Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Luca Bertagna	Date

Reliable direct and inverse methods in computational hemodynamics

by

Luca Bertagna
Doctor of Philosophy

Mathematics and Computer Science

Alessandro Veneziani, Ph.D.
Advisor

Michele Benzi, Ph.D.
Committee member

James Nagy, Ph.D.
Committee member

Leo Rebholz, Ph.D.
Committee member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Reliable direct and inverse methods in computational hemodynamics

by

Luca Bertagna
M.S. in Mathematical Engineering, Politecnico di Milano, 2009
B.S. in Mathematical Engineering, Politecnico di Milano, 2006

Advisor: Alessandro Veneziani, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2015

Abstract

Reliable direct and inverse methods in computational hemodynamics
By Luca Bertagna

In the last 25 years, developments in mathematical models/methods together with the improvements in the data acquisition devices have made possible to use mathematics to study the behavior of the human cardiovascular system. Furthermore, cardiovascular mathematics has not been limited to be used as a descriptive qualitative tool, but instead, has started to be used for quantitative analysis of patients conditions and even treatment design. The robustness of this tool depends on the reliability of the results. Data Assimilation (DA) is a set of techniques that helps to improve the specificity of the model, by incorporating available data (e.g., measurements) into the model and can therefore help to make the results of the simulation patient specific. On the other hand, the numerical methods used in the simulations must be accurate enough to guarantee that the computed solution accurately describes the real behavior of the system.

This work is divided into two parts. In the first, we focus on the estimation of the compliance of a blood vessel using DA techniques. In particular, we use measurements of the displacement of the vessel wall to estimate its Young's modulus. We adopt the variational approach proposed in [69], and we focus on the issue of the computational costs associated with the solution of the inverse problem. The secon part of this work concerns the accurate simulation of flows at moderately large Reynolds numbers. In particular, we focus on the model proposed in [53] for the discretization of the Leray system, and we propose a new interpretation of the method as an operator-splitting scheme, for a perturbed version of the Navier-Stokes equations, and we use heuristic arguments to calibrate one of the main parameters of the model.

For both these parts we will perform numerical experiments, on 3D geometries, to validate the approaches. In particular, for the first part, we will use synthetic measures to validate our approach, while for the second part, we will test the method on a benchmark proposed by the Food and Drug Administration, comparing out results with experimental data.

Reliable direct and inverse methods in computational hemodynamics

by

Luca Bertagna
M.S. in Mathematical Engineering, Politecnico di Milano, 2009
B.S. in Mathematical Engineering, Politecnico di Milano, 2006

Advisor: Alessandro Veneziani, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2015

Acknoledgements

There are a lot of people that I want to thank for their support in the last five and a half years. They all contributed to the completion of my PhD, and I just hope I won't forget anyone. Also, I'm going to thank people in a random order, so don't be sad if you're not the first.

I will start from my family, my parents and my brother, for supporting me when I decided to hop on a plane and try this adventure in America. They never stopped me to wonder if that was a wise choice or not, they always thought I was going to do great and trusted me in my decisions. And not just about my PhD. For this I will always be grateful.

I need to thank my advisor Alessandro. His passion about his work overflows to those working around him, and is really contagious. Thanks to him, I have learned how to tackle an engineering problem, discovering the beauty of the mathematics underlying it without losing the focus on the concrete application.

I have learned a lot also from the people that worked in Alessandro's (extended) group. Some I had the pleasure to work with for a long time, others less, but they all contributed to stregthen my skills and passion. In random order, thank you Umberto, Marta, Tiziano, Lucia, LGG, Mauro, Marina, Annalisa, Leandro, Huanhuan, Boyi, Jim, Gaetano, Adrien, Rodrigo, Ricardo.

I thank the professors in the committee, for all of them have contributed (in many ways) to make me reach the level of knowledge (and confidence) I now have in mathematics. I had the pleasure to have Michele and Jim also as professors in some of my classes, where, in addition of teaching me concepts and notions and methods, they also helped me consider things from different perspectives. I haven't had the pleasure of having Leo as a professor, but I did collaborate with him, which gave me the chance to absorb a (very) little part of his deep knowledge on computational fluid dynamics.

I thank my (other) fellow PhD students, with whom I have shared a lot of my time during the years I spent in grad school. In random order, Victor, Kevin, McKenzie, Megan, Pascal, Veronica, Christine, James, Bastian, Marie, Vindya, Bill, Robert (both), Anastasia, Alex. A special note goes to Hernando, who was a really good friend and helped me a lot when I first arrived in the US. I would have had a much harder time without him.

I thank my fellow aliens (and aliens-wannabe) in Atlanta, with whom I shared a lot of time and run lots of races. They have been a family to me, and it is been a privilege to grow as a person by learning from their cultures. In random order, thank you Umberto, Deborah, Lucien, Alex, Ines, Karla, Carla, Isabel, Rafa, Joan, Tauriq. We had some legendary adventures together.

I thank my friends in Desenzano, who will probably never read this but who cares, I'm thanking them anyways. Going home to my family is always nice, but having great friends waiting for you really makes the trip home a unique moment of the year. In random order, thank you Case, Santi, Seba, Massi, Visco, Scavo, Bomp, Teo, MLNI, Denis, Ivan, Bunelc, Alex, Bugo, Riino.

Also a big thank you to my fellow Ing+Math@work friends, because they know what "operare ad anello chiuso" means. The classic dinner at Christmas is a treat, and seeing them is always great. In random order, Teo, Lorenzo, Ruggi, Armando,

L, Gauss, GabriVale, AndreaFra, Cate, Azzu, Fra, Ale, Max. Also, let me extend the thanks to the Milano family: John, Sonny, Goro, Michea, Beppe, Giuglio, Spartanna.

Finally, to you, Sandy, for suddenly burst into my life as a sharknado and turn it into a dream. You surprise me and make me laugh everyday. I don't care how hard will be the challenges that life will throw at me, as long as I know you will be there.

To the Luca of 6 years ago

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cardiovascular mathematics

The beginning of cardiovascular mathematics, that is, the application of mathematics to the modeling and investigation of the human cardiovascular system, dates back to the XVIII century, when Leonard Euler introduced the equations (which now are called Euler equations) that describe the motion of an inviscid fluid in a compliant domain, with the goal to describe the motion of blood inside human arteries [24].

Other important contributions were given, in the XVIII and XIX centuries, by other mathematicians, physicists and physiologists. Among them, we mention Daniel Bernoulli, who related the increase of the speed of an ideal, inviscid fluid to a decrease of its pressure, Jean Léonard Marie Poiseuille, who gave a formula for the pressure drop of a laminar viscous flow in a pipe, Thomas Young, who characterized the relation between stresses and deformation in an elastic solid through what we now call *Young's modulus* [104] (although the concept was already known to Euler).

However, it is only in the second half of the XX century that mathematics started to rise as an important quantitative tool to help understanding the behavior of the cardiovascular system. First of all, the development of technologies for image acquisition, such as Computed Tomography (CT), ultrasounds and, later on, Optical Coherence Tomography (OCT), allowed doctors to acquire a larger amount of data.

These new techniques allowed the investigation of pathologies on patients using specific information on the geometry of their cardiovascular system.

On the other hand, important advancements were also made in mathematics, and we can identify (at least) two major areas where advancements made cardiovascular mathematics possible: image processing and numerical methods for fluid and structure dynamics. The first area is focused on the *refinement* of the available data, converting it into coherent and ordered data that can be used in further analysis or computations. For instance, when dealing with images of a blood vessel, this refinement process can include the segmentation of the images, in order to identify the region where the blood flows (also called *lumen* of the vessel), and the registration of the images when, for instance, two images corresponding to two different instants in time are compared, and corresponding points are identified with the goal to track the movement of the vessel in time.

On the side of mathematical models for fluid and structural dynamics of the circulatory system, several advancements have been done in the second half of the XX century (see, for instance, [78] and references therein). On one hand, the mechanical properties of the vessel wall have been investigated thoroughly and several models have been proposed, ranging from simple elastic models to more complex rheologies [16, 31, 70]. We also mention important improvements for the modeling of the blood flow such as non-Newtonian rheologies for the blood have been proposed [103], or the study of boundary conditions for the inflow and outflow section of the vessel [63, 98]. The resulting set of equations for fluid and structure and their interaction have also been thoroughly investigated [73, 94].

Finally, the numerical methods for solving the equations governing the fluid and structural dynamics also experienced a terrific improvement. Arguably, the most important advancement was the invention of the Finite Element (FE) method, which naturally reduces the weak formulation of a Partial Differential Equation to a suitable finite dimensional discrete space (see, for instance, [79, 91]). In the last sixty years, the FE method has been massively used in structural mechanics, computational fluid dynamics and, in the last 25 years, for simulating blood flow problems and Fluid-Structure Interaction (FSI) problems arising in cardiovascular ap-

plications. Among other methods used in FSI application, we also mention the *Immersed Boundary Method*, which is particularly successful for problems that involve large displacements of the structure [71].

1.2 Some of the challenges in cardiovascular mathematics

The increasing power of mathematical models and methods, together with the increasing computational power of modern architectures, has led cardiovascular mathematics to become not only a descriptive tool, but more importantly, a tool for diagnosis and treatment design [74, 93]. *In silico* experiments (that is, numerical simulations of possible scenarios) have become an extremely useful resource to analyze a pathology and investigate possible solutions. The advantage of *in silico* experiments compared to *in vitro* experiments is twofold: i) it does not require invasive testing, which for certain patologies and certain regions of the human body may be dangerous, and ii) when investigating possible treatments, it allows to consider and compare multiple solutions *before* they are concretely implemented. This not only reduces the risk of experimental treatments, but sometimes also their costs (for instance, when developing new devices).

However, in order for the numerical results to really help the diagnosis and treatment of the pathology of a patient, two requirements must be satisfied:

- the *inputs* of the mathematical model (such as geometry, parameters, boundary conditions) must be the ones corresponding to the current patient.
- the numerical method must give a solution which is reliable.

As we mentioned in the previous section, advancement in imaging techniques and image processing have made it possible to create geometries (i.e., computational domains) that are specific to the patient at hand [72]. Nevertheless, some of the other inputs are usually not known, or known with limited precision. For instance, when simulating the motion of blood in a vessel, the full velocity (or stress) profile at the inflow is usually not known, and all that is available (from medical

measurements) is some average quantity (such as the pressure or flow rate). Another example is given by the mechanical properties of the vessel: in a linearly elastic framework, these are condensed into two parameters, the Young's modulus and the Poisson's ratio. When simualting a blood flow in a compliant vessel, in order to solve the FSI problem arising from the interaction between blood and vessel wall, these parameters must be known. If the patient is known to be in "good" condition, usually an average value taken from literature is used. This not only gives up the specificity of the numerical experiment, but it may give misleading results in case the patient is in fact not in good condition. As a matter of fact, the ideal scenario would be to have the (unknown) parameters as part of the solution of the problem. This is one of the goals of Data Assimilation (DA), which we will present in more detail in Chapter 2.

The issue of the reliability of the numerical solution is, on the other hand, more subtle. When we use numerical methods to approximate the solution to a problem, say a PDE, we introduce some *discretization parameter*. The method is then said to be *convergent* if, as the discretization parameters vanish, the numerical solution approaches the solution of the original problem (for a more precise definition of these concepts, see, for instance, [75]). In practice, the dimension of the discrete problem increases (often rapidly) as these parameters approach zero, and therefore one typically has some constraints on the minimum magnitude of the discretization parameters. Still, the convergence analysis of the method usually gives, up to some constants, an estimate of the error in the solution.

However, even if the difference between the numerical and the exact solutions is small (measured with respect to some norm), the numerical solution may have some unphysical features. In some cases, these are quite easy to identify, such as instabilities in convection-dominated flows, and suitable *stabilization techniques* can be adopted to fix the problem [79]. The situation is more complicated when the computed solution does not appear wrong, but still features some important differences when compared with the exact solution. An important example is given by the numerical simulation of turbulent flows on underresolved meshes, that is, meshes with element size h larger than the size of the smallest eddies in the flow.

The result is a solution that, despite looking correct, does not dissipate enough energy. In Chapter 5 we will investigate this issue more in detail, showing how an under-resolved numerical solution can remarkably differ from the exact one (in our case, coming from experimental measures).

The goals (and also challenges) of cardiovascular mathematics are therefore to provide results that are reliable and more patient specific. This can be done with a better integration of the mathematical models with the information coming from experimental measures (which we call Data Assimilation) and with the development of more robust methods, which guarantee that the numerical solutions have all the important features of the exact ones. We point out that these are not easy tasks and the computational costs associated may be large, which forces us to focus particularly on the performance of the algorithm when developing new approaches.

1.3 Thesis outline

In this thesis we focus on two problems arising in cardiovascular applications. The first is the estimation of the compliance of a blood vessel using measures in time of the displacement of the wall. The second is the reliability of the numerical solutions of flows with moderately large Reynolds numbers. In particular, the original contributions of this work are presented in **Chapters 4** and **5**.

We start in **Chapter 2** with a quick overview of Data Assimilation (DA). First, we explain what DA does and what its goal is. Then, we will present the two most common approaches currently used to implement DA in practice, which are the *Kalman filter* and the *variational method*. We will focus mostly on the latter, since it will be used in the following chapters. Finally, we will discuss how DA can be beneficial for cardiovascular applications.

In **Chapter 3** we will introduce the problem of the (variational) estimation of the compliance of a vessel, following the idea presented in [69]. This application of DA will lead to an *Inverse Fluid-Structure Interaction* (IFSI) problem, which can be casted in the framework of constrained optimization. We will briefly discuss the

issues related to the solution of the optimization problem, such as the choice between *discretize-optimize* and *optimize-discretize* approaches or the regularization of the problem. Finally, we will present some numerical results for some idealized geometries.

In **Chapter 4** we will present a Reduced Order Model approach for the solution of the inverse problem introduced in **Chapter 3**. In particular, we will adopt a Proper Orthogonal Decomposition (POD) strategy for the reduction of the costs associated with the optimization problem. We will present the algorithm and we will apply it to the same scenarios analyzed in **Chapter 3**, comparing the performance and finding the limitation of the algorithm.

In **Chapter 5** we will tackle the problem of the numerical solutions of the Navier-Stokes equations for moderately large Reynolds numbers. We will follow the idea presented in [53] and we will elaborate on it, reinterpreting the algorithm as an operator splitting technique for a perturbed equation. Moreover, we will also derive a heuristic formula to calibrate one of the important parameters of the method. We will then test the robustness of the method on a benchmark proposed by the Food and Drug Administration (FDA).

Concluding remarks and suggestions for future research directions will be addressed in **Chapter 6**.

Chapter 2

Data Assimilation: methods, examples and applications to cardiovascular mathematics

This chapter contains a brief and general introduction to the topic of Data Assimilation (DA) and some of the techniques used for its application to real problems. In particular, in Section 2.1 we will motivate the use of DA for a wide class of problems, and we will formulate a simple reference problem. After discussing in Section 2.2 the two possible approaches that can be used in DA, namely probabilistic and deterministic, in Sections 2.3 and 2.4 we will give a brief overview of the two main techniques used in practice. Finally, in Section 2.5 we will discuss how DA can improve the quality of a mathematical study of the cardiovascular system.

We mention that parts of this chapter are inspired by the work presented in [54], where further details on the topic can be found.

2.1 Introduction

Most of the practical problems in the field of applied sciences are described using mathematical models. These models are usually derived from basic principles, such as conservation laws, and by a set of constitutive equations involving the state of the system, a set of parameters, and input data. For instance, the motion of a

rigid body in a gravity field with no friction can be described by a simple system of differential equations, involving the position and velocity of the body in time, the mass of the body and the gravity field acting on the body. In this case, the state of the problem is given by position and velocity, the only parameter is the mass of the body and the input datum is the gravity field. Another example is the evolution of the distribution of currents and voltages in a power grid, where the state is given by the voltages and currents across each component of the grid, the parameters are the values of resistance, capacitance and inductance of each component, while the input data are the voltages or currents imposed through generators. In this case, the set of equations includes both differential and algebraic equations (DAE).

In general, we can denote the set of equations governing the problem with

$$\mathcal{A}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{b}) = 0, \tag{2.1}$$

where $\mathbf{x}$ is the state of the problem, $\boldsymbol{\mu}$ is the vector of the parameters of the model, and $\mathbf{b}$ contains the input data that determine the particular configuration for which we have to find a solution.

Remark 2.1. It is worth clarifying the distinction between parameters $\boldsymbol{\mu}$ and input data $\mathbf{b}$, which may otherwise seem vague and arbitrary. In this context, we think of a parameter as a fixed *intrinsic* quantity of one (or more) component of the system, while an input datum is an *extrinsic* given quantity, which is more related to the particular instance of the problem rather than to a component of the system. For instance, if we are to consider an incompressible fluid flowing inside a rigid pipe, the density and viscosity of the fluid would be considered parameters, while the initial velocity distribution or the conditions imposed at the boundary of the computational domain would be considered as given input data.

The quality and reliability of a model can be assessed by comparing the solution of the mathematical model with measurements that are taken on the system. For some problem (such as the motion of a rigid body) this can be done directly on the state $\mathbf{x}$. However, in general, this can only be done on some *derived* or *output* quantity $\mathbf{y}$, which depends on the state $\mathbf{x}$ and the parameters $\boldsymbol{\mu}$ through a relation

that we denote as

$$\mathbf{y} = \mathcal{H}(\mathbf{x}, \boldsymbol{\mu}). \tag{2.2}$$

This relation could consist of a simple algebraic expression, or a more complicated set of integro-differential equations. However, we require that the map $\mathcal{H}$ is a function, that is, for each pair state-parameter $(\mathbf{x}, \boldsymbol{\mu})$, the quantity $\mathbf{y}$ must be uniquely defined.

One of the most important applications of such mathematical models is predicting the outputs of interest $\overline{\mathbf{y}}$ for a specific configuration corresponding to a given set of parameters $\overline{\boldsymbol{\mu}}$ and input data $\overline{\mathbf{b}}$. In particular, once the mathematical model reliability has been established through comparisons with experiments and available data, one is typically interested in using its predicting power in case the value of $\mathbf{y}$ is required for a large number of configurations and/or if the direct measurement of $\mathbf{y}$ is not feasible (for instance, because the experiment is too expensive, or too dangerous). This allows us to use the mathematical model to discover *new* information.

On the other hand, information coming from observations, measures and experiments also contains useful information, and can be used to improve the quality of the mathematical model. As a matter of fact, no mathematical model can perfectly describe the reality we observe, due to the simplifications and the assumptions that are made to derive it. Furthermore, as mentioned before, mathematical models usually contain parameters $\boldsymbol{\mu}$ and input terms $\mathbf{b}$, whose numerical values depend on the specific problem under consideration. The calibration of such quantities can be a complex task, and the value that it is used to solve a particular instance of the problem may be known only with limited precision.

Remark 2.2. A mathematical model and the data can be interpreted as dual sources of knowledge. In particular, mathematical models can be thought of as *background knowledge*, since they are based on past experiments and analyses, and they do not depend on all the details of the particular experiment we are describing. On the other hand, information coming from data and measurements can be thought of as *foreground knowledge*, since it does not rely on any underlying justification (such

as models and theories), and, under specific circumstances, it has the property of reflecting the real behavior of the system observed. For instance, patient-specific measurements are an instance of foreground knowledge. As for any measurement process, this knowledge is affected by noise and errors. On the other hand, a mathematical model for blood flow represents a background knowledge that can be applied to many patients once the right combination of parameters is used

DA is a set of techniques that aims to use information coming from data (e.g., experimental measures) to improve the quality of the mathematical model and, as a consequence, of the computed solution. This approach, developed in the 20^{th} century, has been used massively in weather forecast applications as well as in geophysics [3]. The merging of the *background* and *foreground* knowledges has the double effect of improving the reliability of the quantitative results of the mathematical model and also to partially remove noise and errors that are systematically present in the data (for instance, due to the limited resolution of measuring devices). If we denote the state of the system with $\mathbf{x}$, some output of interest with $\mathbf{y}$, the model parameters with $\boldsymbol{\mu}$ and the (in general, limited and noisy) available data with $\mathbf{d}$, we can identify at least three possible ways in which DA can be used:

i) *reconstruction*: $\mathbf{x}$ and/or $\mathbf{y}$ are reconstructed in full starting from $\mathbf{d}$. This can also be interpreted as a *denoising* of the data $\mathbf{d}$.

ii) *prediction*: using data $\mathbf{d}$ from previous experiments, the model is used to predict the value of $\mathbf{x}$ and/or $\mathbf{y}$ for a new experiment.

iii) *estimation*: $\boldsymbol{\mu}$ is estimated by forcing the model output $\mathbf{y}$ to be "close" to the observed output $\mathbf{d}$, where the meaning of "close" depends on the context.

Regardless of which one of the above is the ultimate goal, DA helps to improve the reliability of the quantitative solutions, by tuning the model in such a way that it better explains the data coming from experiments and measures.

In Section 2.2 we present the two different approaches that can be used in DA, while in Sections 2.3 and 2.4 we will introduce the two most common algorithms

used in practice for DA problems. Then, in Section 2.5 we will briefly discuss what is the current and potential impact of DA in computational hemodynamics.

2.2 Stochastic and deterministic approaches

By looking at the approaches that can be used in DA, we can identify two main categories:

- *stochastic* approaches: in this case, the goal is to find probability distributions and confidence intervals for the system state $\mathbf{x}$ and output $\mathbf{y}$, starting from some *a priori* probabilistic knowledge on the model and the data.
- *deterministic* approaches: in this case, one is typically interested in reconstructing, predicting or estimating with no a priori probabilistic knowledge of the problem of interest.

It is worth noting that, in the second case, one can still incorporate statistical knowledge about the model, as we will discuss in Section 2.4.2.

Among the methods used in the two approaches, of particular importance are the Kalman filter (with its extensions) for the stochastic approach and the variational method for the deterministic one. These constitute the subject of the next two sections.

To make the description of the methods easier, we introduce a model problem to which we will apply the Kalman filter and the variational method. We will consider the following simple (discrete) dynamical problem:

$$\begin{cases} \mathbf{x}^n &= \mathrm{A}(\boldsymbol{\mu})\mathbf{x}^{n-1} + \mathbf{b}^n(\boldsymbol{\mu}) + \boldsymbol{\varepsilon}^n \\ \mathbf{y}^n &= \mathrm{H}(\boldsymbol{\mu})\mathbf{x}^n + \boldsymbol{\nu}^n \end{cases} \tag{2.3}$$

endowed with the initial condition $\mathbf{x}^0 = \mathbf{g} + \boldsymbol{\varepsilon}^0$, with $\mathbf{g}$ given. Here, the first equation represents how the current state $\mathbf{x}^n$ is affected by the previous state $\mathbf{x}^{n-1}$ through the linear operator A, the parameters $\boldsymbol{\mu}$ and the given input $\mathbf{b}^n$. Since, as we mentioned before, no model is perfect, we also have a term $\boldsymbol{\varepsilon}^{n-1}$ that represents errors

in the model. We assume this term behaves as *white noise* and is temporally uncorrelated, that is, $\boldsymbol{\varepsilon}^n \sim \mathcal{N}(\mathbf{0}, \mathrm{Q}_n)$ and $\mathbb{E}\left[\boldsymbol{\varepsilon}^i \boldsymbol{\varepsilon}^j\right] = \delta_{ij}\mathrm{Q}_j$, where $\mathbb{E}$ is the *expectation* operator and δ_{ij} is the Kronecker symbol. The second equation, instead, represents the measurement process: the matrix H, usually called the *observation matrix*, represents how the output $\mathbf{y}^n$ depends on the state $\mathbf{x}^n$ according to the model, while $\boldsymbol{\nu}^n \sim \mathcal{N}(\mathbf{0}, \mathrm{R}_n)$ represents errors in the measures (for instance, due to limitations on the measuring devices). Also for $\boldsymbol{\nu}^n$ we assume that vectors corresponding to different times are statistically uncorrelated. Furthermore, we will assume that a set of measurements $\mathbf{d}^n$ of the output $\mathbf{y}^n$ is available for every n.

Remark 2.3. In (2.3) we are assuming that the parameters of the model are known *exactly*. If this is not the case, as in a parameter estimation problem, dynamics for the parameters have to be introduced. In particular, for time-independent parameters, we can assume $\boldsymbol{\mu}$ to have a stochastic dynamics of the form

$$\boldsymbol{\mu}^n = \boldsymbol{\mu}^{n-1} + \boldsymbol{\eta}^n \tag{2.4}$$

with $\boldsymbol{\eta}^n \sim \mathcal{N}(\mathbf{0}, \mathrm{S}_n)$ not correlated with $\boldsymbol{\nu}^n$ and $\boldsymbol{\varepsilon}^n$. Therefore, we can write the problem as

$$\begin{cases} \mathbf{x}^n &= \mathrm{A}(\boldsymbol{\mu}^n)\mathbf{x}^{n-1} + \mathbf{b}^n(\boldsymbol{\mu}^{n-1}) + \boldsymbol{\varepsilon}^n \\ \boldsymbol{\mu}^n &= \boldsymbol{\mu}^{n-1} + \boldsymbol{\eta}^n \\ \mathbf{y}^n &= \mathrm{H}(\boldsymbol{\mu}^n)\mathbf{x}^n + \boldsymbol{\nu}^n \end{cases} \tag{2.5}$$

This is a non-linear problem, due to the coupling between the state $\mathbf{x}$ and the parameters $\boldsymbol{\mu}$ (unless both A and H do not depend on $\boldsymbol{\mu}$).

2.3 The Kalman Filter

The Kalman Filter (KF) was first introduced for linear problems [49], but has then been extended to non-linear problems with the so called Extended Kalman Filter [44] and Unscented Kalman Filter [48]. An exhaustive analysis of these methods is beyond the scope of this work, and we refer to the literature for further information

[44,48,49]. Here, we limit to a brief overview of the idea behind the KF, applying it to the linear problem (2.3) with the dependency on $\boldsymbol{\mu}$ dropped for the sake of notation.

The KF can be considered as a two-step method. The first is a *prediction* step, whose solution we denote with $\mathbf{x}_p^n$, while the second is a *correction* step, whose solution we denote with $\mathbf{x}_c^n$. In the prediction step, the system evolves according to the deterministic model, that is, without considering the presence of the model errors $\boldsymbol{\varepsilon}^n$. The ideal step would be

$$\mathbf{x}_p^n = \mathrm{A}\mathbf{x}^{n-1} + \mathbf{b}^n. \tag{2.6}$$

However, since the "true" solution $\mathbf{x}^{n-1}$ is not available, the state evolves according to the *best* information available on $\mathbf{x}^{n-1}$, that is,

$$\mathbf{x}_p^n = \mathrm{A}\mathbf{x}_c^{n-1} + \mathbf{b}^n. \tag{2.7}$$

In the correction step, the information coming from the data is used to update $\mathbf{x}_p^n$. More precisely, the KF performs a correction of the form

$$\mathbf{x}_c^n = \mathrm{L}_n\mathbf{x}_p^n + \mathrm{K}_n\mathbf{d}^n \tag{2.8}$$

where the matrix K_n, which is called *Kalman gain matrix*, determines how the information of the data not captured by the model prediction should affect the correction of the state, and it is clearly crucial for the success of the whole method. Forcing the correction to be unbiased[1], it can be shown that the step can be rewritten as

$$\mathbf{x}_c^n = \mathbf{x}_p^n + \mathrm{K}_n\left(\mathbf{d}^n - \mathrm{H}\mathbf{x}_p^n\right). \tag{2.9}$$

The derivation of the expression for K_n is not difficult, but requires a few intermediate steps. We just present the final result, and we refer to [49, 54, 96] for more details on the derivation, noting that the Kalman gain matrix K_n can be derived based on an optimality criterion. In particular, let us define the prediction and cor-

[1] That is, $\mathbb{E}(\mathbf{x}_c^n) = \mathbf{x}^n$, where $\mathbb{E}(\cdot)$ is the expectation operator.

rection errors and their corresponding covariance matrices:

$$\mathbf{e}_p^n = \mathbf{x}^n - \mathbf{x}_p^n \tag{2.10}$$

$$\mathrm{P}_n = \mathbb{E}\left[\mathbf{e}_p^n (\mathbf{e}_p^n)^T\right] \tag{2.11}$$

$$\mathbf{e}_c^n = \mathbf{x}^n - \mathbf{x}_c^n \tag{2.12}$$

$$\mathrm{C}_n = \mathbb{E}\left[\mathbf{e}_c^n (\mathbf{e}_c^n)^T\right]. \tag{2.13}$$

Then, an expression for K_n can be obtained by minimizing the expectation of the squared norm of the correction step error, that is $\mathbb{E}\left[(\mathbf{e}_c^n)^T \mathbf{e}_c^n\right]$ (which is equivalent to minimize the trace of C_n). An iteration of the KF method can then be written in the following form

$$\text{P-step} \begin{cases} \mathbf{x}_p^n &= \mathrm{A}\mathbf{x}_c^{n-1} + \mathbf{b}^n \\ \mathrm{P}_n &= \mathrm{A}\mathrm{C}_{n-1}\mathrm{A}^T + \mathrm{Q}_n \end{cases} \tag{2.14}$$

$$\text{C-step} \begin{cases} \mathrm{K}_n &= \mathrm{P}_n \mathrm{H}^T \left(\mathrm{H}\mathrm{P}_n \mathrm{H}^T + \mathrm{R}_n\right)^{-1} \\ \mathbf{x}_c^n &= \mathbf{x}_p^n + \mathrm{K}_n \left(\mathbf{d}^n - \mathrm{H}\mathbf{x}_p^n\right) \\ \mathrm{C}_n &= (\mathrm{I} - \mathrm{K}_n \mathrm{H})\mathrm{P}_n \end{cases} \tag{2.15}$$

Notice that the algorithm also requires an initial condition for C_0. If $\mathbf{x}^0$ is known exactly, then one can choose $\mathrm{C}_0 = \mathrm{I}$, otherwise it must be constructed using the statistical information on the error of the initial condition $\mathbf{x}^0$.

Remark 2.4. If we assume that $\mathrm{H} \in \mathbb{R}^{m \times n}$ with $rank(\mathrm{H}) = m$, and we drop all the stochastic terms in the expression of K_n, we obtain $\mathrm{K}_n = \mathrm{H}^\dagger$, where $\mathrm{H}^\dagger$ denotes the Moore-Penrose pseudo-inverse of H [43]. This is consistent, since

$$\mathbf{e}_p^n = \mathbf{x}^n - \mathbf{x}_p^n \quad \Longrightarrow \quad \mathrm{H}\mathbf{e}_p^n = \mathrm{H}\mathbf{x}^n - \mathrm{H}\mathbf{x}_p^n = \mathbf{d}^n - \mathrm{H}\mathbf{x}_p^n$$

and therefore, the correction that would give a perfect match with the data is given by

$$\mathbf{e}_p^n = \mathrm{H}^\dagger(\mathbf{d}^n - \mathrm{H}\mathbf{x}_p^n). \tag{2.16}$$

Hence, in a deterministic setting, the correction step would reduce to a simple application of the Moore-Penrose inverse. The Kalman gain matrix K_n can thus be interpreted as a version of the Moore-Penrose inverse adaptated to the DA problem and enhanced by statistical information about the underlying system dynamics (namely, the covariance matrices of the predicted status $\mathbf{x}_p^n$ and of the data $\mathbf{d}^n$).

We conclude this section by listing some of the interesting properties of the KF. We will make use of the following statistical scalar product:

$$\langle \mathbf{u}, \mathbf{v} \rangle_s = tr\left(\Sigma_v^{-1/2}\Sigma_u^{-1/2}\mathbb{E}(\mathbf{u}\mathbf{v}^T)\right). \tag{2.17}$$

where Σ_v and Σ_u are the covariance matrices of $\mathbf{u}$ and$\mathbf{v}$ respectively, and $tr(\mathrm{A})$ denotes the trace of A. According to this definition, for two vectors to be orthogonal is equivalent to be statistically uncorrelated. The following properties hold (see, e.g., [96]):

i) $\langle \mathbf{u}_c^n, \mathbf{e}_c^n \rangle_s = 0$: the error of the corrected solution is orthogonal to the solution itself.

ii) $\langle \mathbf{d}^n - \mathrm{H}\mathbf{u}_p^n, \mathbf{d}^{n-j} - \mathrm{H}\mathbf{u}_p^{n-j} \rangle_s = 0, \forall j \geqslant 1$: the innovation added at the iteration n is orthogonal to the iteration added at the previous iteration.

iii) $\mathrm{P}_n - \mathrm{C_n} \succ 0$: the innovation reduces the variance of the output of the model[2].

iv) given $\mathbf{u}^0, \ldots, \mathbf{u}^{n-1}$, the solution $\mathbf{u}_c^n$ obtained with the KF minimizes the functional

$$\mathcal{J} = \frac{1}{2}\sum_{k=1}^{n} \|\mathbf{u}^k - \mathrm{A}\mathbf{u}^{k-1}\|^2_{\mathrm{Q}_n^{-1}} + \frac{1}{2}\sum_{k=1}^{n} \|\mathbf{d}^k - \mathrm{H}\mathbf{u}^k\|^2_{\mathrm{R}_n^{-1}} \tag{2.18}$$

that underlines a connection between the KF method and the variational method (which we will present in the next section).

Remark 2.5. Here we presented an example of the Kalman filter applied to a *reconstruction* problem, that is, a problem where we reconstructed the state of the system $\mathbf{x}$ from a set of measurements $\mathbf{d}$. As we mentioned in the previous section,

[2]The notation $\mathrm{A} \succ 0$ means that the matrix A is positive definite.

DA can also be used for *estimation* problems, where some of the parameters of the model are not known, such as in (2.5). For this kind of problems, the KF method has to be adapted in order to deal with the non-linearities of the problem.

2.4 Variational method

The variational method is an approach for DA that aims to assimilate the data by means of solving a suitable constrained optimization problem. Usually, the optimization problem consists of the minimization of a functional measuring (in a proper norm) the misfit between the available data and the computed solution, using as a constraint the mathematical model. More precisely, in an abstract framework, let us define a functional $\mathcal{F}$

$$\mathcal{F}(\mathbf{x}, \boldsymbol{\mu}) = \|\mathbf{d} - \mathcal{H}(\mathbf{x}, \boldsymbol{\mu})\|^p \tag{2.19}$$

where $\|\cdot\|$ denotes a suitable norm, and[3] $p \geqslant 1$. For instance, for time-dependent PDE's, $\|\cdot\|$ could be the $L^2(t^0, t^N; L^2)$ norm and $p = 2$. Then, we can write the optimization problem as

$$\begin{aligned} &\min_{u \in \mathcal{U}} \mathcal{F}(\mathbf{x}, \boldsymbol{\mu}) \\ &\text{s.t. } \mathcal{A}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{b}, u) = 0. \end{aligned} \tag{2.20}$$

Here, u is a control variable and $\mathcal{U}$ is the set of admissible values. The choice of what u should be is not obvious and is an issue that does not appear when using the KF method, where the only requirement is some probabilistic knowledge about the distribution of $\boldsymbol{\mu}$, while the assimilation process is carried out by means of a fixed set of linear algebra operations.

To understand how to choose u, it is useful to look at the model problem (2.3). When using the variational method, we are assuming no knowledge of the errors (in both the model and the data), that is, we are dropping the stochastic terms $\boldsymbol{\varepsilon}^n$

[3] Recall that minimizing $f(x)$ is equivalent to minimize $g(f(x))$, for every strictly convex function g.

and $\boldsymbol{\nu}^n$. Nevertheless, we are still aware of the fact that the model is not perfect, meaning that the values for $\boldsymbol{\mu}$ or $\mathbf{b}^n$ may not correspond to the real ones. This suggests that we could use as control variables the subset of $\boldsymbol{\mu}$ and $\mathbf{b}$ whose values we are not confident about. The idea is that the information in the data, through the minimization process, will drive those values towards the "correct" ones.

Remark 2.6. Notice that, at this stage, there is no distinction between linear and non-linear problems; the differences arising from the nature of the problem are deferred to the choice of the routines used to minimize the cost functional. In particular, in the framework of the variational method, there is no distinction in the algorithm between prediction, reconstruction or estimation problems, since the approach is formally equivalent. This marks one difference between the variational method and the KF method, where, due to the nonlinearities, the case of parameter estimation had to be treated with the EKF or UKF methods.

To give a more concrete example, let us consider again the problem (2.3). Suppose, for simplicity, that the matrices A and H do not depend on the parameters $\boldsymbol{\mu}$, which are the only source of uncertainty in the problem. We consider the functional

$$F(\mathbf{x}) = \sum_{n=0}^{N} w_n \|\mathbf{d}^n - \mathrm{H}\mathbf{x}^n\|^2, \tag{2.21}$$

where $\sum_{n=0}^{N} w_n = 1$. Notice that this is the discrete counterpart of

$$\mathcal{F}(\mathbf{x}) = \int_{t^0}^{t^N} \|\mathbf{d} - \mathcal{H}(\mathbf{x})\|^2 dt \tag{2.22}$$

where the integral is approximated with the quadrature formula

$$\int_{t^0}^{t^N} \|\mathbf{u}(t)\|^2 dt \simeq \sum_{n=0}^{N} w_n \|\mathbf{u}^n\|^2. \tag{2.23}$$

The data assimilation problem would then read

$$\begin{aligned} &\min_{\boldsymbol{\mu}\in\mathcal{U}} F(\mathbf{x}) \\ &\text{s.t. } \mathbf{x}^n = \mathrm{A}\mathbf{x}^{n-1} + \mathbf{b}^n(\boldsymbol{\mu}), \quad n = 1,\dots,N. \end{aligned} \tag{2.24}$$

The solution of the minimization problem (especially in the nonlinear case) is itself a challenging task. Several methods have been developed for such problems, such as the simplex method (only for the linear case), sequential quadratic programming or Lagrange multipliers to mention a few (see [65] for more details). Among all the possible methods, the last one is arguably the most widely used. Since this is the method we are going to use in the following chapters, we briefly recall it in the next section.

2.4.1 The method of Lagrange multipliers

Consider the general minimization problem

$$\begin{aligned} &\min_{\mathbf{u}\in\mathcal{U}} F(\mathbf{x},\mathbf{u}) \\ &\text{s.t. } \mathrm{A}(\mathbf{x},\mathbf{u}) = \mathbf{b}(\mathbf{u}) \end{aligned} \tag{2.25}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state variable, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^k$ is the control variable, A is a (possibly nonlinear) operator from $\mathbb{R}^n \times \mathbb{R}^k$ to $\mathbb{R}^m$ and F is a function from $\mathbb{R}^n \times \mathbb{R}^k$ to $\mathbb{R}$. For simplicity, we will assume that both A and F are differentiable for every $\mathbf{x}$ and $\mathbf{u}$. Let us define the *Lagrangian* functional as

$$\mathcal{L}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = F(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T(\mathbf{b}(\mathbf{u}) - \mathrm{A}(\mathbf{x},\mathbf{u})). \tag{2.26}$$

The new variable $\boldsymbol{\lambda}$ is called the *Lagrange multiplier*, and its dimension equals the number of constraints in the problem, that is, $\boldsymbol{\lambda} \in \mathbb{R}^m$. It is important to notice that, at this point, $\mathbf{x}$, $\mathbf{u}$ and $\boldsymbol{\lambda}$ are considered *independent* variables. If we differentiate $\mathcal{L}$ with respect to each variables, and set the partial derivatives to zero, we

obtain the following system of equations

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}}(\mathbf{x},\mathbf{u}) - \frac{\partial \mathrm{A}^T(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\boldsymbol{\lambda} = \mathbf{0} \tag{2.27}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{b}(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{0} \tag{2.28}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{b}(\mathbf{u}) - \mathrm{A}(\mathbf{x},\mathbf{u}) = \mathbf{0}. \tag{2.29}$$

This system is usually referred to as the Karush-Kuhn-Tucker (KKT) conditions[4], or KKT system, and every pair $(\overline{\mathbf{x}},\overline{\mathbf{u}})$ that is optimal for the problem (2.25) must also satisfy these equations (for an opportune Lagrange multiplier $\overline{\boldsymbol{\lambda}}$). In other words, the KKT conditions are *necessary* conditions for optimality. Equation (2.27) is called the *adjoint* equation, (2.28) is the *optimality* condition, while (2.29) is called the *state* equation, and it is always equal to the constraints of the minimization problem. Notice that, even in the non-linear case, the derivatives of A and $\mathbf{b}$ in equations (2.27)-(2.28) are always linear when considered as operators acting on $\boldsymbol{\lambda}$ (i.e., they are matrices).

Let us now extend the Lagrange multipliers method to the minimization problem (2.24) where F is defined in (2.21). Here, the control variable $\mathbf{u}$ coincides with the parameter $\boldsymbol{\mu}$. Notice that, since we have N sets of equations for the constraints, we will also have N different Lagrange multipliers $\boldsymbol{\lambda}^n$, $n = 1,\dots,N$. The Lagrangian would then read

$$\begin{aligned}\mathcal{L}(\mathbf{x}^0,\dots,\mathbf{x}^N,\boldsymbol{\mu},\boldsymbol{\lambda}^0,\dots,\boldsymbol{\lambda}^N) = \\ w_0\|\mathbf{d}^0 - \mathrm{H}\mathbf{x}^0\|^2 + (\boldsymbol{\lambda}^0)^T(\mathbf{g}-\mathbf{x}^0) + \\ \sum_{n=1}^{N}\left(w_n\|\mathbf{d}^n - \mathrm{H}\mathbf{x}^n\|^2 + (\boldsymbol{\lambda}^n)^T(\mathrm{A}\mathbf{x}^{n-1} + \mathbf{b}^n(\boldsymbol{\mu}) - \mathbf{x}^n)\right).\end{aligned} \tag{2.30}$$

As already noted, the state equation corresponds to the constraint of the minimization problem. The optimality condition can be obtained by differentiating

[4]The KKT conditions also apply in the more general setting of inequality constraints [65].

with respect to the control variable $\boldsymbol{\mu}$, yielding

$$\sum_{n=1}^{N} (\boldsymbol{\lambda}^n)^T \frac{\partial \mathbf{b}^n}{\partial \boldsymbol{\mu}} = 0. \tag{2.31}$$

The adjoint equation is more delicate, since, for $1 \leqslant n \leqslant N-1$, there are two equations involving $\mathbf{x}^n$, namely the equation that expresses $\mathbf{x}^n$ in terms of $\mathbf{x}^{n-1}$ and the one that expresses $\mathbf{x}^{n+1}$ in terms of $\mathbf{x}^n$. In addition, $\mathbf{x}^0$ appears in one equation and in the initial condition. Paying attention to this detail, differentiating gives

$$-2w_n \mathrm{H}^T(\mathbf{d}^n - \mathrm{H}\mathbf{x}^n) + \mathrm{A}^T\boldsymbol{\lambda}^{n+1} - \boldsymbol{\lambda}^n = \mathbf{0}, \quad n = 0\ldots, N-1 \tag{2.32}$$

$$-2w_N \mathrm{H}^T(\mathbf{d}^N - \mathrm{H}\mathbf{x}^N) - \boldsymbol{\lambda}^N = \mathbf{0}. \tag{2.33}$$

Notice that the adjoint equation has the structure of a problem backward in time and, in order to solve it, the value of $\mathbf{x}^N$ must be computed first. The full KKT system for this problem reads

$$\text{(State)} \begin{cases} \mathbf{x}^n &= \mathrm{A}\mathbf{x}^{n-1} + \mathbf{b}^n(\boldsymbol{\mu}), \quad n = 1, \ldots, N \\ \mathbf{x}^0 &= \mathbf{g} \end{cases} \tag{2.34}$$

$$\text{(Optimality)} \quad \sum_{n=1}^{N} (\boldsymbol{\lambda}^n)^T \frac{\partial \mathbf{b}^n}{\partial \boldsymbol{\mu}} = 0 \tag{2.35}$$

$$\text{(Adjoint)} \begin{cases} \boldsymbol{\lambda}^n &= \mathrm{A}^T\boldsymbol{\lambda}^{n+1} - 2w_n \mathrm{H}^T(\mathbf{d}^n - \mathrm{H}\mathbf{x}^n) \quad n = N-1, \ldots, 0 \\ \boldsymbol{\lambda}^N &= -2w_N \mathrm{H}^T(\mathbf{d}^N - \mathrm{H}\mathbf{x}^N) \end{cases} \tag{2.36}$$

If $\mathbf{b}(\boldsymbol{\mu})$ is linear, then the KKT system is also linear. Notice, however, that, despite being symmetric, KKT systems are always indefinite, so one cannot use methods such as the Conjugate Gradient, but must resort to more general methods, such as GMRES or MINRES. If $\mathbf{b}(\boldsymbol{\mu})$ is nonlinear, then the KKT system is also nonlinear, due to the optimality condition, and nonlinear routines must be used to tackle the problem. There is a vast literature on ways to tackle the non-linear KKT system [5,6, 39]. In particular, we can identify two main approaches: an all at once approach, in which a Newton or quasi-Newton method is used to solve the whole KKT system,

and the so called reduced space (RS) approach, in which the problem is formally expressed in terms of the control variable [38].

Since in the following chapters we will make use of the second approach, let us show on this simple problem how the KKT conditions can be used to compute the gradient of the functional F. Using the chain rule, we can write

$$\frac{\partial F(\mathbf{x}(\boldsymbol{\mu}))}{\partial \boldsymbol{\mu}} = \frac{\partial F}{\partial \mathbf{x}^n}\frac{\partial \mathbf{x}^n}{\partial \boldsymbol{\mu}}, \tag{2.37}$$

where a summation over n is intended on the right hand side. The first factor reads

$$\frac{\partial F}{\partial \mathbf{x}^n} = -2w_n \mathrm{H}^T(\mathbf{d}^n - \mathrm{H}\mathbf{x}^n).$$

For the second term, we use implicit differentiation on the constraint, to obtain

$$\frac{\partial \mathbf{x}^n}{\partial \boldsymbol{\mu}} = \mathrm{A}\frac{\partial \mathbf{x}^{n-1}}{\partial \boldsymbol{\mu}} + \frac{\partial \mathbf{b}^n}{\partial \boldsymbol{\mu}}$$

For the sake of notation brevity, in the following let $\Delta_n = -2w_n \mathrm{H}^T(\mathbf{d}^n - \mathrm{H}\mathbf{x}^n)$ and $\mathrm{B}_n = \partial_{\boldsymbol{\mu}}\mathbf{b}^n$. Then, the previous recurrence equation can be solved, obtaining

$$\frac{\partial \mathbf{x}^n}{\partial \boldsymbol{\mu}} = \sum_{k=0}^{n-1} \mathrm{A}^k \mathrm{B}_{n-k}, \quad \frac{\partial \mathbf{x}^0}{\partial \boldsymbol{\mu}} = \mathrm{O} \tag{2.38}$$

and therefore

$$\frac{\partial F(\mathbf{x}(\boldsymbol{\mu}))}{\partial \boldsymbol{\mu}} = \sum_{n=1}^{N} \Delta_n \sum_{k=0}^{n-1} \mathrm{A}^k \mathrm{B}_{n-k}. \tag{2.39}$$

Rearranging so that we can factor out the terms B_j with the same index, we obtain

$$\frac{\partial F(\mathbf{x}(\boldsymbol{\mu}))}{\partial \boldsymbol{\mu}} = \sum_{n=1}^{N} \left(\sum_{k=n}^{N} \Delta_k \mathrm{A}^{k-n} \right) \mathrm{B}_n. \tag{2.40}$$

Finally, we claim that the term in parentheses is precisely $(\boldsymbol{\lambda}^n)^T$. In fact,

$$\boldsymbol{\lambda}^n = \mathrm{A}^T\boldsymbol{\lambda}^{n+1} + \Delta_n = \left(\mathrm{A}^T\right)^2 \boldsymbol{\lambda}^{n+2} + \mathrm{A}^T\Delta_{n+1} + \Delta_n = \ldots = \sum_{k=n}^{N} \mathrm{A}^{k-n}\Delta_k, \tag{2.41}$$

where we used the fact that $\boldsymbol{\lambda}^N = \Delta_N$.

Therefore, instead of solving the KKT system, one could just use it to compute the gradient of the functional, and use any unconstrained minimization routine to find the minimum (see, e.g., [38]). For instance, Algorithm 1 shows how this can be done using the BFGS [65] algorithm for the minimization.

Algorithm 1 BFGS with KKT-based gradient

1: $k = 0$, choose $\underline{\boldsymbol{\mu}}_0$ based on some *a priori* knowledge (otherwise randomly)
2: **while** $k < k_{\max}$ and *convergence test==false* **do**
3: Solve state equation with $\boldsymbol{\mu} = \boldsymbol{\mu}_k$ to compute $\mathbf{x}_k = [\mathbf{x}_k^0, \dots, \mathbf{x}_k^N]$
4: Solve adjoint equation with $\mathbf{x} = \mathbf{x}_k$ to compute $\boldsymbol{\lambda}_k = [\boldsymbol{\lambda}_k^0, \dots, \boldsymbol{\lambda}_k^N]$
5: Compute the gradient of F using (2.40)
6: Compute the descent direction $\boldsymbol{\delta}_k$ (using BFGS)
7: $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \tau_k \boldsymbol{\delta}_k$, with τ_k possible damping parameter
8: $k \leftarrow k + 1$
9: **end while**

2.4.2 Regularization

When facing non-linear minimization problems, a question that naturally arises is whether or not the functional has a minimum. Even when minimizing a smooth function in $\mathbb{R}^n$, this can be an issue. In fact, if the region of the admissible solutions is not bounded, the functional may not be bounded from below or, if bounded, may not achieve the minimum at any point. The same is true in the infinite dimensional Hilbert spaces setting.

Furthermore, in the context of DA, the data that we are trying to match are usually affected by noise, due for instance to measurement errors. We can write the data as

$$\mathbf{d} = \mathbf{d}^{true} + \boldsymbol{\nu} \tag{2.42}$$

where $\boldsymbol{\nu}$ is a white noise. In general, $\boldsymbol{\nu}$ does not lie in the space spanned by all the possible solutions to the constraint equations. Nevertheless, the properties of the minimization problem (2.24) deteriorate in the presence of noise, which may impact the convergence of the minimization routine towards the optimum (if any).

A common way to deal with this issue is to modify the functional, adding a term that penalizes admissible solutions with non-desired features. This technique is called *variational regularization*. The analysis of regularization techniques is be-

yond the scope of this work. Here we introduce only the concept of regularization and we refer to [23, 100] for more details.

The new functional to minimize can be written as

$$\mathcal{J}(\mathbf{x}, \mathbf{u}) = \mathcal{F}(\mathbf{x}, \mathbf{u}) + \alpha \mathcal{R}(\mathbf{u}) \tag{2.43}$$

where $\mathbf{u}$ is the control variable and $\alpha > 0$ is the *regularization parameter*, which determines how much the regularization term should affect the minimization process. To calibrate this parameter is not an easy task, and several methods have been proposed, such as Generalized Cross Validation, L-curve or the Discrepancy Principle (see, e.g., [23, 100]).

The choice of $\mathcal{R}$ may change depending on the application. A popular choice is given by *Tikhonov regularization*. In this case, the expression for $\mathcal{R}$ is

$$\mathcal{R} = \|L(\mathbf{u} - \mathbf{u}_{ref})\|^2 \tag{2.44}$$

where $\mathbf{u}_{ref}$ is a reference value for $\mathbf{u}$, and L is a semi-definite operator. The most frequent choices for L are the identity operator, which penalizes admissible solutions with large norm, hence enhancing the convexity of the functional, and the gradient operator, which penalizes highly oscillating solutions.

Another frequently used regularization is the *Total Variation*, given by

$$\mathcal{R}(\mathbf{u}) = \int_{\mathcal{U}} |\nabla \mathbf{u}| dx, \tag{2.45}$$

where $|\cdot|$ denotes the 2-norm.

Remark 2.7. There is a link between the concept of regularization and the a priori statistical knowledge on the solution that is used in the stochastic approach. As a matter of fact, it is possible to show that the Tikhonov regularization

$$\mathcal{R} = \|\mathbf{u} - \mathbf{u}_{ref}\|^2$$

is equivalent, in the stochastic framework, to provide a Gaussian *prior* probabil-

ity distribution for the control variable $\mathbf{u}$ with mean $\mathbf{u}_{ref}$ and covariance matrix $\Sigma = \alpha^2 \mathrm{I}$.

2.5 DA in hemodynamics

To conclude this chapter, we discuss the role that DA has in medicine, with particular attention to hemodynamics. In the last two decades, we have observed a great increase in the use of mathematics in medicine. In particular, when we focus on hemodynamics, we can identify at least two main areas:

- data acquisition: several methods have been developed (or improved) to extract useful information from the increasing amount of data acquired with new technologies, such as Magnetic Resonance Imaging (MRI), ultrasounds, Computer Tomography (CT), to mention a few. Among these methods we mention segmentation, deblurring and registration of images, which have allowed the use of more realistic and patient specific geometries in numerical simulations.

- mathematical framework: new models to describe the physiology of blood and tissues have been developed. In addition, new numerical methods have been devised to compute approximate solutions in a faster and more robust way. Among others, we mention advancements in electrocardiology, fluid-structure interaction and non-Newtonian models.

Both of these areas, separately, have dramatically increased the amount of information and knowledge available to doctors in order for them to better analyze the condition of patients and, ultimately, take better decisions on their treatment. However, both of these areas suffer from a systematic limit. On one hand, image (or, more generally, data) processing alone is limited by the fact of using general purpose (and somehow "static") methods, that is, without considering the fact that the underlying data should obey physical principles, eventually described by a precise mathematical model. On the other hand, mathematical models are limited by the presence of parameters and input data that must be properly tuned in order for the model to correctly describe the particular problem at hand.

However, these drawbacks can be compensated by the improvements in the other area. In particular, the mathematical framework can provide additional constraints to the data processing step, hence allowing for a better data reconstruction. Meanwhile, the acquisition and processing of patient specific data can provide a useful source of information to calibrate the model so that the parameters and input data are sufficiently close (in an appropriate sense) to the patient ones.

In this scenario DA becomes the perfect framework to extract only the most relevant information from the data, and use it to obtain reliable results from the mathematical models. The possible applications in hemodynamics are numerous. For example, the prescription of boundary conditions remains a major challenge for applications where the information available from measurements is not sufficient to close the problem. For instance, in [29, 98] the authors use an optimization approach to handle boundary conditions for flow problems where only the flow rate was available (i.e., the integral of the normal velocity over the inlet/outlet sections). For certain boundary conditions, closure models have been proposed to reduce the prescription of outflow boundary conditions to the opportune tuning of few constants. For instance, for outflow boundary conditions zero-dimensional circuit models have been proposed which require only a few parameters to be calibrated [28]. Nevertheless, the calibration of these parameters is not obvious, especially in cases corresponding to patients with non-standard pathologies. In [4], the Unscented Kalman Filter is used to sequentially estimate the parameters involved in the zero-dimensional circuit model by assimilating measures of the vessel wall displacement. Another application of DA was presented in [19], to the case where the inflow boundary conditions are not known. There, measurements of the velocity field at some locations in the computational domain were used to quantify the wall shear stress on the vessel wall.

In the next chapter we will present the problem of estimating the compliance of the vessel wall, by using a variational method.

Chapter 3

Variational estimation of the compliance of a blood vessel

3.1 Motivation

In this chapter we focus on the application of DA to parameter estimation, and, in particular, to the estimation of the compliance of a blood vessel. We start by giving a medical definition of compliance:

Definition 3.1. The compliance of a vessel, and in general of a hollow tissue, is its ability to change its size when an internal pressure is applied or, equivalently, as its tendency to resist recoil towards its original configuration once a compressing or distending force is removed.

For arteries, we can translate this definition into the more quantitative formula

$$C = \frac{\Delta V}{\Delta P}, \tag{3.1}$$

where ΔV denotes the change of the vessel volume induced by change in the blood pressure ΔP. The concept of compliance is therefore related to the mechanical properties of the vessel; it is of great interest from the medical point of view, since anomalies in its value can indicate the possible presence of pathologies, such as atherosclerotic plaques or tumors [36, 97].

If we consider the vessel wall to behave as an elastic material, then the compliance can be identified with the *Young's modulus* of the tissue. For biological tissues, this parameter can be estimated *in vitro* and also *in vivo* for the past 20 years, with a procedure called *elastography* [92]. This technique recovers the value of the Young's modulus by solving an *inverse elasticity problem*. In particular, two images of the tissue are compared: the first is an image of the tissue at rest, while the second is an image of the tissue deformed under the action of a *known* force. The estimate of the Young's modulus is given by the value that minimizes the mismatch between the observed deformation and the one given by the mathematical model. In the framework of the variational approach for DA introduced in Section 2.4, we can formulate the elastography technique as

$$\begin{aligned} E = \arg\min_{E \in \mathcal{U}} & \|\boldsymbol{\eta}(E) - \boldsymbol{\eta}_{obs}\|_* \\ & \text{s.t. } \mathcal{A}(\boldsymbol{\eta}, E, \mathbf{f}) = 0, \end{aligned} \tag{3.2}$$

where $\boldsymbol{\eta}$ is the deformation of the tissue, E is the Young's modulus of the tissue, $\mathcal{U}$ is a set of admissible values for E, $\|\cdot\|_*$ is a suitable norm, $\mathbf{f}$ is the (known) force acting on the tissue and $\mathcal{A}(\boldsymbol{\eta}, E, \mathbf{f})$ is the elastic model considered for the tissue.

When it comes to estimating the compliance of a large artery inside a patient, this procedure has to be modified. In fact, the vessel is already undergoing a deformation, caused by the pulsatility of the blood flow. This motion is orders of magnitude larger than the one that can be induced using non-invasive techniques, such as ultrasounds. Therefore, we cannot use an external force to induce a deformation of the tissue to be used in the inverse problem, since the observed deformation would be mostly caused by other factors, making the inverse problem extremely ill-posed.

A possible solution to this issue is to take advantage of the force exerted by the blood pulsatility on the wall vessel, using it as the force generating the displacement needed to set up the inverse problem. However, this force is not known, since it is the solution of a set of PDE governing the motion of the fluid. As a result, the equations governing the motion of the fluid as well as the interaction between the fluid

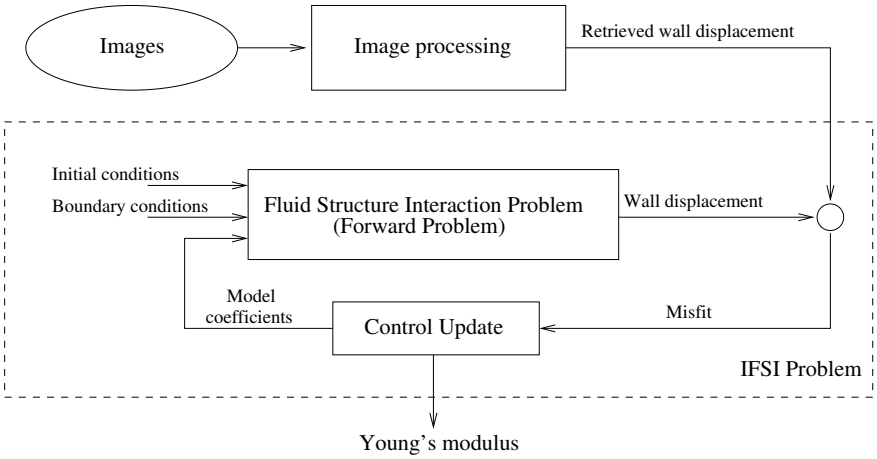


Figure 3.1: Outline of the compliance estimation procedure.

and the structure must be solved. Therefore, instead of a "simple" inverse elasticity problem, we have to solve an *inverse fluid-structure-interaction* (IFSI) problem. This procedure, proposed for the first time in [69], is outlined in Fig. 3.1.

In Section 3.2 we will introduce the set of governing equation for the forward FSI problem. In Section 3.3 we will introduce the inverse problem and the algorithm used to solve it. Finally, in Section 3.4 we will present some numerical results.

3.2 The forward problem

In the following, the subscripts f and s will be used, if needed, to distinguish fluid quantities from solid quantities. We will use bold letters $(\mathbf{x}, \mathbf{u}, \ldots)$ to indicate continuous vector quantities, while double underline $(\underline{\underline{\mathrm{I}}}, \underline{\underline{\sigma}}, \ldots)$ will be used to denote continuous tensorial quantities. A superscript t will denote quantities at time t $(\Omega_f^t, \Sigma^t, \ldots)$, while the superscript T will denote the transpose of a tensor $(\nabla \mathbf{u}^{\mathrm{T}}, \underline{\underline{\mathrm{B}}}^{\mathrm{T}}, \ldots)$. Also, $\partial_t y$ will be sometimes used to denote the partial derivative of y with respect to the variable t.

3.2.1 The fluid-membrane interaction

Let Ω^t be the region occupied by the fluid at time t. We denote by $\mathbf{u}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ the fluid velocity and pressure respectively, and we assume that, at every time, they fulfill the incompressible Navier-Stokes equations in Ω^t. In particular, we assume

that the fluid has Newtonian rheology, with Cauchy stress tensor

$$\underline{\underline{\mathrm{T}}}_f = -p\underline{\underline{\mathrm{I}}} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right), \tag{3.3}$$

where μ is the fluid's *dynamic viscosity*. Since the domain is moving in time, the fluid equations are formulated in an *Arbitrary Lagrangian-Eulerian* (ALE) framework. In particular, for every time t we introduce the invertible map $\mathcal{A}^t$ such that $\Omega^t = \mathcal{A}^t(\Omega^0)$. A quantity defined on the reference domain Ω^0 can then be transported into the current domain by composition with the ALE map $\mathcal{A}^t$. In particular, given a quantity $\widehat{f}$ defined on Ω^0 (or its boundary), we will let $f = \widehat{f} \circ (\mathcal{A}^t)^{-1}$ be the value of $\widehat{f}$ on the current domain Ω^t. Similarly, given a quantity g defined on Ω^t (or its boundary), we will let $\widehat{g} = g \circ \mathcal{A}^t$ be the value of g on the reference domain Ω^0. The domain velocity $\mathbf{w} = \widehat{\mathbf{w}} \circ (\mathcal{A}^t)^{-1}$ is then defined in terms of the ALE map by $\widehat{\mathbf{w}} = \partial_t \mathcal{A}^t$. In the ALE framework, the Navier-Stokes equations read

$$\rho_f \frac{D^A\mathbf{u}}{Dt} + \rho_f\left((\mathbf{u}-\mathbf{w})\cdot\nabla\right)\mathbf{u} - \nabla\cdot\underline{\underline{\mathrm{T}}}_f = \mathbf{f}_f \qquad \text{in } \Omega^t \times (t_0, T) \tag{3.4}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } \Omega^t \times (t_0, T), \tag{3.5}$$

where ρ_f is the fluid density and $\mathbf{f}_f$ accounts for possible external forces (such as gravity). Here, D^A/Dt is the ALE time derivative, that is, the time derivative in the ALE coordinates.

For the vessel wall we consider a simplified model, where the structure is approximated with an elastic two dimensional membrane, which we denote by $\Sigma^t \subset \partial\Omega^t$. This approximation, besides being reasonable in large arteries, where the vessel thickness is much smaller than the diameter, is also motivated by the fact that the lumen of the vessel is usually easier to identify than the whole wall on medical images, due to the contrast between different gray levels. In addition to the two dimensional approximation, we make the following assumptions for the vessel wall:

- linear stress-strain constitutive law
- small deformations

- negligible longitudinal and circumferential displacement

- negligible bending terms

which are also reasonable in large arteries (see, e.g., [16]). Thanks to these assumptions, the linear elasticity equations for the structure reduce to the simple scalar equation (sometimes called "independent rings")

$$\rho_s h_s \frac{\partial^2 \eta}{\partial t^2} + \beta E \eta = f_s \tag{3.6}$$

where η is the membrane displacement (in the normal direction), h_s and ρ_s are the vessel wall thickness and density respectively, E is the Young's modulus and f_s accounts for the external forces, which we assume to consist solely of the fluid stresses on the wall. The parameter β encodes both geometric and physical properties of the membrane. Its expression is given by

$$\beta = \frac{h_s}{1-\nu^2}\left(4k_m^2 - 2(1-\nu)k_g\right) \tag{3.7}$$

where k_m and k_g are the mean and Gaussian curvatures of the membrane respectively. The equation for the structure will be stated in a Lagrangian framework, so that, since the displacements are assumed to be small, we can write the structure equation in the reference domain $\Sigma = \Sigma^0$.

Conditions enforcing the continuity of the velocity and stress fields have to be prescribed on Σ^t. In particular, we impose

$$\mathbf{u} = \frac{\partial \eta}{\partial t}\mathbf{n}, \qquad \underline{\underline{\mathrm{T}}}_f(\mathbf{u}, p) \cdot \mathbf{n} = -f_s \mathbf{n}, \tag{3.8}$$

where $\mathbf{n}$ is the (outward) unit normal vector to the membrane Σ^t. The resulting Fluid-Structure Interaction (FSI) problem is then composed of two coupled subproblems:

i) given the domain Ω^t and the domain velocity $\mathbf{w}$, compute fluid velocity $\mathbf{u}$,

fluid pressure p and membrane displacement η such that

$$\begin{cases} \rho_f \dfrac{D^A \mathbf{u}}{Dt} + \rho_f \left((\mathbf{u} - \mathbf{w}) \cdot \nabla \right) \mathbf{u} - \nabla \cdot \underline{\underline{\mathrm{T}}}_f = \mathbf{f}_f & \text{in } \Omega^t \times (t_0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega^t \times (t_0, T) \\ \rho_s h_s \dfrac{\partial^2 \widehat{\eta}}{\partial t^2} + \beta E \widehat{\eta} = \widehat{f}_s & \text{on } \Sigma \times (t_0, T), \\ \mathbf{u} \cdot \mathbf{n} = \dfrac{\partial \widehat{\eta}}{\partial t} \circ (\mathcal{A}^t)^{-1} & \text{on } \Sigma^t \times (t_0, T), \\ \underline{\underline{\mathrm{T}}}_f(\mathbf{u}, p) \cdot \mathbf{n} = -\left(\widehat{f}_s \circ (\mathcal{A}^t)^{-1} \right) \mathbf{n} & \text{on } \Sigma^t \times (t_0, T), \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t) & \text{on } \partial\Omega_D^t \times (t_0, T) \\ \underline{\underline{\mathrm{T}}}_f(\mathbf{x}, t) \cdot \mathbf{n} = \mathbf{h}(\mathbf{x}, t) & \text{on } \partial\Omega_N^t \times (t_0, T) \end{cases} \tag{3.9}$$

endowed with suitable initial conditions for the fluid velocity, membrane displacement and membrane velocity. Here $\partial\Omega_D^t$ and $\partial\Omega_N^t$ are the portion of the boundary where Dirichlet and Neumann conditions are prescribed, and they are such that $\overline{\partial\Omega_D^t} \cup \overline{\partial\Omega_N^t} \cup \overline{\Sigma^t} = \partial\Omega^t$ and $\partial\Omega_D^t \cap \partial\Omega_N^t = \varnothing$, $\partial\Omega_D^t \cap \partial\Sigma^t = \varnothing$ and $\partial\Omega_N^t \cap \partial\Sigma^t = \varnothing$. Notice that no boundary conditions are required for the structure equation, since no spatial derivatives appear in (3.6). Also, notice that, due to the interface conditions, the initial data must satisfy the compatibility condition $\mathbf{u}(\mathbf{x}, 0) = \partial_t \eta(\mathbf{x}, 0)\mathbf{n}$ on Σ.

ii) Given the membrane displacement $\widehat{\boldsymbol{\eta}} := \widehat{\eta}\widehat{\mathbf{n}}$, we compute the displacement field $\widehat{\mathbf{d}}$ as

$$\begin{cases} S\widehat{\mathbf{d}} = \mathbf{0} & \text{in } \Omega^0 \\ \widehat{\mathbf{d}} = \widehat{\boldsymbol{\eta}} & \text{on } \Sigma \end{cases} \qquad \mathcal{A}^t(\mathbf{x}^0) = \mathbf{x}^0 + \widehat{\mathbf{d}}, \quad \widehat{\mathbf{w}} = \partial_t \mathcal{A}^t(\mathbf{x}^0) \tag{3.10}$$

where S is a suitable smooth extension operator (for instance harmonic) from the membrane to the fluid domain.

For the weak formulation of (3.9) we introduce the following function spaces

$$\begin{aligned} &V_{\mathbf{g}}^t := \left\{ \mathbf{v} \in H^1(\Omega^t) : \mathbf{v}_{|\Sigma^t} \cdot \boldsymbol{\tau} = 0, \mathbf{v}_{|\partial\Omega_D^t} = \mathbf{g} \right\} \qquad Q^t := L^2(\Omega^t) \\ &X := H^{1/2}(\Sigma) \qquad \mathcal{E} := \{ E \in L^\infty(\Sigma) : E \geqslant E_0 > 0 \} \end{aligned} \tag{3.11}$$

where $\boldsymbol{\tau}$ is the unit vector tangent to Σ^t. Furthermore, we introduce the space

$$W^t = \left\{ (\mathbf{v}, \widehat{\psi}) \in V_{\mathbf{0}}^t \times X : \mathbf{v}_{|\Sigma^t} = (\widehat{\psi} \circ (\mathcal{A}^t)^{-1})\mathbf{n} \right\}. \tag{3.12}$$

Remark 3.1. Technically, if $\mathbf{g} \neq \mathbf{0}$, then $V_{\mathbf{g}}^t$ is not a vector space. Therefore, the solution of the problem is usually decomposed as

$$\mathbf{u} = \overset{\circ}{\mathbf{u}} + \mathbf{u}_{\mathbf{g}}, \tag{3.13}$$

with $\overset{\circ}{\mathbf{u}} \in V_{\mathbf{0}}^t$ and $\mathbf{u}_{\mathbf{g}} \in H^1$ is such that $\mathbf{u}_{\mathbf{g}|\partial\Omega_D^t} = \mathbf{g}$. The function $\mathbf{u}_{\mathbf{g}}$ is usually called *lift function* (see, for instance, [79]). Here, when we write $\mathbf{u} \in V_{\mathbf{g}}^t$, we actually mean that $\mathbf{u}$ can be written in the form (3.13).

Using the second relation in the interface conditions (3.8), we can write the weak formulation as follows: given $\mathbf{g} \in H^{1/2}(\partial\Omega_D^t)$ and $\mathbf{h} \in L^2(\partial\Omega_N^t)$, and given $E \in \mathcal{E}$, find $\mathbf{u} \in V_{\mathbf{g}}^t$, $p \in Q^t$ and $\widehat{\eta} \in X$ such that

$$\begin{cases} \left(\rho_f \dfrac{D^A \mathbf{u}}{Dt}, \mathbf{v}\right)_{\Omega^t} + (\rho_f((\mathbf{u}-\mathbf{w})\cdot\nabla)\mathbf{u}, \mathbf{v})_{\Omega^t} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}, \nabla\mathbf{v})_{\Omega^t} + \\ \qquad -(p, \nabla\cdot\mathbf{v})_{\Omega^t} + \left(\rho_s h_s \dfrac{\partial^2\widehat{\eta}}{\partial t^2}, \widehat{\psi}\right)_{\Sigma} + \left(\beta E\widehat{\eta}, \widehat{\psi}\right)_{\Sigma} = (\mathbf{f}_f, \mathbf{v})_{\Omega^t} + (\mathbf{h}, \mathbf{v})_{\partial\Omega_N^t} \\ (q, \nabla\cdot\mathbf{u})_{\Omega^t} = 0 \\ (\widehat{\mathbf{u}}_{|\Sigma}\cdot\widehat{\mathbf{n}} - \partial_t\widehat{\eta}, \widehat{\psi})_{\Sigma} = 0 \end{cases} \tag{3.14}$$

for every $(\mathbf{v}, \widehat{\psi}) \in W^t$ and $q \in Q^t$. Here, the notation $(f, g)_\Omega$ denotes the inner product in $L^2(\Omega)$. In the next section we present the discretization of problem (3.14) that will be used in this chapter and the following one.

3.2.2 The discrete problem

To discretize problem (3.14) we follow the strategy presented in [64]. We start by introducing a set of discretization points for the time dimension $t^n = t_0 + n\Delta t$, $n = 0, \dots N$, with $t^N = T$. Correspondingly, we will denote by f^n the value of the generic quantity $f(t^n)$. We will use Backward Differentiation Formulas of order p (BDFp) for the discretization of the time derivatives, with a semi-implicit treatment of the

convective and geometric non-linearities. In particular, we will denote with Ω^*, $\mathbf{u}^*$ and $\mathbf{w}^*$ the extrapolations of Ω^n, $\mathbf{u}^n$ and $\mathbf{w}^n$ respectively, based on the previous time steps values. For instance, using BDF1 (that is, Backward Euler), we would have $\Omega^* = \Omega^{n-1}$, $\mathbf{u}^* = \mathbf{u}^{n-1}$ and $\mathbf{w}^* = \mathbf{w}^{n-1}$. Accordingly, $V_{\mathbf{g}}^*$, Q^* and W^* will be the counterparts of $V_{\mathbf{g}}^t$, Q^t and W^t when the domain is Ω^*. With these choices, in [64] the authors showed that the FSI problem can be casted into a fluid problem, where the action of the structure is regarded as a Robin boundary condition on Σ^t. The semi-discrete FSI problem reads as follows: given $E \in \mathcal{E}$, $\mathbf{u}^{n-1} \in V_{\mathbf{g}}^*$ and $\widehat{\eta}^{n-1} \in X$, find $(\mathbf{u}^n, p^n, \eta^n) \in V_{\mathbf{g}}^* \times Q^* \times X$ such that

$$
\begin{cases}
\dfrac{\alpha\rho_f}{\Delta t}(\mathbf{u}^n, \mathbf{v})_{\Omega^*} + \rho_f(((\mathbf{u}^* - \mathbf{w}^*)\cdot\nabla)\mathbf{u}^n, \mathbf{v})_{\Omega^*} + \mu(\nabla\mathbf{u}^n + (\nabla\mathbf{u}^n)^{\mathrm{T}}, \nabla\mathbf{v})_{\Omega^*} \\
\qquad -(p^n, \nabla\cdot\mathbf{v})_{\Omega^*} + \displaystyle\int_\Sigma \left(\frac{\rho_s h_s}{\Delta t} + E\beta\Delta t\right)\widehat{\mathbf{u}}^n\cdot\widehat{\mathbf{n}}\,\widehat{\mathbf{v}}\cdot\widehat{\mathbf{n}}\,d\sigma \\
\qquad = F(\mathbf{v}) + \displaystyle\int_\Sigma \left(\frac{\rho_s h_s}{\Delta t}\widehat{\mathbf{u}}^{n-1}\cdot\widehat{\mathbf{n}} - E\beta\widehat{\eta}^{n-1}\right)\widehat{\mathbf{v}}\cdot\widehat{\mathbf{n}}\,d\sigma \\
(q, \nabla\cdot\mathbf{u}^n)_{\Omega^*} = 0 \\
\dfrac{1}{\Delta t}(\widehat{\eta}^n - \widehat{\eta}^{n-1}, \widehat{\psi})_\Sigma = (\widehat{\mathbf{u}}^n\cdot\widehat{\mathbf{n}}, \widehat{\psi})_\Sigma
\end{cases}
\tag{3.15}
$$

for all $(\mathbf{v}, \widehat{\psi}) \in W^*$ and $q \in Q^*$. Here $F(\mathbf{v})$ is a functional accounting for the (possible) non-homogeneous boundary conditions, the (possible) forcing terms, the terms in the BDFp approximation to the ALE time derivative corresponding to previous time steps, while the parameter α is the coefficient of $\mathbf{u}^n$ in such approximation. Notice that, thanks to the semi-implicit treatment of the geometry, $\widehat{\eta}^n$ can be promptly computed from $\mathbf{u}^n$ once the fluid equations has been solved. Finally, we compute the new position of the domain and the corresponding domain velocity by solving the smooth extension problem

$$
\begin{cases}
(\nabla\widehat{\mathbf{d}}^n, \nabla\widehat{\mathbf{z}}) = 0 \quad \text{in } \Omega^0 \\
\widehat{\mathbf{d}}^n = \widehat{\boldsymbol{\eta}}^n \quad \text{on } \Sigma \\
\forall\widehat{\mathbf{z}} \in H^1(\Omega^0) : \widehat{\mathbf{z}}_{|\Sigma} = \mathbf{0}
\end{cases}
\quad\Rightarrow\quad
\begin{cases}
\mathcal{A}^n(\widehat{\mathbf{x}}) = \mathbf{x}^0 + \widehat{\mathbf{d}}^n \\
\Omega^n = \mathcal{A}^n(\Omega^0).
\end{cases}
\tag{3.16}
$$

For the space discretization we use conformal Finite Elements (FE). In particular, we choose *inf-sup* stable FE spaces for the fluid velocity and pressure (for

instance, Taylor-Hood elements [79]), and we use the same polynomial order for the discretization of the membrane displacement. We let V_h^*, Q_h^* and X_h be the finite dimensional subspaces of the spaces V^*, Q^* and X. We will use underlined variables $(\underline{u}, \underline{p}, \underline{\eta}, \ldots)$ to denote the vectors of the coefficients of the FE expansions $(\mathbf{u}_h, p_h, \eta_h, \ldots)$ of the corresponding continuous quantities $(\mathbf{u}, p, \eta, \ldots)$. Furthermore, since there is a bijection between each FE function f_h and the vector $\underline{f} \in \mathbb{R}^n$ of the coefficients in the FE basis expansion, with abuse of notation, we will use underlined variables to denote both the FE function f_h and the vector of the coefficients $\underline{f}$. Finally, $\mathcal{A}_h^n$ will denote the discrete ALE map at the time step t^n.

To formulate the fully discrete FSI problem, let us introduce the matrices M, K, B, P, M_Σ and $\mathrm{M}_\Sigma(E)$, whose entries are defined by

$$\mathrm{M}_{ij} = \int_{\Omega^*} \rho_f \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i dx \tag{3.17}$$

$$\mathrm{K}_{ij} = \int_{\Omega^*} \rho_f \left(((\mathbf{u}_h^* - \mathbf{w}_h^*) \cdot \nabla) \boldsymbol{\varphi}_j \right) \boldsymbol{\varphi}_i dx + \int_{\Omega^*} \mu (\nabla \boldsymbol{\varphi}_j + \nabla \boldsymbol{\varphi}_j^{\mathrm{T}}) : \nabla \boldsymbol{\varphi}_i dx \tag{3.18}$$

$$\mathrm{B}_{ij} = -\int_{\Omega^*} \nabla \cdot \boldsymbol{\varphi}_j \psi_i dx \tag{3.19}$$

$$\mathrm{P}_{ij} = \int_\Sigma \widehat{\boldsymbol{\varphi}}_j \cdot \widehat{\mathbf{n}}\, \widehat{\chi}_i d\sigma \tag{3.20}$$

$$[\mathrm{M}_\Sigma]_{ij} = \int_\Sigma \widehat{\chi}_j \widehat{\chi}_i d\sigma \tag{3.21}$$

$$[\mathrm{M}_\Sigma(E)]_{ij} = \int_\Sigma E \widehat{\chi}_j \widehat{\chi}_i d\sigma \tag{3.22}$$

where $\boldsymbol{\varphi}_i, \psi_i, \widehat{\chi}_i$ are basis functions for the spaces V_h^*, Q_h^* and X_h respectively. Similarly, we introduce the momentum right hand side $\underline{f}^n(E)$, with entries given by

$$\begin{aligned}[\underline{f}^n(E)]_i &= F(\boldsymbol{\varphi}_i) + \int_\Sigma \left(\frac{\rho_s h_s}{\Delta t} \widehat{\mathbf{u}}_h^{n-1} \cdot \widehat{\mathbf{n}} - E\beta \widehat{\eta}_h^{n-1} \right) (\widehat{\boldsymbol{\varphi}}_i \cdot \widehat{\mathbf{n}}) d\sigma = \\ &= F(\boldsymbol{\varphi}_i) + \frac{\rho_s h_s}{\Delta t} \mathrm{P}^{\mathrm{T}} \mathrm{M}_\Sigma \mathrm{P} \underline{u}^{n-1} - \beta \mathrm{P}^{\mathrm{T}} \mathrm{M}_\Sigma(E) \underline{\eta}^{n-1}. \end{aligned} \tag{3.23}$$

The fully discrete problem then reads: given $E \in \mathcal{E}$, $\underline{u}^{n-1} \in V_h$, $\underline{\eta}^{n-1} \in X_h$, find $(\underline{u}^n, \underline{p}^n, \underline{\eta}^n) \in V_h \times Q_h \times X_h$ such that

$$\begin{bmatrix} \mathrm{C}(E) & \mathrm{B}^{\mathrm{T}} & \mathrm{O} \\ \mathrm{B} & \mathrm{O} & \mathrm{O} \\ -\Delta t\mathrm{P} & \mathrm{O} & \mathrm{M}_{\Sigma} \end{bmatrix} \begin{bmatrix} \underline{u}^n \\ \underline{p}^n \\ \underline{\eta}^n \end{bmatrix} = \begin{bmatrix} \underline{f}^n(E) \\ \underline{0} \\ \mathrm{M}_{\Sigma}\underline{\eta}^{n-1} \end{bmatrix} \tag{3.24}$$

where we introduced the momentum matrix

$$\mathrm{C}(E) = \frac{1}{\Delta t}\mathrm{M} + \mathrm{K} + \mathrm{P}^{\mathrm{T}}\left(\frac{\rho_s h_s}{\Delta t}\mathrm{M}_{\Sigma} + \Delta t\beta\mathrm{M}_{\Sigma}(E)\right)\mathrm{P}. \tag{3.25}$$

The choice of $\mathcal{E}$ presented in (3.11) is enough to guarantee that all the bilinear forms are well defined. However, we will focus on two particular scenarios, which are interesting both on the application and mathematical point of view. To present these two scenarios, let s be the curvilinear abscissa along the vessel centerline, and let $\Sigma_i = \{\mathbf{x} \in \Sigma : s_i \leqslant s_{i-1} \leqslant s_i\}$, $i = 1, \ldots, k$. This is a partitioning of the membrane into k regions, depending on the curvilinear abscissa s, with s_0 corresponding to the inflow section of the vessel, while s_k is the outflow section. We will consider the following cases:

i) $E = E(s)$ is constant in Σ_i, that is, the Young's modulus is piecewise constant along the vessel. This could model, for instance, the case where an artificial prostheses in inserted in the vessel, thus introducint a sudden change in the structural properties of the wall.

ii) $E = E(s)$ is continuous on Σ and linear in Σ_i, that is, the Young's modulus is piecewise linear along the vessel. This could model, for instance, the case where an atherosclerotic plaque is present at a certain position in the vessell, affecting the structural properties of the vessel wall in the immediate upstream and downstream sections.

Although the problem only requires E to be a given function of space, in view of the solution of the inverse problem, it is convenient to introduce also a space discretization of the Young's modulus. In particular, the discrete space $\mathcal{E}_h$ for the discrete Young's modulus E_h will be the FE space of piecewise constant (or linear) polynomials. However, to enforce the fact that the Young's modulus is constant (or

linear) on rather extended regions of Σ (depending on the curvilinear abscissa), we introduce an extension matrix R such that the vector of the coefficients $\underline{E}$ in the FE expansion of E_h can be written as $\underline{E} = \mathrm{R}\underline{y}$, with $\underline{y} \in \mathbb{R}^k$, with k equal to the number of independent values of E_h on the membrane, which is assumed to be a rather small number[1]. For instance, for the piecewise constant case, $\mathrm{R}_{ij} = 1$ if the i-th degree of freedom lies in Σ_j and zero otherwise. For the piecewise linear case, the expression is slightly more complicated, with two non-zero coefficients per row, corresponding to the two sections that delimit the region where the i-th degree of freedom is located. This choice allows us to write the membrane weighted mass matrix (3.22) as

$$\mathrm{M}_\Sigma(E) = \sum_{\ell=1}^{k} y_\ell \mathrm{M}_\ell, \qquad \text{with} \qquad [\mathrm{M}_\ell]_{ij} = \int_\Sigma \mathrm{R}_{i\ell}\mathrm{R}_{j\ell}\widehat{\chi}_i\widehat{\chi}_j d\sigma. \tag{3.26}$$

Although this expansion requires us to compute k matrices rather than just one, it will prove useful when deriving the optimality conditions for the inverse problem.

To conclude this section we rewrite the discrete problem (3.24) in a slightly different way, which shall prove useful in chapter 4 to develope a Reduced Basis approach for the solution of the inverse problem. In particular, we adopt an *incremental* approach for the pressure [41]: given $\underline{p}^*$, a suitable extrapolation of $\underline{p}^n$, we let $\underline{p}^n = \underline{p}^* + \underline{\delta p}^n$, and we rewrite the problem as

$$\begin{bmatrix} \mathrm{C}(\underline{y}) & \mathrm{B}^{\mathrm{T}} & \mathrm{O} \\ \mathrm{B} & \mathrm{O} & \mathrm{O} \\ -\Delta t\mathrm{P} & \mathrm{O} & \mathrm{M}_\Sigma \end{bmatrix} \begin{bmatrix} \underline{u}^n \\ \underline{\delta p}^n \\ \underline{\eta}^n \end{bmatrix} = \begin{bmatrix} \underline{f}^n(\underline{y}) - \mathrm{B}^{\mathrm{T}}\underline{p}^* \\ \underline{0} \\ \mathrm{M}_\Sigma\underline{\eta}^{n-1} \end{bmatrix} \tag{3.27}$$

where, in view of the solution of the inverse problem, we stressed the dependence of C and $\underline{f}$ on the Young's modulus as a dependence on $\underline{y}$ rather than on E.

[1] In the piecewise constant case, k is simply the number of regions that the membrane has been divided into, while in the piecewise linear case it will be equal to that number plus one.

3.3 The inverse problem

In this section we present the inverse problem for the estimation of the Young's modulus, following the approach proposed in [69]. In particular, concerning the framework presented in the previous chapter, we will adopt a variational approach. The starting point is to define a *cost functional*

$$\mathcal{J}(E) = \frac{1}{2}\int_{t_0}^{T}\int_{\Sigma} |\widehat{\eta} - \widehat{\eta}_m|^2 d\sigma dt + \gamma \mathcal{R}(E), \tag{3.28}$$

where $\widehat{\eta}_m$ is the displacement field on the membrane retrieved from the images (the measures) and $\mathcal{R}$ is a regularization term. For instance, we could use a Tikhonov regularization term of the form

$$\mathcal{R}(E) = \int_{\Sigma} |E - E_r|^2 d\sigma, \tag{3.29}$$

where E_r is a (known) reference value for E such as, for example, the value of a healthy vessel taken from literature. As explained in Section 2.4.2, the role of $\mathcal{R}$ is to enhance the convexity of the functional. The inverse problem then reads

$$\begin{aligned} E^{opt} &= \arg\min \mathcal{J}(E) \\ &\text{s.t. } FSI(\mathbf{u}, p, \eta, E) \end{aligned} \tag{3.30}$$

where $FSI(\mathbf{u}, p, \eta, E)$ indicates that $\mathbf{u}, p, \eta$ and E must solve the system of equations (3.9)-(3.10). This means that the FSI problem is regarded as a constraint for the minimization of the functional $\mathcal{J}$.

When we look for a numerical approximation of the inverse problem, we face the issue of deciding the order in which to perform the two operations, discretization and optimization. One approach is to write the optimality conditions for (3.30), for instance using the Lagrange multipliers method, and then discretize them. This approach is usually referred to as Optimize-then-Discretize (OD). Alternatively, one can discretize the functional and the constraint in (3.30) and then write the optimality conditions for the discrete system. This approach is called Discretize-then-

Optimize (DO). Both of the approaches eventually lead to a set of (possibly non-linear) equations to solve that, in general, may differ and it is not clear if one approach is better than the other. However, as pointed out in [38], the DO approach has the advantage that the solution is optimal for the discrete problem, while the OD approach may lead to a solution that is not optimal for the continuous or discrete problem.

For time-dependent problems, optimization is an even harder issue. In fact, when using Lagrange multipliers, one has to solve also an *adjoint* problem in order to compute the gradient of the functional, and is a backward in time problem. From the storage point of view, this is demanding, since it requires storing the solution of the forward problem at every time step, while the adjoint problem is solved. Storage costs can be partially reduced using the so called *Check-pointing* technique [17], but this causes an increase in the computational costs[2]. Moreover, for the case of the estimation of the compliance, the adjoint problem would also involve the calculation of the so called *shape derivatives* [87], which are also expensive.

In [69] the authors proposed a trade-off approach, tackling the inverse problem with an approach that we call *semi-discretize-then-optimize-then-fully-discretize*. In particular, they discretized the problem in time, introducing an *instantaneous* cost functional

$$\mathcal{J}^n(E) = \int_\Sigma |\eta^n - \eta_m^n|^2 d\sigma + \gamma \mathcal{R}(E) \tag{3.31}$$

and performed the minimization at every time step, treating the geometry explicitly. Finally, they discretized the problem in space to obtained a discrete KKT system. This approach has the advantage of dramatically reduce the computational and storage costs, at the price that every time step will in general lead to a different estimate for E. In [69], the authors overcome this disadvantage by taking an average (possibly weighted) of the estimates obtained at all the time steps. Furthermore, they also perform a well posedness analysis, showing that the choices of a piecewise linear or piecewise constant Young's modulus guarantee the existence of an optimum solution.

[2] Since only a few time steps are stored, the forward problem has to be solved again to recover the solution at the remaining time steps.

Remark 3.2. If using Tikhonov regularization of the type (3.29), this approach also offers a good candidate for E_r, namely the estimate obtained at the previous time step, or a weighted combination of the last $k \geqslant 1$ estimates, to prevent the case where a bad estimate at a certain time step excessively influences the next estimate.

Here we follow a genuine DO approach, discretizing the problem both in space and time, and performing the optimization at each time step. We use a Tikhonov regularization where the reference value at time step t^n is set to the average of the previous m estimates (or all the estimates, if $n < m$), for a predefined value of m (in our experiments we used $m = 5$). The instantaneous discrete cost functional is given by

$$J^n(\underline{y}) = \frac{1}{2}\|\underline{\eta}^n - \underline{\eta}^n_m\|_\Sigma + \frac{\gamma}{2}\|\mathrm{R}(\underline{y} - \underline{y}^n_{ref})\|_\Sigma \tag{3.32}$$

where $\|\underline{v}\|_\Sigma = \underline{v}^{\mathrm{T}}\mathrm{M}_\Sigma\underline{v}$ is the discrete counterpart of the continuous L^2 norm on Σ and $\underline{y}^n_{ref}$ is the reference value for the Tikhonov regularization at time step t^n. The optimization problem in the discrete space reads as follows

$$\begin{aligned} \underline{y}^n_{opt} &= \arg\min J^n(\underline{y}) \\ \text{s.t.} \quad & \begin{bmatrix} \mathrm{C}(\underline{y}) & \mathrm{B}^{\mathrm{T}} & \mathrm{O} \\ \mathrm{B} & \mathrm{O} & \mathrm{O} \\ -\Delta t\mathrm{P} & \mathrm{O} & \mathrm{M}_\Sigma \end{bmatrix} \begin{bmatrix} \underline{u}^n \\ \underline{\delta p}^n \\ \underline{\eta}^n \end{bmatrix} = \begin{bmatrix} \underline{f}^n(\underline{y}) - \mathrm{B}^{\mathrm{T}}\underline{p}^* \\ \underline{0} \\ \mathrm{M}_\Sigma\underline{\eta}^{n-1} \end{bmatrix} \end{aligned} \tag{3.33}$$

We will refer to this problem as the Inverse Fluid-Membrane Interaction (IFMI) problem. We decide to tackle the IFMI problem with the Lagrange multipliers method. To this end, we introduce the Lagrangian function

$$\begin{aligned} \mathcal{L}(\underline{u}^n, \underline{\delta p}^n, \underline{\eta}^n, \underline{y}, \underline{\lambda}_u, \underline{\lambda}_{\delta p}, \underline{\lambda}_\eta) =& J^n(\underline{y}) + \underline{\lambda}_u^{\mathrm{T}}\left(\mathrm{C}(\underline{y})\underline{u}^n + \mathrm{B}^{\mathrm{T}}\underline{\delta p}^n - \underline{f}^n(\underline{y}) + \mathrm{B}^{\mathrm{T}}\underline{p}^*\right) + \\ &+ \underline{\lambda}_{\delta p}^{\mathrm{T}}\mathrm{B}\underline{u}^n + \underline{\lambda}_\eta^{\mathrm{T}}\left(\mathrm{M}_\Sigma\underline{\eta}^n - \Delta t\mathrm{P}\underline{u}^n - \mathrm{M}_\Sigma\underline{\eta}^{n-1}\right). \end{aligned} \tag{3.34}$$

Then, we differentiate the Lagrangian with respect to each variable, to obtain the

KKT system

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial\{\underline{u}^n, \underline{\delta p}^n, \underline{\eta}^n\}} = 0 \quad \Rightarrow \begin{bmatrix} \mathrm{C}^{\mathrm{T}}(\underline{y}) & \mathrm{B}^{\mathrm{T}} & -\Delta t \mathrm{P}^{\mathrm{T}} \\ \mathrm{B} & \mathrm{O} & \mathrm{O} \\ \mathrm{O} & \mathrm{O} & \mathrm{M}_{\Sigma} \end{bmatrix} \begin{bmatrix} \underline{\lambda}_u \\ \underline{\lambda}_{\delta p} \\ \underline{\lambda}_{\eta} \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ -\mathrm{M}_{\Sigma}(\underline{\eta}^n - \underline{\eta}^n_m) \end{bmatrix} \\ \dfrac{\partial \mathcal{L}}{\partial y_i} = 0 \qquad\qquad \Rightarrow \gamma \mathrm{M}_{R}(\underline{y} - \underline{y}^n_r) + \underline{\lambda}_u^{\mathrm{T}} \left(\dfrac{\partial \mathrm{C}}{\partial y_i} \underline{u}^n - \dfrac{\partial \underline{f}^n}{\partial y_i} \right) = 0 \quad i = 1, \ldots, k \\ \dfrac{\partial \mathcal{L}}{\partial\{\underline{\lambda}_u, \underline{\lambda}_{\delta p}, \underline{\lambda}_{\eta}\}} = 0 \quad \Rightarrow \begin{bmatrix} \mathrm{C}(\underline{y}) & \mathrm{B}^{\mathrm{T}} & \mathrm{O} \\ \mathrm{B} & \mathrm{O} & \mathrm{O} \\ -\Delta t \mathrm{P} & \mathrm{O} & \mathrm{M}_{\Sigma} \end{bmatrix} \begin{bmatrix} \underline{u}^n \\ \underline{\delta p}^n \\ \underline{\eta}^n \end{bmatrix} = \begin{bmatrix} \underline{f}^n(\underline{y}) \\ \underline{0} \\ \mathrm{M}_{\Sigma} \underline{\eta}^{n-1} \end{bmatrix} \end{cases} \tag{3.35}$$

where we have set $\mathrm{M}_{R} = \mathrm{R}^{\mathrm{T}} \mathrm{M}_{\Sigma} \mathrm{R}$. The expression for the derivatives $\dfrac{\partial \mathrm{C}}{\partial y_i}$ and $\dfrac{\partial \underline{f}^n}{\partial y_i}$ can be computed from (3.23-3.25) and (3.26):

$$\frac{\partial \mathrm{C}}{\partial y_i} = \Delta t \beta \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \mathrm{P}, \qquad \frac{\partial \underline{f}^n}{\partial y_i} = -\beta \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \underline{\eta}^{n-1}. \tag{3.36}$$

For the solution of the KKT system (which is non-linear) we use the approach we presented in Section 2.4.1. In particular, we minimize $J^n(\underline{y})$ with the BFGS method (see, for instance, [65]), where the gradient of the functional is computed as

$$\begin{aligned} \frac{\partial J^n(\underline{y})}{\partial y_i} &= \underline{\lambda}_u^{\mathrm{T}} \left(\frac{\partial \mathrm{C}}{\partial y_i} \underline{u}^n - \frac{\partial \underline{f}^n}{\partial y_i} \right) + \gamma \mathrm{M}_{R}(\underline{y} - \underline{y}^n_r) = \\ &= \underline{\lambda}_u^{\mathrm{T}} \left(\Delta t \beta \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \mathrm{P} \underline{u}^n + \beta \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \underline{\eta}^{n-1} \right) + \gamma \mathrm{M}_{R}(\underline{y} - \underline{y}^n_r). \end{aligned} \tag{3.37}$$

In order to compute the gradient, we must therefore solve, in order, the forward problem and the adjoint problem. Algorithm 2 summarizes the steps to estimate y^n at time t^n.

Remark 3.3. The BFGS method is a routine for unconstrained minimization, and therefore can lead to solutions $\underline{y}^n$ with negative components. However, since this is clearly not physical, and, moreover, can lead to instabilities of the linear system, in practice we made a change of variables for the minimization routine. In particular,

Algorithm 2 Full space BFGS

1: Compute the reference value $\underline{y}_r^n$ as the average of previous m estimates
2: $k = 1$, let $\underline{y}_k = \underline{y}_r^n$
3: **while** $k < k_{\max}$ and *convergence test==false* **do**
4: Solve state equation with $\underline{y} = \underline{y}_k$ to compute the state variables
5: Solve adjoint equation with $\underline{y} = \underline{y}_k$ to compute the adjoint variables
6: Compute the gradient of J^n using (3.37)
7: Compute the descent direction $\boldsymbol{\delta}_k$ (using BFGS)
8: Perform line search along $\boldsymbol{\delta}_k$ and update the estimate: $\underline{y}_{k+1} = \underline{y}_k + \tau_k \boldsymbol{\delta}_k$
9: $k \leftarrow k + 1$
10: **end while**

we set

$$\underline{\widehat{y}} = \ln(\underline{y}), \tag{3.38}$$

where the logarithm is to be intended component-wise. With this change of variables, we can allow $\underline{\widehat{y}}$ to be negative, while the real (physical) value of $\underline{y}$ will stay positive. The new functional to minimize is therefore

$$\widehat{J}(\underline{\widehat{y}}) = J^n(e^{\underline{\widehat{y}}}) = J^n(\underline{y}), \tag{3.39}$$

so that there is no difference in evaluating $\widehat{J}^n(\underline{\widehat{y}})$ versus evaluating $J^n(\underline{y})$. For the derivatives we obtain

$$\frac{\partial \widehat{J}^n(\underline{\widehat{y}})}{\partial \widehat{y}_i} = \frac{\partial J^n(e^{\underline{\widehat{y}}})}{\partial \widehat{y}_i} = \left(\frac{\partial J^n(\underline{y})}{\partial y_i} \Big|_{\underline{y}=e^{\underline{\widehat{y}}}} \right) e^{\widehat{y}_i}. \tag{3.40}$$

3.4 Results

In this section we present some numerical results on the problem of the estimation of the compliance. In particular, since the focus of this work is on the Reduced Order Model approach for the reduction of the computational costs associated with the inverse problem, here we will present only two examples: the first on a cylindrical domain and the second an idealized aortic arch. All the simulations have been performed using LifeV, a C++ FE library [1].

Remark 3.4. We underline the fact that the variational approach for the estimation of the compliance was already proposed in [69], where several results were

presented using what we called *semi-discretize-then-optimize-then-fully-discretize* approach. The results we present here are obtained, instead, with a genuine DO approach. Furthermore, in [69], the authors mostly focused on the case where the Young's modulus is constant in the whole domain, that is, $\underline{y} \in \mathbb{R}$, with only one test for the case where $\underline{y} \in \mathbb{R}^3$, corresponding to a piecewise linear distribution. Here, we always focus on the multi-dimensional case, with $\underline{y} \in \mathbb{R}^3$, and analyze both piecewise constant and piecewise linear distributions of the Young's modulus.

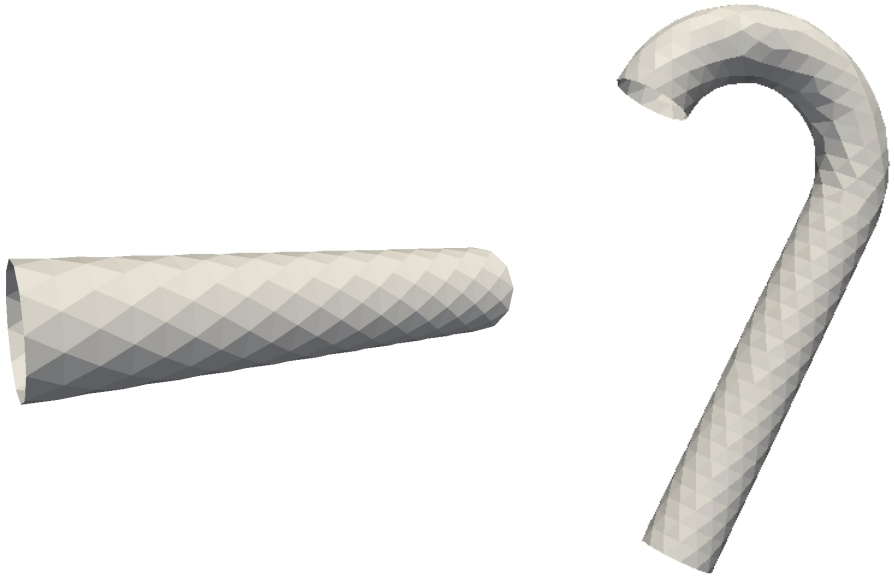
For these simulations we will make use of synthetic data. This means that we are going to generate data by solving the forward problem with a given (realistic) value for the Young's modulus, and we will use the computed membrane displacement as 'measures' for the inverse problem. In order to avoid the so called 'inverse crime', we are going to generate the data on a mesh with a finer granularity than the one used in the inverse problem, and we will add random noise to the computed solution. The resulting vectors will then be projected on a coarser mesh, where the inverse problem will be solved. If we denote with $\underline{\eta}_f$ the solution obtained on a finer mesh and with Π_f^c a projector from the fine to the coarse mesh, the measures used in the inverse problem can be expressed as

$$\underline{\eta}_m = \Pi_f^c \left(\underline{\eta}_f + \xi \|\underline{\eta}_f\|_\infty \underline{e} \right), \tag{3.41}$$

where $\underline{e}$ is a random vector obtained from the uniform distribution $\mathcal{U}([-1,1])$ and ξ is the reciprocal of the *signal to noise ratio* (SNR) and represents the amount of noise in the measures as a fraction of the magnitude of the 'exact' measures. For data retrieved from medical images, the magnitude of the noise can be as large as 20% of that of the wall displacement [14], which means that the SNR can be as low as 5 and ξ as large as 0.2.

3.4.1 Cylinder case

The geometry for the cylindrical test case is shown in Figure 3.2(a). In particular, the length of the cylinder is $L = 5cm$, while the radius is $R = 0.5cm$. For the Young's modulus we will consider a piecewise linear distribution, with $\underline{y} \in \mathbb{R}^3$. In particular,



(a) Cylinder (b) Idealized aortic arch

Figure 3.2: Geometries for the cylinder (a) and idealized aortic arch (b) test cases.

y_1 and y_3 corresponds to the value of the Young's modulus at the inflow and outflow sections respectively, while y_2 corresponds to the value of the Young's modulus in the middle of the cylinder. The values used for the generation of the synthetic measures are $[y_1, y_2, y_3] = [1.3, 1.8, 1.6] \cdot 10^6\, dyn/cm^2$, and the corresponding distribution is depicted in Figure 3.3.

The initial condition for the fluid is at rest (zero velocity and pressure), while at the inflow (Γ_{in}) and outflow (Γ_{out}) sections we impose a Neumann boundary condition of the type

$$-p\mathbf{n} + \left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\mathbf{n} = g(t)\mathbf{n}, \tag{3.42}$$

where

$$g(t) = \begin{cases} 500\sin(100\pi t) & \text{on } \Gamma_{in} \\ 0 & \text{on } \Gamma_{out}, \end{cases} \tag{3.43}$$

and $0 \leqslant t \leqslant 0.06$, so that the inflow pressure wave completes three periods in a simulation.

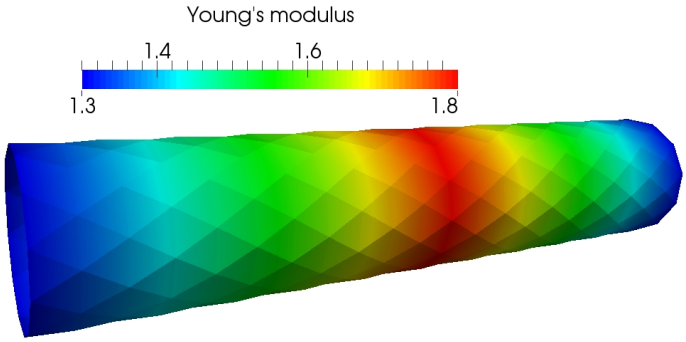


Figure 3.3: Piecewise linear distribution of the Young's modulus for the cylinder test case.

For the minimization of the functional with BFGS we used tolerances on both the functional and its gradient. In particular, we picked relative tolerances of 10^{-3} and absolute tolerances of 10^{-5} on both. These values were determined experimentally: larger values for tolerances led to inaccurate results, while smaller values led to eccessive computational costs, without significantly increasing the accuracy of the results. We also mention that we used the Tikhonov regularization 3.29, with regularization parameter $\gamma = 10^{-2}$. Also this value has been determined experimentally. In particular, smaller values made the functional oscillate more, and be more sensitive to noise. On the other hand, larger values were also not satisfactory, since they excessively affected the functional. The reference value in (3.29) is taken as the average of the previous 5 estimates (or the available ones, for the first five time steps).

The history of the estimates of the Young's modulus is reported in Figure 3.4. In particular, the blue line represents the estimates for y_1, the red line the estimates for y_2 and the green line the estimates for y_3. The two dashed lines represent the correct values of the components of $\underline{y}$ that were used to generate the synthetic measures.

It is evident from the figure that the first component to approach the correct value is y_1, followed by y_2 and, only after several time steps, by y_3. This is expected and reasonable, since the pressure wave propagates in the cylinder with finite velocity, and therefore the displacement of the membrane in the downstream sections is very small at the beginning. As a consequence, at the beginning, a change in the value of the Young's modulus in the downstream sections does not signifi-

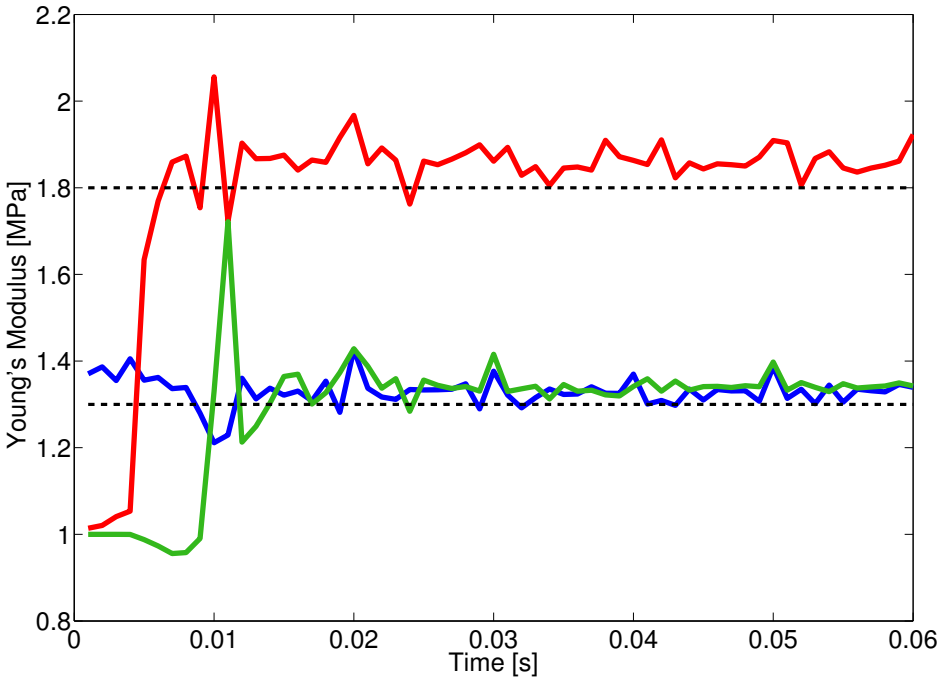


Figure 3.4: Young's modulus estimates for the cylinder test case. Here, SNR=10 (meaning 10% noise).

cantly influence the value of the functional. However, after a few time steps, the displacement in the downstream sections starts to be relevant enough to influence the functional and its gradient in the minimization routine.

In Table 3.1 we reported a summary of the minimization routine. On average, the BFGS routine took around 4 iterations to converge, as it can be inferred from the number of evaluation of the functional and its gradient. In fact, each iteration of the BFGS algorithm require one evaluation of the gradient (to compute the descent direction) and one or more evaluation of the functional (depending on whether or not the line search is successful at the first evaluation). In our simulation, we used the Armijo's criterion [65] for the line search, with a target reduction of 1%, which seldom required more than one evaluation of the functional to achieve a sufficient reduction of the functional.

Another interesting observation that we can extract from Table 3.1 is that the assimilation procedure gave an average of the estimates that has a relative error of less than 5%. This is interesting especially if compared to the value of the SNR,

Summary	
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.33, 1.87, 1.35]·10^6
rel. error	3.27%
exec. time	2175s
Functional eval.	251
Gradient eval.	251

Table 3.1: Summary of the minimization of the functional using BFGS for the cylindrical test case.

which was 10, meaning that 10% of the measures consisted of noise. This fact, as we will see, is observed also on other geometries and also in the ROM approach, which means that the constrained optimization is able to reduce the error, when interpreted as a map from the measures to the parameters.

3.4.2 Idealized aortic arch case

The geometry for the cylindrical test case is shown in Figure 3.2(b). The idealized aortic arch consists of half a torus completed by a cylinder. In particular, the major and minor radii of the torus (i.e. the distance between the center of the torus and the centerline of the pipe and the radius of the pipe respectively) are $R = 1.5cm$ and $r = 0.5cm$, while the length of the cylindrical part is $L = 5cm$. For this case we consider, instead of a piecewise linear, a piecewise constant distribution for the Young's modulus, with $\underline{y} \in \mathbb{R}^3$. More precisely, y_1 and y_2 correspond, respectively, to the value of the Young's modulus in the first and second quarter of the torus, while y_3 corresponds to the value of the Young's modulus in the cylindrical part.

The synthetic measures are generated with $\underline{y} = [1.3, 1.8, 1.3] \cdot 10^6\ dyn/cm^2$. The corresponding distribution of the Young's modulus is pictured in Figure 3.5. The initial condition is at rest, while at the inflow (Γ_{in}) and outflow (Γ_{out}) sections we impose the same Neumann boundary conditions that we used in the cylindrical case:

$$-p\mathbf{n} + \left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\mathbf{n} = g(t)\mathbf{n}, \tag{3.44}$$

where

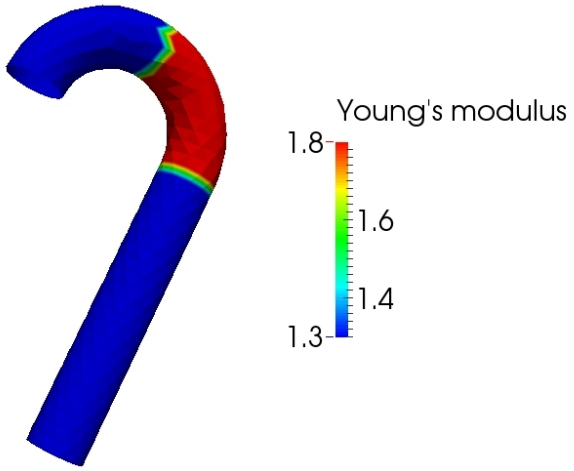


Figure 3.5: Piecewise constant distribution of the Young's modulus for the idealized aortic arch test case.

$$g(t) = \begin{cases} 500 \sin(100\pi t) & \text{on } \Gamma_{in} \\ 0 & \text{on } \Gamma_{out}, \end{cases} \tag{3.45}$$

and $0 \leqslant t \leqslant 0.06$, so that the inflow pressure wave completes three periods in a simulation.

The tolerances used for the minimization with the BFGS method are the same as for the cylinder test case, namely relative tolerances of 10^{-3} and absolute tolerances of 10^{-5}, on both the functional and its gradient. The functional is regularized with Tikhonov regularization (3.29), with coefficient $\gamma = 10^{-2}$, with reference value taken as the average of the previous 5 estimates (or the available ones, for the first five time steps).

The estimates of the three components of $\underline{y}$ are plotted in Figure 3.6. In particular, the blue line refers to the estimates of y_1, the red line to the estimates of y_2 and the green line to the estimates of y_3. As we already noticed in the cylinder test case, the estimates for y_1 are the first to approach the correct value used to generate the synthetic measures, followed by the estimates of y_2 and y_3. In this test case, however, the estimates are somewhat more oscillatory around the correct value.

A summary of the performance of the minimization routine is reported in Table 3.2. As for the cylinder test case, the number of BFGS iterations per time step is, on

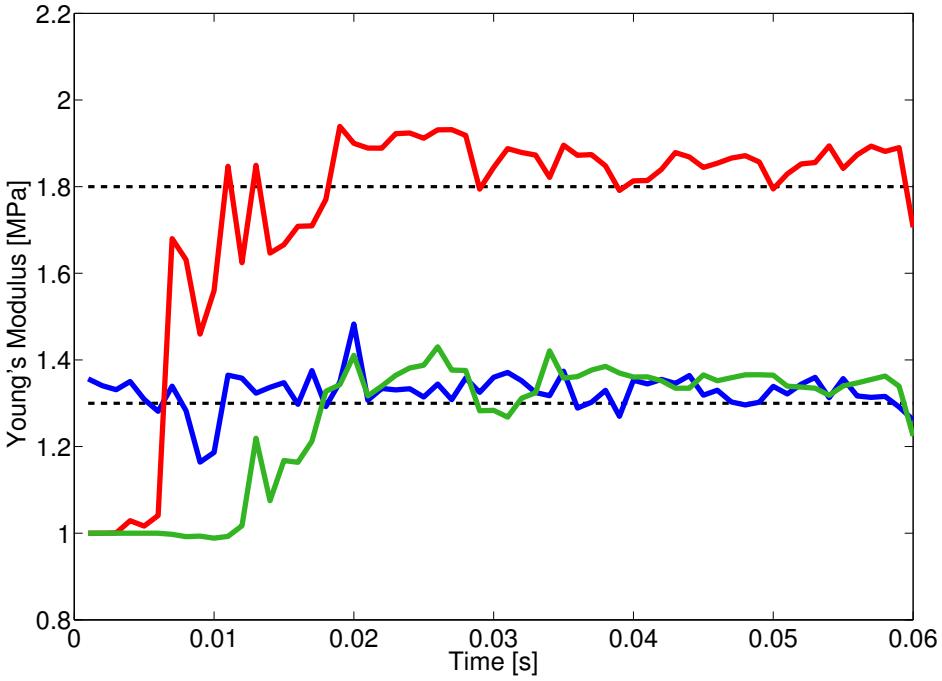


Figure 3.6: Young's modulus estimates for the idealize aortic arch test case. Here, SNR=10 (meaning 10% noise).

Summary	
$\underline{y}^{ex}$	$[1.3, 1.8, 1.3]\cdot 10^6$
$\underline{y}^*$	$[1.33, 1.84, 1.31]\cdot 10^6$
rel. error	1.91%
exec. time	3176s
Functional eval.	246
Gradient eval.	246

Table 3.2: Summary of the minimization of the functional using BFGS for the idealized aortic arch test case.

average, four. The relative error of the estimates is also significantly smaller than the error in the measures, as it was the case in the cylinder test case, passing from 10% in the measures to less than 2% in the estimates.

The computational costs associated with the solution of the inverse problem are however elevated. The simulations for both the cylinder and idealized aortic arch were performed on a personal laptop, with 2 processors (1.7GHz per processor). The number of unknowns was somewhat limited: for the fluid velocity, we used 9186 total d.o.f. for the cylinder case and 13017 total d.o.f. for the idealized aortic arch. In the next chapter we will present a POD approach for the reduction of these costs.

Chapter 4

Reduced Order Modeling for the compliance estimation problem

In this chapter we show how Reduced Order Models (ROM) can be used to reduce the computational costs in the solution of the inverse problem presented in the previous chapter. First, we will introduce the concept of ROMs in general, illustrating the two most common approaches used to build the reduced model. Then we will present the ROM that we used for the problem of the estimation of the compliance. Finally we will show some results, comparing the ROM approach with the Full Order Model (FOM) approach that we presented in the previous chapter.

4.1 Reduced Order Models

When using a mathematical model to describe a complex system, in order to have an accurate solution, the number of variables (and equations) must usually be fairly large and, in addition, may have to be solved several times, as often happens in optimization problems. This means that tackling the problem is expensive both from the storage and, most importantly, the computational point of view. To reduce the computational costs, there are two main strategies:

- reduce the complexity of the model
- reduce the size of the space where a solution is sought.

In the first case, further assumptions are made, so that the set of equations to be solved simplifies to a smaller set of equations or to a different (and simpler) type of equation. This is precisely what we presented in Section 2.4, where a 2D model has been used for the vessel wall instead of a full 3D one. However, the model cannot be simplified beyond a certain level, otherwise the quality and reliability of the solution would deteriorate. In the example presented in (2.3), a simplification of the model corresponds to an increase of the model error term ε^n.

If the model cannot be further simplified without losing reliability, then additional reduction in the computational costs can be achieved by reducing the dimension of the space of solutions. To explain the concept, consider the simple algebraic problem

$$\mathrm{A}(\mu)\underline{x} = \underline{b}(\mu), \tag{4.1}$$

where $\mu \in \mathbb{R}$ is a parameter, $\mathrm{A} \in \mathbb{R}^{n\times n}$ and $\underline{x}, \underline{b} \in \mathbb{R}^n$. In general, different values of μ would lead to different solutions. It is convenient at this point to think of $\underline{x}$ as a *function* of the parameter μ. Then, the map that at each parameter associates a solution defines a trajectory (or a hyperplane, if μ is a vector) in the space of solutions $\mathbb{R}^n$. Clearly, each point along this trajectory has a representation with respect to the canonical basis, of the form

$$\underline{x}(\mu) = \sum_{i=1}^{n} x_i(\mu)\underline{e}_i. \tag{4.2}$$

If we assume that the dependence of $\underline{x}$ on μ is *smooth*, then, for $\mu_{min} \leqslant \mu \leqslant \mu_{max}$, it is reasonable to look for a solution of the form

$$\underline{x}(\mu) = \sum_{i=1}^{N} c_i(\mu)\underline{v}_i, \tag{4.3}$$

where $N << n$, $\underline{v}_i \in \mathbb{R}^n$, and, without loss of generality[1], $\underline{v}_i^T \underline{v}_j = \delta_{ij}$.

Definition 4.1. The vectors $\underline{v}_1, \ldots, \underline{v}_N$ are called the *reduced basis* and the space they span is called the *reduced space*. Congruently, we will sometimes refer to the ambient space $\mathbb{R}^n$ as the *full space*.

[1] In fact, we can always ortho-normalize the vectors first.

If we let $\mathrm{W} = [\underline{v}_1, \ldots, \underline{v}_N]$ be the matrix containing the new basis, then (4.3) can be written as

$$\underline{x}(\mu) = \mathrm{W}\underline{c}, \tag{4.4}$$

where $\underline{c} \in \mathbb{R}^N$ is the unknown in the reduced space. If we plug this expression in (4.1), we obtain

$$\mathrm{A}(\mu)\mathrm{W}\underline{c} = \underline{b}(\mu). \tag{4.5}$$

This is an overdetermined system, which does not, in general, admit a solution. To overcome this issue, one has two strategies:

a) Least Squares: minimize the residual in the full space.

b) Galerkin: solve exactly the problem in the reduced space.

This alternative is similar to the one that arises when using Krylov methods to solve linear systems. In that framework, choice a) leads to the GMRES method, while choice b) leads to the Conjugate Gradient method (provided the matrix is symmetric and positive definite).

In the context of reduced basis, the first approach leads to the system of the *normal equations*, projected onto the reduced space

$$\mathrm{W}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{A}\mathrm{W}\underline{c} = \mathrm{W}^{\mathrm{T}}\underline{b}(\mu), \tag{4.6}$$

with the drawback that the condition number would increase. The second approach, on the other hand, leads to the linear system

$$\mathrm{A}_r(\mu)\underline{c} = \underline{b}_r(\mu), \tag{4.7}$$

where $\mathrm{A}_r(\mu) = \mathrm{W}^T\mathrm{A}(\mu)\mathrm{W}$ and $\underline{b}_r(\mu) = \mathrm{W}^T\underline{b}(\mu)$. This linear system, which is of order m, in general features a dense matrix A_r and can be tackled with a direct method. Let us denote with $\underline{x}_r$ the projection to the full space of the solution computed in the reduced space, that is $\underline{x} = \mathrm{W}\underline{c}$. Clearly, the crucial question is: how well does $\underline{x}_r$ approximate $\underline{x}$? To answer this question, let P and R be, respectively, the projector

from $\mathbb{R}^n$ onto the reduced space and its orthogonal complement. In terms of the matrix W, these projectors are given by

$$\mathrm{P} = \mathrm{W}\mathrm{W}^T \qquad \mathrm{R} = \mathrm{I} - \mathrm{W}\mathrm{W}^T. \tag{4.8}$$

Then, the reduced basis approximation error is given by

$$\begin{aligned}
\underline{x} - \mathrm{W}\underline{c} =& \underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\underline{b} = \underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\underline{x} = && (4.9)\\
=& \underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}(\mathrm{P}+\mathrm{R})\underline{x} = && (4.10)\\
=& \underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\mathrm{W}\mathrm{W}^T\underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\mathrm{R}\underline{x} = && (4.11)\\
=& \underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{A}_r\mathrm{W}^T\underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\mathrm{R}\underline{x} = && (4.12)\\
=& \underline{x} - \mathrm{W}\mathrm{W}^T\underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\mathrm{R}\underline{x} = && (4.13)\\
=& \mathrm{R}\underline{x} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A}\mathrm{R}\underline{x} = (\mathrm{I} - \mathrm{W}\mathrm{A}_r^{-1}\mathrm{W}^T\mathrm{A})\mathrm{R}\underline{x}, && (4.14)
\end{aligned}$$

where we omitted the dependency on μ for the sake of brevity, and we used the fact that $\mathrm{P} + \mathrm{R} = \mathrm{I}$. This expression shows that the error is small if either $\mathrm{W}A_r^{-1}\mathrm{W}^T$ is a good approximation of A^{-1} or if $\mathrm{R}\underline{x}$ is small. Since the matrix A may be large and depend on μ, the first requirement appears much harder to fulfill in general. On the other hand, the projector R has norm one, so the term $\mathrm{R}\underline{x}$ cannot be small for every $\underline{x}$. However, if we limit our attention to the vectors $\underline{x}$ that are solutions of (4.1) with $\mu_{min} \leqslant \mu \leqslant \mu_{max}$, then it is possible to construct a basis that fulfills the second requirement.

A ROM approach consists then of two stages:

- *offline*: it is the stage where the RB is constructed. The computational costs of this stage may be rather high.
- *online*: it is the stage where the offline work is exploited to solve the problem (4.1) for several values of the parameter μ. The solution of each instance of problem (4.7) should be remarkably cheaper compared to problem (4.1).

In Sections 4.1.1 and 4.1.2 we will present two of the most common methods used in literature to construct the RB, namely the so-called *greedy approach* and the

Proper Orthogonal Decomposition (POD). For both of these methods, the starting point is the construction of a *training set* for the parameter μ:

$$S = \{\mu_1, \dots, \mu_m\}. \tag{4.15}$$

Then, using the solution corresponding to one (or all) of these values, these methods build a basis that is guaranteed to approximate any solution $\underline{x}(\mu)$ within a desired tolerance for every $\mu \in S$.

4.1.1 Greedy Reduced Basis

In the so called *Greedy Reduced Basis* (GRB) approach, the basis is built *bottom-up*; that is, we start with an empty basis, and then add vectors according to a particular rule, until some stopping criterion is satisfied. In particular, given a current RB of size k, the algorithm estimates the RB approximation error for every value of the parameter in S, then adds to the RB the solution corresponding to the value of the parameter that maximizes the error estimates. Since in this thesis we will not use the GRB approach, we provide only a quick overview of it, in order to highlight the differences with the POD method, and we refer to the literature for further details (see e.g., [15, 26, 55, 83] and references therein).

Let us consider the problem (4.1), that is a steady discrete linear problem. We will denote by V_N the reduced space generated by the basis vectors $\underline{v}_1, \dots, \underline{v}_N$. Given a training set S for the parameter, and a tolerance $\tau > 0$, the GRB procedure is summarized in Algorithm 3.

Algorithm 3 GRB

1: Set $N = 0$, $V_0 = \varnothing$ and $\varepsilon = \infty$.
2: **while** $\varepsilon > \tau$ and $N < n$ **do**
3: Compute $\underline{x}(\mu_j)$ for all $\mu_j \in S$.
4: Compute $\Delta_N(\mu_j)$ for all $\mu_j \in S$.
5: Find $\mu_j = \arg\max_S \Delta_N(\mu_j)$ and set $\varepsilon = \Delta_N(\mu_j)$
6: Solve the full order model $\mathrm{A}(\mu_j)\underline{x} = \underline{b}(\mu_j)$
7: Expand the reduced space: $V_{N+1} = V_N \oplus span\{\underline{x}\}$
8: $N \leftarrow N + 1$
9: **end while**

Here we denoted with $\Delta_N(\mu)$ the estimate of the RB approximation error for the solution corresponding to the parameter μ when the size of the reduced space is N.

Clearly, the error estimator $\Delta_N(\mu)$ plays a key role in this process. In order to be reliable, the estimate must be sharp and effective. On the other hand, since at every iteration it must be computed for every parameter in the training set, it must also be cheap to evaluate, otherwise the construction time overhead may defeat the purpose of the whole procedure. For these reasons, its construction must be tuned on the problem at hand and, depending on the nature of the problem, may involve the estimation of many factors. As a matter of fact, there is a vast literature on problem specific *a posteriori* error estimators (see e.g., [20, 50, 82]). However, in most of the cases, the main idea is to exploit the *error relation*

$$\mathrm{A}(\mu)\underline{e}_N = \underline{r}_N(\mu), \tag{4.16}$$

where

$$\underline{e}_N := \underline{x}_N - \underline{x}, \qquad \underline{r}_N = \underline{b}(\mu) - \mathrm{A}(\mu)\underline{x}_N, \tag{4.17}$$

with $\underline{x}_N$ being the RB solution corresponding to the basis $\underline{v}_1, \ldots, \underline{v}_N$. For instance, in the case wher A is *positive definite*, multiplying both side by $\underline{e}_N$ and using the Schwarz inequality, one obtains

$$\alpha(\mu)\|\underline{e}_N\|^2 \leqslant \underline{e}_N^T \mathrm{A}(\mu)\underline{e}_N = \underline{e}_N^T \underline{r}_N(\mu) \leqslant \|\underline{e}_N\|\|\underline{r}_N(\mu)\|, \tag{4.18}$$

where $\alpha(\mu) = \min \mathcal{R}e\,[\lambda(\mathrm{A}(\mu))]$. If $\mathrm{A}(\mu)$ is also symmetric, then $\alpha(\mu) = \lambda_{min}(\mathrm{A}(\mu))$. The error estimator is then defined as

$$\Delta_N(\mu) = \frac{\|\underline{r}_N(\mu)\|}{\alpha_{LB}(\mu)}, \tag{4.19}$$

where $\alpha_{LB}(\mu)$ is a *lower bound* for $\alpha(\mu)$ which can be computed in a reasonable time. One possibility is to use the so called *Successive Constraint Method* (SCM), which allows to compute a lower bound for $\alpha(\mu)$ for every $\mu \in S$, by means of solving a certain number of linear optimization problems [46].

The positive and negative aspects of the GRB method can be summarized as follows (see, for instance, [68]):

- ⊕ The full order model is solved only for a limited number of parameters.
- ⊕ The RB approximation error decreases monotonically with respect to the basis dimension.
- ⊖ The procedure needs a sharp and effective *a posteriori* error estimator.
- ⊖ The basis is not optimal, in the sense that there may be a basis of smaller size that achieves the same approximation error.

The dependence of the GRB method on an error estimator constrains its use to a limited class of problems, where such an estimator is available. Although in the last ten years this class has been enlarged, this still remains the major limitation of the method.

4.1.2 Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) approaches the problem of the construction of a RB from a different that can be regarded as the dual of that of the GRB method. The GRB method starts from an empty basis and progressively adds the vectors that help to improve the approximation error estimates. On the other hand, the POD starts from the set of all the solutions corresponding to the training set parameters, and discards redundant information.

POD is a technique that have been used for decades in different fields, and has been named in different ways, depending on the field: in statistics it is often called Principal Component Analysis (PCA) or Hotelling transform [47], in the theory of stochastic processes it is known as Karhunen-Loéve transform [57], in whether forecast and meteorology is known as Empirical Orthogonal Functions analysis [58], in matrix analysis it is known as Eckart-Young theorem [43] (although the result was first shown by Schmidt [86]).

To see how the POD method works, assume that we have a given set of values for the parameter $S = \{\mu_1, \ldots, \mu_m\}$, which we will call *training set*, and that we have

solved the problem (4.1) for all the parameters in S. We can gather the solutions in the so called *snapshots matrix*

$$\mathrm{X} = [\underline{x}_1, \ldots, \underline{x}_m], \tag{4.20}$$

where $\underline{x}_i$ is the solution corresponding to the parameter μ_i. In the following, we will always assume that $m \leqslant n$, that is, the number of snapshots is smaller than the dimension of each snapshot[2]. Furthermore, without loss of generality, we will assume that the average of all the snapshots is $\underline{0}$. Otherwise, one has to subtract the average to every snapshot and apply the POD to the matrix $\mathrm{X} = [\underline{x}_1 - \overline{\underline{x}}, \ldots, \underline{x}_m - \overline{\underline{x}}]$, where $\overline{\underline{x}}$ is the average of all the snapshots.

The POD aims to find a low dimensional approximation of the range of X. In particular, we look for a matrix X_r such that

$$\|\mathrm{X} - \mathrm{X}_r\| \leqslant \tau \|\mathrm{X}\|, \quad rank(\mathrm{X}_r) = r \tag{4.21}$$

where τ is a given tolerance. In general, different choices for the matrix norm would lead to different matrices X_r. However, if we restrict our attention to *unitarily invariant* norms[3], then the matrix X_r has a precise structure in terms of the *singular value decomposition* (SVD) of X. In particular, let

$$\mathrm{X} = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}} \tag{4.22}$$

be the SVD of X, where $\mathrm{U} \in \mathbb{R}^{n\times n}$ and $\mathrm{V} \in \mathbb{R}^{m\times m}$ are unitary and $\Sigma \in \mathbb{R}^{n\times m}$ is such that $\Sigma_{ij} = 0$ for $i \neq j$, and $\Sigma_{11} \geqslant \Sigma_{22} \geqslant \cdots \geqslant \Sigma_{mm} \geqslant 0$. Moreover, if $rank(A) = k < m$, then $\Sigma_{ii} = 0$ for $i > k$. The numbers $\sigma_i := \Sigma_{ii}$ are called the *singular values* of X, while the columns of U and V are the left and right *singular vectors* respectively. The non-zero singular values can be characterized as the non-negative square roots of the non-zero eigenvalues of $\mathrm{A}^{\mathrm{T}}\mathrm{A}$ (or, equivalently, $\mathrm{A}\mathrm{A}^{\mathrm{T}}$).

[2] This is not a requirement, but rather a way to fix the notation: it allows us to write, in the following, the matrix Σ in the SVD of X as a matrix with more rows than columns.

[3] $\|\cdot\|$ is unitarily invariant if $\|\mathrm{UAV}\| = \|\mathrm{A}\|$ for every unitary matrix U, V (of the proper dimensions).

Let us decompose U, Σ and V as

$$\mathrm{U} = \begin{bmatrix} \mathrm{U}_r & \mathrm{U}_{n-r} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_r & \\ & \Sigma_{n-r} \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} \mathrm{V}_r & \mathrm{V}_{m-r} \end{bmatrix} \tag{4.23}$$

where $\mathrm{U}_r \in \mathbb{R}^{n\times r}$ and $\mathrm{V}_r \in \mathbb{R}^{m\times r}$ contain the first r columns of U and V respectively, while U_{n-r} and V_{m-r} contain the remaining columns of U and V respectively. The matrix $\Sigma_r \in \mathbb{R}^{r\times r}$ is the restriction of Σ to the first r rows and r columns. In [61] it is shown that for a unitarily invariant norm $\|\cdot\|$, the matrix of rank r that minimizes (4.21) is given by

$$\mathrm{X}_r = \mathrm{U}_r \Sigma_r \mathrm{V}_r^{\mathrm{T}}, \tag{4.24}$$

which is usually referred to as *truncated* SVD (TSVD). Moreover, the corresponding approximation error depends solely on the singular values of X and r. In fact, since we are focusing on a unitarily invariant norm, we have that $\|\mathrm{X}\| = \|\Sigma\|$, and therefore we can express (4.21) as

$$\tau\|\Sigma\| \geqslant \left\|\mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}} - \mathrm{U}_r\Sigma_r\mathrm{V}_r^{\mathrm{T}}\right\| = \left\|\Sigma - \mathrm{U}^{\mathrm{T}}\mathrm{U}_r\Sigma_r\mathrm{V}_r^{\mathrm{T}}\mathrm{V}\right\| = \tag{4.25}$$

$$= \left\| \begin{bmatrix} \Sigma_r & \\ & \Sigma_{n-r} \end{bmatrix} - \begin{bmatrix} \mathrm{I}_r \\ \mathrm{O} \end{bmatrix} \Sigma_r \begin{bmatrix} \mathrm{I}_r & \mathrm{O} \end{bmatrix} \right\| = \|\Sigma_{n-r}\|. \tag{4.26}$$

John von Neumann showed [101] that if $\|A\|$ is a unitarily invariant matrix norm, then $\|A\| = \|\!|\underline{\sigma}|\!\|$, where $\|\!|\cdot|\!\|$ is a vector norm and $\underline{\sigma}$ is the vector of the singular values of A[4]. Therefore, the POD first step is to find r such that

$$\|\!|(\sigma_1, \ldots, \sigma_r)|\!\| \geqslant \tau \|\!|(\sigma_1, \ldots, \sigma_m)|\!\|. \tag{4.27}$$

Let us show what this condition translates into for the unitarily invariant matrix norms that are more frequently used:

i) spectral norm: $\|\mathrm{A}\|_2 = \max_{\|\underline{x}\|_2=1} \|\mathrm{A}\underline{x}\|_2$. It is possible to show[5] that this definition

[4] More precisely, he showed that to each unitarily invariant norm $\|\mathrm{A}\|$ corresponds a *symmetric gauge function* g such that $\|\mathrm{A}\| = g(\underline{\sigma})$.

[5] For instance, using Lagrange multipliers.

leads to $\|\mathrm{A}\|_2 = \sqrt{\rho(\mathrm{A}^{\mathrm{T}}\mathrm{A})}$, where $\rho(\mathrm{A})$ is the *spectral radius* of A. Therefore, $\|\mathrm{A}\|_2 = \sigma_1$ and, since the singular values are ordered, $\|\mathrm{A}\|_2 = \|\underline{\sigma}\|_\infty$. The POD truncation criterion for the spectral norm case is therefore

$$\frac{\sigma_r}{\sigma_1} \geqslant \tau. \tag{4.28}$$

ii) Frobenius norm: $\|\mathrm{A}\|_F = \sqrt{\sum |\mathrm{A}_{ij}|^2}$. It is easy to show that $\|\mathrm{A}\|_F = \sqrt{tr(\mathrm{A}^{\mathrm{T}}\mathrm{A})}$ and, since the trace is the sum of the eigenvalues, $\|\mathrm{A}\|_F = \sqrt{\sum \lambda_i(\mathrm{A}^{\mathrm{T}}\mathrm{A})} = \sqrt{\sum \sigma_i^2}$, by the characterization of the singular values. The Frobenius norm is therefore equivalent to the 2-norm on $\underline{\sigma}$, that is $\|\mathrm{A}\|_F = \|\underline{\sigma}\|_2$. The POD truncation criterion for the Frobenius norm case is

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2} \geqslant \tau^2. \tag{4.29}$$

iii) nuclear norm: $\|\mathrm{A}\|_* = \sum \sigma_i$. The nuclear norm is, by definition, the sum of the singular values, and is therefore equivalent to the 1-norm on $\underline{\sigma}$. The POD stopping criterion is therefore

$$\frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{m} \sigma_i} \geqslant \tau. \tag{4.30}$$

Once the SVD of X has been computed and the rank of the approximating matrix X_r has been determined, the construction of the reduced basis needs no additional time. In fact, we know from the property of the SVD that the left singular vectors form an orthonormal basis for the range of X. Therefore, one simply takes $\mathrm{W} = \mathrm{U}_r$.

The POD from a statistical perspective

It is interesting to analyze the difference between the three unitarily invariant norms we presented, when used inside the POD method. In particular, if we forget the fact

that the snapshots are the solutions to the problem (4.1) corresponding to different values of μ, and we regard them simply as points in $\mathbb{R}^n$, we may consider them as realization of the random variable $\underline{x}$. From the statistical point of view, the POD is a techniques that aims to identify privileged directions, that capture most of the variability of this dataset. In statistics, this analysis is usually called Principal Component Analysis (PCA): through a SVD, the dataset is rotated on the reference frame generated by the left singular vectors of X. In this context, the singular values are the variability of the dataset along the direction of the corresponding left singular vector. In the PCA terminology, the singular vectors are called *principal components* and the singular values correspond to the *sample standard deviation* along the corresponding principal component. The three norms presented in the previous section correspond to different ways to measure the total variability of the dataset when projected on a subset of the principal components.

The spectral norm is remarkably different from the other two. In fact, it measures the variability of a dataset based only on the variability along the first principal component. Therefore, once the direction with the largest variance has been identified, each other direction is marked as statistically relevant (or irrelevant) based solely on how its variability compares with σ_1. No information about the variance along the other principal components is needed.

The Frobenius and nuclear norm, on the other hand, both consider the dataset variance along all the principal components to establish if a component is needed to describe the dataset variability. However, they weigh the variabilities along the principal components in a different way. The Frobenius norm considers the squares of the singular values, which, in statistical terms, represent the sample variances along each principal direction. On the other hand, the nuclear norm considers the singular values themselves, that is, the sample standard deviations. The Frobenius norm is therefore more meaningful from the statistical point of view, and in fact, it is the norm mostly used in PCA.

4.2 A POD approach for the compliance estimation problem

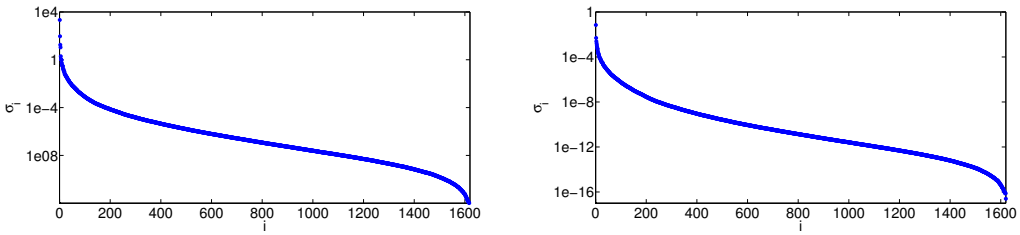
In this section we present how the POD method can be applied to the problem of the estimation of the compliance of a vessel. In particular, we want to reduce the computational costs associated with the solution of the optimization problem (3.33).

To motivate the choice of using a RB method for this problem, it is useful to look at the correlation between solutions corresponding to different values of $\underline{y}$ and at different instants in time. In particular, let us consider the fluid-membrane interaction problem (3.15), with $y_i \in \{1,2,3\} \cdot 10^6 dyn/cm^2$, and inflow and outflow boundary conditions given by

$$\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right) \mathbf{n} + p\mathbf{n} = g(t)\mathbf{n}, \tag{4.31}$$

with $g(t) = 0$ at the outflow (homogenous Neumann conditions) and $g(t) = 500(1 + \sin(100\pi t))$ at the inflow. We chose $\Delta t = 0.001$ and store the results for 60 time steps and build the corresponding snapshots matrix. Here, we consider the solutions as if the time t were itself a parameter. In other words, $\underline{u}^{n+1}$ and $\underline{u}^n$ are considered as different snapshots, even if they are computed using the same Young's modulus. Nevertheless, we expect a rather smooth dynamics in time, so that the singular values of the snapshots matrix should decay of several orders of magnitude rather quickly. In Figure 4.1 we show the behavior of the singular values of the snapshots matrices of velocity and displacement. As we expected, the magnitude of the singular values decays rather quickly at the beginning and, although there is no gap in the spectrum, it is reasonable to infer that $\underline{u}$ and $\underline{\eta}$ can be well approximated by vectors belonging to spaces of dimensions much lower than the corresponding finite element ones (which, in this example, are $N_h = 36,624$ for the velocity and $N_h = 7,760$ for the displacement).

To construct a Reduced Basis (RB) we decide to use the POD approach since, at the best of our knowledge, no *a priori* error estimator is known for the fluid-membrane interaction problem (3.14). Moreover, we want to take advantage of the



(a) Velocity (b) Displacement

Figure 4.1: Singular values of the velocity (left) and membrane displacement (right) snapshot matrices. The snapshots have been first amended of the non-homogeneous boundary conditions and cenetered around the sample average.

fact that the snapshots are divergence free. To understand how, let us assume we already computed the RB for the velocity space and let us denote by W_u the matrix whose columns are the RB vectors. If we project the momentum equation onto the velocity reduced space, we obtain

$$\mathrm{W}_u^{\mathrm{T}}\mathrm{C}(\underline{y})\underline{u}^n + \mathrm{W}_u^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\underline{\delta p}^n = \mathrm{W}_u^{\mathrm{T}}\underline{f}(\underline{y}) - \mathrm{W}_u^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\underline{p}^*. \tag{4.32}$$

Should the geometry be fixed in time, then the term $\mathrm{W}_u^{\mathrm{T}}\mathrm{B}^{\mathrm{T}} = (\mathrm{B}\mathrm{W}_u)^{\mathrm{T}}$ would be identically zero, due to the (weak) divergence-free nature of the snapshots. In the case of a moving domain, this is no longer true, in general, for each snapshots is divergence-free only in the geometry in which it was computed. However, for small displacements, we expect the product BW to be small. Furthermore, for small time steps, we expect $\underline{\delta p}^n$ to be also small. For the sake of the computational costs, we drop the pressure term in (4.32). Notice that this is equivalent to use a time-advancing scheme where the pressure is treated explicitly. Once the momentum equation has been solved and the velocity field has been computed, the pressure can be recovered by solving a least squares problem in the full space:

$$\underline{p}^n = \underset{\underline{p}\in\mathbb{R}^n}{\arg\min}\|\underline{f}^n - \mathrm{C}\underline{u}^n - \mathrm{B}^{\mathrm{T}}\underline{p}\|^2. \tag{4.33}$$

Notice that (4.33) is well posed since the FE spaces for velocity and pressure are assumed to satisfy the *inf-sup* condition.

Remark 4.1. In order to be able to project a saddle point problem onto a reduced space without resorting to an explicit treatment of the pressure, one has to make sure that the *inf-sup* condition holds also for the reduced problem. A possible way to enforce this is to enrich the velocity reduced space, by adding the so called *supremizers*: for each pressure snapshot $\underline{p}_i$, an element $\underline{z}_i$ is added to the velocity snapshots. The new element $\underline{z}_i$ solves

$$\mathrm{H}\underline{z}_i = \mathrm{B}^{\mathrm{T}}\underline{p}_i, \tag{4.34}$$

where H is the matrix corresponding to the discretization of the inner product in H^1. In other words, $\underline{z}_i$ is the Riesz representative (in the discrete norm induced by H) of the gradient of the pressure snapshot $\underline{p}_i$. An example of this procedure can be found in [20, 82].

However, in the case of the fluid-membrane interaction problem, even adding the supremizers to the velocity snapshots does not guarantee that the projection of the saddle point problem will be *inf-sup* stable for every possible value of the Young's modulus. For this reason, in this work we decided to circumvent the *inf-sup* condition and formulate the problem in the sole velocity space.

We therefore solve the forward problem for a given set of Young's moduli $\underline{y}_1, \dots, \underline{y}_M$ and store the corresponding velocity and membrane displacement fields, $\underline{u}_i$ and $\underline{\eta}_i$. In case of non-homogeneous Dirichlet boundary conditions at the inflow and/or outflow section, we modify the velocity snapshots as

$$\hat{\underline{u}}_i = \underline{u}_i - \underline{u}_\ell, \tag{4.35}$$

where $\underline{u}_\ell$ is a (weakly) divergence-free velocity field, computed as a solution of a steady Stokes problem and used as a boundary datum lift function. This allows us to maintain the divergenge-free nature of the snapshots.

Next, the snapshots (possibly amended by the lifting) are collected in the velocity and displacement snapshots matrices X_u and X_η. We perform the POD, as described in Section 4.1.2, using the Frobenius norm to establish the truncation in-

dex in the SVD, and we collect the velocity and displacement RB in the matrices W_u and W_η.

Projecting the IFMI problem (3.33) onto the reduced space, we obtain

$$\begin{aligned} \underline{y}^n = &\underset{\underline{y}\in\mathbb{R}^k}{\arg\min}\, J_r^n(\underline{y}) = \frac{1}{2}\|\mathrm{W}_\eta\underline{\eta}^n - \eta_{meas}^n\|_\Sigma + \frac{\gamma}{2}\|\mathrm{R}(\underline{y} - \underline{y}_{ref}^n)\|_\Sigma \\ &\text{s.t.} \begin{bmatrix} \mathrm{C}_r & \mathrm{O} \\ -\Delta t\mathrm{P}_r & \mathrm{M}_{\Sigma,r} \end{bmatrix} \begin{bmatrix} \underline{u}_r^n \\ \underline{\eta}_r^n \end{bmatrix} = \begin{bmatrix} \underline{f}_r^n \\ \mathrm{M}_{\Sigma,r}\underline{\eta}_r^{n-1} \end{bmatrix} \end{aligned} \tag{4.36}$$

where, $\mathrm{C}_r = \mathrm{W}_u^{\mathrm{T}}\mathrm{C}\mathrm{W}_u$, $\mathrm{M}_{\Sigma,r} = \mathrm{W}_\eta^{\mathrm{T}}\mathrm{M}_\Sigma\mathrm{W}_\eta$, $\mathrm{P}_r = \mathrm{W}_\eta^{\mathrm{T}}\mathrm{P}\mathrm{W}_u$ and $\underline{f}_r^n = \mathrm{W}_u^{\mathrm{T}}\left(\underline{f}^n - \mathrm{B}^{\mathrm{T}}\underline{p}^{n-1}\right)$.

Remark 4.2. Notice that, since the projection onto the reduced space is orthogonal, for the sake of the minimization we can replace the first part of the functional with

$$\frac{1}{2}\left\|\underline{\eta}_r^n - \mathrm{W}^{\mathrm{T}}\underline{\eta}_{meas}^n\right\|_{\Sigma,r} \tag{4.37}$$

where $\|\underline{v}\|_{\Sigma,r} = \underline{v}^{\mathrm{T}}\mathrm{M}_{\Sigma,r}\underline{v}$ is the discrete L^2 norm in the reduced space. This allows to evaluate the functional with a number of operations that depends solely on the dimension of the reduced spaces, since the term $\mathrm{W}^{\mathrm{T}}\underline{\eta}_{meas}^n$ can be precomputed before the optimization loop.

The optimization problem (4.36) can be solved using, again, the Lagrange multipliers method. The Lagrangian for the reduced problem reads

$$\mathcal{L}(\underline{u}_r^n, \underline{\eta}_r^n, \underline{y}^n, \underline{\lambda}_u, \underline{\lambda}_\eta) = \mathcal{J}_r^n(\underline{y}) + \underline{\lambda}_u^{\mathrm{T}}\left(\mathrm{C}_r\underline{u}_r^n + \underline{f}_r^n\right) + \underline{\lambda}_\eta^{\mathrm{T}}\left(\underline{\eta}^n - \Delta t\mathrm{P}_r\underline{u}^n - \underline{\eta}^{n-1}\right) \tag{4.38}$$

where $\underline{\lambda}_u$ and $\underline{\lambda}_\eta$ are the Lagrange multipliers corresponding to the momentum and

displacement equations respectively. The KKT system then reads

$$
\text{(KKT)}\begin{cases}
\begin{bmatrix} \mathrm{C}_r^{\mathrm{T}} & -\Delta t \mathrm{P}_r^{\mathrm{T}} \\ \mathrm{O} & \mathrm{M}_{\Sigma,r} \end{bmatrix} \begin{bmatrix} \underline{\lambda}_u \\ \underline{\lambda}_\eta \end{bmatrix} = \begin{bmatrix} \underline{0} \\ -\mathrm{M}_{\Sigma,r}\underline{\eta}_r^n + \underline{d}_r^n \end{bmatrix} & \text{Adjoint} \\
\gamma \left[\mathrm{M}_R(\underline{y}^n - \underline{y}_{ref}^n) \right]_i + \underline{\lambda}_u^{\mathrm{T}} \left(\dfrac{\partial \mathrm{C}_r}{\partial y_i} \underline{u}_r^n - \dfrac{\partial \underline{f}_r^n}{\partial y_i} \right) = 0 \quad i = 1, \dots, k & \text{Optimality} \\
\begin{bmatrix} \mathrm{C}_r & \mathrm{O} \\ -\Delta t \mathrm{P}_r & \mathrm{M}_{\Sigma,r} \end{bmatrix} \begin{bmatrix} \underline{u}_r^n \\ \underline{\eta}_r^n \end{bmatrix} = \begin{bmatrix} \underline{f}_r^n \\ \mathrm{M}_{\Sigma,r}\underline{\eta}_r^{n-1} \end{bmatrix} & \text{State}
\end{cases}
\tag{4.39}
$$

where, as we did in (3.35), we set $\mathrm{M}_R = \mathrm{R}^{\mathrm{T}}\mathrm{M}_\Sigma\mathrm{R}$. The expression for the derivatives $\dfrac{\partial \mathrm{C}_r}{\partial y_i}$ and $\dfrac{\partial \underline{f}_r^n}{\partial y_i}$ is the projection onto the reduced space of the derivatives in (3.36), namely

$$
\frac{\partial \mathrm{C}_r}{\partial y_i} = \Delta t \beta \mathrm{W}_u^{\mathrm{T}} \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \mathrm{P} \mathrm{W}_u \qquad \frac{\partial \underline{f}_r^n}{\partial y_i} = -\beta \mathrm{W}_u^{\mathrm{T}} \mathrm{P}^{\mathrm{T}} \mathrm{M}_i \mathrm{W}_\eta \underline{\eta}_r^{n-1}. \tag{4.40}
$$

As we did for the full order problem in Section 3.3, we can tackle the KKT system (4.39) with the BFGS method, using the KKT system to evaluate the functional and its gradient.

Algorithm 4 shows the steps of the POD at the generic time step t^n.

Algorithm 4 POD-ROM

1: Compute the reference value $\underline{y}_r^n$ as the average of previous m estimates
2: Assemble the FE matrices/vectors
3: Load the measures and project them onto the reduced space
4: Project the FE matrices onto the reduced space
5: Using BFGS, compute $\underline{u}_r^n$, $\underline{\eta}_r^n$ and $\underline{y}^n$
6: Compute $\underline{u}^n$ and $\underline{\eta}^n$ by extending their reduced space counterparts
7: Compute $\underline{p}^n$ by solving the least squares problem (4.33)

Remark 4.3. The choice of the parameter τ in the POD is a delicate issue. In the next section we will show a numerical sensitivity study for this parameter to justify the value that we used. In [38], it is suggested to use a value 'close' to unity, but there is no unique and universal way to define what 'close' should be in practice. A possible strategy could be to look at the singular value of the snapshots matrix to see if it

is possible to identify a corner, where the singular values flatten out. However, in practice, this cutoff may lead to a RB af size much larger than what is needed to get good estimates of the parameter. As a matter of fact, it is important to point out that the reduced matrices are in general full, therefore, to achieve good computational speeds in the optimization loop, the size of the RB must stay controlled, otherwise the costs of the direct solvers used to solve the reduced system may become too large. In our experiments, even with a threshold as high as $\tau = 0.999$ we never observed a RB larger than 50 vectors.

4.3 Results

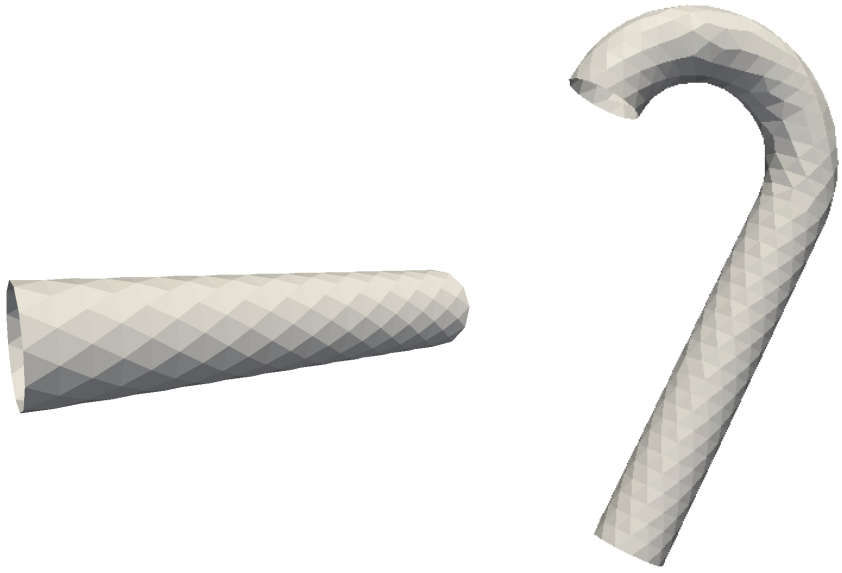
We present numerical results of the POD approach for the estimation of the compliance. As we did for the Full Space approach, we are going to use synthetic data. Let us recall how this procedure works. First, we solve the forward problem with a given (realistic) value for the Young's modulus on a mesh which is finer than that used to solve the inverse problem, in order to avoid the so called 'inverse crime'. Finally, after adding random noise to the computed solution, we project the vectors on the coarser mesh, where the inverse problem will be solved. If we denote with $\underline{\eta}_f$ the solution obtained on a finer mesh and with Π_f^c a projector from the fine to the coarse mesh, the measures used in the inverse problem can be expressed as

$$\underline{\eta}_m = \Pi_f^c \left(\underline{\eta}_f + \xi \|\underline{\eta}_f\|_\infty \underline{e} \right), \tag{4.41}$$

where $\underline{e}$ is a random vector obtained from the uniform distribution $\mathcal{U}([-1,1])$ and ξ is the reciprocal of the *signal to noise ratio* (SNR) and represents the amount of noise in the measures as a fraction of the magnitude of the 'exact' measures. We recall that, for data retrieved from medical images, the magnitude of the noise can be as large as 20% of that of the wall displacement [14], which means that the SNR can be as low as 5 and ξ as large as 0.2

To assess the effectiveness and robustness of the method with respect to noise level and availability of measures, we will perform different simulations, consider-

ing the two geometries that we used for the experiments in Section 3.4 (see Fig. 3.2). We will test the algorithm with different values of SNR, to verify how the algorithm behaves with respect to the noise level. We will also compare some results obtained with the Reduced Space (RS) approach with those presented in Section 3.4 for the Full Space (FS) approach. All the simulations have been performed using LifeV, a C++ FE library [1].



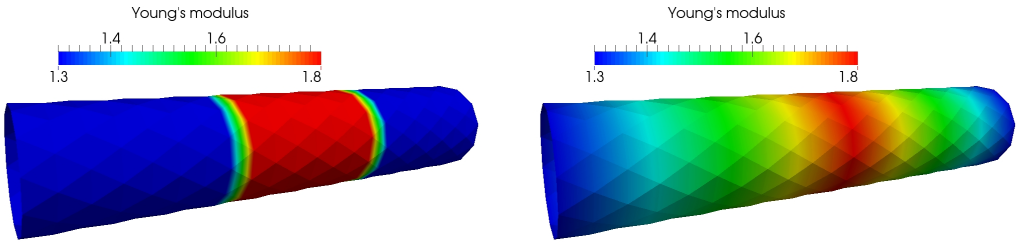
(a) Cylinder (b) Idealized aortic arch

Figure 4.2: Geometries for the cylinder (left) and idealized aortic arch (right) test cases.

4.3.1 Cylinder case

The geometry for the cylindrical test case is shown in Figure 4.2(a). It is the same geometry used in Section 3.4.1, namely a $5cm$ long cylinder with radius $0.5cm$. For the Young's modulus we will consider both the piecewise linear and piecewise constant case, with $\underline{y} \in \mathbb{R}^3$. The two possible scenarios are pictured in Figure 4.3. The initial condition is at rest, while at the inflow (Γ_{in}) and outflow (Γ_{out}) sections we impose a Neumann boundary condition of the type

$$-p\mathbf{n} + \left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\mathbf{n} = g(t)\mathbf{n}, \tag{4.42}$$



(a) Piecewise constant (b) Piecewise linear

Figure 4.3: Possible distributions of the Young's modulus: piecewise constant (left) and piecewise linear (right).

where

$$g(t) = \begin{cases} 500 \sin(100\pi t) & \text{on } \Gamma_{in} \\ 0 & \text{on } \Gamma_{out}, \end{cases} \tag{4.43}$$

and $0 \leqslant t \leqslant 0.06$, so that the inflow pressure wave completes three periods in a simulation.

Piecewise linear Young's modulus

For the piecewise linear test, we generate the synthetic measures with $\underline{y} = [1.3, 1.8, 1.3] \cdot 10^6\ dyn/cm^2$, and we added random noise with SNR=10. The RB has been generated with the training set given by

$$S = \{\underline{y} \in \mathbb{R}^3 : y_i \in \{1, 2\} \cdot 10^6\ dyn/cm^2\}, \tag{4.44}$$

that is a set consisting of 8 possible configurations for $\underline{y}$.

Remark 4.4. Notice that the value of $\underline{y}$ used to generate the measures lies inside $conv(S)$, the convex hull of S. This is not a coincidence, but rather a desired scenario. As a matter of fact, we noticed that with this choice the algorithm behaves better. Another way to see this is that, in this case, we are trying to *interpolate* the exact solution with the snapshots generated with $\underline{y} \in S$, while if the exact value lied outside $conv(S)$, the process would be more similar to an *extrapolation*. It may appear like we are using information about the exact solution to build the RB, which is not the case in a real application, where the exact solution is unknown. Neverthe-

τ	0.9	0.95	0.99
N_u	3	6	17
N_η	3	4	7

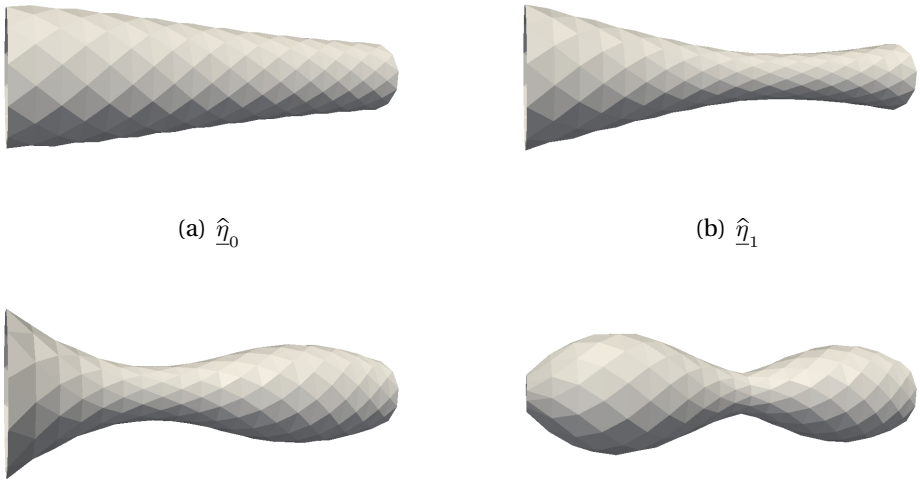
Table 4.1: Dimension of the fluid velocity and membrane displacement RB for different values of the POD threshold for the cylinder test case with piecewise linear Young's modulus distribution.

less, it is reasonable to assume that a "good-guess" range for the exact value of the parameters is available, so that one can build a training set S in such a way that the exact value of the parameters are "expected" to lie inside $conv(S)$. We also notice that, a good "guess-range" for the parameters also provide a reasonable choice for the initial guess for the minimization, namely $\underline{y}_0$ equal to the baricenter of S. In our experiments, however, we always chose $\underline{y}_0$ on the boundary of $conv(S)$, to better assess the robustness of the method.

In Table 4.1 we report the dimension of the RB size for different choices of the parameter τ used in the POD approximation (4.29). We observe that the dimension of the velocity RB grows faster than the dimension of the displacement RB. This is reasonable, if we consider that the fluid dynamics is more complex than the structure one, and that the dimension of the velocity FE space is larger than that of the displacement FE space.

In Figure 4.4 we show the first four modes of the membrane displacement RB. It is interesting to notice the increasing number of valleys and peaks along the axial direction. This reflects the fact that the singular vectors in the SVD decomposition of the snapshots matrices become more oscillatory as the index moves forward.

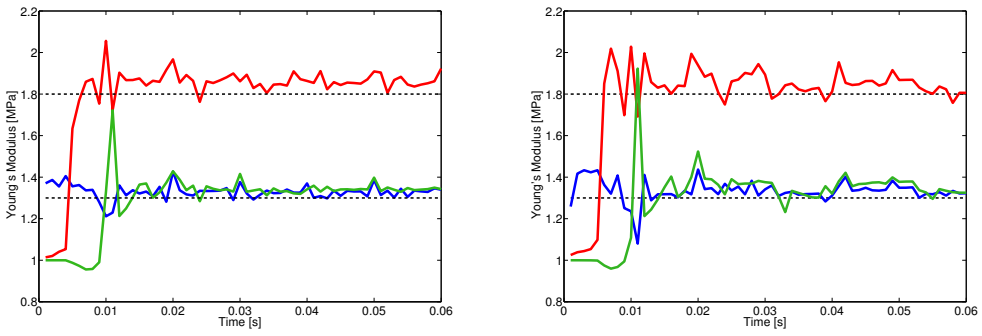
In Figure 4.5 we show the history of the Young's modulus estimates for both the FS approach and the RS approach. In particular, the RB was obtained using $\tau = 0.95$ as threshold for the POD method. The graph on the left is the same as the one in Figure 3.4. In particular, there, as well as in the next figures, the blue line refers to the estimates of y_1, the red line to the estimates of y_2 and the green line to the estimates of y_3. As we can see, the estimates obtained with the RS approach are comparable to those obtained with the FS. In particular, we notice how the delay in the estimates of the values of the Young's modulus corresponding to the downstream section is



(a) $\hat{\underline{\eta}}_0$ (b) $\hat{\underline{\eta}}_1$

(c) $\hat{\underline{\eta}}_2$ (d) $\hat{\underline{\eta}}_3$

Figure 4.4: First four modes of the membrane displacement RB for the piecewise linear Young's modulus distribution. The deformation has been amplified by a factor 3 for display purposes.



(a) Full Space (b) Reduced Space

Figure 4.5: Estimates for the piecewise linear Young's modulus distribution: FS (left), and RS (right) with $\tau = 0.95$.

reproduced similarly in both cases. This qualitative similarity is confirmed by the numerical values of the estimates, which are reported in Table 4.2, together with other information on the performance of the minimization algorithm.

In this table, the number of Navier-Stokes solves for the FS approach includes both forward and adjoint problems, while for the RS approach it refers to the Navier-Stokes solves needed in the *offline* stage to build the reduced basis. We observe that the error of the estimate obtained with the RS approach is roughly equivalent to that of the estimate obtained with the FS approach, even slightly better. We will come back on this aspect later. The execution time of the RS approach is roughly one order of magnitude smaller than that of the FS approach, although it is still non negligible. The residual computational costs are mainly due to the fact that the FE matrices have to be assembled at every time step before the problem can be projected onto the RS, due to the change in the geometry and the presence of the convective term. Although we point out that the assembly of the FE matrices is the most scalable part, it would be preferable to have online costs fully independent on the dimension of the FE spaces used in the offline stage. In Chapter 6 we will briefly discuss this issue and address a possible way to cut these costs.

We also point out that a comparison of the online execution times for the two approaches is not fair. In fact, with the RS approach, for a given new patient (hence, geometry) we first have to compute the reduced basis before we can start the optimization routine. Therefore, it is more fair to count the number of Navier-Stokes solves needed by the two approaches, that is, we also consider the offline costs for the RB approach. If we do this, we find out that the two approaches have roughly the same costs. In the concluding remarks in Chapter 6 we will discuss how these offline costs can be damped on patient specific cases.

It is also interesting to observe how the error on the estimate behave with respect to the threshold τ used in the POD for the construction of the RB. In Figure 4.6 we show the history of the estimates of the Young's modulus obtained with the RS approach using $\tau = 0.9, 0.95, 0.99$. We observe that, for $\tau = 0.9$, the method cannot estimate the value of y_3. This can be explained looking at the modes in Figure 4.4: the RB for $\tau = 0.9$ has only 3 modes, for both velocity and displacement. As we can

approach	FS	RS
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.33, 1.87, 1.35]·10^6	[1.33, 1.85, 1.36]·10^6
rel. error	3.27%	3.13%
exec. time (online)	2175s	203s
NS solves	502	480

Table 4.2: Summary of the minimization of the functional using BFGS for the cylindrical test case, for both FS and RS approaches.

see, the first modes favor deformation towards the inflow section, where most of the displacement is concentrated. Therefore, a small basis cannot accurately describe the displacement of the structure in the downstream sections.

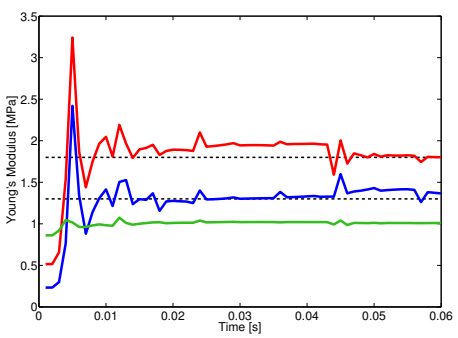
In Table 4.3 we report the time average of the estimates obtained with $\tau = 0.9$, 0.95, 0.99. The averages are computed excluding the first 10 estimates, since we want to avoid an average which is clearly biased by the initial condition. As we can observe, the estimate improve as τ approaches one, which is what we expected. For this case, however, it appears as there is no significant improvement when τ is increased from 0.95 to 0.99. Notice that, for the case $\tau = 0.9$, the large magnitude of the error is mostly due to the fact that y_3 is not estimated correctly (with a relative error of about 30%).

	τ = 0.9	τ = 0.95	τ = 0.99
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.34, 1.89, 1.01]*10^6	[1.33, 1.85, 1.36]*10^6	[1.35, 1.83, 1.33]*10^6
rel. error	11.87%	3.22%	2.57%

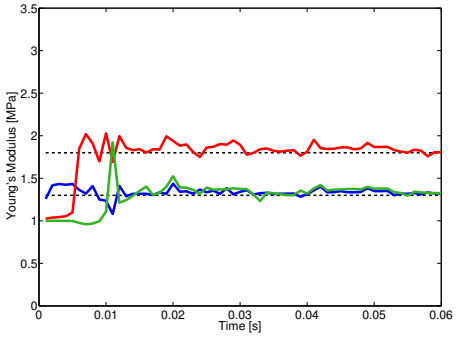
Table 4.3: Time average of the estimates and relative error for different values of the POD threshold for the cylinder test case.

Piecewise constant Young's modulus

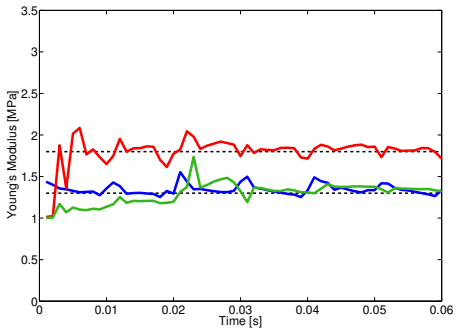
We also test our method on the same geometry but with a piecewise constant distribution of the Young's modulus. In particular, we generate the synthetic measures with $\underline{y} = [1.3, 1.8, 1.3] \cdot 10^6\ dyn/cm^2$, but the meaning of these values is clearly not the same as for the piecewise linear case: here, y_1, y_2 and y_3 represent, repsectively, the constant value of the Young's modulus in the first, second and third portion of



(a) $\tau = 0.9$



(b) $\tau = 0.95$



(c) $\tau = 0.99$

Figure 4.6: Estimates of the Young's modulus for different choices of the threshold τ in the POD method, for the case of piecewise linear Young's modulus.

τ	0.9	0.95	0.99
N_u	4	7	20
N_η	4	5	10

Table 4.4: Dimension of the fluid velocity and membrane displacement RB for different values of the POD threshold for the cylinder test case with piecewise constant Young's modulus distribution.

the cylinder, as shown in Figure 4.3(a). The RB is generated using the same training set we used in the piecewise lienar case, namely

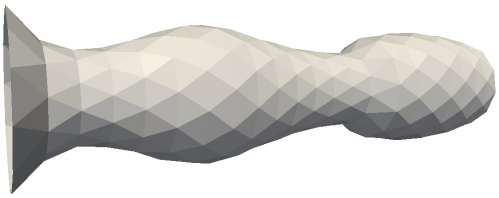
$$S = \{\underline{y} \in \mathbb{R}^3 : y_i \in \{1,2\} \cdot 10^6 \, dyn/cm^2\}, \tag{4.45}$$

that is a set consisting of 8 possible configurations for $\underline{y}$.

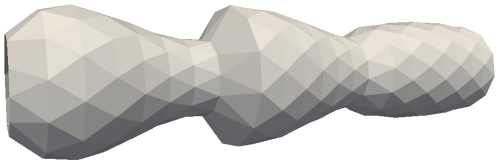
In Table 4.4 we report the dimension of the RB size for different choices of the parameter τ used in the POD approximation (4.29). As we observed also for the piecewise linear Young's modulus case, the dimension of the velocity RB grows faster than the dimension of the displacement RB. In addition, we can also observe that the dimension of the RB is slightly larger than in the case of piecewise linear Young's modulus, probably due to the fact that a piecewise constant distribution allows for more irregular modes. In particular, the first 4 modes look similar to those in Figure 4.4, but further modes allow for more irregular displacements. In Figure 4.7 we show, for instance, the fifth and sixth mode of the membrane displacement RB, where we can observe the precence of more rapid changes along the axial direction.

In Figure 4.8 we show the history of the estimates of the Young's modulus for different values of the SNR. The RB was obtained with $\tau = 0.95$. If we compute the average of the estimates, omitting, as we did before, the first 10 estimates (which, at least for y_3 are still clearly affected by the choice of the initial condition), we see that this is not true. As we can see from Table 4.5, the error in the estimate increases as the signal to noise ratio deteriorates. Nevertheless, it does not explode, and remains of the same order of magnitude as the POD truncation error.

Remark 4.5. In Figure 4.8 we observe some spikes in the estimates, especially for y_2 and y_3, with the largest spike always occurring at $t = 0.01s$ (or at the time step right



(a) $\widehat{\underline{\eta}}_4$



(b) $\widehat{\underline{\eta}}_5$

Figure 4.7: Fifth and sixth modes of the membrane displacement RB for the piecewise constant Young's modulus distribution. The deformation has been amplified by a factor 3 for display purposes.

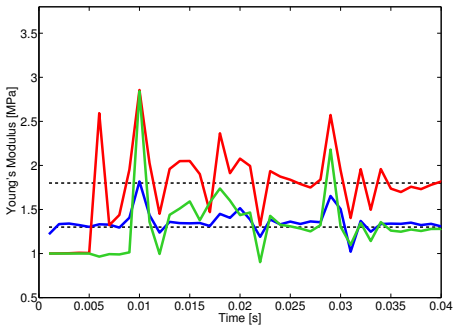
after). This can be explained by looking at the inflow boundary condition, which at $t = 0.01s$ vanishes. The inflow boundary condition is the only external forcing term. The fact that it vanishes decreases the controllability of the system. and, as a consequence, the quality of the estimates deteriorates.

	SNR = 20	SNR = 10	SNR = 5
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.35, 1.85, 1.36]*10^6	[1.36, 1.86, 1.37]*10^6	[1.37, 1.90, 1.41]*10^6
rel. error	3.76%	4.32%	6.30%

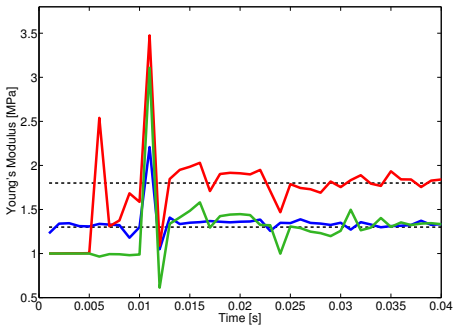
Table 4.5: Time average of the estimates and relative error for different values of SNR for the cylinder test case.

4.3.2 Idealized aortic arch case

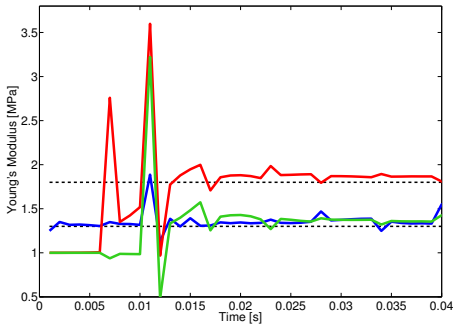
The geometry for the cylindrical test case is shown in Figure 4.2(b), and it is the same used in Section 3.4.2, namely a $5cm$ long cylinder joint with half a torus, with major and minor radii given by $R = 1.5cm$ and $r = 0.5cm$ respectively. For this case we will consider only the the piecewise constant case, with $\underline{y} \in \mathbb{R}^3$. In particular, y_1



(a) SNR=20



(b) SNR=10



(c) SNR=5

Figure 4.8: Estimates of the Young's modulus for different values of SNR for the case of piecewise constant Young's modulus.

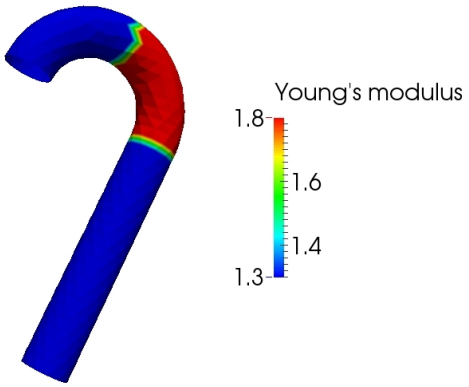


Figure 4.9: Piecewise constant distribution of the Young's modulus for the idealized aortic arch test case.

and y_2 correspond, respectively, to the value of the Young's modulus in the first and second quarter of the torus, while y_3 corresponds to the value of the Young's modulus in the cylindrical part. A distribution of a piecewise constant Young's modulus is depicted in Figure 4.9. The initial condition is at rest, while at the inflow (Γ_{in}) and outflow (Γ_{out}) sections we impose the same Neumann boundary conditions that we used in the cylindrical case:

$$-p\mathbf{n} + \left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\mathbf{n} = g(t)\mathbf{n}, \tag{4.46}$$

where

$$g(t) = \begin{cases} 500\sin\left(100\pi t\right) & \text{on } \Gamma_{in} \\ 0 & \text{on } \Gamma_{out}, \end{cases} \tag{4.47}$$

and $0 \leqslant t \leqslant 0.06$, so that the inflow pressure wave completes three periods in a simulation.

We generate the synthetic measures with $\underline{y} = [1.3, 1.8, 1.3] \cdot 10^6\ dyn/cm^2$ and we added random noise with SNR=10. The RB has been generated with the training set given by

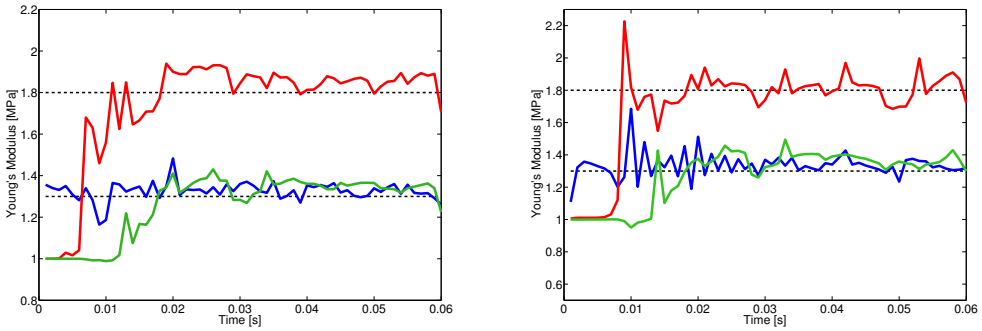
$$S = \{\underline{y} \in \mathbb{R}^3 : y_i \in \{1, 2\} \cdot 10^6\ dyn/cm^2\}, \tag{4.48}$$

that is, again, a set consisting of 8 possible configurations for $\underline{y}$[6]. In Table 4.6 we

[6]In fact, there are 3 components of $\underline{y}$, and each of them can attain 2 possible value. Hence, the training set contains 2^3 different configurations.

τ	0.9	0.95	0.99
N_u	5	8	22
N_η	5	7	12

Table 4.6: Dimension of the fluid velocity and membrane displacement RB for different values of the POD threshold for the idealized aortic arch test case with piecewise linear Young's modulus distribution.



(a) Full Space (b) Reduced Space

Figure 4.10: Estimates for the piecewise constant Young's modulus distribution: FS (left), and RS (right) with $\tau = 0.95$.

report the dimension of the RB for different choices of the threshold parameter τ in the POD method. We observe that the dimension of the basis is larger than in the case of the cylinder test case. This is due in part to the fact that the total number of degrees of freedom is larger (for instance, for the velocity, we have 13017 total d.o.f for the idealized aortic arch, versus 9186 for the cylinder test case), but also to the fact that the flow pattern is more complex in this case, due to the curved pipe.

In Figure 4.10 we show the history of the Young's modulus estimates for both the FS approach and the RS approach. In particular, the RB was obtained using $\tau = 0.95$ as threshold for the POD method. The graph on the left is the same as the one in Figure 3.6. In particular, there, as well as in the next figures, the blue line refers to the estimates of y_1, the red line to the estimates of y_2 and the green line to the estimates of y_3. As we can see, the estimates obtained with the RS approach are comparable to those obtained with the FS. This is also confirmed by the numerical value of the estimates averages, which we report in Table 4.7, together with additional information on the performance of the minimization algorithm.

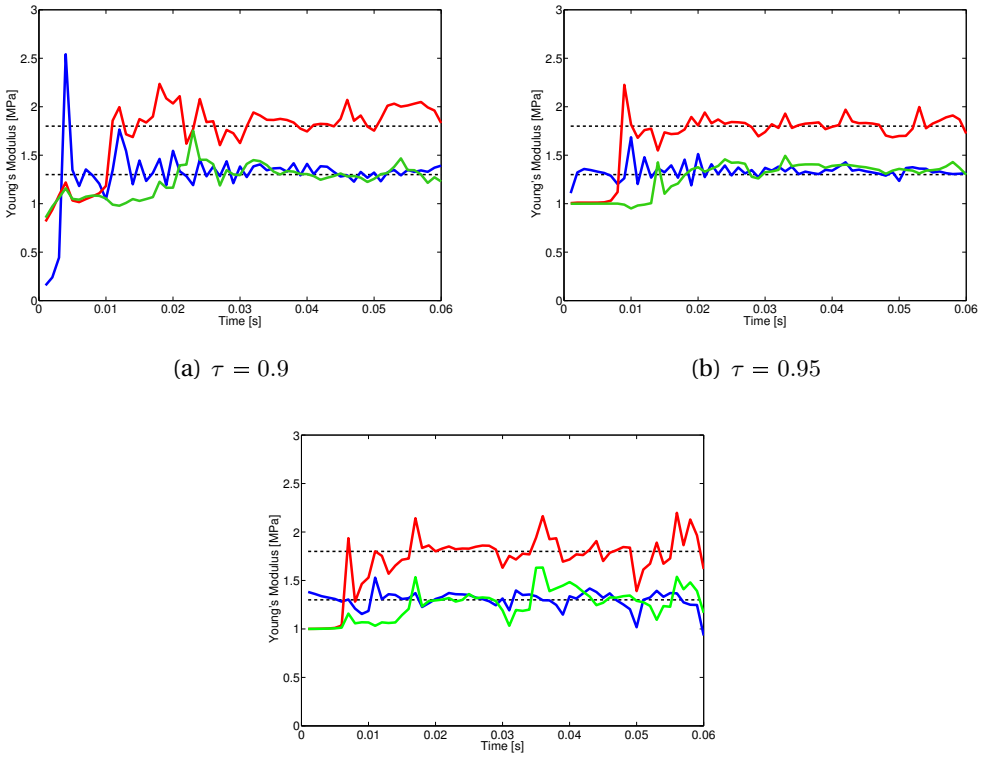
approach	FS	RS
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.33, 1.84, 1.31]·10^6	[1.34, 1.80, 1.33]·10^6
rel. error	1.91%	2.01%
exec. time (online)	3176s	277s
NS solves	492	480

Table 4.7: Summary of the minimization of the functional using BFGS for the idealized aortic arch test case, for both FS and RS approaches.

In this table, as we did for the cylinder test case, we report the average of the estimates (discarding the first ten estimates), the online execution time and the total number of Navier-Stokes solves. For the FS approach, this number accounts for both the forward and adjoint problems solves, while for the RS approach it accounts for the forward solves needed to construct the basis in the offline stage. We notice that the quality of the estimates is similar, and in both cases the error is remarkably smaller than the error in the measures (around 2% versus 10%).

Finally, also for this test case we study the behavior of the estimates with respect to the POD threshold τ. In Figure 4.11 we show the history of the estimates of the Young's modulus obtained for different choices of τ, with SNR=10. Differently from what we observed in the cylinder test cases, here even with $\tau = 0.9$ the RB is rich enough to give reasonable estimates for y_3. In fact, comparing Table 4.6 with Table 4.4, we notice that, even with $\tau = 0.9$, the RB has a larger size in this case.

In Table 4.8 we report the time average of the estimates obtained with $\tau = 0.9$, 0.95, 0.99. As we did for the cylinder test case, the averages are computed excluding the first 10 estimates, since we want to avoid an average which is clearly biased by the initial condition. As we can observe, the estimate improve as τ approaches one, which is what we expected. For the cylinder test case, the improvement was not significant when τ changed from 0.95 to 0.99. Here, instead, we notice that the quality of the estimate keep improving when τ increases. We justify this difference by noticing, as we did before, that the number of degrees of freedom is larger in the idealized aortic arch test case, and, furthermore, the dynamics of the flow is more complicated. Hence, an increase of τ from 0.95 to 0.99 can still add significan information, useful to correctly reconstruct the flow pattern.



(a) $\tau = 0.9$

(b) $\tau = 0.95$

(c) $\tau = 0.99$

Figure 4.11: Estimates of the Young's modulus for different choices of the threshold τ in the POD method, for the case of piecewise constant Young's modulus.

	$\tau = 0.9$	$\tau = 0.95$	$\tau = 0.99$
$\underline{y}^{ex}$	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6	[1.3, 1.8, 1.3]·10^6
$\underline{y}^*$	[1.34, 1.89, 1.01]*10^6	[1.33, 1.85, 1.36]*10^6	[1.35, 1.83, 1.33]*10^6
rel. error	11.87%	3.22%	2.57%

Table 4.8: Time average of the estimates and relative error for different values of the POD threshold for the idealized aortic arch test case.

Chapter 5

Deconvolution-based filtering schemes

In this chapter we focus on the issue of the reliability of the numerical solution of the Navier-Stokes equations when the Reynolds number is large enough to trigger disturbances or even turbulent effects. After a quick introduction of the topic, we will present the non-linear Leray model, used to describe the flow and the so called *indicator function*, which plays a big role in the model. Then we will present a reinterpretation of an existing discretization algorithm for the non-linear Leray model as an operator splitting method, which will allow us to obtain a practical way to calibrate the parameters of the model. Finally, we will present some numerical results, related to a benchmark recently proposed by the Food and Drug Administration (FDA) [40].

5.1 Motivation: numerical simulation of turbulent flows

Consider the unsteady Navier-Stokes (NS) equations for Newtonian fluids in a rigid domain Ω

$$\begin{cases} \rho\dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} - 2\mu\Delta^s\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega\times(t_0,T), \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega\times(t_0,T), \end{cases} \tag{5.1}$$

where $\mathbf{u}$ and p are the fluid velocity and pressure, ρ andμ are the fluid density and (dynamic) viscosity and $\mathbf{f}$ accounts for external forces (such as gravity). Here we defined $\Delta^s\mathbf{u} = \nabla \cdot \nabla^s\mathbf{u}$, where $\nabla^s\mathbf{u} = (\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}})/2$ is the deformation tensor. The system (5.1) is endowed with the initial and boundary conditions

$$\begin{cases} \mathbf{u} = \mathbf{u}_0 & \text{in } \Omega \times \{t_0\}, \\ \mathbf{u} = \mathbf{u}_{_D} & \text{on } \Gamma_{_D} \times (t_0, T), \\ 2\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}})\mathbf{n} - p\mathbf{n} = \mathbf{g} & \text{on } \Gamma_{_N} \times (t_0, T), \end{cases} \tag{5.2}$$

where $\overline{\Gamma_{_D}} \cup \overline{\Gamma_{_N}} = \overline{\partial\Omega}$ and $\overset{\circ}{\Gamma}_{_D} \cap \overset{\circ}{\Gamma}_{_N} = \varnothing$.

Remark 5.1. In the continuous formulation, $\nabla \cdot \nabla\mathbf{u}^{\mathrm{T}} = \nabla(\nabla \cdot \mathbf{u}) = 0$, due to the continuity equation. Therefore, the Navier-Stokes equations are often formulated with the operator Δ rather than Δ^s. However, the contribution of the term $\nabla\mathbf{u}^{\mathrm{T}}$ does not vanish when problem (5.1) is formulated in its *weak* form - as done in the finite element approximation [37]. For some applications, such as fluid-structure interaction problems, the contribution of $\nabla\mathbf{u}^T$ is essential to get the correct behavior of the flow.

As it is well known, the NS equations feature several challenging aspects, such as the non-linearity or the saddle-point nature of the problem, that makes the analysis quite complicated [94][1]. These aspects reflects in a discrete counterpart when numerical methods are used to compute approximate sulutions. For instance, the saddle-point nature of the problem impose a condition that the spaces where the approximate velocity and pressure are sought must satisfy, the so-called as *inf-sup* or LBB condition [12]. When resorting to numerical methods, other difficulties also arise, peculiar to the discrete world, such as instabilities in convection dominated flow. These can be avoided by restrictions on the discretization parameters or by adopting suitable stabilization techniques (e.g., [13]).

An additional issue related to the discretization arises only in particular flow

[1]The uniqueness of the solution of (5.1) has been proven in 2D, but it is still an open problem in 3D.

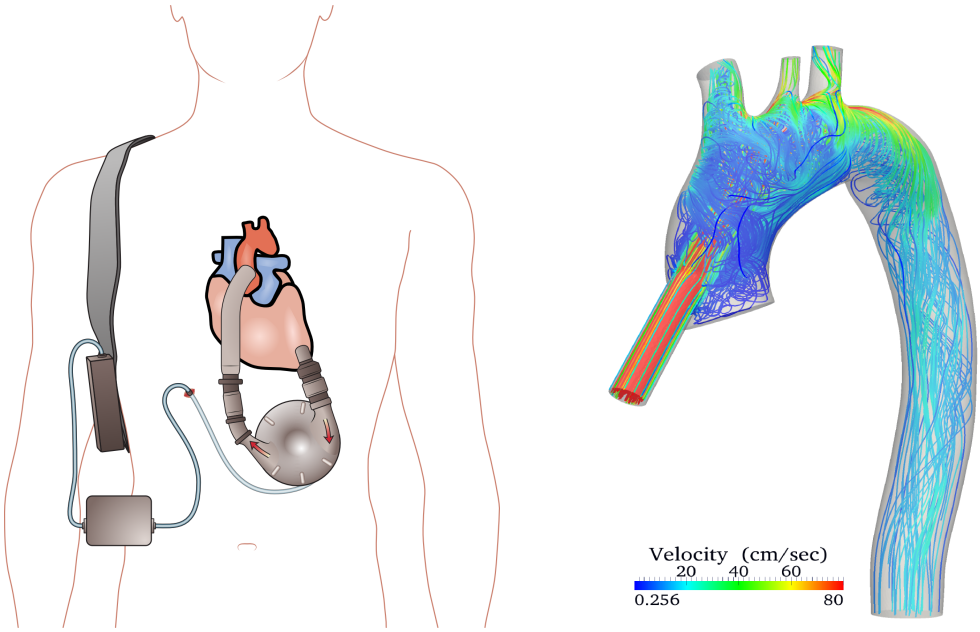
conditions. In particular, let

$$Re = \frac{\rho U L}{\mu} \tag{5.3}$$

be the *Reynolds number*, where U and L are a macroscopic velocity and length associated with the flow. For instance, for a steady flow in a pipe, U would be the average velocity on the centerline and L the pipe diameter. The Reynolds number can be interpreted as a comparison between inertial and viscous forces: for small Reynolds numbers ($Re = O(1)$), inertial forces dominate and we observe a *laminar* flow; for large Reynolds numbers ($Re = O(10^3)$ and above) the inertial forces dominate and the flow is much more irregular and features the presence of structures of a large variety of space scales. This regime is usually referred to as *turbulent*.

Remark 5.2. In cardiovascular mathematics, the Reynolds number associated with the flow is usually small, so that the flow can be considered laminar. However, there are a few scenarios where it could be large enough to trigger turbulent effects. In physiological conditions, the only region where this can happen is the aortic arch, where the blood, coming from the left ventricle, can reach velocities that give a Reynolds number between 1000 and 2000, depending on the person and the conditions. However, more recently, with the development of new medical implants, it is possible to reach even larger values. For instance, consider a patient with a Left Ventricular Assist Device (LVAD). The LVAD is an artificial external pump that is used in case of pathological defects in the aortic valve (see Figure 5.1(a)). This pump replaces the left ventricle and injects the blood directly into the aorta by means of a graft (called cannula) usually sutured in the ascending aortic arch. Since the diameter of the cannula is smaller than that of the aorta (around $0.5cm$, compared to the $2.5cm$ of the aorta), in order to deliver the same flow rate (of about 5 liter per minute, for a healthy adult patient), the velocity of the blood in the cannula must be larger than that usually present in aorta (see Figure 5.1(b)), leading to Reynolds number of up to 3000-3500.

From the numerical point of view, a flow at a large Reynolds number is more complicated to simulate, since the numerical discretization should be fine enough to capture all the scales involved in the dynamics. Unfortunately, as the Reynolds



(a) LVAD concept (b) velocity streamlines

Figure 5.1: A depiction of the concept of the LVAD (a) and an example of a CFD study performed on a possible graft configuration (b) (courtesy of Dr. Divya Gupta, Emory University Hospital).

number increases, this requirement quickly becomes very restrictive, and Direct Numerical Simulation (DNS) may be computationally unaffordable.

A possible way to keep the computational costs under control while maintaining sufficient accuracy is to resort to different models, which properly transfer the effects of the presence of the small scales (which are not solved) to larger scales. For instance, the NS equations can be averaged in time, leading to the so called Reynolds-Averaged Navier-Stokes equations (RANS), or in space, leading to Large Eddy Simulation (LES) techniques (see, e.g., [25]).

Here, we consider a variant of the so called *Leray model* [56], where small scale effects are described by an additional set of equations to be added to the discrete NS equations, which can be interpreted as a differential filter. In the next section we present the continuous model, originally proposed in [9], and its actual implementation, which was first presented in [53].

5.2 Non-linear Leray models

To motivate the introduction of the Leray model, it is useful to look at the behavior of the turbulent kinetic energy (TKE) of the fluid, which is the kinetic energy associated with eddies in the turbulent flow. In the framework of the Kolmogorov 1941 (K41) theory [51, 52], the TKE is injected in the system at the large scales (low wave numbers). Since the large scale eddies are unstable, they break down, transferring the energy to smaller eddies. Finally, the TKE is dissipated by the viscous forces at the small scales (high wave numbers). This process is usually referred to as *energy cascade*. The scale at which the viscous forces dissipate energy is referred to as *Kolmogorov scale* and can be expressed as

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4}, \tag{5.4}$$

where $\nu = \mu/\rho$ is the *kinematic viscosity*, and ε is the time-average of the rate at which the energy is dissipated (see e.g. [30]). Formally, ε is defined as

$$\varepsilon := \limsup_{T\to\infty} \frac{1}{T|\Omega|} \int_{t_0}^{T} \nu ||\nabla \mathbf{u}||_{L^2}^2 dt. \tag{5.5}$$

The dissipation rate has to be of the same magnitude of the production rate, which is the rate at which the TKE is supplied to the small scales. A common way to express ε in terms of the macro-scale variables is $\varepsilon \sim U^3/L$ [95], leading to the expression

$$\eta = Re^{-3/4} L. \tag{5.6}$$

This scaling law pinpoints the difficulty of the numerical solution of the NS equation at high Reynolds numbers: a thrustworthy numerical simulation must correctly dissipate energy, and therefore DNS should use grids with spacing $h \sim \eta$. Clearly, as the Reynolds number increases, this requirement causes the number of unknowns to dramatically increase, and the computational costs become prohibitive. On the other hand, when the mesh size h fails to resolve the Kolmogorov scale, nonphysical wiggles are observed in the computed velocity due to an under-

diffusion in the simulation.

A common way to overcome this problem without increasing the number of unknowns is to introduce a model which uses filter and/or averaging techniques to extract information on the small scales dynamics using only information at the large scales. Such models can be thought as a way to revert the energy cascade by transferring the energy dissipated at the small (unresolved) scales towards larger (resolved) scales. An example is the so-called "generalized Leray model" that couples the Navier-Stokes equations with a differential filter.

5.2.1 The continuous problem

The non-linear Leray model can be written as

$$\begin{cases} \rho\dfrac{\partial \mathbf{u}}{\partial t} + \rho(\overline{\mathbf{u}}\cdot\nabla)\mathbf{u} - 2\mu\Delta^s\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega\times(t_0,T), \quad (5.7)\\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega\times(t_0,T), \quad (5.8)\\ -2\delta^2\nabla\cdot(a(\mathbf{u})\nabla^s\overline{\mathbf{u}}) + \overline{\mathbf{u}}\nabla\lambda = \mathbf{u} & \text{in } \Omega\times(t_0,T), \quad (5.9)\\ \nabla\cdot\overline{\mathbf{u}} = 0 & \text{in } \Omega\times(t_0,T), \quad (5.10) \end{cases}$$

The first equation is a variation of the NS momentum equation, where the convective field has been replaced with $\overline{\mathbf{u}}$. This new variable is in turn related to the fluid velocity by the third equation, which can be interpreted as a *differential filter*, with δ as the *filtering radius*[2]. The action of the filter is to extract information from the unresolved scales, using the information on the resolved scales in a neighborhood of radius δ. We therefore refer to $\overline{\mathbf{u}}$ as the the *filtered velocity*, or *regularized velocity*. The variable λ is a Lagrange multiplier used to enforce the incompressibil-

[2] In literature, this is usually denoted with α. Here we used a different notation, because, later on, as we did in Chapter 3, we will use α to denote the leading coefficient in the BDF approximation of the time derivative of $\mathbf{u}$.

ity constraint for $\overline{\mathbf{u}}$. Finally, $a(\cdot)$ is a scalar function such that

$$
\begin{aligned}
&a(\mathbf{u}) \geqslant 0 \text{ everywhere} \\
&a(\mathbf{u}) \simeq 0 \text{ where the velocity } \mathbf{u} \text{ does not need regularization} \\
&a(\mathbf{u}) \simeq 1 \text{ where the velocity } \mathbf{u} \text{ does need regularization.}
\end{aligned}
$$

This function is usually called *indicator function*, and clearly it is of the utmost importance for the success of the Leray model. We will discuss the possible choices that have been proposed in literature for this function in Section 5.3. For now, we only point out that the choice $a(\mathbf{u}) \equiv 1$ corresponds to the classic Leray-α model, proposed in [18]. The Leray-α model has a rather limited effectiveness, because it introduces a uniform regularization in the whole domain. Nevertheless, the fact that $a(\mathbf{u}) \equiv 1$ makes the filter equations linear and constant in time makes the Leray-α model useful for preliminary investigations of possible numerical methods used to discretize (5.7-5.10).

The Leray system consists of two coupled saddle point problems: the NS equations (5.7-5.8), where the advection field is equal to the filtered velocity $\overline{\mathbf{u}}$, and the filter problem (5.9-5.10). Notice that the latter has a structure that resembles that of a generalized Stokes problem (apart from the non-constant viscosity). We will return on this fact in Section 5.5, where we will introduce the preconditioners used in the solution of the discretized problem.

The filter equations also require suitable boundary conditions. These are chosen to be

$$
\begin{cases}
\overline{\mathbf{u}} = \mathbf{u}_D & \text{on } \partial\Gamma_D \times (t_0, T) \\
(2\delta^2 a(\mathbf{u}) \nabla^s \overline{\mathbf{u}} - \lambda \underline{\underline{\mathrm{I}}}) \mathbf{n} = \mathbf{0} & \text{on } \partial\Gamma_N \times (t_0, T)
\end{cases}
\tag{5.11}
$$

while no initial condition is required for $\overline{\mathbf{u}}$, since its time derivative does not appear in (5.7-5.10). The impact of non-Dirichlet boundary conditions so far has not been investigated thoroughly in literature. In Section 5.4, we will discuss the effect of these boundary conditions on the solution of the problem.

5.2.2 The time-discrete problem

To discretize in time problem (5.7)-(5.10), let $\Delta t \in \mathbb{R}$, $t^n = t_0 + n\Delta t$, with $n = 0, ..., N_T$ and $T = t_0 + N_T \Delta t$. Moreover, we denote by y^n the approximation of a generic quantity y at the time t^n. In the following, we will assume that all the equations hold in Ω, unless otherwise specified.

For the discretization of the time derivative we adopt a Backward Differentiation Formula of order p (BDFp) [75]. The system (5.7-5.10) discretized in time then reads: given $\mathbf{u}^0$, for $n \geqslant 0$ find $(\mathbf{u}^{n+1}, p^{n+1}, \overline{\mathbf{u}}^{n+1}, \lambda^{n+1})$ such that

$$\begin{cases} \rho \dfrac{\alpha}{\Delta t}\mathbf{u}^{n+1} + \rho(\overline{\mathbf{u}}^{n+1} \cdot \nabla)\mathbf{u}^{n+1} - 2\mu\Delta^s \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{b}^{n+1}, & (5.12) \\ \nabla \cdot \mathbf{u}^{n+1} = 0, & (5.13) \\ -2\delta^2 \nabla \cdot \left(a(\mathbf{u}^{n+1})\nabla^s \overline{\mathbf{u}}^{n+1}\right) + \overline{\mathbf{u}}^{n+1} + \nabla \lambda^{n+1} = \mathbf{u}^{n+1}, & (5.14) \\ \nabla \cdot \overline{\mathbf{u}}^{n+1} = 0, & (5.15) \end{cases}$$

where α is a coefficient that depends on the order of BDF chosen and $\mathbf{b}^{n+1}$ contains the forcing term $\mathbf{f}^{n+1}$, the possible contributions of the non homogeneous boundary conditions and the solution at the previous time steps used to approximate the time derivative of $\mathbf{u}$ at time t^{n+1}. For example, when using BDF2, we have

$$\partial_t \mathbf{u} \simeq \frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t}, \tag{5.16}$$

thus $\alpha = 3/2$ and $\mathbf{b}^{n+1} = \mathbf{f}^{n+1} - (-4\mathbf{u}^n + \mathbf{u}^{n-1})/(2\Delta t) + bc^{n+1}$.

To tackle problem (5.12-5.15) in a monolithic fashion, one has to introduce subiterations to work around the non-linear terms. For instance Newton or, more generally, fixed point iterations can be used. However, this would remarkably increase the computational costs of each time step, especially if the expression of $a(\cdot)$ is rather complex. This would make the advantage of the Leray model compared to a DNS questionable. To save computational time, we can decouple the two saddle point problems in one of the following two ways:

1. *Filter-Evolve*: we first solve the filter equations with $\mathbf{u}^{n+1}$ replaced by a suitable

extrapolation $\mathbf{w}^*$ both in the right hand side and in the indicator function, and then solve the NS equations.

2. *Evolve-Filter* : we first solve the NS equations with the advection field $\overline{\mathbf{u}}^{n+1}$ replaced by a suitable extrapolation $\mathbf{w}^*$, and then solve the filter problem.

Although these two approaches look different, if we use BDF to compute the extrapolation $\mathbf{w}^*$ we find out that they are rather similar, and that the only substantial difference between the two lies in the convective term used in the solution of the NS equations. In particular, we would obtain

1. *Filter-Evolve*: $(\overline{\mathbf{u}}^{n+1}\cdot\nabla)\mathbf{u}^{n+1} \simeq (\overline{(2\mathbf{u}^n-\mathbf{u}^{n-1})}\cdot\nabla)\mathbf{u}^{n+1}$.

2. *Evolve-Filter*: $(\overline{\mathbf{u}}^{n+1}\cdot\nabla)\mathbf{u}^{n+1} \simeq ((2\overline{\mathbf{u}}^n-\overline{\mathbf{u}}^{n-1})\cdot\nabla)\mathbf{u}^{n+1}$.

Notice that the two are not equivalent in the case of non-linear filters. The first approach has been investigated thoroughly in literature, using the linearly extrapolated Crank-Nicolson approximation $\mathbf{u}^{n+1} \simeq (3\mathbf{u}^n-\mathbf{u}^{n-1})/2$ [8]. The second approach, as we stated, has been less investigated. As a matter of fact, the dynamics of $\overline{\mathbf{u}}$ is induced by that of $\mathbf{u}$, so an extrapolation of the filtered velocity appears less meaningful. In order to avoid extrapolations of the filtered velocity, one could take $\mathbf{w}^*$ to be an extrapolation of $\mathbf{u}^{n+1}$ also the Evolve-Filter approach. However, this would cause $\mathbf{u}^{n+1}$ to not depend anymore on the filtered velocity, ultimately making the scheme a discretization for the NS equations rather than the Leray system.

To overcome this impasse, in [53] the authors proposed a three steps algorithm, which they called Evolve-Filter-Relax (EFR), in which a relaxation step is introduced after the solution of the filter problem. The EFR algorithm, as proposed in [53], reads as follows: given the velocities $\mathbf{u}^k$ ($k=n-p+1,\dots,n$) needed for the approximation of $\partial_t\mathbf{u}$ by BDFp at t^{n+1}, and given an extrapolation $\mathbf{v}^*$ of $\mathbf{u}^{n+1}$ (for instance, using the BDFp),

i) evolve: find intermediate velocity and pressure $(\mathbf{v}^{n+1}, q^{n+1})$ such that

$$\begin{cases} \rho\dfrac{\alpha}{\Delta t}\mathbf{v}^{n+1}+\rho(\mathbf{v}^*\cdot\nabla)\mathbf{v}^{n+1}-2\mu\Delta^s\mathbf{v}^{n+1}+\nabla q^{n+1}=\mathbf{b}^{n+1}, & (5.17)\\ \nabla\cdot\mathbf{v}^{n+1}=0, & (5.18)\end{cases}$$

ii) filter: find $(\overline{\mathbf{v}}^{n+1}, \lambda^{n+1})$ such that

$$\begin{cases} -2\delta^2 \nabla \cdot \left(a(\mathbf{v}^{n+1}) \nabla^s \overline{\mathbf{v}}^{n+1}\right) + \overline{\mathbf{v}}^{n+1} + \nabla \lambda^{n+1} = \mathbf{v}^{n+1} & (5.19) \\ \nabla \cdot \overline{\mathbf{v}}^{n+1} = 0 & (5.20) \end{cases}$$

iii) relax: set

$$\begin{cases} \mathbf{u}^{n+1} = (1-\chi)\mathbf{v}^{n+1} + \chi \overline{\mathbf{v}}^{n+1}, & (5.21) \\ p^{n+1} = q^{n+1}, & (5.22) \end{cases}$$

where $\chi \in [0,1]$ is a relaxation parameter.

The choice of using an extrapolation of $\mathbf{u}^{n+1}$ rather than $\overline{\mathbf{u}}^{n+1}$ in the convective term of the NS equations comes at the price of the additional parameter χ, which has to be properly tuned. It was shown in [66] that the EFR algorithm is equivalent to a certain viscosity model in Large Eddy Simulation (LES). In [53] the authors use energy arguments to support the choice $\chi = O(\Delta t)$, in order to keep the numerical dissipation introduced by the filter step under control, but they do not specify a practical way to tune the parameter.

In Section 5.4, we will rewrite step (ii) so that we will be able to interpret the EFR algorithm as an *operator splitting* method. On one hand, this reinterpretationi will allow us to use the Lagrange multiplier for the incompressibility constraint in the filter equations to relax also the pressure found at step (i); on the other hand, we will be able to give an argument on how to tune the relaxation parameter χ based on the physical and numerical parameters of the problem. Moreover, we will discuss how step (ii) affects the boundary conditions that are satisfied by $\mathbf{u}^{n+1}$ and p^{n+1} (in particular in the case of Neumann boundary conditions).

5.3 Indicator functions

Although the breaking down of eddies into smaller ones until they get damped is highly nonlinear, most current models for eddy viscosity use linear filters to select

the eddies to be damped. Linear filter based stabilization, developed by Boyd [10] and Fischer and Mullen [27, 62], has been widely studied over the past years (see, e.g., [32, 59, 99]). Only in a recent publication by Layton and coauthors [53] nonlinear filters have been introduced: eddies are selected for damping based on knowledge of how nonlinearity acts in real flow problems.

The success of nonlinear filters such as (5.7)-(5.10) in simulations ultimately depends on the quality of the indicator function. One of the most mathematically convenient indicator function is $a(\mathbf{u}) = \|\nabla^s \mathbf{u}\|_F$ (suitably normalized [7]), where $\|\cdot\|_F$ denotes the Frobenius norm, because of it is strongly monotone, easy to implement and stable, and allows us to recover variants of the so called *Smagorinsky model*. This choice, however, is known to be too dissipative (the corresponding artificial viscosity is not bounded for large gradients) and not sufficiently selective. For instance, it selects laminar shear flow, where $|\nabla \mathbf{u}|$ is constant but large, as regions of the domain with large turbulent fluctuations.

In the following, we report on some indicator functions that have been proposed in the literature. We group them into two categories: physical phenomenology based and deconvolution based. In Sections 5.3.1 and 5.3.2 (respectively), we will highlight their strengths and limitations.

5.3.1 Physical phenomenology based indicator functions

The indicator functions proposed in [53] are defined on physical quantities that are known to vanish for coherent flow structures, that is structures that do not break into smalle ones, following the energy cascade process. One of the most popular methods for deduction of coherent vortices is the Q criterion [45], which identifies as persistent and coherent structures those regions where

$$Q(\mathbf{u}, \mathbf{u}) = \frac{1}{2}(\nabla^{ss}\mathbf{u} : \nabla^{ss}\mathbf{u} - \nabla^s \mathbf{u} : \nabla^s \mathbf{u}) > 0, \tag{5.23}$$

where $\nabla^{ss}\mathbf{u} = (\nabla \mathbf{u} - \nabla \mathbf{u}^{\mathrm{T}})/2$ is the spin tensor. So, $Q > 0$ occurs in those regions where spin dominates deformation. Since dissipation occurs where deformation dominates spin, we can build an indicator by rescaling $Q(\mathbf{u}, \mathbf{u})$ so that the condition

$Q(\mathbf{u},\mathbf{u}) > 0$ implies $a(\mathbf{u}) \simeq 0$, that is regularization is not needed. The Q-criterion based indicator function is given by:

$$a_Q(\mathbf{u}) = \frac{1}{2} - \frac{1}{\pi}\arctan\left(\delta^{-1}\frac{Q(\mathbf{u},\mathbf{u})}{|Q(\mathbf{u},\mathbf{u})| + \delta^2}\right). \tag{5.24}$$

A second indicator uses an eddy viscosity coefficient formula proposed by Vreman [102] that vanishes for 320 types of flow structures known to be coherent. The Vreman based indicator function is given by:

$$a_V(\mathbf{u}) = \sqrt{\frac{B(\mathbf{u})}{|\nabla\mathbf{u}|_F^4}}, \tag{5.25}$$

where $B(\mathbf{u})$ is defined as

$$B = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2, \quad \beta_{ij}(\mathbf{u}) = \sum_{m=1,2,3}\frac{\partial u_i}{\partial x_m}\frac{\partial u_j}{\partial x_m}. \tag{5.26}$$

Since $0 \leqslant B(\mathbf{u})/|\nabla\mathbf{u}|_F^4 \leqslant 1$, $a_V(\mathbf{u}) \in [0,1]$. The Vreman based indicator function was shown to be successful in [8].

Another physics-based indicator function uses the relative helicity density RH, which is a local quantity, its macroscopic counterpart being the helicity H. The two quantities H and RH are defined as

$$H = \frac{1}{|\Omega|}\int_\Omega \mathbf{u}\cdot\boldsymbol{\omega}\, d\Omega, \quad RH = \frac{\mathbf{u}\cdot\boldsymbol{\omega}}{|\mathbf{u}||\boldsymbol{\omega}|} \tag{5.27}$$

with $\boldsymbol{\omega}$ denoting vorticity, i.e. $\boldsymbol{\omega} = \nabla\times\mathbf{u}$. From the Navier-Stokes equations in rotational form, it is possible to see that local high helicity suppresses local turbulent dissipation caused by breakdown of eddies into smaller ones. The helicity based indicator is developed by adjusting relative helicity density so that values of RH near one imply $a(\mathbf{u}) \simeq 0$:

$$a_H(\mathbf{u}) = 1 - \left|\frac{\mathbf{u}\cdot\boldsymbol{\omega}}{|\mathbf{u}||\boldsymbol{\omega}| + \delta^2}\right|. \tag{5.28}$$

Notice that other (more selective) indicator functions can also be obtained by taking the geometric average of two (or more) indicator functions.

All the indicator functions discussed in this section have the advantage of requiring only algebraic operations on $\mathbf{u}$ and its derivatives. Depending on how the Navier-Stokes solver is coded and vectors/matrices are handled, the implementation of these indicators could be trivial. However, the major drawback of the physics based indicator functions is that they do not allow for a rigorous convergence theory to verify the robustness of the associated filtering method. This limitation is overcome by the mathematics-based (rather than physics-based) indicators discussed in the next section.

5.3.2 Deconvolution based indicator functions

To motivate the deconvolution-based indicator functions, let F be a linear, invertible, self-adjoint, compact operator from a Hilbert space V (such as $H^1(\Omega)$ or $H_0^1(\Omega)$) to itself. The spectral theorem gives (see, for instance, [85])

$$F\mathbf{x} = \sum_{i=0}^{\infty} \lambda_i \langle \mathbf{x}, \mathbf{e}_i \rangle \mathbf{e}_i, \tag{5.29}$$

$$F^{-1}\mathbf{y} = \sum_{i=0}^{\infty} \frac{1}{\lambda_i} \langle \mathbf{y}, \mathbf{e}_i \rangle \mathbf{e}_i, \tag{5.30}$$

where $\mathbf{e}_i$ are the eigenfunctions of F, which form an orthonormal basis for V.

Since F is compact, we know that F^{-1} is unbounded. Nevertheless, since the inverse is defined, we have that $\mathbf{x} = F^{-1}F\mathbf{x}$. Let D be a bounded regularized approximation of F^{-1}, whose action on $\mathbf{y}$ is given by

$$D\mathbf{y} = \sum_{i=0}^{\infty} \phi\left(\frac{1}{\lambda_i}\right) \langle \mathbf{y}, \mathbf{e}_i \rangle \mathbf{e}_i \tag{5.31}$$

where the function ϕ is such that

$$\phi\left(\frac{1}{\lambda_i}\right) \simeq \begin{cases} \dfrac{1}{\lambda_i} & \text{if } i \text{ is “small”}, \\ 0 & \text{if } i \text{ is “large”}. \end{cases} \tag{5.32}$$

Then we have that

$$\|\mathbf{x} - DF\mathbf{x}\| \text{ is } \begin{cases} \text{small} & \text{if } \mathbf{x} \text{ is “smooth”}, \\ \text{large} & \text{if } \mathbf{x} \text{ is not “smooth”}, \end{cases} \tag{5.33}$$

where “smooth” is intended with respect to the eigenfunction of the operator F. In particular, $\mathbf{x}$ is smooth if $\langle \mathbf{x}, \mathbf{e}_i \rangle$ is significantly different from zero only for small values of i. The composition of the two operators F and D can be interpreted as a low-pass filter. This motivates the indicator function

$$a_{_D}(\mathbf{u}) = |\mathbf{u} - DF\mathbf{u}|\,. \tag{5.34}$$

Remark 5.3. In order to ensure that $a(\mathbf{u}) \in [0,1]$, indicator function (5.34) may be rescaled

$$a_{_D}(\mathbf{u}) = \frac{|\mathbf{u} - DF\mathbf{u}|}{\max(1, ||\mathbf{u} - DF\mathbf{u}||_\infty)}. \tag{5.35}$$

A popular choice for D is the Van Cittert deconvolution operator D_N, defined as

$$D_N = \sum_{n=0}^{N} (I - F)^n. \tag{5.36}$$

The evaluation of the indicator function corresponding to the Van Cittert deconvolution of order N requires then to apply the filter F a total of $N+1$ times. If $I - F$ is spectrally bounded by 1, then D_N can be seen as the truncated Neumann expansion of F^{-1}, and D_N approaches F^{-1} as $N \to \infty$. We will see later on that the bound on $I - F$ is actually true if the mesh is fine enough, even if this does not introduce a limitation for this method to be applied.

In practice, N is chosen to be small, as the result of a trade-off between accuracy (for a laminar solution) and filtering (for a turbulent solution). For instance, for $N = 0, 1$, the indicator function (5.34) becomes

$$a_{_{D_0}}(\mathbf{u}) = |\mathbf{u} - F(\mathbf{u})|\,, \quad a_{_{D_1}}(\mathbf{u}) = |\mathbf{u} - 2F(\mathbf{u}) + F(F(\mathbf{u}))|\,. \tag{5.37}$$

Remark 5.4. The Van Cittert deconvolution D_N can also be interpreted also as the

$N\text{-}th$ iteration of a Richardson scheme to solve the problem $F(\mathbf{u}) = \mathbf{b}$. In fact, letting $\mathbf{u}_N = D_N(\mathbf{b})$, we have

$$\mathbf{u}_{N+1} = \sum_{n=0}^{N+1} (I-F)^n \mathbf{b} = \mathbf{b} + \sum_{n=1}^{N+1} (I-F)^n \mathbf{b} =$$
$$= \mathbf{b} + (I-F) \sum_{n=0}^{N} (I-F)^n \mathbf{b} = \mathbf{b} + (I-F)\mathbf{u}_N = \mathbf{u}_N + (\mathbf{b} - F\mathbf{u}_N).$$

From this perspective, we can further support the choice of limiting the order of the Van Cittert deconvolution. In fact, being the operator F compact, we know that its eigenvalues accumulate to 0, and that its inverse leads to an ill-posed problem. Among the possible regularization techniques used to deal with ill-posed problem, one is precisely to use iterative methods (such as Richardson method) limiting the number of iterations. We can therefore see the Van Cittert deconvolution as an iterative regularization of the inverse problem $F\mathbf{u} = \mathbf{b}$.

In this work, we use as F the linear Helmholtz filter operator F_H [34], defined by

$$F = F_H \equiv \left(I + \delta^2 L\right)^{-1}, \qquad L = -\Delta \equiv -\sum_{i=1}^{3} \frac{\partial^2}{\partial x_j^2}, \tag{5.38}$$

From [22], we know that

$$\mathbf{v} - D_N F_H \mathbf{v} = \delta^{2N+2} L^{N+1} F_H^{N+1} \mathbf{v} = (-1)^{N+1} \delta^{2N+2} \Delta^{N+1} F_H^{N+1} \mathbf{v}. \tag{5.39}$$

Therefore, $a_{D_N}(\mathbf{u})$ is close to zero in the regions of the domain where $\mathbf{u}$ is smooth. Indicator function (5.34) with $D = D_N$ and $F = F_H$ has been recently proposed in [8], however the idea of using van Cittert approximate deconvolution in fluid models to increase accuracy is well established and mathematically grounded [22, 89, 90].

We conclude this section by pointing out that the Van Citttert-Helmholtz deconvolution operators D_0 and D_1 can be conveniently interpreted in a different manner, in view of the theory of maximal monotone operators and their Yosida regularized operator stated in [11], Chapt. 7. Following the theory and notation there in, if we let $L = -\Delta$, then F_H is the *resolvent* J_{δ^2} of L. Correspondingly, we introduice

the *Yosida approximation* (or *regularization*) of L, as

$$L_{\delta^2} \equiv \delta^{-2}\left(I - J_{\delta^2}\right) = \delta^{-2}\left(I - F_{_H}\right) \tag{5.40}$$

From here we obtain

$$a_{D_0}(\mathbf{u}) = \delta^2 \left|L_{\delta^2}\mathbf{u}\right|, \quad a_{D_1}(\mathbf{u}) = \left|(I - F_{_H})^2\mathbf{u}\right| = \delta^4 \left|L_{\delta^2}^2\mathbf{u}\right|. \tag{5.41}$$

Here, we list some properties we infer from Proposition 7.2 in [11]. We assume that the argument v of each operator is selected properly

i) $L_{\delta^2}v = LJ_{\delta^2}v$

ii) $L_{\delta^2}v = J_{\delta^2}Lv$

iii) $|L_{\delta^2}v| \leqslant |Lv|, \quad \delta^2|L_{\delta^2}v| \leqslant |v|$

iv) $(L_{\delta^2}v, v) \geqslant 0$

v) $\lim\limits_{\delta^2\to 0} J_{\delta^2}v = v$

vi) $\lim\limits_{\delta^2\to 0} L_{\delta^2}v = Lv \quad (\Rightarrow \lim\limits_{\delta^2\to 0} \delta^2 L_{\delta^2}v = 0)$.

From the first of these properties we can deduce straightforwardly (5.39).

5.4 EFR as an operator-splitting algorithm

In this section we will rewrite the EFR scheme introduced in Section 5.2.2 as an operator-splitting method, with indicator function a given by (5.35), $D = D_{_N}$ and $F = F_{_H}$. We will derive an empirical formula, based on physical arguments, to tune the relaxation parameter χ. Moreover, we will investigate how the filter equations affect the boundary conditions that are satisfied by the end-of-step velocity and pressure (with particular attention to the case where non-Dirichlet boundary conditions are prescribed).

For the space discretization we will use the Finite Element Method, and in particular we will use *inf-sup* stable Finite Element spaces for velocity and pressure (for

instance, Taylor-Hood elements [80]). The details concerning space discretization will be covered in section 5.5. However, for the content of this section, we need to define the discretization parameter h, which we take to be the length of the shortest edge in the mesh. Coherently, we will denote by $\mathbf{f}_h$ the discrete FE approximation of a generic continuous function $\mathbf{f}$.

With the goal of expressing the EFR algorithm as an operator-splitting method, let $\mathbf{u}_h^*$ be an extrapolation of $\mathbf{u}_h^{n+1}$, and let $(\mathbf{v}_h^{n+1}, q_h^{n+1})$ be the solution of the *evolve* step

$$\begin{cases} \rho\dfrac{\alpha}{\Delta t}\mathbf{v}_h^{n+1} + \rho(\mathbf{u}_h^* \cdot \nabla)\mathbf{v}_h^{n+1} - \nabla \cdot (2\mu\nabla^s \mathbf{v}_h^{n+1}) + \nabla q_h^{n+1} = \mathbf{b}_h^{n+1}, & (5.42) \\ \nabla \cdot \mathbf{v}_h^{n+1} = 0, & (5.43) \\ \mathbf{v}_h^{n+1} = \mathbf{u}_D, \quad \text{on } \Gamma_D & (5.44) \\ (2\mu\nabla^s \mathbf{v}_h^{n+1} - q_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} = \mathbf{g}, \quad \text{on } \Gamma_N & (5.45) \end{cases}$$

Next, let us then denote by $(\overline{\mathbf{v}}_h^{n+1}, \overline{q}_h^{n+1})$ the solution to the filter step, which we rewrite as

$$\begin{cases} \rho\dfrac{\overline{\mathbf{v}}_h^{n+1}}{\Delta t} - \nabla \cdot \left(2\overline{\mu}_h \nabla^s \overline{\mathbf{v}}_h^{n+1}\right) + \nabla \overline{q}_h^{n+1} = \rho\dfrac{\mathbf{v}_h^{n+1}}{\Delta t}, & (5.46) \\ \nabla \cdot \overline{\mathbf{v}}_h^{n+1} = 0, & (5.47) \\ \overline{\mathbf{v}}_h^{n+1} = \mathbf{u}_D, \quad \text{on } \Gamma_D & (5.48) \\ (2\overline{\mu}_h \nabla^s \overline{\mathbf{v}}_h^{n+1} - \overline{q}_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} = \mathbf{g}, \quad \text{on } \Gamma_N & (5.49) \end{cases}$$

where we defined the filter viscosity

$$\overline{\mu}_h = \rho\frac{\delta^2}{\Delta t} a(\mathbf{v}_h^{n+1}). \tag{5.50}$$

Here, the parameter δ is chosen to be of the order of h. Notice that with this formulation, $\overline{\mu}_h$ is dimensionally a dynamic viscosity and the Lagrange multiplier $\overline{q}_h^{n+1}$ is dimensionally a pressure.

Finally, the relaxation step for the velocity reads

$$\mathbf{u}_h^{n+1} = (1-\chi)\mathbf{v}_h^{n+1} + \chi\overline{\mathbf{v}}_h^{n+1}. \tag{5.51}$$

To further analyze the method, let us define the operators

$$\mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{v}_h = \rho(\mathbf{u}_h^* \cdot \nabla)\mathbf{v}_h - \nabla \cdot (2\mu\nabla^s \mathbf{v}_h), \tag{5.52}$$

$$\mathscr{L}_F[\mathbf{v}_h]\overline{\mathbf{v}}_h = -\nabla \cdot (2\overline{\mu}_h \nabla^s \overline{\mathbf{v}}_h). \tag{5.53}$$

Here, the notation $A[\mathbf{v}]\mathbf{u}$ means that the operator A is computed at $\mathbf{v}$ and then applied to the function $\mathbf{u}$. Notice that the operator $\mathscr{L}_F$ depends on $\mathbf{v}_h$ through the artificial viscosity $\overline{\mu}_h$ in (5.50). If we multiply (5.46) by $\chi\alpha$ and add it to (5.42) we obtain

$$\rho\frac{\alpha}{\Delta t}((1-\chi)\mathbf{v}_h^{n+1} - \chi\overline{\mathbf{v}}_h^{n+1}) + \mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{v}_h^{n+1} + \chi\alpha\mathscr{L}_F[\mathbf{v}_h^{n+1}]\overline{\mathbf{v}}_h^{n+1} + \nabla(q_h + \chi\alpha\overline{q}_h) = \mathbf{b}_h^{n+1}. \tag{5.54}$$

Then, using the relaxation for the velocity (5.51), we obtain that $\mathbf{u}_h^{n+1}$ satisfies

$$\rho\frac{\alpha}{\Delta t}\mathbf{u}_h^{n+1} + \mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{v}_h^{n+1} + \chi\alpha\mathscr{L}_F[\mathbf{v}_h^{n+1}]\overline{\mathbf{v}}_h^{n+1} + \nabla(q_h + \chi\alpha\overline{q}_h) = \mathbf{b}_h^{n+1}. \tag{5.55}$$

Using one more time (5.51), we obtain

$$\begin{aligned}\rho\frac{\alpha}{\Delta t}\mathbf{u}_h^{n+1} + \mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{u}_h^{n+1} + \nabla(q_h^{n+1} + \alpha\chi\overline{q}_h^{n+1}) + \\ - \chi(\mathscr{L}_{NS}[\mathbf{u}_h^*](\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) + \alpha\mathscr{L}_F[\mathbf{v}_h^{n+1}]\overline{\mathbf{v}}_h^{n+1}) = \mathbf{b}_h^{n+1}.\end{aligned} \tag{5.56}$$

The continuity equation for $\mathbf{u}_h^{n+1}$ is automatically satisfied, since both $\mathbf{v}_h^{n+1}$ and $\overline{\mathbf{v}}_h^{n+1}$ are divergence free.

A few interesting observations can be made by looking at equations (5.55)-(5.56). First of all, they suggest a relaxation for the end-of-step pressure, namely

$$p_h^{n+1} = q_h^{n+1} + \alpha\chi\overline{q}_h^{n+1}. \tag{5.57}$$

Next, consider the following perturbed version of the discrete Navier-Stokes equations

$$\rho\frac{\alpha}{\Delta t}\mathbf{u}_h^{n+1} + \mathscr{L}[\mathbf{u}_h^*, \mathbf{u}_h^{n+1}]\mathbf{u}_h^{n+1} + \nabla p_h^{n+1} = \mathbf{b}_h^{n+1}, \tag{5.58}$$

where the operator $\mathscr{L}$ is defined by

$$\mathscr{L}[\mathbf{u}_h^*, \mathbf{u}_h^{n+1}]\mathbf{u}_h^{n+1} = \mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{u}_h^{n+1} + \alpha\chi\mathscr{L}_F[\mathbf{u}_h^{n+1}]\mathbf{u}_h^{n+1}. \tag{5.59}$$

The perturbation in eq. (5.58) with respect to the Navier-Stokes equations is due to the operator $\mathscr{L}_F$, which provides additional diffusion through a nonlinear artificial viscosity. A possible strategy to solve eq. (5.58) is the following:

1. split the operator $\mathscr{L}$ according to eq (5.59)
2. solve operator $\mathscr{L}_{NS}$ to obtain the intermediate velocity $\mathbf{v}_h^{n+1}$
3. solve operator $\mathscr{L}_F$, linearized with respect to $\overline{\mathbf{v}}_h^{n+1}$, to obtain $\overline{\mathbf{v}}_h^{n+1}$
4. relax the two solutions according to (5.51).

It is easy to show that the end-of-step solution of such strategy satisfies eq. (5.55). Therefore, we can look at the EFR algorithm as an operator-splitting method for solving the perturbed Navier-Stokes equations (5.58). In particular, the operator-splitting scheme involves three steps:

i) in the first step, given by equations (5.42)-(5.43), the intermediate velocity and pressure are computed using the Navier-Stokes operator $\mathscr{L}_{NS}$ with the physical viscosity μ;

ii) in the second step, given by equations (5.46)-(5.47), the filtered velocity and pressure are computed using the filter operator $\mathscr{L}_F$ with the artificial viscosity $\overline{\mu}_h$;

iii) finally, we combine the solutions found at steps i) and ii) with (5.51), (5.57) to get the end-of-step velocity and pressure.

In the third step, the velocity and pressure found at the first step are *corrected* by taking into account the energy dissipated at the scales that were not resolved with the given mesh in the Navier-Stokes step. More precisely, looking at the diffusive terms in (5.55), we observe that there are two contributions:

i) $\nabla \cdot \left(2\mu\nabla^s \mathbf{v}_h^{n+1}\right)$: this is the physical dissipation, corresponding to the viscosity of the fluid μ.

ii) $\alpha\chi\nabla \cdot \left(2\overline{\mu}_h\nabla^s \overline{\mathbf{v}}_h^{n+1}\right)$: this is the numerical dissipation, induced by the filter step, corresponding to the artificial viscosity $\alpha\chi\overline{\mu}_h$.

Finally, we remark that in eq. (5.56) the perturbation with respect to the usual Navier-Stokes momentum equation is modulated by χ. As mentioned in Section 5.2.2, in [53] the authors suggest the scaling $\chi = O(\Delta t)$. Moreover, we recall that, if $\Delta t \to 0$ no faster than h^2 (as it is typically the case), thanks to eq. (5.39) we obtain that $\overline{\mu}_h \to 0$ and $\overline{\mathbf{v}}_h^{n+1} \to \mathbf{v}_h^{n+1}$ as $h \to 0$. Therefore, as the discretization parameters approach zero, the perturbation of the Navier-Stokes equation vanishes. This behavior is consistent with the fact that a mesh with characteristic length h smaller than the Kolmogorov length η is able to fully resolve the flow, and no stabilization should be added to account for missing dissipation (at under-resolved scales).

Remark 5.5. Eq. (5.57) is not the only possible choice for the end-of-step pressure. For example, another possible relaxation is given by

$$p_h^{n+1} = (1-\chi)q_h^{n+1} + \chi\overline{q}_h^{n+1}, \tag{5.60}$$

with the advantage of treating velocity and pressure in the same way, which may be useful from the implementation point of view. If we adopt this relaxation instead of (5.57), then eq. (5.56) becomes

$$\begin{aligned}
&\rho\frac{\alpha}{\Delta t}\mathbf{u}_h^{n+1} + \mathscr{L}_{NS}[\mathbf{u}_h^*]\mathbf{u}_h^{n+1} + \nabla p_h^{n+1} \\
&+ \chi(\mathscr{L}_{NS}[\mathbf{u}_h^*](\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) + \alpha(1-\chi)\mathscr{L}_F[\mathbf{v}_h^{n+1}](\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1})) \\
&+ \chi\nabla(q_h^{n+1} + (\alpha-1)\overline{q}_h^{n+1}) = \mathbf{b}_h^{n+1}.
\end{aligned} \tag{5.61}$$

Notice that this choice introduces an additional perturbation (last term at the at the left-hand side in eq. (5.61)), which is still multiplied by χ. For this reason, we preferred to use the pressure relaxation given by (5.57) for the numerical simulations reported in Section 5.6.

5.4.1 The choice of the relaxation parameter χ

In [53], the authors observe that, since $\overline{\mu}_h$ in (5.50) features Δt in the denominator, the choice $\chi = O(\Delta t)$ guarantees that the numerical dissipation vanishes as $h \to 0$, regardless of Δt. Given the properties of $a(\cdot)$ shown in Section 5.3, we can claim that $\overline{\mu} \to 0$ when $h \to 0$, provided that $h^2/\Delta t$ stays bounded (meaning that $\Delta t \to 0$ no faster than h^2, as it is typically the case). Therefore, the scaling $\chi = O(\Delta t)$ leads to end-of-step velocity and pressure that satisfy the Navier-Stokes equations up to a residual which vanishes with the discretization parameters (provided $h^2/\Delta t$ is bounded, as mentioned above).

However, this scaling rule does not provide a practical way to tune the proportionality constant between χ and Δt, and the natural choice $\chi \simeq \Delta t$ does not provide enough numerical dissipation in realistic applications (see Section 5.6). We also point out that, although consistency is an important property for a numerical method, in this case one is not interested in the behavior of the scheme as $h \to 0$, since for h small the mesh is already fine enough to capture all the scales and the filter is not needed.

In order to find a proper formula for χ we use a heuristic argument. We want to find χ such that the viscous stress in (5.58) on an under-resolved mesh of size h provides the same amount of dissipation as the viscous term in (5.42) on a fully resolved mesh of size η, η being the Kolmogorov length scale defined in (5.6).

We identify an *equivalent* stress tensor in the perturbed Navier-Stokes equations (5.58):

$$\underline{\underline{\sigma}}^{n+1} = -p_h^{n+1}\underline{\underline{\mathrm{I}}} + 2\mu\nabla^s\mathbf{v}_h^{n+1} + 2\alpha\chi\overline{\mu}\,\nabla^s\,\overline{\mathbf{v}}_h^{n+1}. \tag{5.62}$$

Then, we require that the viscous contribution of this tensor matches the viscous contribution of the usual stress tensor of a Newtonian fluid on a mesh of size η. In

other words,

$$(\mu + \alpha\chi\overline{\mu}_h)\nabla_h^s \mathbf{u}_h^{n+1} \simeq \mu\nabla_\eta^s \mathbf{u}_h^{n+1}, \tag{5.63}$$

where ∇_a^s $(a = h, \eta)$ denotes the operator ∇^s on a mesh of size a. Assuming that all the above velocities have the same order of magnitude, and approximating ∇_a^s with a^{-1}, we obtain

$$(\mu + \alpha\chi\overline{\mu}_h)\frac{1}{h} \simeq \mu\frac{1}{\eta}, \tag{5.64}$$

and thus

$$\chi \simeq \frac{\mu}{\alpha\overline{\mu}}\left(\frac{h}{\eta} - 1\right) \simeq \frac{\mu}{\alpha\rho}\frac{\Delta t}{||a||_\infty \delta^2}\left(\frac{h}{\eta} - 1\right). \tag{5.65}$$

Here, $||a||_\infty$ is the infinity norm of the indicator function. Notice that, except for $||a||_\infty$, the value of χ in (5.65) depends only on physical parameters, discretization parameters (recall that, in practice, $\delta \simeq h$), and the Kolmogorov length scale, which are all available quantities.

The expression (5.65) is compatible with the scaling $\chi = O(\Delta t)$ proposed in [53], but, in addition, it provides a practical way to compute a reasonable value for χ, requiring little *a priori* knowledge of the solution. In particular, one only needs to provide the Kolmogorov length η, which can be estimated through equation (5.6). Moreover, as we will see in Section 5.6, the ratio $\chi/\Delta t$ can be significantly larger than one.

5.4.2 The boundary conditions

Here we check which boundary conditions are satisfied by the end-of-step solution, and we justify the choice of (5.49) for the case of Neumann boundary conditions.

By combining (5.51), (5.44), and (5.48), we see that:

$$\mathbf{u}_h^{n+1} = \mathbf{u}_D, \quad \text{on } \partial\Gamma_D. \tag{5.66}$$

As for the Neumann part of the boundary Γ_N, from (5.51) and (5.57), we have

$$\begin{aligned}(2\mu\nabla^s\mathbf{u}_h^{n+1} - p_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} &= (2\mu\nabla^s((1-\chi)\mathbf{v}_h^{n+1} + \chi\overline{\mathbf{v}}_h^{n+1}) - (q_h^{n+1} + \alpha\chi\overline{q}_h^{n+1})\underline{\underline{\mathrm{I}}})\mathbf{n} \\ &= (2\mu\nabla^s\mathbf{v}_h^{n+1} - q_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} + \chi(2\mu\nabla^s(\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) - \alpha\overline{q}_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n}.\end{aligned}$$

Then, using (5.45) and (5.49), we find that

$$(2\mu\nabla^s\mathbf{u}_h^{n+1} - p_h^{n+1}\mathbf{I})\mathbf{n} = \mathbf{g} - \chi(2\mu\nabla^s(\mathbf{v}_h^{n+1} - \overline{\mathbf{v}}_h^{n+1}) + 2\alpha\overline{\mu}_h\,\nabla^s\overline{\mathbf{v}}_h^{n+1})\mathbf{n}, \tag{5.67}$$

which states that the Neumann boundary condition satisfied by the end-of-step quantities is a perturbation of the prescribed one. We recall that, thanks to eq. (5.39), as the mesh size approaches zero, $\overline{\mu}_h \to 0$ and $\overline{\mathbf{v}}_h^{n+1} \simeq \mathbf{v}_h^{n+1}$. Therefore, as the mesh size approaches zero, the perturbation introduced in the Neumann boundary condition vanishes.

Although less common, it is possible to have, on a portion of the boundary, more general Robin boundary conditions of the form

$$(2\mu\nabla^s\mathbf{u}_h^{n+1} - p_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} + \gamma\mathbf{u}_h^{n+1} = \mathbf{g}^{n+1} \text{ on } \Gamma_R,$$

where $\gamma > 0$. Among the scenarios where this kind of boundary condition can arise, we mention domain decomposition techniques in geometric multiscale approaches (see, e.g., [28], Chapter 11), and the use of the so called *transpiration* techniques for Fluid-Structure Interaction problems (see, e.g., [21]).

It can be shown that, for this type of boundary condition, the same perturbation as in (5.67) can still be achieved by imposing the following boundary condition for the filter equations:

$$(2\overline{\mu}_h\nabla^s\overline{\mathbf{v}}_h^{n+1} - \overline{q}_h^{n+1}\underline{\underline{\mathrm{I}}})\mathbf{n} + \frac{\gamma}{\alpha}\overline{\mathbf{v}}_h^{n+1} = \frac{\gamma}{\alpha}\mathbf{v}_h^{n+1} \text{ on } \Gamma_R,$$

where, we recall, α is the coefficient of $\mathbf{u}_h^{n+1}$ in the approximation of $\partial_t\mathbf{u}_h$ at t^{n+1} by BDFp.

5.5 Discretization of the operator-splitting algorithm

As we mentioned in the previous section, we use the Finite Elements Method for the space discretization. We start by introducing a conformal and quasi-uniform partition $\mathcal{T}_h$ of Ω made up of a certain number of tetrahedra. For the approximation of velocity and pressure, we will resort to the *inf-sup* stable FE pair $\mathbb{P}_2$-$\mathbb{P}_1$. For more details concerning the discretization of the Navier-Stokes problem, we refer, e.g., to [80]. We do not use any stabilization for the convective term; in fact, we will show how the non-linear filtering of the velocity not only improves the quality of the solution, but also allows to use coarser meshes compared to standard Navier-Stokes equations, therefore introducing a stabilizing effect.

For the time discretization we will use BDF2 (5.16), with the corresponding convective term extrapolation

$$\mathbf{u}_h^* = 2\mathbf{u}_h^n - \mathbf{u}_h^{n-1}.$$

In order to write the algebraic form of the problem, let us introduce the notation for both the evolve and filter steps. For the evolve step, we will denote by M the mass matrix, K the diffusion matrix, N the matrix associated with the discretization of the convective term, and B the matrix associated with the discretization of the operator $(-\nabla\cdot)$. Furthermore, we will denote with $\underline{v}$ the vector of coefficients in the FE expansion of $\mathbf{v}_h$. The full discretization of problem (5.42)-(5.43) with BDF2 yields the following system

$$\rho\frac{3}{2\Delta t}\mathrm{M}\underline{v}^{n+1} + \rho\mathrm{N}\underline{v}^{n+1} + \mu\mathrm{K}\underline{v}^{n+1} + \mathrm{B}^{\mathrm{T}}\underline{q}^{n+1} = \underline{b}_u^{n+1}, \tag{5.68}$$

$$\mathrm{B}\underline{v}^{n+1} = \underline{0}, \tag{5.69}$$

The array $\underline{b}_u^{n+1}$ accounts for the contributions of solution at the previous time steps and the contribution that the boundary nodes give to the internal nodes (concerning the non-homogeneous boundary conditions), while the constant $3/2$ in the first term comes from the formula (5.16).

Setting $\mathrm{C} = \rho\frac{3}{2\Delta t}\mathrm{M} + \rho\mathrm{N} + \mu\mathrm{K}$, we can rewrite (5.68)-(5.69) in the form

$$\mathrm{A}\underline{x}^{n+1} = \underline{b}^{n+1}, \tag{5.70}$$

where

$$\mathrm{A} = \begin{bmatrix} \mathrm{C} & \mathrm{B}^{\mathrm{T}} \\ \mathrm{B} & 0 \end{bmatrix}, \quad \underline{x}^{n+1} = \begin{bmatrix} \underline{v}^{n+1} \\ \underline{q}^{n+1} \end{bmatrix}, \quad \underline{b}^{n+1} = \begin{bmatrix} \underline{b}_u^{n+1} \\ \underline{0} \end{bmatrix}. \tag{5.71}$$

For the filter step, we still denote by M the mass matrix and by B the discretization of the (negative) divergence operator. We let $\overline{\mathrm{K}}$ be the matrix associated with the discretization of the diffusive term in (5.46). The full discretization of problem (5.46)-(5.49) then yields to the system:

$$\frac{\rho}{\Delta t}\mathrm{M}\overline{\underline{v}}^{n+1} + \overline{\mathrm{K}}\overline{\underline{v}}^{n+1} + \mathrm{B}^{\mathrm{T}}\overline{\underline{q}}^{n+1} = \frac{\rho}{\Delta t}\mathrm{M}\underline{v}^{n+1}, \tag{5.72}$$

$$\mathrm{B}\overline{\underline{v}}^{n+1} = \underline{0}, \tag{5.73}$$

where $\overline{\underline{v}}^{n+1}$ and $\overline{\underline{q}}^{n+1}$ are the nodal values of the filter step velocity and pressure.

Setting $\overline{\mathrm{C}} = \frac{\rho}{\Delta t}\mathrm{M} + \overline{\mathrm{K}}$, we can rewrite (5.72)-(5.73) in the form

$$\overline{\mathrm{A}}\overline{\underline{x}}^{n+1} = \overline{\underline{b}}^{n+1}, \tag{5.74}$$

where

$$\overline{\mathrm{A}} = \begin{bmatrix} \overline{\mathrm{C}} & \mathrm{B}^{\mathrm{T}} \\ \mathrm{B} & 0 \end{bmatrix}, \quad \overline{\underline{x}}^{n+1} = \begin{bmatrix} \overline{\underline{v}}^{n+1} \\ \overline{\underline{q}}^{n+1} \end{bmatrix}, \quad \overline{\underline{b}}^{n+1} = \begin{bmatrix} \frac{\rho}{\Delta t}\mathrm{M}\underline{v}^{n+1} \\ \underline{0} \end{bmatrix}. \tag{5.75}$$

At every time level t^{n+1}, to solve systems (5.70) and (5.74) we use the left preconditioned GMRES method. To precondition both systems, we use an upper-triangular variant of the pressure corrected Yosida splitting [33, 84]. For the matrix A, this preconditioner reads:

$$\mathrm{P}_{\mathrm{A}} = \begin{bmatrix} \mathrm{C} & \mathrm{B}^{\mathrm{T}} \\ 0 & \mathrm{S}(\mathrm{S} + \mathrm{BH}(\mu\mathrm{K} + \rho\mathrm{N})\mathrm{HB}^{\mathrm{T}})^{-1}\mathrm{S} \end{bmatrix}, \quad \mathrm{H} = \frac{2\Delta t}{3\rho}\mathrm{M}^{-1}, \quad \mathrm{S} = -\mathrm{BHB}^{\mathrm{T}}. \tag{5.76}$$

The above preconditioner is a suitable approximation of the U factor in the exact

block LU factorization of matrix A in (5.71):

$$\mathrm{A} = \mathrm{LU}, \quad \mathrm{L} = \begin{bmatrix} \mathrm{I} & 0 \\ \mathrm{BC}^{-1} & \mathrm{I} \end{bmatrix}, \quad \mathrm{U} = \begin{bmatrix} \mathrm{C} & \mathrm{B}^{\mathrm{T}} \\ 0 & -\mathrm{BC}^{-1}\mathrm{B}^{\mathrm{T}} \end{bmatrix}. \tag{5.77}$$

See [35,76,77] for more details. For the matrix $\overline{\mathrm{A}}$, the preconditioner has a similar structure, namely:

$$\mathrm{P}_{\overline{\mathrm{A}}} = \begin{bmatrix} \overline{\mathrm{C}} & \mathrm{B}^{\mathrm{T}} \\ 0 & \overline{\mathrm{S}}(\overline{\mathrm{S}} + \mathrm{B}\overline{\mathrm{H}}(\overline{\mathrm{K}})\overline{\mathrm{H}}\mathrm{B}^{\mathrm{T}})^{-1}\overline{\mathrm{S}} \end{bmatrix}, \quad \overline{\mathrm{H}} = \frac{\Delta t}{\rho}\mathrm{M}^{-1}, \quad \overline{\mathrm{S}} = -\mathrm{B}\overline{\mathrm{H}}\mathrm{B}^{\mathrm{T}}. \tag{5.78}$$

The application of the preconditioner requires to solve two linear systems in both C and S for P_{A}, and two linear systems in both $\overline{\mathrm{C}}$ and $\overline{\mathrm{S}}$ for $P_{\overline{A}}$. To solve each of these systems, we use a Krylov method with a general purpose preconditioner, such as incomplete LU or algebraic multilevel. It is worth mentioning that, while C is in general non-symmetric because of the convective term, the matrix $\overline{\mathrm{C}}$ is symmetric (and positive definite). Therefore, while for P_{A} the (1,1) block is solved with GMRES method, for $\mathrm{P}_{\overline{A}}$ it can be solved with the Conjugate Gradient method.

All the linear solvers use a stopping criterion based on the relative residual. In particular, in the outer solvers (for the matrices A and $\overline{\mathrm{A}}$) the tolerance is set to 10^{-9}, while for the inner solvers (for the matrices C, $\overline{\mathrm{C}}$, S and $\overline{\mathrm{S}}$) the tolerance was set to 10^{-3}.

5.6 Numerical experiments

In order to demonstrate the effectiveness of the approach described in the previous sections, we have selected a benchmark from the U.S. Food and Drug Administration (FDA). This benchmark consists in simulating the flow of an incompressible and Newtonian fluid with prescribed density and viscosity ($\rho = 1056$ kg/m^3 and $\mu = 0.0035$ Pa·s) in an idealized medical device shaped like a nozzle (see Fig. 5.2) at different Reynolds numbers. In the geometry pictured in Fig. 5.2, the fluid flows from left to right, passing through a cylindrical entrance region, a conical conver-

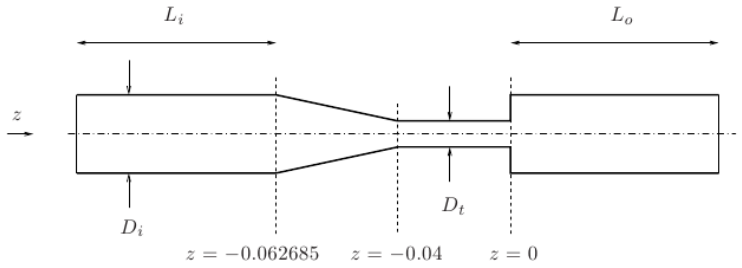


Figure 5.2: A section of the computational domain, with $D_i = 0.012$, $D_t = 0.004$, $L_i = 4D_i$ and $L_o = 12D_i$. The units are meter.

Re_t	Re_i	flow rate Q (m^3/s)	η
3500	1167	3.6444e-5	2.6371e-5
5000	1667	5.2062e-5	2.0182e-5

Table 5.1: Throat Reynolds number Re_t, inlet Reynolds number Re_i, flow rate Q, and Kolmogorov length scale η for the flow regimes under consideration.

gent, a cylindrical throat, and a sudden expansion into a larger cylinder.

The complete FDA benchmark requires to study this system for a variety of conditions, including laminar, transitional, and turbulent regimes: the results of the published inter-laboratory experiments refer to values of the Reynolds numbers in the throat (defined as in (5.3)) of $Re_t = 500, 2000, 3500, 5000, 6500$. In a previous work [67], the authors have successfully validated LifeV [1] against this benchmark for Re_t up to 3500 using direct numerical simulations.

To test our methodology, we focus on Reynolds numbers $Re_t = 3500, 5000$. In Table 5.1, we report the throat Reynolds number Re_t, the corresponding inlet Reynolds number Re_i, flow rate Q, and the Kolmogorov length scale η for the flow regimes that we are going to consider. The value of η was found by plugging into (5.6) the value of Re_t and the diameter of the expansion channel D_i as characteristic length. Notice that for both flow regimes the flow in the entrance region is laminar, Re_i being below the critical Reynolds number for transitional flow in a straight pipe ($Re \simeq 2000$ [81]).

On the lateral surface of the computational domain we prescribe a no-slip boundary condition. For the two flow regimes in Table 5.1, at the inlet section we prescribe a Poiseuille velocity profile to get the desired flow rate, a choice which is justified by the considered values of Re_i. The length of the inlet chamber L_i was set to four

times its diameter. At the outlet section, we prescribe a stress-free (natural) boundary condition. This is an artificial condition, which does not correspond to the experimental set up of the FDA benchmark (a closed flow loop [40]). However, this choice is expected to alter the computed solution only in a confined region of the computational domain close to the outlet section [42]. The results of the flow analysis are not affected, provided that the expansion channel is long enough. For all the simulations, the length of the expansion channel (L_o in Fig. 5.2) was set to 12 times its diameter and we checked that in all the cases the velocity components reach a plateau before the outlet.

As for the initial condition, we start our simulations with fluid at rest, i.e., $p = 0$ and $\mathbf{u} = \mathbf{0}$ everywhere in Ω. We use a smooth increase of the velocity profile at the inlet to transition from the fluid at rest to the regime flow conditions.

For both flow regimes in Table 5.1, we considered several meshes with different levels of refinement. The selection of the time step was driven by accuracy considerations solely. In fact, even though the semi-implicit treatment of the convective term in eq.(5.42) does not guarantee the unconditional stability in time of the numerical scheme, we encounter no time stability issues in the numerical experiments.

We compare the experimental data provided by the FDA with our numerical simulations for the flow regimes listed in Table 5.1. The experimental data were acquired by three independent laboratories and one of the laboratory ran three trials, so that for each case we have five sets of data. The comparison is made in terms of normalized axial component of the velocity and normalized pressure difference along the centerline. The axial component of the velocity u_z is normalized with respect to the average axial velocity at the inlet $\bar{u}_i$:

$$u_z^{norm} = \frac{u_z}{\bar{u}_i}, \quad \text{with} \quad \bar{u}_i = \frac{Q}{\pi D_i^2/4}, \tag{5.79}$$

where Q is the volumetric flow rate calculated from the throat Reynolds number (see Table 5.1). The pressure difference data are normalized with respect to the dynamic pressure in the throat:

$$\Delta p^{norm} = \frac{p_z - p_{z=0}}{1/2\rho\bar{u}_t^2}, \quad \text{with} \quad \bar{u}_t = \frac{Q}{\pi D_t^2/4}, \tag{5.80}$$

where p_z denotes the wall pressure along the z axis and $p_{z=0}$ is the wall pressure at $z = 0$. To compute the value of Δp^{norm}, we probed the pressure value at the corresponding location on the axis of the domain, since we observed pressure values being approximately uniform on axial cross-sections.

The graphs with the above comparisons are reported in Sec. 5.6.1 and 5.6.2 for $Re_t = 3500$ and $Re_t = 5000$, respectively.

5.6.1 Case $Re_t = 3500$

Among the Reynolds numbers considered by the FDA benchmark, $Re_t = 3500$ is the lowest above the critical Reynolds number for transitional flow in a straight pipe. As mentioned earlier, in [67] the authors already investigated this case, showing that DNS with a properly refined mesh is able to capture with precision the jet breakdown observed in the experiments. Here, we start from the mesh used in [67] for the DNS at $Re_t = 3500$ and make it progressively coarser to understand the performances of the EFR algorithm described in Sec. 5.4.

The meshes we considered and the associated time step used in the simulations are reported in Table 5.2. The name of each mesh is the number of elements. After several numerical experiments, in [67] we managed to identify a time step value and a mesh sufficiently refined in the different regions of the domain such that the results obtained with DNS were in excellent agreement with the experimental data. Mesh $1900k$ features that level of refinement and is associated with the same Δt used for the simulations in [67]. The time step associated with all the other meshes in Table 5.2 was chosen such that the ratio $h_{min}/\Delta t$ is kept roughly constant.

We noticed that, starting from fluid at rest, the turbulent regime is fully developed already passed $t \simeq 0.3$ s. So, we let the simulations run till after that time and then save around 10 snapshots, one every 100 time steps, we compute their average and we compared it with the experimental data. In fact, since the measurements of a turbulent flow are averaged over time [40], we average also the numerical re-

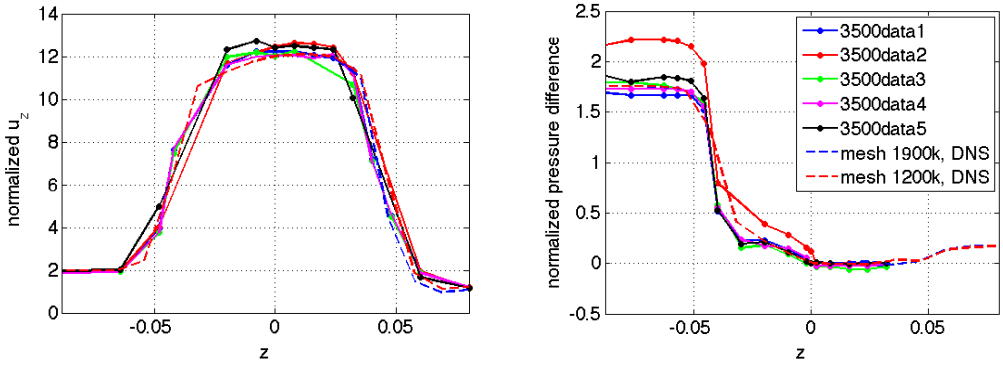
mesh name	h_{min}	h_{avg}	h_{max}	# nodes	# tetrahedra	Δt
$1900k$	1.06e-4	5.15e-4	1.49e-3	3.7e5	1.9e6	1e-4
$1200k$	1.08e-4	5.46e-4	1.63e-3	2.3e5	1.2e6	1e-4
$900k$	1.09e-4	5.16e-4	1.87e-3	1.8e5	9.0e5	1e-4
$330k$	2.23e-4	9.48e-4	1.93e-3	6.5e4	3.3e5	2e-4
$140k$	3.39e-4	1.11e-3	3.09e-3	3.1e4	1.4e5	3e-4

Table 5.2: Case $Re_t = 3500$: meshes used for the simulations, with their minimum diameter h_{min}, average diameter h_{avg}, maximum diameter h_{max}, and number of nodes and tetrahedra. We also report the time step Δt used for the simulations with each mesh.

sults for a fair comparison. We noticed that averaging over more than 10 snapshots would not change the average value.

A DNS is possible only with meshes $1900k$ and $1200k$. A DNS with mesh $900k$ does not even reach regime conditions because the instabilities in the computed velocity due to mesh under-resolution cause the simulation to crash. We report the comparison for the normalized axial velocity (5.79) along the z axis (Fig. 5.3(a)) and the normalized pressure difference (5.80) along the z axis (Fig. 5.3(b)). In Fig. 5.3, we plotted a dot for every measurement and a solid line to linearly interpolate the five sets of measurements, while we used a dashed line for the numerical results obtained with meshes $1900k$ and $1200k$. We notice that the axial velocities computed with both meshes match the measurements all along the portion of the z axis under consideration ($-0.088 < z < 0.08$), capturing with precision the jet breakdown point observed in experiments. As shown in Fig. 5.3(b), also the simulated pressure difference on both meshes is in very good agreement with the experimental data, except in the convergent where the simulated pressure difference overestimates almost all the measurements. The reason for this pressure difference overestimation is explained in [67]. Fig. 5.3 shows that even mesh $1200k$ has a sufficient level of refinement to obtain numerical results in excellent agreement with the measurements in terms of average quantities, despite its h_{avg} is roughly 20 times larger than the Kolmogorov length scale at $Re_t = 3500$ (see Tables 5.1 and 5.2).

We test the EFR algorithm described in Sections 5.4 and 5.5 with deconvolution of order $N = 0$ on all the meshes in Table 5.2 that are coarser than mesh $1200k$. We report the comparison between computed and measured normalized axial velocity



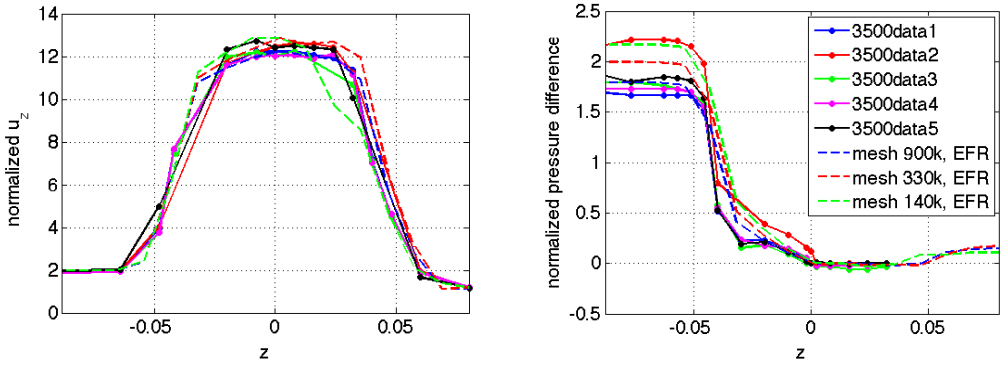
(a) normalized axial velocity along z (b) normalized pressure difference along z

Figure 5.3: Case $Re_t = 3500$, DNS with two different meshes: comparison between experimental data (solid lines) and numerical results (dashed lines) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.

(5.79) and pressure difference (5.80) in Fig. 5.4(a) and 5.4(b), respectively. Remarkably, the EFR algorithm succeeds in curing the convective term instabilities even on mesh $140k$, which has 88% less elements than mesh $1200k$ (the coarsest mesh that allowed for a DNS) and an average diameter more than 42 times larger than η at $Re_t = 3500$. From Fig. 5.4(a), we see that the axial velocity computed on meshes $900k$ and $330k$ are in agreement with the measurements all along the z axis. With mesh $140k$, the jet starts to breakdown closer to the sudden expansion than the jets obtained with the other two meshes. Nonetheless, the total jet length computed with mesh $140k$ agrees very well with the measured total jet length. As for the pressure difference, we see in Fig. 5.4(b) that the pressure differences computed on the three meshes fall within the measurements. In particular, the pressure difference simulated with mesh $900k$ is in very good agreement with the experimental data, again with the exception of the convergent region.

In [88], none of the presented CFD results were able to reproduce the correct jet breakdown point, because DNS predicted a longer jet (likely due to an incorrect simulation setup) while simulations with turbulence models under-predicted the jet length. Thus, all the computed axial velocities in Fig. 5.4(a) are reasonable, regardless of the mesh.

For a qualitative comparison, we report in Fig. 5.5 the velocity magnitude com-

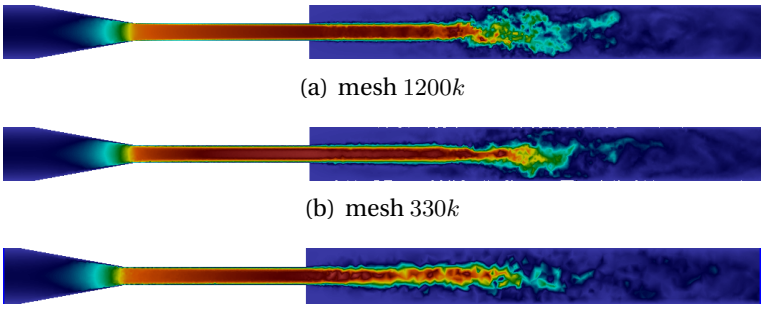


(a) normalized axial velocity along z (b) normalized pressure difference along z

Figure 5.4: Case $Re_t = 3500$, EFR with three different meshes, $N = 0$: comparison between experimental data (solid lines) and numerical results (dashed lines) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.

puted with meshes $1200k$, $330k$, and $140k$ on a section of the domain after the turbulent regime is fully established. We remind that the results with mesh $1200k$ (in Fig. 5.5(a)) have been obtained with DNS and therefore show a high level of detail. With meshes $330k$ and $144k$, the finer details of the smaller turbulent structures are lost, yet thanks to the EFR algorithm the average behavior of the flow is well captured (see Fig. 5.5(b) and 5.5(c)) at a fraction of the computational cost. In fact, a time step of the DNS with mesh $1200k$ takes around 240 s on 80 CPUs, while a time step of the EFR algorithm with mesh $330k$ takes around 80 s (50 s for the evolve step plus 30 s for the filter step) on 48 CPUs and with mesh $140k$ around 65 s (50 s for the evolve step plus 15 s for the filter step) on 24 CPUs. These computational times refer to simulations run on Maxwell, a cluster of the Research Computing Center at the University of Houston.

Next, we set the deconvolution order N to 1 and repeat the simulations on meshes $900k$, $330k$, and $140k$. We report the comparison for the normalized axial velocity (5.79) along the z axis in Fig. 5.6(a) and the normalized pressure difference (5.80) along the z axis in Fig. 5.6(b). From Fig. 5.6(a), we see that, while the axial velocity computed on mesh $900k$ is still in excellent agreement with the experimental data, the jet obtained with coarser meshes is too long. It seems that at $Re_t = 3500$ the choice $N = 1$ leads to too much artificial viscosity if the mesh is too coarse with

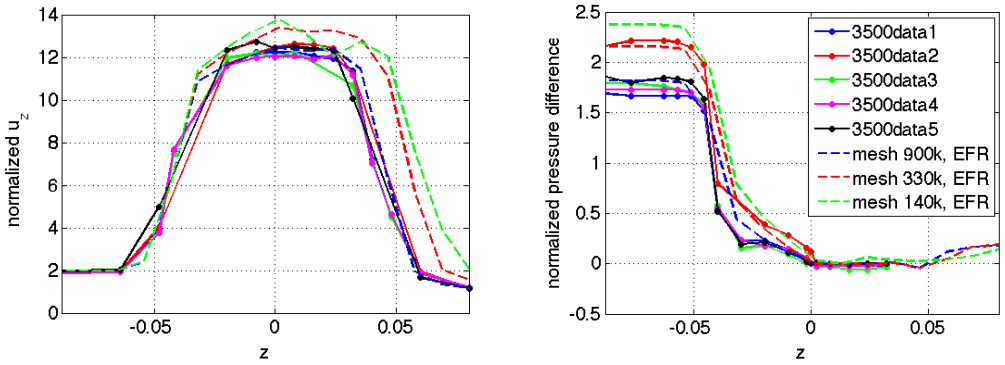


(a) mesh $1200k$

(b) mesh $330k$

(c) mesh $140k$

Figure 5.5: Case $Re_t = 3500$: velocity magnitude computed with meshes (a) $1200k$, (b) $330k$, and (c) $140k$ on a section of the domain after the turbulent regime is fully established. The results with mesh $1200k$ have been obtained with DNS, while the results with meshes $330k$ and $140k$ have been obtained with the EFR algorithm and $N = 0$.



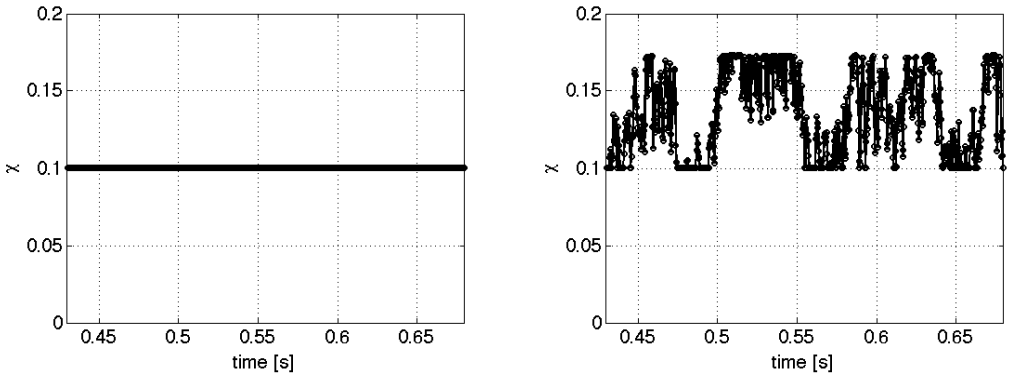
(a) normalized axial velocity along z

(b) normalized pressure difference along z

Figure 5.6: Case $Re_t = 3500$, EFR with three different meshes, $N = 1$: comparison between experimental data (solid lines) and numerical results (dashed lines) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.

respect to those that allow for DNS. Concerning Fig. 5.6(b), we see that the pressure difference gets more and more overestimated in the entrance region as the mesh gets coarser (up to 25% overestimation on mesh $140k$ with respect to the average measured pressure difference).

In order to understand the differences in the results obtained with $N = 0$ and $N = 1$, we track the value of χ over the time interval $[0.43, 0.68]$ s for mesh $140k$. Note that $||a||_\infty$ is the only quantity in the definition of χ (5.65) that varies in time. In Fig. 5.7(a), we see that for $N = 0$ the value of χ does not change over interval $[0.43, 0.68]$ s and it is equal to 0.1, which corresponds to $||a||_\infty = 1$. When $N = 1$, the



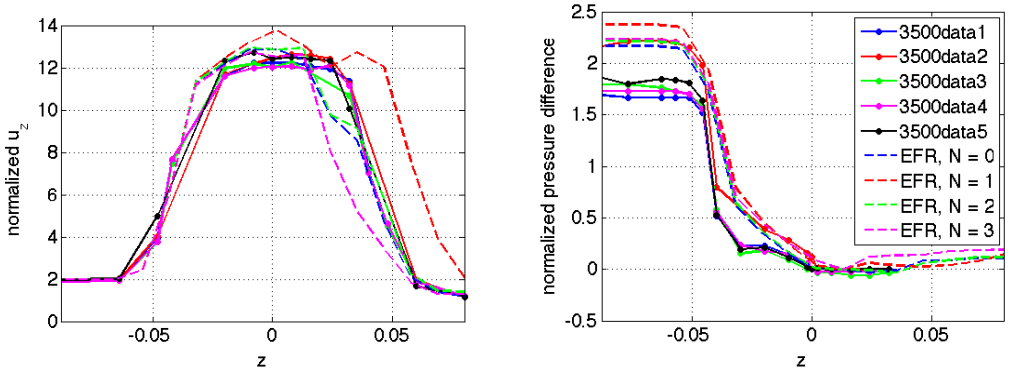
(a) $N = 0$ (b) $N = 1$

Figure 5.7: Case $Re_t = 3500$, EFR with mesh $140k$: value of χ over time interval $[0.43, 0.68]$ s for (a) $N = 0$ and (b) $N = 1$.

value of χ oscillates between 0.1 and 0.17 (see Fig. 5.7(b)). The value of χ is chosen so that the dissipation given by the physical and artificial viscosity amounts to the physical dissipation on a properly refined mesh, i.e. with $h \simeq \eta$ (see relation (5.63)). However, the larger the value of χ, the more the filtered velocity $\overline{\mathbf{v}}_h$ weights in the end-of-step velocity (5.51). This explains the longer jets in Fig. 5.6 that we get with meshes $330k$ and $140k$ for $N = 1$.

Notice that the values of χ computed with eq. (5.65) and shown in Fig. 5.7 are two orders of magnitude larger than Δt used for mesh $140k$ (see Table 5.2). For the problem under consideration, the choice $\chi \simeq \Delta t$ is not appropriate, since it would lead to an under-diffused flow that would not match the experimental data.

By comparing Fig. 5.4 and 5.6, one could think that the artificial viscosity introduced by the filtering step increases as N increases. To this purpose, we compare the normalized axial velocity (5.79) along the z axis and the normalized pressure difference (5.80) along the z axis obtained with mesh $140k$ and $N = 0, 1, 2, 3$ in Fig. 5.8(a) and 5.8(b), respectively. From Fig. 5.8(a), we see that the jet length reduces as N is increased from 1 to 3, while the let length for $N = 0$ is comparable to that obtained for $N = 2$. As for the pressure difference, the results obtained with $N = 0, 2, 3$ are fairly close to each other, while the pressure difference computed with $N = 1$ is larger in the entrance region. The sensitivity of the solution to N (among other parameters) is currently under investigation and will be object of a future publication.



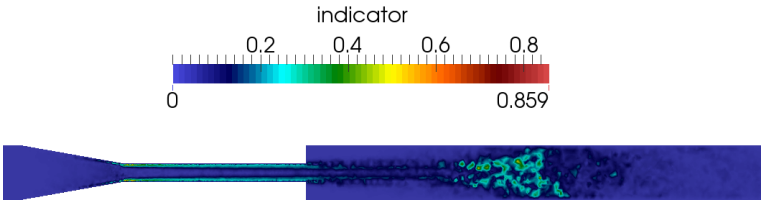
(a) normalized axial velocity along z (b) normalized pressure difference along z

Figure 5.8: Case $Re_t = 3500$, EFR with mesh $140k$ and four different values of the deconvolution order $N = 0, 1, 2, 3$: comparison between experimental data (solid lines) and numerical results (dashed lines) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.

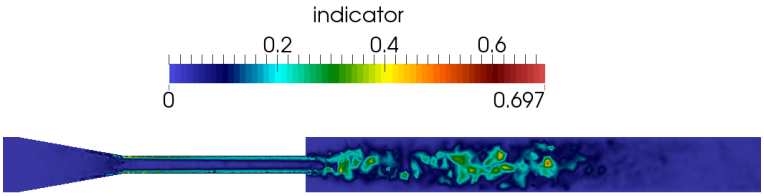
We tried also an extremely coarse mesh with roughly $7.2 \cdot 10^4$ elements and $h_{avg} = 1.25 \cdot 10^{-3}$m. The simulation with the EFR algorithm on such a mesh crashed less than 0.1 s after reaching regime conditions regardless of the deconvolution order. This indicates that with an extremely coarse mesh (with approximately 95% less elements than the coarsest mesh that allowed for a DNS) the EFR algorithm does not provide enough artificial diffusion to cure the convective term instabilities.

A key role in the EFR algorithm is played by the indicator function. We show in Fig. 5.9 the indicator function a_{D_0} (see (5.37) and Remark 5.3 for the definition), computed with meshes $330k$, and $140k$ at the same time step as the velocity magnitudes reported in Fig. 5.5(b) and 5.5(c), respectively. For both cases in Figure 5.9, the indicator function takes its largest value in the boundary layer at the entrance of the throat. Moreover, on mesh $140k$ it takes fairly large values all along the jet, while on mesh $330k$ larger values are taken only where the jet breaks down. Figure 5.9 shows that a_{D_0} is a suitable indicator function since it correctly selects the regions of the domain where the velocity does need regularization.

Remark 5.6. The exact choice of the filtering radius δ is an open problem when using non-uniform grids. If the mesh has been properly generated, it is fine where the smaller scales are expected and coarser where larger scales are expected. In this



(a) mesh $330k$



(b) mesh $140k$

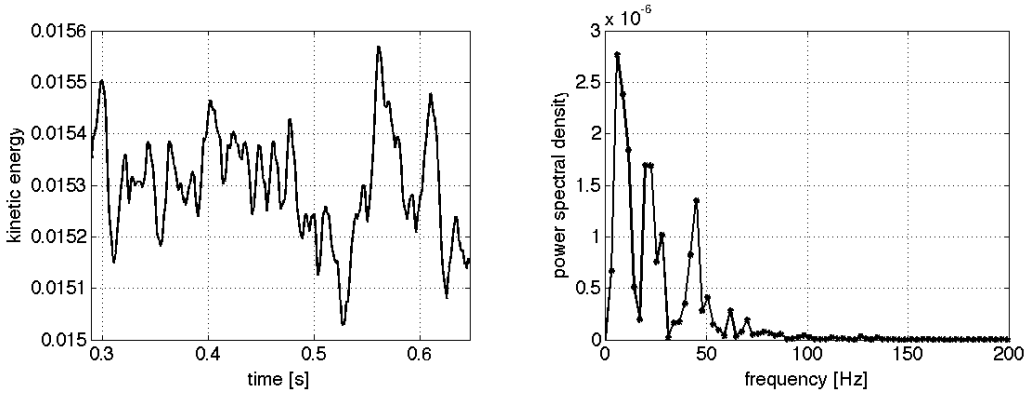
Figure 5.9: Case $Re_t = 3500$, EFR with $N = 0$: indicator function a_{D_0} computed with meshes (a) $330k$ and (b) $140k$ at the same time step as the velocity magnitudes reported in Figure 5.5(b) and 5.5(c), respectively.

case, it is reasonable to set $\delta = h_{min}$. So, where the smaller scales are expected the velocity will be smoothed but not oversmoothed (which happens if $\delta > h_{min}$), while where the larger scales are expected and the local h is greater than δ the filtering is essentially not affecting the flow. For all the simulations in this work, we have set $\delta = h_{min}$.

Remark 5.7. The operator ∇^s in eq. (5.42) and (5.46) could be replaced by the operator ∇ only if the mesh is not too coarse with respect to the meshes that allow for DNS. This option is appealing because it reduces the computational costs[3], but if the mesh is too coarse it leads to instabilities that make the simulation crash. The only mesh with which we could use operator ∇ is $900k$, where we obtained results almost superimposed to those we got with operator ∇^s, shown in Figure 5.4. As mentioned in Remark 5.1, at the discrete level the contribution of $\nabla \mathbf{u}^{\mathrm{T}}$ is essential.

Remark 5.8. When $||a||_\infty$ is very small, the value of χ becomes large. In order to avoid dealing with a large value of χ, in our solver the filter is turned on only when the velocity is sufficiently large to make the current Kolmogorov length scale (which is computed with the current Reynolds number) smaller than h_{min}.

[3]In fact, using the operator ∇^s rather than ∇ for the stress tensor creates a coupling between the different velocity components, which causes the pattern of the momentum matrix to be full rather than block diagonal.



(a) kinetic energy (b) power spectral density

Figure 5.10: $Re = 3500$, DNS with mesh $1900k$: kinetic energy over time interval $[0.29, 0.64]$s (a) and the velocity power spectral density (b).

We conclude this section by analyzing the evolution of the kinetic energy of the system. More precisely, we study the *power spectral density* of the velocity field. Recall that the power spectral density of a time-series $x(t)$ is simply the distribution in the frequency domain of its variance. In this case, the time-series is the velocity field, and its variance is the kinetic energy related to the velocity fluctuations [2]. This allows us to establish the periodic characterof the solution in time. We let the DNS with mesh $1900k$ run for several tenths of seconds. The computed kinetic energy is shown in Figure 5.10(a): it oscillates around the mean value 0.0153. The power spectral density of the signal in Figure 5.10(a) is reported in Figure 5.10(b): there are several peaks for frequencies up to 50 Hz. In Figure 5.11, we see the power spectral density of the velocity field computed with the EFR algorithm on meshes $900k$ and $140k$. Not all the peaks that appear in the spectrum of the kinetic energy obtained with DNS are present in the spectra of the kinetic energy given by the EFR algorithm and, in particular, the coarser the mesh the less frequencies are excited. Nonetheless, the frequencies that do get excited appear also in the in the power spectrum in Figure 5.10(b).

5.6.2 Case $Re_t = 5000$

The second flow regime we consider features a throat Reynolds number $Re_t = 5000$. As for $Re_t = 3500$, turbulence downstream of the sudden expansion was observed

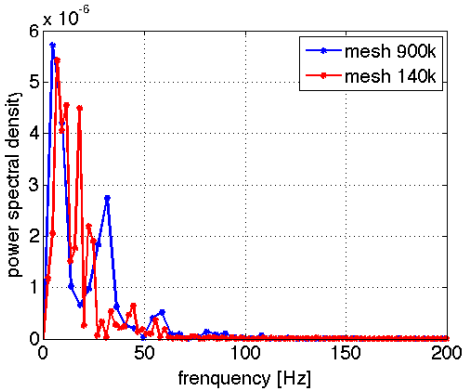


Figure 5.11: $Re = 3500$, EFR with meshes $900k$ and $140k$: power spectral density of the velocity field.

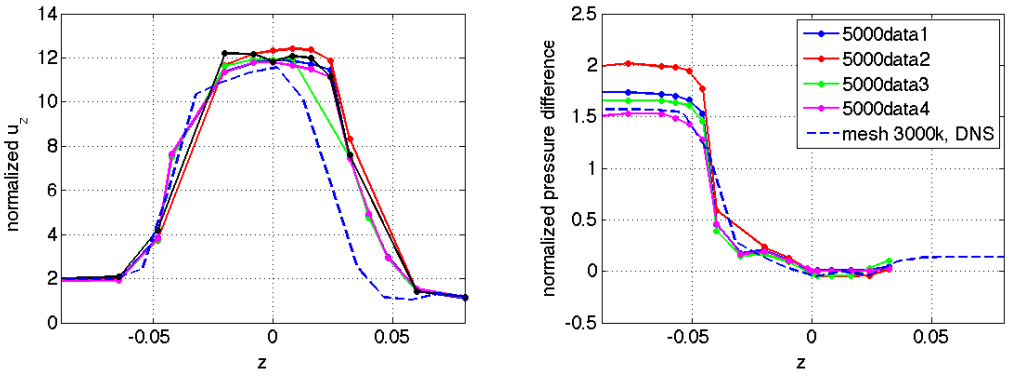
mesh name	h_{min}	h_{avg}	h_{max}	# nodes	# tetrahedra	Δt
$3000k$	1.17e-4	4.76e-4	9.64e-4	5.5e5	3.0e6	1e-4
$1900k$	1.06e-4	5.15e-4	1.49e-3	3.7e5	1.9e6	1e-4
$900k$	1.09e-4	5.16e-4	1.87e-3	1.8e5	9.0e5	1e-4
$330k$	2.23e-4	9.48e-4	1.93e-3	6.5e4	3.3e5	2e-4

Table 5.3: Case $Re_t = 5000$: meshes used for the simulations, with their minimum diameter h_{min}, average diameter h_{avg}, maximum diameter h_{max}, and number of nodes and tetrahedra. We also report the time step Δt used for the simulations with each mesh.

in all the experiments with a reproducible jet breakdown point.

We consider several meshes with different levels of refinement. Table 5.3 contains the details of all the meshes under consideration, together with the associated time step used in the simulations. Meshes $1900k$, $900k$, $330k$ are the same used for the simulations at $Re_t = 3500$, while the finest mesh was generated for this specific case. Despite the large number of nodes and tetrahedra, mesh $3000k$ has an average diameter roughly 24 times larger than the Kolmogorov scale at $Re_t = 5000$ (see Tables 5.1 and 5.3) and is not refined enough for a DNS to give results in good agreement with the experimental data. In fact, we see in Figure 5.12(a) that the computed velocity underestimates the measured velocity in the throat and the computed jet length is too small due to insufficient diffusion. A better agreement is found between computed and measured pressure difference (see Figure 5.12(b)). We remark that the lab whose data are labeled as "data5" provided measurements for for the axial velocity but not of the pressure at $Re_t = 5000$.

We have seen in Sec. 5.6.1 that the results obtained at $Re_t = 3500$ with the EFR al-

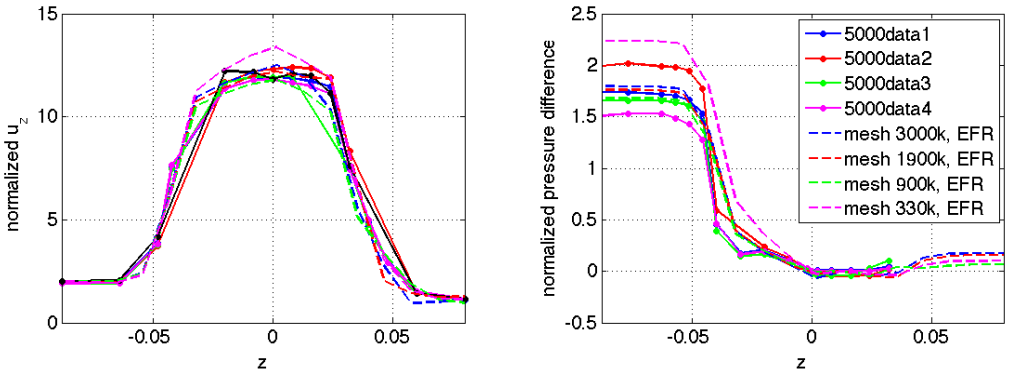


(a) normalized axial velocity along z (b) normalized pressure difference along z

Figure 5.12: Case $Re_t = 5000$, DNS with mesh $3000k$: comparison between experimental data (solid lines) and numerical results (dashed line) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.

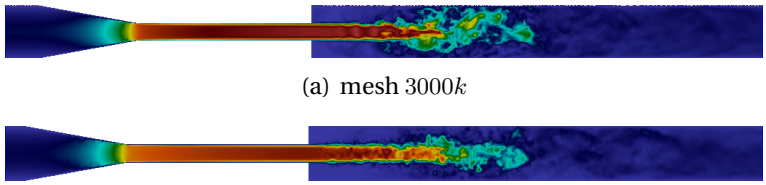
gorithm compared well with the measurements when $N = 0$, while the choice $N = 1$ featured too much artificial viscosity. Since we are mostly using the meshes employed at $Re_t = 3500$, we expect the EFR algorithm to perform better at $Re_t = 5000$ for $N = 1$ because additional dissipation is needed to compensate for the smaller scales that are filtered out by each mesh. We report the comparison between computed and measured normalized axial velocity (5.79) and pressure difference (5.80) for all the meshes in Table 5.3 in Figure 5.13(a) and 5.13(b), respectively. Indeed, from those figures we see that the jet length is very well captured with all the mesh, and the axial velocity and pressure difference computed on meshes $3000k$, $1900k$, and $900k$ are in excellent agreement with the respective measured quantities. For the results on mesh $330k$, we observe that the axial velocity gets overestimated in the throat and the pressure difference is overestimated in the entrance region. Notice that this same behavior for the solution on coarser meshes was observed at $Re_t = 3500$ for $N = 1$ (see Figure 5.6).

From Figure 5.13, we notice that there is little difference between the results obtained with mesh $3000k$ and those obtained with mesh $900k$, as they both compare well with the experimental data. See also Figure 5.14 for a qualitative comparison of the velocity magnitude computed in the two cases on a section of the domain after the turbulent regime is fully established. Obviously, the computational time



(a) normalized axial velocity along z (b) normalized pressure difference along z

Figure 5.13: Case $Re_t = 5000$, EFR with four different meshes, $N = 1$: comparison between experimental data (solid lines) and numerical results (dashed lines) for (a) normalized axial velocity (5.79) along the z axis and (b) normalized pressure difference (5.80) along the z axis. The legend in (b) is common to both subfigures.



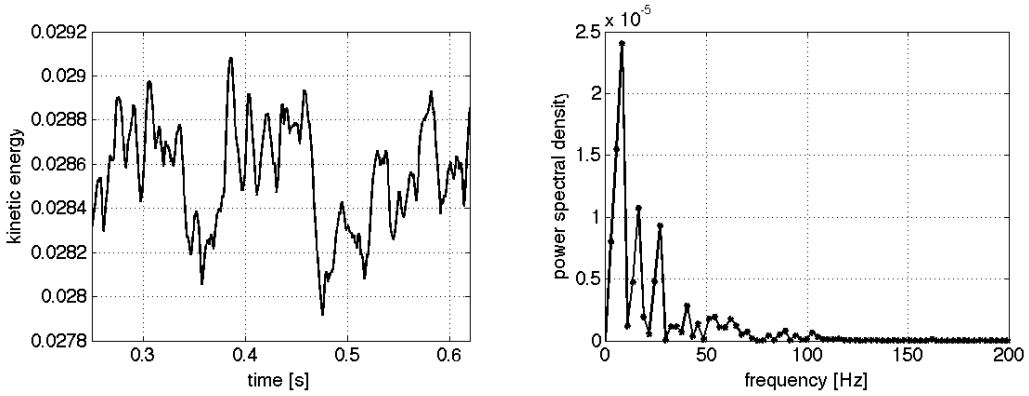
(a) mesh $3000k$

(b) mesh $900k$

Figure 5.14: Case $Re_t = 5000$, EFR with $N = 1$: velocity magnitude computed with (a) mesh $3000k$ and (b) mesh $900k$ on a section of the domain after the turbulent regime is fully established.

needed for the two simulations is very different: a time step of the EFR algorithm with mesh $3000k$ takes around 280 s (220 s for the evolve step plus 60 s for the filter step) on 208 CPUs and with mesh $900k$ around 220 s (165 s for the evolve step plus 55 s for the filter step) on 96 CPUs. Again the computational times refer to simulations run on Maxwell. Better computational times were achieved on Stampede, a cluster of the XSEDE consortium (around 52 s per iteration on 256 CPUs for mesh $3000k$).

In order to analyze the evolution of the kinetic energy of the system, we let the EFR algorithm with mesh $1900k$ run for several tenths of seconds. In Figure 5.15(a) we show the computed kinetic energy, while in Figure 5.15(b) we show the power spectral density of the velocity field. At $Re_t = 5000$, the kinetic energy oscillates around the mean value 0.0285. As expected, more peaks appear in Figure 5.15(b)



(a) kinetic energy (b) power spectral density

Figure 5.15: $Re = 5000$, EFR with mesh $1900k$: kinetic energy over time interval $[0.25, 0.62]$s (a) and velocity field power spectral density (b).

than in Figure 5.10(b), which has been obtained with the same mesh at $Re_t = 3500$: at $Re_t = 5000$ ($Re_t = 3500$, resp.) frequencies up to 100 Hz (50 Hz, resp.) have power spectral density peaks higher than $5 \cdot 10^{-7}$.

Chapter 6

Conclusions and future directions

The recent development of medical imaging equipment and mathematical models/methods has opened the way for a massive use of mathematics in clinical diagnosis and treatment. In particular, for the cardiovascular field, numerical simulations have proved to be an important tool to understand the flow of blood inside vessels and its interaction with the vessel walls. Furthermore, in recent years, cardiovascular mathematics has changed from a tool for describing and understanding the behavior of the cardiovascular system to a tool for quantitatively analyzing it and even design possible treatments. However, the quantitative use of mathematical tools demands more accurate and patient specific results, which means improvements in the reliability of numerical solutions. This, in turn, reflects on the choice of the numerical discretizations and, on the other hand, on the accurate tuning of the parameters in the model.

Data Assimilation

Data Assimilation is a collection of techniques that, after having proved successful in other engineering fields, are recently rising as a complementary tool in biomedical application, making the results of numerical simulations more reliable for clinical purposes. In this work we followed the idea proposed in [69] for the variational estimation of the compliance of a vessel by means of assimilating measures of the displacement of the vessel wall. In particular, we proposed a POD-based Reduced

Order Model (ROM) approach for the reduction of the costs associated with the solution of the inverse problem, and we compared the estimates with those obtained by solving the inverse problem in the Full Order Model (FOM). The algorithm had proven to be robust with respect to the errors in the measurements while adding a regularizing effect to the inverse problem. This is expected, since the ROM approach forces the solution to lie in a vector space of dimension remarkably smaller, compared to that of the corresponding FOM. This constraint ultimately helps to filter out a considerable amount of the noise present in the measurements. In particular, we emphasize the robustness of the method with respect to the noise in the measures, with good results for Signal to Noise Rations as low as 5 (corresponding to an intensity of noise up to 20% of the intensity of the data).

On the computational costs side, the approach contributed to reduce the costs of the online stage by roughly an order of magnitude. Most of the residual online costs are associated to the assembly of the Finite Element matrices and to the recovery of the pressure at the end of the time step, which, ultimately, depend on the fact that the geometry changes in time. In the case of small displacements, we can identify a possible research direction for further damp the online costs. To this end, let $\mathrm{A}(\underline{\eta})$ denote a generic matrix assembled in the geometry corresponding to the wall displacement $\underline{\eta}$. If $\underline{\eta}$ is a linear combination of the displacement Reduced Basis vectors $\underline{w}_1, \ldots, \underline{w}_N$, then we can write

$$\mathrm{A}(\underline{\eta}) = \mathrm{A}\left(c_1\underline{w}_1 + \cdots c_N\underline{w}_N\right). \tag{6.1}$$

If the displacements are small, one can approximate $\mathrm{A}(\underline{\eta})$ with a linear Taylor polynomial with respect to the coefficients $c_1, \ldots, c_N$, as

$$\mathrm{A}(\underline{\eta}) \simeq \mathrm{A}(\underline{0}) + \sum_{i=1}^{N} c_i \frac{\partial \mathrm{A}}{\partial \underline{w}_i}, \tag{6.2}$$

where $\mathrm{A}(\underline{0})$ denotes the matrix assembled in the undeformed geometry, while $\dfrac{\partial \mathrm{A}}{\partial \underline{w}_i}$ can be computed using the *shape derivatives* or approximated by simple finite differences. This approach would allow us, in the offline stage, to precompute these

matrices and project them onto the corresponding reduced space. Then, in the online stage, the reduced system matrix can be obtained again using the expression (6.2), where each of the matrices in the linear expansion is of the dimension of the reduced space, hence making the assembly cost independent of the dimension of the FE space used in the offline stage.

Another important direction that would be worth exploring is the reusability of the Reduced Basis. As a matter of fact, since the geometry is patient specific, it is necessary to compute a RB for every new geometry. A possible way to damp the offline costs would be to construct a geometry *atlas*, consisting of a limited number of domain configurations. The RB construction and the solution of the inverse problem would then happen only on the geometries in the atlas. When the inverse problem has to be solved for a new set of measures corresponding to a new geometry, we would then map the measures on the atlas, find the *closest* geometry available in the atlas, and then solve the inverse problem with the corresponding RB. This approach has already been used for building ROM for biomedical applications [60], and it relies on the development of a reliable and efficient mapping criterion.

Large Reynolds numbers and filtering techniques

Although blood flow in the human cardiovascular system is usually laminar, there are a few scenarios where the Reynolds number may be large enough to trigger turbulent effects, as we pointed out in the Remark 5.2. The accuracy and the computational costs of simulating flows at large Reynolds number represent a challenging aspect, and Direct Numerical Simulation (DNS) can become a big burden in some cases. In such scenarios, stabilization techniques that allow to obtain accurate results while keeping the number of unknowns limited become mandatory.

In this work we analyzed the model proposed in [53], for the discretization of the generalized Leray system, with a deconvolution-based indicator function. In particular, we reinterpreted the method as an operator-splitting scheme for the discretization of a perturbed version of the Navier-Stokes equations, and we used this to derive a heuristic tuning of one of the main parameters involved in the model.

We tested the effectiveness of the algorithm on a realistic 3D flow problem, corresponding to a benchmark proposed by the Food and Drug Administration, and we compared the results with experimental data. We emphasize, in particular, the fact that the method allows to use meshes of much smaller dimension (even by an order of magnitude) than those neded with a DNS approach, while still providing reliable results.

Furthermore, the interpretation of the algorithm as an operator-splitting scheme opens the door to a variety of techniques, such as incomplete factorization and correction techniques, that may lead, in the future, to more accurate schemes.

References

[1] *LifeV - A parallel Finite Element library*, `www.lifev.org`.

[2] F. Autieri, N. Parolini, and L. Quartapelle, *Numerical investigation on the stability of singular driven cavity flow*, Journal of Computational Physics **183** (2002), 1–25.

[3] P. Bergthorssón and B. R. Döös, *Numerical weather map analysis*, Tellus **7** (1955), 329–340.

[4] C. Bertoglio, P. Moireau, and J.F. Gerbeau, *Sequential parameter estimation for fluid-structure problems: Application to hemodynamics*, International Journal for Numerical Methods in Biomedical Engineering **28** (2012), 434–455.

[5] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: the Krylov-Schur solver*, SIAM Journal on Scientific Computing **27** (2005).

[6] ———, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: the Lagrange-Newton solver and its application to optimal control of steady viscous flows*, SIAM Journal on Scientific Computing **27** (2005).

[7] J. Borggaard, T. Iliescu, and J.P. Roop, *A bounded artificial viscosity large eddy simulation model*, SIAM J. Numer. Anal. **47** (2009), 622–645.

[8] A. L. Bowers and L. G. Rebholz, *Numerical stydy of a regularization model for incompressible flow with deconvolution-based adaptive nonlinear filter-*

ing, Computer Methods in Applied Mechanics and Engineering **258** (2013), 1–12.

[9] A. L. Bowers, L. G. Rebholz, A. Takhirov, and C. Trenchea, *Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering*, Int. J. Num. Meth. Fluids **70** (2012), 805–828.

[10] J. P. Boyd, *Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: Preserving boundary conditions and interpretation of the filter as a diffusion*, Journal of Computational Physics **143** (1998), no. 1, 283–288.

[11] H. Brezis, *Functional analysis, sobolev spaces and partial differential equations*, Springer, 2010.

[12] F. Brezzi, *On the existence, uniqueness and approximation of saddle point problems arising from Lagrange multipliers*, RAIRO Analyse Numèrique **8** (1974), 129–151.

[13] A.N. Brooks and T.J.R. Hughes, *Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation*, Computer Methods in Applied Mechanics and Engineering **32** (1982), 199–259.

[14] M. Brummer, private communication, 2009, Emory Children Healthcare of Atlanta.

[15] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici, *A priori convergence of the greedy algorithm for the parametrized reduced basis method*, ESAIM: Mathematical Modelling and Numerical Analysis **46** (2012), 595–603.

[16] T. Canfield and P. Dobrin, *Static elastic properties of blood vessels*, Handbook of Bioengineering (R. Skalak and S. Chen, eds.), McGraw-Hill, New York, 1987.

[17] I. Charpentier, *Checkpointing schemes for adjoint codes: Application to the meteorological model meso-nh*, SIAM Journal on Scientific Computing **22** (2000), 2135–2151.

[18] A. Cheskidov, D. D. Holm, E. Olson, and E. S. Titi, *On a Leray-α model of turbulence*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **461** (2005), no. 2055, 629–649.

[19] M. D'Elia, *Assimilation of velocity data into fluid dynamic simulations, an application to computational hemodynamics*, Ph.D. thesis, Emory University, Atlanta, December 2011.

[20] S. Deparis, *Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach*, SIAM Journal on Numerical Analysis **46** (2008).

[21] S. Deparis, M. A. Fernandez, and L. Formaggia, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions*, ESAIM: Mathematical Modelling and Numerical Analysis **37** (2003), 601–616.

[22] A. Dunca and Y. Epshteyn, *On the Stolz-Adams deconvolution model for the large-eddy simulation of turbulent flows*, SIAM Journal on Mathematical Analysis **37** (2005), no. 6, 1980–1902.

[23] H.W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Kluwer Academic Publishers, 2000.

[24] L. Euler, *Principia pro motu sanguinis per arterias determinando*, vol. 16, pp. 178–196, Birkhäuser Verlag", Basel, 1989.

[25] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*, Springer-Verlag, Berlin.

[26] J.P. Fink and W.C. Rheinboldt, *On the error behavior of the reduced basis technique for nonlinear finite element approximations*, Journal of Applied Mathematics and Mechanics **63** (1983), 21–28.

[27] P. Fischer and J. Mullen, *Filter-based stabilization of spectral element methods*, Comptes Rendus de l'Academie des Sciences - Series I - Mathematics **332** (2001), no. 3, 265–270.

[28] L. Formaggia, A. Quarteroni, and A. Veneziani, *Cardiovascular mathematics. modeling and simulation of the cyrculatory system*, Modeling, Simulation and Applications, vol. 1, Springer, 2009.

[29] L. Formaggia, A. Veneziani, and C. Vergara, *A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics*, SIAM Journal on Numerical Analysis **46** (2008), 2769–2794.

[30] U. Frisch, *Turbulence: The legacy of A. N. Kolmogorov*, Cambridge University Press, Cambridge, 1996.

[31] Y. Fung, K. Fronek, and P. Patitucci, *Pseudoelasticity of arteries and the choice of its mathematical expression*, Americal Journal of Physics **237** (1979), H620–H631.

[32] E. Garnier, N. Adams, and P. Sagaut, *Large eddy simulation for compressible flows*, Springer, Berlin, 2009.

[33] A. Gauthier, F. Saleri, and A. Veneziani, *A fast preconditioner for the incompressible Navier Stokes equations*, Computing and Visualization in Science **6** (2004), no. 2, 105–112.

[34] M. Germano, *Differential filters of elliptic type*, Physiscs of Fluids **29** (1986), 1757–1758.

[35] P. Gervasio, F. Saleri, and A. Veneziani, *Algebraic fractional-step schemes with spectral methods for the incompressible Navier-Stokes equations*, Journal of Computational Physics **214** (2006), no. 1, 347–365.

[36] E. Giuliani, B. Gersh, M. McGoon, D. Hayes, and H. Schaff, *Mayo clinic practice of cardiology*, Mosby, St Louis, 1996.

[37] R. Glowinski, *Finite element methods for incompressible viscous flow, in: P.G.Ciarlet, J.-L.Lions (Eds), Handbook of numerical analysis*, vol. 9, North-Holland, Amsterdam, 2003.

[38] M.D. Gunzburger, *Perspectives in flow control and optimization. advances in design and control*, SIAM, Philadelphia, 1987.

[39] E. Haber and U.M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems **17** (2001).

[40] P. Hariharan, M. Giarra, V. Reddy, S. W. Day, K. B. Manning, S. Deutsch, S. F. C. Stewart, M. R. Myers, M. R. Berman, G. W. Burgreen, E. G. Paterson, and R. A. Malinauskas, *Multilaboratory particle image velocimetry analysis of the FDA benchmark nozzle model to support validation of computational fluid dynamics simulations*, Journal of Biomechanical Engineering **133** (2011), no. 4.

[41] M. O. Henriksen and J. Holmen, *Algebraic splittings for incompressible Navier-Stokes equations*, Journal of Computational Physics **175** (2002), 438–453.

[42] J. Heywood, R. Rannacher, and S. Turek, *Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids **22** (1996), 325–352.

[43] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, 1985.

[44] J. Humpherys, P. Redd, and J. West, *A fresh look at the Kalman filter*, SIAM Review **54** (2012), 801–823.

[45] J. C. Hunt, A. A. Wray, and P. Moin, *Eddies stream and convergence zones in turbulent flows*, Tech. Report CTR-S88, CTR report, 1988.

[46] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, Analyse Numèrique **345** (2007), 473–478.

[47] R. A. Johnson and D. W. Wichern, *Applied multivariate statistical analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1998.

[48] S. J. Julier and J. K. Uhlmann, *Unscented filtering and nonlinear estimation*, Proceedings of the IEEE **92** (2004), 401–422.

[49] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Transactions of the ASME Journal of Basic Engineering **82** (1960), 35–45.

[50] D. J. Knezevic, N. C. Nguyen, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for the parametrized unsteady boussinesq equations*, Mathematical Models and Methods in Applied Sciences **21** (2011).

[51] A. N. Kolmogorov, *Dissipation of energy in isotropic turbulence*, Doklady Akademii Nauk SSSR **32** (1941), 19–21.

[52] ———, *The local structure of turbulence in incompressible viscous fluids at very large reynolds numbers*, Doklady Akademii Nauk SSSR **30** (1941), 299–303.

[53] W. Layton, L. G. Rebholz, and C. Trenchea, *Modular nonlinear filter stabilization of methods for higher reynolds numbers flow*, Journal of Mathematical Fluid Mechanics **14** (2012), 325–354.

[54] L.Bertagna, M.D'Elia, M.Perego, and A.Veneziani, *Data assimilation in computational hemodynamics*, Fluid-Structure Interaction and Biomedical Applications (T. Bodnár, G. P. Galdi, and Š. Nečasová, eds.), Birkäuser, 2014.

[55] M. Y. L. Lee, *Estimation of the error in the reduced-basis method solution of differential algebraic equations*, SIAM Journal on Numerical Analysis **28** (1991), 512–528.

[56] J. Leray, *Essai sur le mouvement d'un fluide visqueux emplissant l'espace*, Acta Mathemathica **63** (1934), 193–248.

[57] M. Loéve, *Probability theory*, D. Van Nostrand Co., Inc., 1963.

[58] E. N. Lorenz, *Empirical orthogonal functions and statistical weather prediction*, http://eaps4.mit.edu/research/Lorenz/Empirical_Orthogonal_Functions_1956.pdf, 1956, Massachusetts Institute of Technology, Scientific Report No. 1.

[59] J. Mathew, R. Lechner, H. Foysi, J. Sesterhenn, and R. Friedrich, *An explicit filtering method for large eddy simulation of compressible flows*, Physics of Fluids **15** (2003), no. 8, 2279–2289.

[60] K. McLeod, A. Caiazzo, M. A. Fernández, T. Mansi, I. E. Vignon-Clementel, M. Sermesant, X. Pennec, Y. Boudjemline, and J. F. Gerbeau, *Atlas-based reduced models of blood flows for fast patient-specific simulations*, STACOM/CESC, 2010, pp. 95–104.

[61] L. Mirsky, *Symmetric gauge functions and unitarily invariant norms*, Quarterly Journal of Mathematics **11** (1960), 50–59.

[62] J. Mullen and P. Fischer, *Filtering techniques for complex geometry fluid flows*, Communications in Numerical Methods in Engineering **15** (1999), no. 1, 9–18.

[63] W. Nichols and M. O'Rourke, *Mc Donald's blood flow in arteries*, 3rd ed., Edward Arnold Ltd, London, 1990.

[64] F. Nobile and C. Vergara, *An effective fluid-structure interaction formulation for vascular dynamics by generalized Robin conditions*, SIAM Journal on Scientific Computing **30** (2008).

[65] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, 1999.

[66] M. A. Olshanskii and X. Xiong, *A connection between filter stabilization and eddy viscosity models*, Numerical Methods for Partial Differential Equations **29** (2013), no. 6, 2061–2080.

[67] T. Passerini, A. Quaini, U. Villa, A. Veneziani, and S. Canic, *Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels*, International Journal for Numerical Methods in Biomedical Engineering **29** (2013), no. 11, 1192–1213.

[68] A. T. Patera and G. Rozza, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*, http://augustine.mit.edu/methodology/methodology_book.htm, 2007.

[69] M. Perego, A. Veneziani, and C. Vergara, *A variational approach for estimating the compliance of the cardiovascular tissue: an inverse fluid-structure interaction problem*, SIAM Journal on Scientific Computing **33** (2011).

[70] K. Perktold and G. Rappitsch, *Matematical modeling of local arterial flow and vessel mechanics*, Computational Methods for Fluid Structure Interaction (J. Crolet and R. Ohayon, eds.), Longman Scientific & Technical, Harlow, UK, 1994.

[71] C. Peskin and D. McQueen, *A three-dimensional computational method for blood flow in the heart. i. immersed elastic fibers in a viscous incompressible fluid*, Journal of Computational Physics **82** (1989), 372–405.

[72] M. Piccinelli, L. Mirabella, T. Passerini, E. Haber, and A. Veneziani, *4d image-based cfd simulation of a compliant blood vessel*, Tech. Report TR-2010-27, Department of Mathematics and Computer Science, Emory University, 2010.

[73] G. Prouse, *On the motion of a viscous incompressible fluid in a tube with permeable and deformable wall*, Rendiconti della Accademia Nazionale dei Lincei, Classe di Scienze Mathematiche, Fisiche e Naturali, 1971.

[74] A. Quarteroni, L. Formaggia, and A. Veneziani, *Complex systems in biomedicine*, Springer, 2006.

[75] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical mathematics*, Springer Verlag, 2007.

[76] A. Quarteroni, F. Saleri, and A. Veneziani, *Analysis of the Yosida method for the incompressible Navier-Stokes equations*, Journal de Mathématiques Pures et Appliquées **78** (1999), 473–503.

[77] ———, *Factorization methods for the numerical approximation of Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering **188** (2000), 505–526.

[78] A. Quarteroni, M. Tuveri, and A. Veneziani, *Computational vascular fluid dynamics: problems, models and methods*, Computing and Visualization in Science **2** (2000), 163–197.

[79] A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*, Springer-Verlag, 1994.

[80] ———, *Domain decomposition methods for partial differential equations*, Oxford Science Publications, 1999.

[81] O. Reynolds, *An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous and of the law of resistance in parallel channels*, Philosophical Transactions of the Royal Society of London **35** (1883), 84–99.

[82] G. Rozza, D. B. P. Huynh, and A. Manzoni, *Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants*, Numerische Matematik **125** (2013), 115–152.

[83] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Computational Methods in Engineering **15** (2008).

[84] F. Saleri and A. Veneziani, *Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations*, SIAM Journal on Numerical Analysis **43** (2006), no. 1, 174–194.

[85] S. Salsa, *Partial differential equations in action. from modeling to theory*, Springer, 2009.

[86] E. Schmidt, *Zur theorie der linearen und nichtlinearen integralgleichungen*, Matematische Annalen **63** (1907), 433–476.

[87] J. Sokolowski and J. P. Zolesio, *Introduction to shape optimization*, Springer-Verlag, Berlin, 1992.

[88] S. F. C. Stewart, E. G. Paterson, G. W. Burgreen, P. Hariharan, M. Giarra, V. Reddy, S. W. Day, K. B. Manning, S. Deutsch, M. R. Berman, M. R. Myers, and R. A. Malinauskas, *Assessment of CFD performance in simulations of an idealized medical device: Results of FDA's first computational inter laboratory study*, Cardiovascular Engineering and Technology **3** (2012), no. 2, 139–160.

[89] S. Stolz and N. A. Adams, *An approximate deconvolution procedure for large-eddy simulation*, Physics of Fluids **11** (1999), no. 7, 1699–1701.

[90] S. Stolz, N. A. Adams, and L. Kleiser, *An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows*, Physics of Fluids **13** (2001), no. 4, 997–1015.

[91] G. Strang and G. J. Fix, *An analysis of the finite element method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[92] T. Sugimoto, S. Ueha, and K. Itoh, *Tissue hardness measurement using the radiation forced of focused ultrasound*, Proceedings of the 1990 IEEE Ultrasonics Symposium (1990).

[93] C. A. Taylor and M. T. Draney, *Experimental and computational methods in cardiovascular mathematics*, Annual Review of Fluid Mechanics **36** (2004), 197–231.

[94] R. Temam, *Navier-Stokes equations. theory and numerical analysis*, North-Holland, Amesterdam, 1986.

[95] H. Tennekes and J. L. Lumley, *A first course in turbulence*, MIT Press, Cambridge, 1972.

[96] R. Todling, *Estimation theory and foundations of atmospheric data assimilation*, Tech. Report DAO Office Note 1999-01, Goddard space flight center, 1999.

[97] E. J. Topol and R. M. Califf (eds.), *Textbook of cardiovascular medicine*, Lippincot-Raven, Philadelphia, 1998.

[98] A. Veneziani and C. Vergara, *Flow rate defective boundary conditions in haemodynamics simulations*, International Journal for Numerical Methods in Fluids (2005), 803–816.

[99] M. R. Visbal and D. P. Rizzetta, *Large eddy simulation on curvilinear grids using compact differencing and filtering schemes*, Journal of Fluids Engineering **124** (2002), 836–847.

[100] C.R. Vogel, *Computational methods for inverse problems*, SIAM, 2002.

[101] J. von Neumann, *Collected works*, Pergamon Press, New York, 1962.

[102] A. W. Vreman, *An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications*, Physics of Fluids **16** (2004), no. 10, 3670–3681.

[103] K. Yeleswarapu, *Evaluation of continuum models for characterizing the constitutive behavior of the blood*, Ph.D. thesis, University of Pittsburgh, 1996.

[104] T. Young, *A course of lectures on natural philosophy and the mechanical arts*, vol. 2, London: Printed for J. Johnson, 1807.