nicola Demo et al. «Shape Optimization by means of Proper Orthogonal Decomposition and Dynamic Mode Decomposition». In: *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*. IOS Press. Trieste, Italy: IOS Press, 2018. Chap. 212. DOI: [10.3233/978-1-61499-870-9-212](https://doi.org/10.3233/978-1-61499-870-9-212). URL: <http://ebooks.iospress.nl/publication/49229>.
- [11] J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996. DOI: [10.1137/1.9781611971200](https://doi.org/10.1137/1.9781611971200). eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611971200>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611971200>.
- [12] Tiffany Fan et al. *Solving Inverse Problems in Steady-State Navier-Stokes Equations using Deep Neural Networks*. 2020. arXiv: [2008.13074](https://arxiv.org/abs/2008.13074) [math.NA].
- [13] Joel Ferziger, Milovan Perić, and Robert Street. *Computational Methods for Fluid Dynamics*. Springer International Publishing, Jan. 2020. ISBN: 978-3-319-99691-2. DOI: [10.1007/978-3-319-99693-6](https://doi.org/10.1007/978-3-319-99693-6).
- [14] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2013. ISBN: 9781118723180.
- [15] R. Fletcher and C. M. Reeves. «Function minimization by conjugate gradients». In: *The Computer Journal* 7.2 (Jan. 1964), pp. 149–154. ISSN: 0010-4620. DOI: [10.1093/comjnl/7.2.149](https://doi.org/10.1093/comjnl/7.2.149). eprint: <https://academic.oup.com/comjnl/article-pdf/7/2/149/959725/070149.pdf>. URL: <https://doi.org/10.1093/comjnl/7.2.149>.
- [16] Gilles Fourestey and Marwan Moubachir. «Solving inverse problems involving the Navier–Stokes equations discretized by a Lagrange–Galerkin method». In: *Computer Methods in Applied Mechanics and Engineering* 194 (Feb. 2005), pp. 877–906. DOI: [10.1016/j.cma.2004.07.006](https://doi.org/10.1016/j.cma.2004.07.006).
- [17] A.-L. Gerner and Karen Veroy. «Certified Reduced Basis Methods for Parametrized Saddle Point Problems». In: *SIAM Journal on Scientific Computing* 34 (Jan. 2011). DOI: [10.1137/110854084](https://doi.org/10.1137/110854084).
- [18] S. F. Gilyazov. «Regularizing algorithms based on the conjugate-gradient method». In: *Ussr Computational Mathematics and Mathematical Physics* 26 (1987), pp. 8–13.
- [19] W. R. Graham, Jaime Peraire, and Kongzheng Tang. «OPTIMAL CONTROL OF VORTEX SHEDDING USING LOW-ORDER MODELS. PART II-MODEL-BASED CONTROL». In: *International Journal for Numerical Methods in Engineering* 44 (1999), pp. 973–990.
- [20] Carmen Gräßle, Michael Hinze, and Stefan Volkwein. «2 Model order reduction by proper orthogonal decomposition». In: *Volume 2 Snapshot-Based Methods and Algorithms: Volume 2: Snapshot-Based Methods and Algorithms*. De Gruyter, 2020, pp. 47–96. DOI: [10.1515/9783110671490-002](https://doi.org/10.1515/9783110671490-002). URL: <https://doi.org/10.1515/9783110671490-002>.
- [21] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. <http://eigen.tuxfamily.org>. 2010.

- 
- [22] Max Gunzburger, Janet Peterson, and J. Shadid. «Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data». In: *Computer Methods in Applied Mechanics and Engineering* 196 (Jan. 2007), pp. 1030–1047. DOI: [10.1016/j.cma.2006.08.004](https://doi.org/10.1016/j.cma.2006.08.004).
- [23] MD Gunzburger and HG Wood. «Perspectives in Flow Control and Optimization». eng. In: *Applied mechanics reviews* 56.6 (2003), B82–B83. ISSN: 0003-6900.
- [24] J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Alpha Editions, 2020. ISBN: 9789354180712. URL: <https://books.google.it/books?id=fq7zzQEACAAJ>.
- [25] Martin Hanke. *Conjugate gradient type methods for ill-posed problems*. Chapman and Hall/CRC, 2017.
- [26] Charles R. Harris et al. «Array programming with NumPy». In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [27] Roland Herzog and Karl Kunisch. «Algorithms for PDE-Constrained Optimization». In: *GAMM-Mitteilungen* 33 (Oct. 2010), pp. 163–176. DOI: [10.1002/gamm.201010013](https://doi.org/10.1002/gamm.201010013).
- [28] Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. 1st ed. Springer Briefs in Mathematics. Switzerland: Springer, 2015, p. 135. ISBN: 978-3-319-22469-5.
- [29] Saddam Hijazi et al. «Data-driven POD-Galerkin reduced order model for turbulent flows». In: *Journal of Computational Physics* 416 (Sept. 2020), p. 109513. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2020.109513](https://doi.org/10.1016/j.jcp.2020.109513). URL: <http://dx.doi.org/10.1016/j.jcp.2020.109513>.
- [30] Saddam Hijazi et al. *Non-Intrusive Polynomial Chaos Method Applied to Full-Order and Reduced Problems in Computational Fluid Dynamics: a Comparison and Perspectives*. 2019. arXiv: [1901.02285](https://arxiv.org/abs/1901.02285) [math.NA].
- [31] Saddam Hijazi et al. *The Effort of Increasing Reynolds Number in Projection-Based Reduced Order Methods: from Laminar to Turbulent Flows*. 2018. arXiv: [1807.11370](https://arxiv.org/abs/1807.11370) [math.NA].
- [32] Saddam N Y Hijazi. «Reduced order methods for laminar and turbulent flows in a note volume setting: projection-based methods and data-driven techniques.» PhD thesis. SISSA, 2020. URL: <http://hdl.handle.net/20.500.11767/114353>.
- [33] CH Huang and MN Özisik. «Inverse problem of determining unknown wall heat flux in laminar flow through a parallel plate duct». In: *Numerical Heat Transfer* 21.1 (1992), pp. 55–70.
- [34] J. D. Hunter. «Matplotlib: A 2D graphics environment». In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [35] A Iollo, M Ferlauto, and L Zannetti. «An aerodynamic optimization method based on the inverse problem adjoint equations». In: *Journal of Computational Physics* 173.1 (2001), pp. 87–115.

- [36] Angelo Iollo and Manuel D. Salas. «Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks». In: *Journal of Computational Physics* 125.1 (1996), pp. 124–134. ISSN: 0021-9991. DOI: <https://doi.org/10.1006/jcph.1996.0083>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999196900832>.
- [37] V. Isakov. *Inverse Problems for Partial Differential Equations*. Applied Mathematical Sciences. Springer International Publishing, 2017. ISBN: 9783319516585. URL: <https://books.google.it/books?id=cT02DgAAQBAJ>.
- [38] Barbara Kaltenbacher, Andreas Neubauer, and Otmar Scherzer. *Iterative regularization methods for nonlinear ill-posed problems*. de Gruyter, 2008.
- [39] Andreas Karageorghis and Daniel Lesnic. «Identification of obstacles immersed in a stationary Oseen fluid via boundary measurements». In: *Inverse Problems in Science and Engineering* 28.7 (2020), pp. 950–967.
- [40] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Applied Mathematical Sciences. Springer New York, 1996. ISBN: 9780387945309. URL: <https://books.google.it/books?id=1lNUaSKHj3gC>.
- [41] Karl Kunisch and Stefan Volkwein. «Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics». In: *SIAM J. Numerical Analysis* 40 (July 2002), pp. 492–515. DOI: [10.1137/S0036142900382612](https://doi.org/10.1137/S0036142900382612).
- [42] T. Lassila et al. «A reduced computational and geometrical framework for inverse problems in hemodynamics». In: *International Journal for Numerical Methods in Biomedical Engineering* 29.7 (2013), pp. 741–776. DOI: [10.1002/cnm.2559](https://doi.org/10.1002/cnm.2559).
- [43] Jacques Louis Lions. *Optimal control of systems governed by partial differential equations*. eng. Die Grundlehren der mathematischen Wissenschaften 170. Berlin: Springer, 1971.
- [44] Stefano Lorenzi et al. «POD-Galerkin Method for Finite Volume Approximation of Navier-Stokes and RANS Equations». In: *Computer Methods in Applied Mechanics and Engineering* 311 (2016), pp. 151–179. DOI: [j.cma.2016.08.006](https://doi.org/10.1016/j.cma.2016.08.006).
- [45] OpenCFD Ltd. *OpenFOAM (Open-source Field Operation And Manipulation)*. Version 2006. June 2020. URL: <https://www.openfoam.com/>.
- [46] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 2015. ISBN: 3319188410.
- [47] Andrea Manzoni. «Reduced models for optimal control, shape optimization and inverse problems in haemodynamics». In: *2nd ECCOMAS Young Investigators Conference (YIC 2013)*. ECCOMAS PhD Award. Bordeaux, France, Sept. 2013. URL: <https://hal.archives-ouvertes.fr/hal-00855846>.
- [48] Tomohito Matsuo, Hikari Shimadera, and Akira Kondo. «Identification of multiple contamination sources using variational continuous assimilation». In: *Building and Environment* 147 (2019), pp. 422–433.
- [49] Giovanni Monegato. *Metodi e algoritmi per il calcolo numerico / Giovanni Monegato*. ita. 2. ed. Torino: CLUT, 2020. ISBN: 9788879924672.

- [50] Umberto Morelli et al. «A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation». In: *International Journal for Numerical Methods in Engineering* 122 (Apr. 2021). DOI: [10.1002/nme.6713](https://doi.org/10.1002/nme.6713).
- [51] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*. 1st. Springer Publishing Company, Incorporated, 2015. ISBN: 3319168738.
- [52] F. Negri et al. «Reduced basis method for parametrized elliptic optimal control problems». In: *SIAM Journal on Scientific Computing* 35.5 (2013), A2316–A2340. DOI: [10.1137/120894737](https://doi.org/10.1137/120894737).
- [53] Jorge Nocedal. *Numerical optimization / Jorge Nocedal, Stephen J. Wright*. eng. 2nd ed. Springer series in operation research and financial engineering. New York: Springer, 2006. ISBN: 978-0-387-30303-1.
- [54] OpenCFD. *OpenFOAM - The Open Source CFD Toolbox - User's Guide*. 2106th ed. OpenCFD Ltd. United Kingdom, June 2021. URL: <https://www.openfoam.com/documentation/user-guide>.
- [55] S.V Patankar and D.B Spalding. «A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows». In: *International Journal of Heat and Mass Transfer* 15.10 (1972), pp. 1787–1806. ISSN: 0017-9310. DOI: [https://doi.org/10.1016/0017-9310\(72\)90054-3](https://doi.org/10.1016/0017-9310(72)90054-3). URL: <https://www.sciencedirect.com/science/article/pii/0017931072900543>.
- [56] Anthony T Patera and Gianluigi Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. MIT-Pappalardo Graduate Monographs in Mechanical Engineering. Preface and Part I. Cambridge, MA, US: (C) MIT, Massachusetts Institute of Technology, 2007. URL: <http://infoscience.epfl.ch/record/124801>.
- [57] M. J. D. Powell. «A Hybrid Method for Nonlinear Equations». In: *Numerical Methods for Nonlinear Algebraic Equations*. Ed. by P. Rabinowitz. Gordon and Breach, 1970.
- [58] Alfio Quarteroni. *Numerical Models for Differential Problems*. Springer Publishing Company, Incorporated, Jan. 2017. ISBN: 978-3-319-49315-2. DOI: [10.1007/978-3-319-49316-9](https://doi.org/10.1007/978-3-319-49316-9).
- [59] Mathias Richter. *Inverse Problems: Basics, Theory and Applications in Geophysics*. 1st. Lecture notes in geosystems mathematics and computing. Cham, Switzerland: Birkhäuser, 2016, p. 240.
- [60] Gianluigi Rozza, David Phuong, and Andrea Manzoni. «Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: Roles of the inf-sup stability constants». In: *Numerische Mathematik* 125 (Sept. 2013). DOI: [10.1007/s00211-013-0534-8](https://doi.org/10.1007/s00211-013-0534-8).
- [61] Gianluigi Rozza and Karen Veroy. «Veroy, K.: On the stability of reduced basis method for Stokes equations in parametrized domains. Comp. Meth. Appl. Mech. and Eng.196, 1244-1260». In: *Computer Methods in Applied Mechanics and Engineering* 196 (Jan. 2007), pp. 1244–1260. DOI: [10.1016/j.cma.2006.09.005](https://doi.org/10.1016/j.cma.2006.09.005).

- [62] Gianluigi Rozza et al. *Volume 2 Snapshot-Based Methods and Algorithms*. De Gruyter, Dec. 2020. ISBN: 9783110671407. DOI: [10.1515/9783110671490](https://doi.org/10.1515/9783110671490).
- [63] B.P. Rynne and M.A. Youngson. *Linear Functional Analysis*. Springer undergraduate mathematics series. Springer, 2000. ISBN: 9781852332570. URL: [https://books.google.it/books?id=WVg4G%5C\\_6gnXkC](https://books.google.it/books?id=WVg4G%5C_6gnXkC).
- [64] Giovanni Stabile and Gianluigi Rozza. «Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations». In: *Computers & Fluids* (2018). DOI: [10.1016/j.compfluid.2018.01.035](https://doi.org/10.1016/j.compfluid.2018.01.035).
- [65] Giovanni Stabile, Matteo Zancanaro, and Gianluigi Rozza. «Efficient Geometrical parametrization for finite-volume based reduced order methods». In: *International Journal for Numerical Methods in Engineering* 121 (2020), pp. 2655–2682. DOI: [10.1002/nme.6324](https://doi.org/10.1002/nme.6324). URL: <https://arxiv.org/abs/1901.06373>.
- [66] Giovanni Stabile et al. «POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder». In: *Communications in Applied and Industrial Mathematics* 8.1 (2017), pp. 210–236. DOI: [10.1515/caim-2017-0011](https://doi.org/10.1515/caim-2017-0011).
- [67] Maria Strazzullo, Francesco Ballarin, and Gianluigi Rozza. *POD-Galerkin Model Order Reduction for Parametrized Nonlinear Time Dependent Optimal Flow Control: an Application to Shallow Water Equations*. 2020. arXiv: [2003.09695](https://arxiv.org/abs/2003.09695) [math.NA].
- [68] Maria Strazzullo, Francesco Ballarin, and Gianluigi Rozza. «POD-Galerkin Model Order Reduction for Parametrized Time Dependent Linear Quadratic Optimal Control Problems in Saddle Point Formulation». In: *Journal of Scientific Computing* 83 (2020). DOI: [10.1007/s10915-020-01232-x](https://doi.org/10.1007/s10915-020-01232-x).
- [69] Maria Strazzullo et al. «Reduced order methods for parametrized non-linear and time dependent optimal flow control problems, towards applications in biomedical and environmental sciences». In: *ENUMATH2019 proceedings*. Springer. Springer, 2020. URL: <https://arxiv.org/abs/1912.07886>.
- [70] Michael Ulbrich et al. *Optimization with PDE constraints*. eng. Mathematical modelling theory and applications 23. Heidelberg: Springer Science + Business Media, 2009. ISBN: 9781402088384.
- [71] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [72] Stefan Volkwein. «Optimal Control of a Phase-Field Model Using Proper Orthogonal Decomposition». In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 81 (Feb. 2001), pp. 83–97. DOI: [10.1002/1521-4001\(200102\)81:2<83::AID-ZAMM83>3.0.CO;2-R](https://doi.org/10.1002/1521-4001(200102)81:2<83::AID-ZAMM83>3.0.CO;2-R).
- [73] Ralph A Willoughby. «Solutions of ill-posed problems (AN Tikhonov and VY Arsenin)». In: *SIAM Review* 21.2 (1979), p. 266.

- [74] Zakia Zainib. «Reduced order parameterized viscous optimal flow control problems and applications in coronary artery bypass grafts with patient-specific geometrical reconstruction and data assimilation.» PhD thesis. SISSA, 2019. URL: <http://search.ebscohost.com/login.aspx?direct=true&db=ir01172a&AN=sissa.103036&site=eds-live>.
- [75] Zakia Zainib et al. «Reduced order methods for parametric optimal flow control in coronary bypass grafts, toward patient-specific data assimilation». In: *International Journal for Numerical Methods in Biomedical Engineering* (May 2020). DOI: [10.1002/cnm.3367](https://doi.org/10.1002/cnm.3367).