SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

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

Model order reduction for compressible turbulent flows: hybrid approaches in physics and geometry parametrization

Candidate:
Matteo Zancanaro

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

Academic Year 2020-21

"Il mio stile è vecchio
come la casa di Tiziano
a Pieve di Cadore."
F.B.

"Il falco pellegrino incombe sul vasto campo
della pagina stampata, nascondetevi care idee,
mie prede e mio nutrimento, non soltanto vi mangeró,
ma vi assimileró, futuro cibo della mia testa."
D.P.

"Alexander, Macedonum rex, discere geometriam
coeperat, infelix, sciturus, quam pusilla terra esset,
ex qua minimum occupaverat."
L.A.S.

Abstract

This thesis provides a dissertation about efficient and reliable methods developed to deal with fluid flows problems, discretized by the use of finite volume approaches. In general increasing complexity dynamics are taken into consideration and suited strategies are utilized to overcome arising hurdles. The basic idea behind this work is the construction of reduced order models capable of providing fully consistent solutions with respect to the high fidelity flow fields. Full order solutions are often obtained through the use of segregated solvers, employing slightly modified conservation laws so that they can be decoupled and then solved one at a time. Classical reduction architecture, on the contrary, rely on the Galerkin projection of a complete Navier-Stokes system to be projected all at once, causing a mild discrepancy with the high order solutions. In this thesis three different segregated reduced order algorithms are presented for the resolution of laminar, turbulent and compressible flows respectively.

Turbulent flows are frequently approached by the employment of Reynolds averaged Navier-Stokes equations. Since this set of equations is not self closed, an additional modeling is required for some terms related with turbulence. In particular in this thesis we will rely on eddy viscosity models. Since there are a variety of different turbulence models for the approximation of this supplementary viscosity, one of the aims of this work is to provide reduced order models which are independent on this selection. This goal is reached by the application of hybrid methods where Navier-Stokes equations are projected in a standard way while the viscosity field gets approximated by the use of data-driven interpolation methods or by the evaluation of a properly trained neural network. By exploiting the aforementioned expedients it is possible to resolve fluid flow problems characterized by high Reynolds numbers and elevated Mach numbers in a less costly and more general way.

Contents

List of Figures

1 Introduction

In the last decades fluid flows simulations have progressively enlarged their applicability and their influence in many different research fields. Nowadays Computational Fluid Dynamics (CFD) applications have reached widely spread ambits as, for example, shape optimization for naval/automotive/aerospace engineering, cardiovascular in real time diagnosis, chemistry industrial processes or weather forecasts ([6, 139, 20]).

Various choices are possible for a discrete approximation of the solution. Among them we can cite the Finite Element Method (FEM) [150], the Finite Difference Method (FDM) [97], the Spectral Element Method (SEM) [102] or the Finite Volume Method (FVM) [48]. In this work we opted for the last one.

While increasing the demand on reliability and usability of CFD, the computational capability of the employed hardware architectures is no more sufficient to cover the simulation demand in due time. Even though computational facilities are rapidly increasing their resources and power, the resolution cost for a Direct Numerical Simulation (DNS) is in many cases unaffordable due to the huge dimension of the discretized problem (see [92, 93, 47]). A possible choice to overcome this issue is an adaptive refinement of the mesh capable of providing a more accurate resolution of the fields for those areas into the domain where fully resolved dynamics are needed as for example in [67].

For this reason the search for new efficient methods, proficient in reducing computational time, keeps on cover a relevant amount of CFD research efforts. A popular research field, related to the aforementioned applications, is the analysis of the dynamics related to Parametrized Partial Differential Equations (PPDEs). In this case a possible infinite number of solutions is admissible for every slightly different value of the parameter. Some specific ambits require many realizations to be evaluated in order to find out the best performing one, in terms of prefixed requirements fulfillment. The cost of such a procedure would increase by order of magnitude with respect to the cost of a single solution, leading to the impossibility of relying on standard numerical discretization techniques.

Many different techniques have been taken into consideration to overtake this problem. Galerkin projection reduction method has widely been employed to develop new reduction strategies capable of exploiting the information of a certain number of full order solutions for different parameter values in order to perform efficient, accurate and computationally cheaper solutions for a different configuration of the problem. The main idea behind these approaches is the possibility

to disclose solution manifolds describable by a much smaller number of degrees of freedom with respect to the dimension of a high fidelity resolution. Those degrees of freedom represent the possible linear combinations of a certain amount of modal basis functions, composing the desired solution. The cited basis functions can be obtained in a variety of ways. The most employed one is probably the Proper Orthogonal Decomposition (POD) method, in which case we will refer to the projection technique as a POD-Galerkin strategy. Many interesting publications rely on it. We can cite [2, 9, 17, 28, 74, 121]. Other possible choices are the Proper Generalized Decomposition (PGD) [46, 34] or the Dynamic Mode Decomposition (DMD) [122, 75]. Another possible approach is the use of a Greedy algorithm for a hierarchical selection of needed basis, one at a time, selecting the most relevant solutions in the discretized parameter set [59, 135, 104]. Good overviews are reported in [112, 59, 15, 14, 16]. Some specific application of reduced order methods in a FV framework can be found in [54, 56, 57, 55, 81, 29].

These strategies usually have to face stability issues when applied to fluid flows problems. The fulfillment of the InfSup condition can be ensured by the enrichment of the velocity space with supremizer solutions as introduced in [118] and explained in [10], where this technique was firstly developed for FEM environments. The same approach has been extended to FV architecture in [129]. In [8] the authors propose a stabilization technique consisting on both an offline part and an online one to cure proper orthogonal decomposition reduced problems instabilities caused by advection dominated flows.

Classical Galerkin-projection ROMs only succeed on providing a good reliability for those problems characterized by solution manifolds expressible as a linear combination of modal basis functions. To overtake this problem in the last years also non-linear approaches have been analyzed, extending the applicability of ROMs to new problems [71, 151, 96, 69].

The aforementioned approaches can all be classified as intrusive methods. Also non-intrusive strategies have been developed for the same purpose [145, 147, 33]. A possible example is the use of Radial Basis Functions (RBF) interpolation [144] or some other types of interpolation procedures for the evaluation of POD expansion coefficients [95].

Lately a new research branch has risen in this sector, belonging to this category: machine learning. By the use of Neural Networks (NN), the resolution of parametric PDEs has surely become more accessible [60, 138, 91, 88].

Both the projection methods and machine learning techniques have some valu-

able aspects together with critical points to be analyzed. Projection techniques are strongly connected with the physical laws of the problem since they use modal basis functions obtained by real solutions to extract the main dynamics and they employ those modes to project and reconstruct conservation equations solution manifolds. Unfortunately non-linearity together with non-affinity of the parametric formulation can be difficult issues to be carefully treated by the use of specific approaches as, e.g., the Discrete Empirical Interpolation Method (DEIM) [31, 105, 44].

Moreover sometimes it may happen that the equations are not directly available and in that case projection methods are not employable. Again, in the literature one may find many examples of very slow decay of the eigenvalues for specific problems that leads to the requirement of many modal basis functions to reach a decent approximation for the solutions, wasting, de facto, the whole possible gain coming from such an architecture. On the contrary artificial intelligence techniques are very versatile. They just need a set of solutions to be trained and, no matter the complexity of the mathematical formulation of the starting problem, they can be modeled to provide good approximations in a short time. The dark side of these approaches is the fact that they have no direct connection with the real physics of what they are approximating and the actual meaning of their architecture is arguably comprehensible. For this reason they may give inaccurate results thanks to impossibility in having a deeper control on networks responses.

A typical example for slow decreasing eigenvalues is the one related to turbulence in fluid dynamics applications. Turbulence is mainly treated by the application of Large Eddy Simulation (LES) [53, 23, 106, 18] or Reynolds Average Navier-Stokes equations (RANS) [3, 32]. The scope of these techniques is to avoid the resolution of DNS turbulent problems that would require a too expensive computation for the discretized formulation. A solution for turbulent ROMs can be found in Variational Multi-Scale (VMS) with VMS-ROMs examples in [65, 127, 38]. Specific implementations for reduced order LES and RANS can be found in [134, 114] and [81, 123, 62] respectively. One of the main disadvantages of this projection techniques developed for turbulent flows is that they are usually set up to deal with one single turbulence closure model which causes a large limitation on the applicability (see e.g. [115]).

A furthermore step towards complexity in the fluid dynamics ROMs field is given by compressibility. The addition of compressibility leads to the necessity of equipping the Navier-Stokes system with new equations capable of supplying thermodynamics evolution to the problem. Many different architecture have been devel-

oped to deal with both subsonic and supersonic flows by counting on intrusive and non-intrusive methods. See for example [11, 152, 73],[146, 148] and [4, 142].

Also neural networks found wide usage for the resolution of compressible fluid flows [84, 83, 152].

For what concerns real applications and industrial interests, reduced order models can be a very useful tool in many different fields. In particular geometry parametric problems play a very important role for example in shape optimization procedures or for fluid-structure interaction studies or, again, for "in real time" estimations (see for example [5, 50, 120] and [66, 19]).

Taking all the aforementioned examinations under consideration, a good choice for a resolution strategy would be a mixed technique where good properties of both projection methods and data-driven or non-intrusive methods are retained. Some mixed approaches have already been employed. In [62] and [52], the authors use a Galerkin-projection technique for what concerns the Navier-Stokes equations while eddy viscosity related terms are reconstructed through an interpolation strategy based on Radial Basis Functions (RBF). [26] provides, on the contrary, an example of mixture between projection based methods for the reduction of the Navier-Stokes system and machine learning strategies for the propagation of time dependent coefficients. The authors in [113] propose a method for the resolution of incompressible Navier-Stokes equations where only the pressure equation gets projected by the usage of proper orthogonal decomposition basis while momentum equation is retained in its full order form: a neural network provide the information exchange between high fidelity and reduced order worlds.

Thesis contribution

This thesis focuses on efficient methods for the resolution of parametric fluid flows problems. In particular the goal is the development of accurate and reliable methods for fluid dynamics PPDEs in a reduced order model framework. The general idea is to propose an architecture proficient in dealing with different types of parametrizations in different flow regimes.

Chapter 2 introduces the meaning of parametric partial differential equations and reduced order models. More in details section 2.1 provides an introduction on high fidelity discretization methods for PPDEs while section 2.2 shows the approach we utilize for the finite volume resolution of the problems. Section 2.3

points out the reconstruction of the reduced solution manifold by the use of proper orthogonal decomposition techniques which is summarized into algorithm 1. Paragraph 2.4 closes the chapter by summing up the aforementioned contents.

In chapter 3 we give an overview on conservation laws describing laminar incompressible flows related dynamics or rather fluid flows where the ratio between the particles velocity and the speed of sound, i.e. the *Mach* number, is below 0.3 and the ratio between inertia forces against viscous forces, i.e. the *Reynolds* number, is below 10^4. Section 3.1 contains a detailed description of the needed equations together with all the required assumptions and provides the finite volume discretization to be used for the applications. The paragraph 3.2 is composed by an explanation about the real core of the chapter that is the reduced Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm we use to obtain ROM evaluations. The main features of this strategy are enclosed into algorithm 2. Section 3.3 gives an introduction on physical parametric problems together with a first test case where a back step problem with a variable viscosity is resolved. On the contrary in section 3.4 geometric parametrization is handled with a discussion on mesh motion and a description of the adopted proper orthogonal decomposition procedure, supported by two test cases related to the flow around aerofoils where the angle of attack and the surface of the body can be varied. Finally, section 3.5 draws conclusions on what has been treated in the chapter.

Since in general laminar problems are a very restricted set, chapter 4 deals with turbulent flows and turbulence closure models. More in details, section 4.1 supplies a description of turbulent phenomena showing how to treat turbulent flows by the usage of Reynolds averaged Navier-Stokes equations. Given that this set of equations require a closure model for turbulence, few different choices are presented. Section 4.2 explains the method we developed for the resolution of reduced order turbulent flows, the most relevant steps being collected into algorithm 3. Two different variants are possible for the reduced treatment of turbulence. The first one is based on radial basis functions interpolation and is shown in section 4.3 where a test case focused on a back step with a parametrized viscosity is used to prove its performances. The second choice is the employment of a neural network for the reconstruction of the reduced eddy viscosity and it is presented in section 4.4. This solution gets selected for more complex test cases: in this section it is cast, together with a different selection for the snapshots, for a geometrically parametrized back step problem where the slope of the step can be varied. Section 4.5 tries to analyze the goals of the chapter.

For flow problems characterized by a *Mach* number higher than 0.3, chapter 5

has been devoted to compressible Navier-Stokes equations. In a deeper analysis, section 5.1 deals with the needed conservation laws and their finite volume discretization. A step by step averaging procedure is reported taking advantage of both Reynolds and Favre strategies. Section 5.2 reports the approach utilized in this work for the resolution of reduced compressible turbulent flow problems, exploiting the points listed in algorithm 5. The last part of the chapter is dedicated to the exposure of two different test cases, both related with the flow around a NACA0012 aerofoil, where in section 5.3 a viscosity parametrization is considered while in section 5.4 a shape deformation problem is investigated. Section 5.5 is then employed to briefly summarize the arguments analyzed in this chapter.

The last chapter of this thesis, the number 6, presents a general overview on the global contribution of this work and provides some possible future improvements and extensions for the developed methods.

2 Reduced Order Models

2.1 Parametrized Partial Differential Equations and high fidelity solutions

Let us suppose to deal with a generic abstract problem expressed into its strong form as follows:

$$\boldsymbol{a}(t, \boldsymbol{v}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{\mu}), \tag{1}$$

where $\boldsymbol{v} \in \mathbb{V}$ is a generic variable belonging to the space $\mathbb{V}$ that does not require to be specified at this point, $\boldsymbol{\mu} \in \mathbb{P}$ is a generic parameter belonging to a regular enough parameter domain $\mathbb{P}$, t indicates the time dependency, $\boldsymbol{a} : \mathbb{V} \times \mathbb{P} \to \mathbb{R}^d$ is the form representing the problem under investigation while $\boldsymbol{f} : \mathbb{P} \to \mathbb{R}^d$ stands for the forcing right hand side term, being d the physical dimension of the problem.

The dependency of the problem on the parameter can be here both explicit in the formulation or implicit. For example the problem can be dependent on $\boldsymbol{\mu}$ because of a change on physical parameters describing the behaviour of the solution (see subsection 3.3, subsection 4.3, subsection 5.3) or the solution can be implicitly depend on the parameter because of a change of the domain of interest (see subsection 3.4,subsection 4.4 and subsection 5.4).

Clearly Equation 1 is here indicating the continuum formulation of the problem but this is not what we are interested on. The resolution of Equation 1 is not possible in a closed form in most of the cases. For this reason, a discrete approximation of the solution is required.

To this scope we introduce here the tessellation $\mathcal{T} = \{\Omega_i\}_1^{N_h}$ covering the whole domain of definition of the problem, namely Ω, by a certain number N_h of non-convex polyhedral cells Ω_i so that $\bigcup_{i=1}^{N_h} \Omega_i = \Omega$ while $\Omega_i \bigcap \Omega_j = \varnothing$ for every $i \neq j$. The discrete solution $\boldsymbol{v}_h$ evaluated on $\mathcal{T}$ can then be defined.

$\boldsymbol{v}_h$ can be obtained in many different ways. The most common and widespread techniques are the Finite Differences method (see [79]), probably the first strategy developed for Partial Differential Equations but still used and studied, and the Finite Element method (see [37]) that is for sure the most common and employed discretized resolution technique for many different ambits.

For what concerns this work, the discretization procedure we will employ is the

Finite Volume (FV) method. This technique has firstly been introduced in [89, 86] and is nowadays very used especially for fluid dynamics problems. Few distinctive features are to be underlined:

- volume integrals over each cell containing divergence terms are reshaped into surface integrals exploiting the divergence theorem;
- quantities of interest are considered to be constant over each cell, at least for classical applications (high order Finite Volume schemes are also applicable, see e.g. [36, 42]), assuming their evaluation at the cell-center. This means that the degrees of freedom of the discrete problem are strictly equal to the number of cells N_h of the tessellation $\mathcal{T}$;
- by considering surface integrals, fluxes play a relevant role: they are evaluated in such a way that the method is conservative by construction;
- all the previous considerations allow the usage of unstructured polyhedral meshes (see Figure 2) which becomes particularly efficient for difficult geometries.

For all these reasons FV methods increased their usage a lot in the last decades especially in all the fields where a strong conservation property is required, e.g. fluid dynamics simulations. The Finite Volume solutions are obtained, in this work, by the use of the open access FV library OpenFOAM ([99]). We will assume the $\boldsymbol{v}_h$ solutions to be high fidelity solutions in the sense that they are able to provide an approximation of $\boldsymbol{v}$ such that

$$||\boldsymbol{v} - \boldsymbol{v}_h||_{\mathbb{V}} < \tau \ , \tag{2}$$

for every possible τ provided that a sufficiently refined mesh $\mathcal{T}$ is given.

What is then required by the high fidelity problem is to find a solution $\boldsymbol{v}_h$ so that

$$\boldsymbol{a}(t, \boldsymbol{v}_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = \boldsymbol{f}_h(\boldsymbol{\mu}) \ , \tag{3}$$

where $\boldsymbol{v}_h \in \mathbb{V}_h \subset \mathbb{V}$, $\boldsymbol{f}_h$ is the discrete counterpart of $\boldsymbol{f}$, namely its evaluation for every cell-center, and Equation 2 is satisfied. We have:

$$\boldsymbol{v}_h \in \mathbb{R}^{N_h \times d} \qquad\qquad \boldsymbol{f}_h \in \mathbb{R}^{N_h} .$$

We will assume the truth solution problem to be well posed and the high fidelity method to be reliable enough to be able to satisfy Equation 3: the scope of this work is not focused on high fidelity discretization techniques. For this reason we will give for granted the high accuracy of $\boldsymbol{v}_h$ with respect to the real solution $\boldsymbol{v}$.

2.2 Semi-Implicit Method for Pressure Linked Equations

This work is devoted to the resolution of different fluid dynamics problems related to different thermodynamics conditions. The equations describing the physical behaviour of the systems we are interested on are the well known Navier-Stokes equations ([131, 35]). These conservation laws are here employed in different formulations because of the difference in the characteristics of the flows (see subsection 3.1, subsection 4.1 and subsection 5.1). For all these different applications, a similar strategy has been employed for what concerns the offline high fidelity algorithms: the FV methods employed here are different approaches for the same basic Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm ([101]).

Some aspects related to these methods are here underlined to justify such a choice:

- the NS equations are manipulated so that it is possible to segregate them. In this way they are no more coupled between each others and they are no more to be solved as a block system; they can be easily solved one by one in an iterative way until convergence is reached. This may become useful for big meshes because a storage of huge matrices can be in this way avoided by only collecting some much smaller ones;
- by the usage of segregated approaches, the saddle-point formulation is somehow circumvented in the sense that the Ladyzhenskaya–Babuška–Brezzi condition [21, 43] is not strictly required anymore since no coupled problems are here to be solved. This aspect is strongly relevant especially for the reduced order part since no stabilization is needed (for a deeper understanding compare [130] and [10]);
- by the use of a basic under-relaxation these methods show good stability and reliability properties.

Two main classes of SIMPLE algorithms are used in the literature:

- pressure based algorithms, mostly used for incompressible and low compressibility flows;
- density based algorithms, mostly used for high compressibility flows.

The pressure based algorithms rely on the resolution of a suited pressure equation, obtained from a combination between the continuity equation together with the momentum equation, while density is explicitly calculated through the state equation. For what concerns the density based algorithms they are constructed exactly the other way around and they rely on a suited equation for density while

pressure is derived from the state equation.

In this thesis only pressure based algorithms are taken into consideration since we will not be dealing with highly compressible flows: no discontinuities will be here analyzed and no shock waves will be studied in the following paragraphs.

2.3 Proper Orthogonal Decomposition

The background idea leading to reduction techniques is the possibility of using a certain amount of high fidelity solutions to get as much information as possible, so that a decrease in computational cost will take place for some eventual solutions coming next: a compression strategy is needed, able to retain only the essential basic dynamics from the whole content provided by the high fidelity solutions. A remark on this last sentence has to be pointed out here: such a procedure is not always applicable. These techniques, for some reasons that will result more comprehensible at the end of this paragraph, fail when the solutions manifold can not be approximated as a linear combination of some basis. We will suppose these technique to be effective for the problem we are interested on.

We can define here $\mathbb{P}_h \subset \mathbb{P}$ as a finite dimensional parameter set of dimension $N_{\boldsymbol{\mu}}$. It can be extracted from $\mathbb{P}$ in many different distributions. The easiest one is a random picking from the original continuum set but shrewder strategy are possible, depending on the problem of interest. Equation 3 can then be solved for every $\boldsymbol{\mu} \in \mathbb{P}_h$. We have:

$$\boldsymbol{s}_i = \boldsymbol{v}_h(\boldsymbol{\mu}_i) \quad \text{for } i = 1, \dots, N_{\boldsymbol{\mu}},$$

$$\boldsymbol{S} = \left\{ \boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_{\boldsymbol{\mu}}} \right\} = \begin{bmatrix} s_{1_1} & s_{2_1} & \dots & s_{N_{\boldsymbol{\mu}_1}} \\ \vdots & \vdots & \vdots & \vdots \\ s_{1_{d\,N^h}} & s_{2_{d\,N^h}} & \dots & s_{N_{\boldsymbol{\mu}\,d\,N^h}} \end{bmatrix} \in \mathbb{R}^{d\,N_h \times N_{\boldsymbol{\mu}}},$$

where $\boldsymbol{s}_i$ stands for the i-th snapshot or rather the solution of Equation 3 for $\boldsymbol{\mu}_i$ while $\boldsymbol{S}$, namely the snapshots matrix, is the matrix collecting all those solutions together.

The goal for a reduction method is the possibility to obtain a solution $\boldsymbol{v}_r$ in such a way that the distance between it and its high fidelity counterpart is minimized in a predefined norm, where $\boldsymbol{v}_r \in \mathbb{V}_r \subset \mathbb{V}_h$ being $\mathbb{V}_r$ the reduced space. Nowadays a plenty of different methods are available for the construction of $\mathbb{V}_r$, e.g. the greedy algorithm ([109, 137, 103]), the Proper Generalized Decomposition ([34, 98]), the Proper Orthogonal Decomposition ([85, 125, 51]) or the Dynamic Mode

Decomposition ([108, 76, 122]). Some comparisons can be found in [14] and in [59].

The whole thing, for all the aformentioned methods, reduces to the search for the reduced space $\mathbb{V}_r$ as one of the possible subspaces of span$\{\boldsymbol{s}_i|\boldsymbol{\mu}_i \in \mathbb{P}_h\}$ (see Figure 1), capable of minimizing the quantity

$$\sqrt{\frac{1}{N_{\boldsymbol{\mu}}}\sum_{i=1}^{N_{\boldsymbol{\mu}}}||\boldsymbol{s}_i - \boldsymbol{v}_r(\boldsymbol{\mu}_i)||^2_{\mathbb{V}}}. \tag{4}$$

By following what is prescribed for an SVD procedure we have:

$$\boldsymbol{\mathcal{M}}^{\frac{1}{2}}\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}_{svd}\boldsymbol{W}^T \ ,$$

in which

- $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{d\,N_h \times d\,N_h}$ is the diagonal mass matrix (repeated for vector valued variables) containing, for FV methods, the volumes of every cell in $\mathcal{T}$:

$$\boldsymbol{\mathcal{M}}_{ii} = \text{vol}(\mathcal{T}_i);$$

- $\boldsymbol{\Lambda} \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times N_{\boldsymbol{\mu}}}$ is the diagonal matrix containing the eigenvalues λ_i:

$$\boldsymbol{\Lambda}_{svd_{ii}} = \lambda_i \ ;$$

- $\boldsymbol{U} \in \mathbb{R}^{d\,N_h \times d\,N_h}$ is an orthogonal matrix of left singular vectors;
- $\boldsymbol{\Lambda}_{svd} \in \mathbb{R}^{d\,N_h \times N_{\boldsymbol{\mu}}}$ is a diagonal matrix containing the singular values of $\boldsymbol{S}$;
- $\boldsymbol{W} \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times N_{\boldsymbol{\mu}}}$ is an orthogonal matrix of right singular vectors.

Columns of $\boldsymbol{U}$ can be used as basis functions for a linear construction of the space $\mathbb{V}_r$. What is not desirable in this procedure is the use of snapshots matrices characterized by a number of rows much higher than the number of columns since the decomposition would become too expensive. Usually this is the case since the number of offline solutions is much lower than the number of degrees of freedom of the high fidelity problem.

For this reason in this work we will focus on Proper Orthogonal Decomposition (POD) in its snapshots method only, since it is the one that best fits our application purposes (see [125]). In particular we will report in the following the procedure employed for steady state problems since the applications exposed in the next paragraph will be devoted to the resolution of steady state problems. The strong assumption we are taking here is $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x};\boldsymbol{\mu})$.

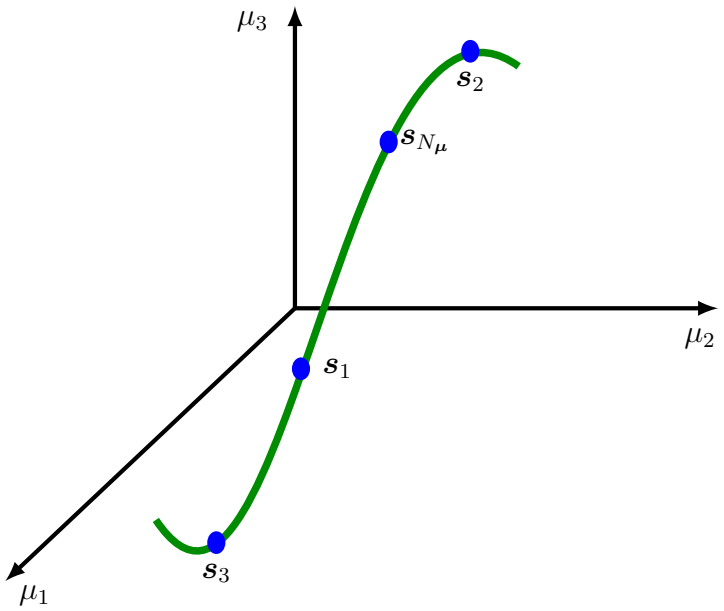


Figure 1: Disclosure of the hidden dynamics by the use of a Proper Orthogonal Decomposition where the green line represents the hidden space obtained as the $\text{span}\{\boldsymbol{s}(\boldsymbol{\mu}_i)|\boldsymbol{\mu}_i \in \mathbb{P}_h\}$ for $\dim(\boldsymbol{\mu}) = 3$.

A correlation matrix $\boldsymbol{C} \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times N_{\boldsymbol{\mu}}}$ can be constructed to tackle the correspondences between the snapshots into $\boldsymbol{S}$:

$$\boldsymbol{C}_{ij} = \boldsymbol{s}_i^T \boldsymbol{\mathcal{M}} \boldsymbol{s}_j \ .$$

The main information content can be extracted from $\boldsymbol{C}$ by the use of an eigenproblem construction:

$$\boldsymbol{C}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda} \ ,$$

where $\boldsymbol{V} \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times N_{\boldsymbol{\mu}}}$ is the matrix containing the eigenvectors while $\Lambda \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times \boldsymbol{\mu}}$ is the diagonal matrix containing the eigenvalues λ_i so that $\Lambda_{ii} = \lambda_i$. The correlation matrix could also be constructed the other way around as $\boldsymbol{C}_{ij} = \boldsymbol{s}_i \boldsymbol{\mathcal{M}} \boldsymbol{s}_j^T$, what would have resulted in a much bigger eigenvalue problem. The choice is due to the necessity of compressing the information as much as possible, following what the snapshots method prescribes.

A set of basis functions for the reconstruction of the reduced space $\mathbb{V}_r$ can be obtained as a combinations of the snapshots in the following way:

$$\boldsymbol{\xi}_i(\boldsymbol{x}) = \frac{1}{N_{\boldsymbol{\mu}} \lambda_i} \sum_{j=1}^{N_{\boldsymbol{\mu}}} \boldsymbol{V}_{ji} \boldsymbol{s}_j(\boldsymbol{x}) \quad \text{for } i = 1, \dots, N_{\boldsymbol{\mu}} \ .$$

The basis functions $\boldsymbol{\xi}_i \in \mathbb{R}^{d\, N_h}$ obtained in this way are ordered by following an "energy content" principle: the first eigenvectors are related to the biggest eigenvalues and they are, thus, the ones containing the highest information content, according to the chosen norm, L^2 in our case, as well as the most relevant dynamics of the system. For this reason the use of all the modes is not mandatory: a subset can be selected to recover the reduced space $\mathbb{V}_r$. This strategy clearly is neglecting part of the compressed information obtained from the snapshots $\boldsymbol{S}$ but, on the other hand, it is focusing all the reduced solutions on the most relevant dynamics. The number N_r of selected modes has to be properly chosen and it will be based on the required accuracy, the eigenvalues decay and many more different aspects depending on the application case.

The final reduced space can then be expressed as:

$$\mathbb{V}_r = \text{span}\{\boldsymbol{\xi}_i | i = 1, \dots, N_r\} \ .$$

It can be proven that the quantity (4) reaches its minimum when $\boldsymbol{v}_r \in \mathbb{V}_r$ is considered, being $\mathbb{V}_r$ constructed as exposed in this paragraph ([25, 124]).

The basis functions $\boldsymbol{\xi}_i = \boldsymbol{\xi}_i(\boldsymbol{x})$ are only dependent on the position $\boldsymbol{x}$: they are calculated once and for all.

The variable $\boldsymbol{v}_r \in \mathbb{V}_r$ can then be expressed as a linear combination of those basis functions. We have:

$$\boldsymbol{v}_r = \sum_{i=1}^{N_r} \alpha_i(\boldsymbol{\mu})\boldsymbol{\xi}_i(\boldsymbol{x}) = \boldsymbol{\Xi}(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{\mu}) \ , \tag{5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{N_r}$ is the vector containing the coefficients for the linear combination, depending on the parameter $\boldsymbol{\mu}$ only, while $\boldsymbol{\Xi} \in \mathbb{R}^{d\,N_h \times N_r}$ is the matrix containing the selected basis functions, depending on the position $\boldsymbol{x}$ only. Once the POD procedure has been performed, the reduced problem reduces to finding the coefficients $\boldsymbol{\alpha}$ so that the quantity $||\boldsymbol{v}_h(\boldsymbol{\mu}) - \boldsymbol{v}_r(\boldsymbol{\mu})||_{\mathbb{V}}$ is minimum.

In this work for the evaluation of the reduced solutions, i.e. the coefficients of the linear expansion, is performed by the use of Galerkin projection techniques (see subsection 3.2, subsection 4.2, subsection 5.2). In particular, reduced problems are treated by the use of the open access library ITHACA-FV ([126]).

Algorithm 1 The Proper Orthogonal Decomposition algorithm

Input: snapshots matrix $\boldsymbol{S} \in \mathbb{R}^{d\,N_h \times N_{\boldsymbol{\mu}}}$
Output: basis functions matrix $\boldsymbol{\Xi}(\boldsymbol{x})$ containing the modal basis $\boldsymbol{\xi}(\boldsymbol{x})$

1: construct the correlation matrix for the snapshots:

$$\boldsymbol{C}_{ij} = \boldsymbol{s}_i^T \boldsymbol{\mathcal{M}} \boldsymbol{s}_j \ ,$$

or equivalently

$$\boldsymbol{C} = \boldsymbol{S}^T \boldsymbol{\mathcal{M}} \boldsymbol{S} \ ,$$

where $\boldsymbol{s}$ stands for a single snapshot, i.e. a column of $\boldsymbol{S}$, and $\mathcal{M}$ is the diagonal mass matrix containing the volumes of all the cells belonging to $\mathcal{T}$;

2: solve the eigenvalue problem

$$\boldsymbol{C}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda}$$

where $\boldsymbol{V}$ is the matrix containing the eigenvectors while $\boldsymbol{\Lambda}$ is the diagonal matrix containing the eigenvalues λ_i;

3: construct the modal basis functions as a combinations of the snapshots contained in $\boldsymbol{S}$:

$$\boldsymbol{\xi}_i(\boldsymbol{x}) = \frac{1}{N_{\boldsymbol{\mu}} \lambda_i} \sum_{j=1}^{N_{\boldsymbol{\mu}}} \boldsymbol{V}_{ji} \boldsymbol{s}_j(\boldsymbol{x}) \quad \text{for } i = 1, \dots, N_{\boldsymbol{\mu}} \ ;$$

4: collect the first N_r basis functions as columns of the basis functions matrix $\boldsymbol{\Xi}$.

2.4 Context and overview

In this section we tried to give a general introduction to Model Order Reduction. The subsection 2.1 gives a brief description of what we refer to, when talking about parame-
trized Partial Differential Equations. The continuum formulation has been exposed and the discretization technique employed in this thesis has been defined, trying to give a justification for choosing a Finite Volume approach. A more specific survey on what a FV method consists of, is presented in subsection 3.1, subsection 4.1 and subsection 5.1 for laminar, turbulent and compressible Navier Stokes equations respectively.

In subsection 2.2 the choice of the general class of algorithms identified for the high fidelity solutions in this thesis is discussed, while subsection 3.2, subsection 4.2 and subsection 5.2 will contain more detailed statements on the specific algorithms employed for different physical problems.

Since we rely on a "solve-then-compress" strategy for the construction of the reduced order model, subsection 2.3 presents possible paths for the same scope and explains in details the steps for the implementation of a Proper Orthogonal Decomposition algorithm, resumed in Algorithm 1. A slightly different perspective will be given for POD procedure, employed for moving domains test cases, in subsubsection 3.4.2.

The reader interested on a deeper knowledge of possible choices for ROMs can have a look at [49, 58, 59, 14] for what concerns intrusive methods while an overview for non-intrusive approaches can be found in [145, 60, 33]. A very deep and widely complete description of the Finite Volume approach for high fidelity solutions of the Navier Stokes equations is contained in [94].

3 Incompressible laminar Navier-Stokes equations

In this section we will discuss fluid flow problems characterized by a low velocity and a motion where viscous forces overwhelm inertia forces. We will suppose to have:

$$M < 0.3 \,, \qquad Re < 10^4 \,,$$

where $M = \dfrac{\boldsymbol{u}_\infty}{C}$ stands for the *Mach* number [87], i.e. the ratio between the magnitude of the free stream velocity of the fluid $\boldsymbol{u}_\infty$ and the velocity of sound C while $Re = \dfrac{\boldsymbol{u}_\infty L}{\nu}$ stands for the *Reynolds* number [116], i.e. the ratio between inertia forces, given by the product between the magnitude of the free stream velocity and a characteristic length of the problem L, and viscous forces represented by the kinematic viscosity ν.

This kind of physical phenomena arise in many different fields, especially when dealing with strongly viscous media, for example when working with melted glass or molten metals.

Under the aforementioned conditions, some simplifications can be taken over the description of the fluid dynamics. The thermodynamics related to the problem under interest can be completely neglected since the interactions between the fluid particles as well as the energy dissipation are negligible.

A complete description of fluid motions can be found in [7, 100]. A fascinating historical fluid flows representation in pictures is contained in [136].

3.1 Equations and Finite Volume discretization

The incompressible laminar Navier-Stokes equations can be easily derived from the complete continuum conservation laws, written for both mass and momentum, in their differential forms. We will draw them in this paragraph while their Finite Volume discretization will be reported in the following.

CONTINUITY EQUATION:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \boldsymbol{u}] = 0 \,.$$

In case of incompressible flows and homogeneous density field, all the derivatives related to density vanish and the continuity constraint reduces to:

$$\nabla \cdot \boldsymbol{u} = 0 \,.$$

MOMENTUM EQUATION:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \mathbb{S}(\boldsymbol{u}) - \nabla p \, ,$$

where $\nabla \cdot \mathbb{S}(\boldsymbol{u})$ represents here the viscous force. If we introduce the strain rate tensor $\mathbb{D}(\boldsymbol{u})$, it can be defined as:

$$\mathbb{D}(\boldsymbol{u}) = \frac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2} \, .$$

It is then possible to write the viscous stress tensor as $\mathbb{S}(\boldsymbol{u}) = \mathbb{S}(\mathbb{D}(\boldsymbol{u}), \boldsymbol{u})$. In this work we will consider just newtonian fluids that means we are only considering fluids characterized by a linear constitutive law, i.e. we have a linear relation between the viscous stress tensor and the strain rate tensor. In particular we have:

$$\mathbb{S}(\boldsymbol{u}) = 2\mu \mathbb{D}(\boldsymbol{u}) + \lambda (\nabla \cdot \boldsymbol{u}) \mathbb{I}$$

where μ is the shear viscosity while λ is the first Lamè constant (for a clear and very complete treatment of this argument see [111]).

In the Stokes hypothesis (see [27]), a linear constant relation is present for what concerns the linking between shear and bulk viscosity, leading to:

$$\lambda - \frac{2}{3}\mu = 0 \Rightarrow \mathbb{S}(\boldsymbol{u}) = 2\mu \left[\mathbb{D}(\boldsymbol{u}) - \frac{1}{3} (\nabla \cdot \boldsymbol{u}) \mathbb{I} \right] \, .$$

We can substitute this last expression for the viscous stress tensor into the momentum equation. We obtain:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \left[\mu \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right) \right] - \frac{2}{3} \nabla \left(\mu \nabla \cdot \boldsymbol{u} \right) - \nabla p \, .$$

For incompressible flows the incompressibility constraint simplifies the bulk contribution to the viscosity leading to the following final expression for the momentum equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \left[\nu \left(\nabla \boldsymbol{u} \right) \right] - \nabla p \, ,$$

where $p = \dfrac{p}{\overline{\rho}}$ for sake of simplicity, indicating by $\overline{\rho}$ the constant and homogeneous density.

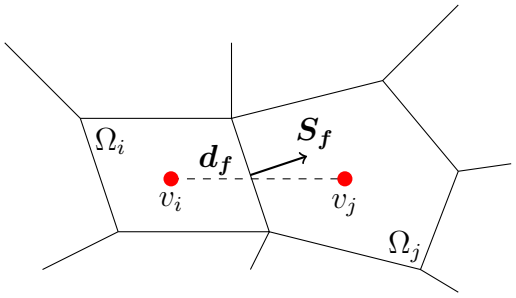


Figure 2: Scheme of the relation between two neighbor cells of the tessellation $\mathcal{T}$ for a certain variable v.

The Navier-Stokes system for incompressible laminar flows in the newtonian fluid hypothesis and Stokes relation then reads:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot [\nu \, (\nabla \boldsymbol{u})] + \nabla \overline{p} = 0 \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} . \tag{6}$$

In this chapter only steady-state fluid flows will be taken into consideration: Equation 6 can be rewritten considering $\boldsymbol{u}(t, \boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{u}(\boldsymbol{x}; \boldsymbol{\mu})$, $p(t, \boldsymbol{x}; \boldsymbol{\mu}) = p(\boldsymbol{x}; \boldsymbol{\mu})$ and they will be referred to as just $\boldsymbol{u}$ and p for sake of simplicity.

To approximate the problem by the use of the Finite Volume technique, the domain $\Omega(\boldsymbol{\mu})$ has to be divided into a tessellation $\mathcal{T}(\boldsymbol{\mu}) = \{\Omega_i(\boldsymbol{\mu})\}_1^{N_h}$ so that every cell Ω_i is a non-convex polyhedron and $\bigcup_{i=1}^{N_h} \Omega_i(\boldsymbol{\mu}) = \Omega(\boldsymbol{\mu})$. For sake of brevity, from now on, we will refer to $\Omega_i(\boldsymbol{\mu})$ as Ω_i.

Momentum equation can be rewritten into its integral form over each cell of the tessellation:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) \, dV + \int_{\Omega_i} \boldsymbol{\nabla} p \, dV - \int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[(\nu + \nu_t) \left(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T \right) \right] \, dV = 0 \, .$$

Every term of the previous expression will now be analyzed one by one to obtain the Finite-Volume formulation. The convective term can be treated by the use of the Gauss' theorem:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) \, dV = \int_{S_i} \boldsymbol{u} \otimes \boldsymbol{u} \cdot d\boldsymbol{S} \simeq \sum_f \boldsymbol{S}_f \cdot \boldsymbol{u}_f \otimes \boldsymbol{u}_f = \sum_f \boldsymbol{F}_f \boldsymbol{u}_f \, ,$$

where S_i is the total surface related to the cell i, $\boldsymbol{S}_f$ is the oriented surface dividing the two neighbor cells i and f, $\boldsymbol{u}_f$ is the velocity evaluated at the center of the face S_f and $\boldsymbol{F}_f = \boldsymbol{S}_f \cdot \boldsymbol{u}_f$ is the flux of the velocity through the face S_f (see Figure 2). Two considerations have to be underlined for this procedure. The first one is that $\boldsymbol{u}_f$ is not straight available in the sense that all the variables of the problem are evaluated at the center of the cells while here an evaluation for the velocity is required at the center of the face. Many different techniques are available to obtain it but the basic idea behind them all is that the face value is obtained by interpolating the values at the center of the cells. The second clarification is about fluxes: during an iterative process for the resolution of the equations, they are calculated by the use of the velocity obtained at previous step so that the non-linearity is easily resolved.

Two main choices are possible for what concerns the treatment of the pressure gradient term:

- by the use of a least square approach:
  1. pressure can be evaluated at a cell center i while it can be extrapolated to neighbouring cell centers j employing the gradient at i;
  2. values in j obtained by extrapolation can be compared with real values in j, since in the cell centers pressure value are always available, to construct an error estimation;
  3. by minimizing the sum of the square of weighted errors for all the cells surrounding the i-th cell while changing the gradient a good approximation for ∇p can be obtained.

  The first step is the evaluation of the tensor

$$\boldsymbol{G} = \sum_f \omega_f^2 \boldsymbol{d}_f \boldsymbol{d}_f \ ,$$

  being $\omega_j = \dfrac{1}{|\boldsymbol{d}_f|}$. We finally get:

$$(\nabla p)_i = \sum_f \omega_f^2 \boldsymbol{G}^{-1} \cdot \boldsymbol{d}_f (p_j - p_i) \ ,$$

- by the use of the Gauss gradient theorem:

$$\int_{\Omega_i} \boldsymbol{\nabla} p \, dV = \int_{S_i} p \, d\boldsymbol{S} \simeq \sum_f \boldsymbol{S}_f p_f \ ;$$

where p_f refers to pressure evaluated at the center of the face S_f. Once again these terms are obtained through an interpolation starting from pressure evaluation at the center of the cells. This second approach is the one that has been employed for this work.

The last term to deal with is the diffusive one:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[\nu \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right)\right] \, dV \simeq \nu_i \int_{\Omega_i} \boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) \, dV$$
$$= \nu_i \int_{\Omega_i} \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla u}) \, dV = (\nu)_i \int_{S_i} \boldsymbol{\nabla u} \cdot d\boldsymbol{S} \simeq \sum_f \left[\nu_f \left(\boldsymbol{\nabla u}\right)_f\right] \cdot \boldsymbol{S}_f \, ,$$

where ν_i stands for the kinematic viscosity in the i-th cell, ν_f indicates the effective viscosity at the center of the face S_f and $(\boldsymbol{\nabla u})_f$ corresponds to the gradient of the velocity located at the center of the face S_f. In case of orthogonal meshes, we can easily "transport" the needed gradient of the velocity from the cells centers to the faces centers. This is what is required:

$$\boldsymbol{S}_f \cdot (\boldsymbol{\nabla u})_f \simeq |\boldsymbol{S}_f| \frac{\boldsymbol{u}_i - \boldsymbol{u}_j^f}{|\boldsymbol{d}_f|} \, ,$$

where $\boldsymbol{d}_f$ is the vector connecting the centers of cells i and j and $\boldsymbol{u}_j^f$ is the velocity evaluated at the center of the cell j divided from the cell i by the face f. When the mesh is not orthogonal (see Figure 2), the last expression has to be corrected:

$$\boldsymbol{S}_f \cdot (\boldsymbol{\nabla u})_f \simeq |\boldsymbol{\pi}_f| \frac{\boldsymbol{u}_i - \boldsymbol{u}_j^f}{|\boldsymbol{d}_f|} + \boldsymbol{\omega}_f \cdot (\boldsymbol{\nabla u})_f \, ,$$

where $\boldsymbol{S}_f$ has been decomposed into a parallel and an orthogonal component with respect to $\boldsymbol{d}_f$, namely $\boldsymbol{\pi}_f$ and $\boldsymbol{\omega}_f$ respectively. The term $(\boldsymbol{\nabla u})_f$ can be then approximated by the use of an interpolation between $(\boldsymbol{\nabla u})_i$ and $(\boldsymbol{\nabla u})_j^f$ at the centers of the surrounding cells.

The complete momentum equation then reads:

$$\sum_i^{N_h} \left[\sum_f^{N_h} \boldsymbol{F}_f \boldsymbol{u}_f + \sum_f^{N_f} \boldsymbol{S}_f p_f - \sum_f^{N_f} (\nu + \nu_t)_f \left[|\boldsymbol{\pi}_f| \frac{\boldsymbol{u}_i - \boldsymbol{u}_j^f}{|\boldsymbol{d}_f|} + \boldsymbol{\omega}_f \cdot (\boldsymbol{\nabla u})_f \right] \right] = 0 \, , \quad (7)$$

where N_f is the number of faces related to the cell i. By exploiting the interpolation to faces centers it is possible to write down the whole equations into their matrix form:

$$\begin{bmatrix} \boldsymbol{A}_u & \boldsymbol{B}_p \\ \boldsymbol{\nabla}(\cdot) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h \\ \boldsymbol{p}_h \end{bmatrix} = \boldsymbol{0} \, , \tag{8}$$

indicating by $\boldsymbol{A}_u$ the matrix containing the terms related to velocity for the discretized momentum equation, by $\boldsymbol{B}_p$ the matrix containing the terms related to pressure for the same equation and by $\boldsymbol{\nabla}(\cdot)$ the matrix representing the incompressibility constraint operator, being $\boldsymbol{u}_h$ the vector where all the $\boldsymbol{u}_i$ variables are collected and $\boldsymbol{p}_h$ the vector where all the p_i are stored, having $\boldsymbol{u}_h \in \mathbb{U}_h \subset \mathbb{R}^{d\, N_h}$ and $\boldsymbol{p}_h \in \mathbb{Q}_h \subset \mathbb{R}^{N_h}$ with d spacial dimension of the problem. Some deeper explanations on Finite Volume discretizations for incompressible Navier-Stokes equations can be found in [72, 22, 94].

3.1.1 Incompressible pressure equation

In this work, for what concerns the offline phase, a segregated pressure-based approach has been selected. All the formulations will then rely on this formulations for what concerns the high fidellity solutions. In particular, the *Semi-Implicit Method for Pressure-Linked Equations* (SIMPLE) algorithm is here employed in its laminar form.

To better understand the procedure, let us report here the crucial points about this algorithm, they will be very useful later during the description of the ROM technique in this thesis.

First of all we can divide the operator related to velocity into a diagonal and an extra-diagonal parts so that

$$\boldsymbol{A}_u \boldsymbol{u}_h = \boldsymbol{A}\boldsymbol{u}_h - \boldsymbol{H}(\boldsymbol{u}_h) \, .$$

After that, recalling Equation 8, we can reshape the momentum equation as follows:

$$\boldsymbol{A}\boldsymbol{u}_h = \boldsymbol{H}(\boldsymbol{u}_h) - \boldsymbol{B}_p \boldsymbol{p}_h \Rightarrow \boldsymbol{u}_h = \boldsymbol{A}^{-1} \left[\boldsymbol{H}(\boldsymbol{u}_h) - \boldsymbol{B}_p \boldsymbol{p}_h \right] \, .$$

In an iterative algorithm, we can express both velocity and pressure as their value at previous iteration plus a correction term:

$$\boldsymbol{u}_h = \boldsymbol{u}^* + \boldsymbol{u}' \qquad \boldsymbol{p}_h = \boldsymbol{p}^* + \boldsymbol{p}' \,,$$

where $\square^*$ terms are the old ones while $\square'$ are the correction terms. With some approximations for the mixed terms, the following relation holds:

$$\boldsymbol{u}_h = \boldsymbol{A}^{-1} \left[\boldsymbol{H}(\boldsymbol{u}^*) + \boldsymbol{H}(\boldsymbol{u}') - \boldsymbol{B}_p \boldsymbol{p}^* - \boldsymbol{B}_p \boldsymbol{p}' \right] \,.$$

Into the SIMPLE algorithm a big assumption is taken since the extra-diagonal term $\boldsymbol{H}(\boldsymbol{u}')$ is discarded and put to zero. Of course this makes the whole procedure no more consistent but on the counterpart it makes the resolution of the so-called *pressure correction step* much easier. We then get:

$$\boldsymbol{u}_h = \boldsymbol{A}^{-1} \left[\boldsymbol{H}(\boldsymbol{u}^*) - \boldsymbol{B}_p \boldsymbol{p}_h \right] \,. \tag{9}$$

If we now apply the divergence operator to both sides of Equation 9, we end up with a Poisson equation for pressure by exploiting the incompressibility constraint:

$$\left[\boldsymbol{\nabla}(\cdot)\right] \boldsymbol{u}_h = \left[\boldsymbol{\nabla}(\cdot)\right] \left\{ \boldsymbol{A}^{-1} \left[\boldsymbol{H}(\boldsymbol{u}^*) - \boldsymbol{B}_p \boldsymbol{p}_h \right] \right\} \Rightarrow \left[\boldsymbol{\nabla}(\cdot)\right] \boldsymbol{A}^{-1} \boldsymbol{B}_p \boldsymbol{p}_h = \left[\boldsymbol{\nabla}(\cdot)\right] \boldsymbol{A}^{-1} \boldsymbol{H}(\boldsymbol{u}^*) \,. \tag{10}$$

3.2 Reduced order algorithm for incompressible laminar flows

As explained in subsection 2.2 and pointed out again in the previous paragraph, all the high fidelity solutions are obtained by the employment of a segregated algorithm iterating Equation 7 and Equation 10 until convergence is reached. In subsection 2.3 we anticipated the purpose to rely on Galerkin projection for the construction of the reduced order method we are looking for.

The first idea one may have for such a scope is to construct the complete coupled discretized Navier-Stokes system (8) as the discrete steady state counterpart of Equation 6. Once the system is assembled it can be projected. For such a scope let us introduce here the reduced expansions of the velocity and pressure fields respectively:

$$\boldsymbol{u}_r = \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu}) \boldsymbol{\psi}_i(\boldsymbol{x}) = \boldsymbol{\Psi} \boldsymbol{a} \qquad p_r = \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu}) \varphi_i(\boldsymbol{x}) = \boldsymbol{\Phi} \boldsymbol{b} \,,$$

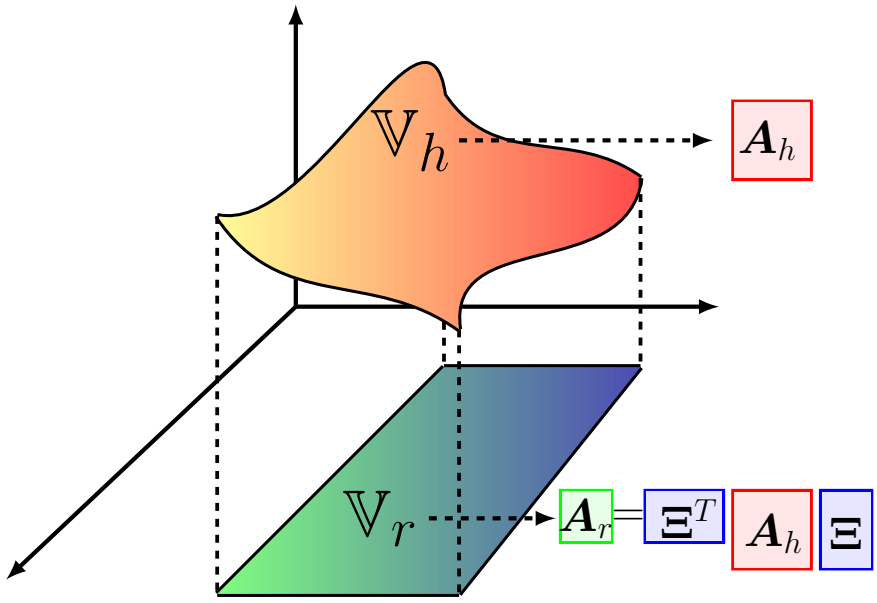


Figure 3: Projection of the generic full order space $\mathbb{V}_h$ over the reduced one $\mathbb{V}_r$ spanned by the generic basis functions Ξ where $\boldsymbol{A}_h$ and $\boldsymbol{A}_r$ are the full order matrix and its reduced order counterpart related to the considered problem respectively.

where N_u and N_p are the numbers of basis functions selected for the reconstruction of velocity and pressure solutions respectively, $\boldsymbol{a} \in \mathbb{R}^{N_u}$ is the vector containing the coefficients for the velocity expansion while the same reads for pressure with respect to $\boldsymbol{b} \in \mathbb{R}^{N_p}$ being $\boldsymbol{\psi}_i$ and φ_i the corresponding vector and scalar basis functions collected into the matrices $\boldsymbol{\Psi} \in \mathbb{R}^{d\,N_h \times N_u}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{N_h \times N_p}$ respectively.

This procedure would lead to a new matrix $\boldsymbol{A}_r \in \mathbb{R}^{(N_u+N_p)\times(N_u+N_p)}$ obtained by projecting the momentum equation over the subspace created by spanning the basis functions $\boldsymbol{\psi}_i$ and the continuity constrain over the subspace created by spanning the basis functions φ_i so that the reduced problem we would obtain is:

$$\boldsymbol{A}_r \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \boldsymbol{F} \,, \tag{11}$$

where $\boldsymbol{F}$ stands for a possible appearing forcing term.

It is clear that the resulting system can be solved by the use of, e.g., a Newton method or any other kind of non-linear iterative solver. It has been done in many

different works, see for example [128, 61]. The perspective in this work is slightly different: we would like to rely on a method capable of being as coherent as possible with respect to the high fidelity algorithm. Since the decoupling of the system to reshape Equation 6 into Equation 7 and Equation 10 is not fully consistent, the solution coming from the SIMPLE algorithm may not fit in the best way the dynamics of the reduced system (11).

For this reason some efforts have been spent to develop an algorithm that emulates the steps of the full order scheme. The main steps for the reduced method related to incompressible laminar flows are reported in Algorithm 2.

To reach the stability of the method it is necessary to under relax the variables, as pointed out in Algorithm 2, so that they are not fully updated at each step. For what concerns pressure we have:

$$p^{ur} = p^{\star} + \alpha_p(p^{\star\star} - p^{\star}) \,,$$

where α_p is the pressure under relaxation factor, $p^{\star}$ stands for the pressure at the previous iteration while $p^{\star\star}$ corresponds to the last evaluated one.Velocity is not under relaxed in the same way: for what concerns velocity, the momentum equation gets modified in full before fluxes evaluation.The under relaxed momentum equation reads:

$$\frac{1-\alpha_u}{\alpha_u}\boldsymbol{A}\boldsymbol{u}^{ur} + \boldsymbol{A}\boldsymbol{u}^{ur} = \boldsymbol{H}(\boldsymbol{u}^{\star}) - \boldsymbol{\nabla}p^{\star} + \frac{1-\alpha_u}{\alpha_u}\boldsymbol{A}\boldsymbol{u}^{\star} \,,$$

where α_u stands for the velocity under relaxation factor, $\boldsymbol{u}^{ur}$ stands for the under-relaxed velocity while $\boldsymbol{u}^{\star}$ is the velocity evaluated at the previous step. Notice that, at convergence, $\boldsymbol{u}^{ur} = \boldsymbol{u}^{\star}$ and the two terms that have been added to under-relax the momentum equation cancel each others. We can reshape the last expression so that the new terms are collected into the matrices $\boldsymbol{A}$ and $\boldsymbol{H}$:

$$\boldsymbol{A}^{ur}\boldsymbol{u}^{ur} = \boldsymbol{H}^{ur}(\boldsymbol{u}^{\star}) - \boldsymbol{\nabla}p^{\star} \,, \tag{12}$$

being $\boldsymbol{A}^{ur}$ and $\boldsymbol{H}^{ur}$ the under relaxed matrices. The conservative fluxes into Algorithm 2 are then calculated extracting the face velocity $\boldsymbol{u}_f$ from cell center velocities $\boldsymbol{u}^{ur} = (\boldsymbol{A}^{ur})^{-1}\left[\boldsymbol{H}^{ur}(\boldsymbol{u}^{\star}) - \boldsymbol{\nabla}p^{\star}\right]$.

Under relaxation factors have to be carefully selected because of their crucial role. Guide lines to do that are:

$$\begin{aligned} 0 < \alpha_p < 1 \,, \\ 0 < \alpha_u < 1 \,, \\ \alpha_p \simeq \frac{1}{2}\alpha_u \,, \end{aligned}$$

Algorithm 2 The Reduced Order SIMPLE algorithm

Input: first attempt reduced pressure and velocity coefficients $\boldsymbol{b}^\star$ and $\boldsymbol{a}^\star$; modal basis functions matrices for pressure and velocity Φ and Ψ

Output: reduced pressure and velocity fields p_r and $\boldsymbol{u}_r$

1: From $\boldsymbol{b}^\star$ and $\boldsymbol{a}^\star$, reconstruct reduced fields $p^\star$ and $\boldsymbol{u}^\star$:

$$p^\star = \Phi \boldsymbol{b}^\star, \qquad \boldsymbol{u}^\star = \Psi \boldsymbol{a}^\star \ ;$$

2: Momentum predictor step : assemble the momentum equation, project and solve it to obtain a new reduced velocity coefficients $\boldsymbol{a}^{\star\star}$:

$$(\boldsymbol{\psi}_i, \boldsymbol{A}\left[\boldsymbol{\Psi}\boldsymbol{a}^{\star\star}\right] - \boldsymbol{H}(\boldsymbol{u}^\star) + \boldsymbol{\nabla} p^\star)_{L^2(\Omega)} = 0;$$

3: Reconstruct the new reduced velocity $\boldsymbol{u}^{\star\star}$ and calculate the off-diagonal component $\boldsymbol{H}(\boldsymbol{u}^{\star\star})$;

4: Pressure correction step: project pressure equation to get new reduced pressure coefficients $\boldsymbol{b}^{\star\star}$:

$$(\varphi_i, \boldsymbol{\nabla} \cdot [\boldsymbol{A}^{-1} \boldsymbol{\nabla} (\boldsymbol{\Phi} \boldsymbol{b}^{\star\star})] - \boldsymbol{\nabla} \cdot [\boldsymbol{A}^{-1} \boldsymbol{H}(\boldsymbol{u}^{\star\star})])_{L^2(\Omega)} = 0;$$

Then correct the velocity explicitly after having reconstructed the new pressure $p^{\star\star}$;

5: Relax the pressure field and the velocity equation with the prescribed under-relaxation factors α_p and α_u, respectively. The under-relaxed fields are called p^{ur} and $\boldsymbol{u}^{ur}$;

6: **if** convergence **then**

7: $\quad \boldsymbol{u}_r = \boldsymbol{u}^{ur}$ and $p^\star = p^{ur}$;

8: **else**

9: $\quad$ Assemble the conservative face fluxes F_f:

$$F_f = \boldsymbol{u}_f \cdot \boldsymbol{S}_f;$$

10: $\quad$ set $\boldsymbol{u}^\star = \boldsymbol{u}^{ur}$ and $p^\star = p^{ur}$;

11: $\quad$ iterate from step 1.

12: **end if**

$$\alpha_u + \alpha_p \simeq 1 \ .$$

In broad terms the last rules are the ones usually prescribed in the literature but they have to be trimmed every time in a case-specific way.

3.3 Incompressible laminar Navier-Stokes problems: physical parametrization

Many problems are subjected to different behaviours when changing some physical aspects characterizing the problem. For fluid dynamics applications, very classical examples are:

- variations on the *Reynolds* number;
- variations on the boundary conditions;
- variations on the forcing term;
- a combination of the previous ones.

These changes may induce very different dynamics to rise or to get extinguished. The wider is the range where the parameters describing the problem can be selected, the higher is the number of offline solutions required to get enough information, the larger is the amount of modal basis functions needed to obtain a satisfactory enough coverage of the solution manifold. Moreover, as explained in subsection 2.3, reduction techniques can be employed only for problems characterized by a solution manifold describable as a linear combination of some modal basis functions. When the range of solutions is too wide this is not the case and a single set of basis functions would not be able to describe all the involved dynamics.

It is clear that the changes introduced into the system must be not large enough to cause a mutation on the regime we are working at. In particular a steep increase of the *Reynolds* number would require the introduction of turbulence treatments, as explained in section 4. On the contrary, a significant increase in, e.g., inlet velocity may lead to the necessity for a deeper thermodynamics treatment, as explained in section 5.

In this section we will present a test case where the reduction techniques explained in section 2 are employed. In particular the resolution of a very classical back step test case will be shown, where the viscosity of the problem is parametrized.

3.3.1 Back step problem with a variable viscosity

This paragraph is devoted to the presentation of the results obtained for a reduced back step problem characterized by a variable viscosity. In particular we want to study an inner flow constituted by a fluid moving into a channel where a back step is present, giving rise to a recirculation zone.

With reference to Figure 4 we have total length of the duct $L = 7$, total height

$H = 1.7$, step height $h = 0.7$ and pipe length after the step $l = 5$. Fluid is flowing from left to right so that Γ_{in} is the inlet boundary while Γ_{out} stands for the outlet boundary.

The problem we want to solve is described by the following equations:

$$\begin{cases} \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot [\nu (\nabla \boldsymbol{u})] + \nabla p = 0 & \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \\ \boldsymbol{u} = [1, 0]^T & \text{on } \Gamma_{in} \\ \nu \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} - p\boldsymbol{n} = 0 & \text{on } \Gamma_{out} \end{cases} .$$

The kinematic viscosity ν is here parametrized so that $\nu \in [0.01, 1] \equiv \mathbb{P}$. This means that the *Reynolds* number is varying by two orders of magnitude:

$$Re = \frac{\boldsymbol{u}_{in} L}{\nu} \in [7, 700] .$$

It is sufficiently low to guarantee the absence of significant turbulent phenomena but still high enough to require the use of Navier-Stokes equations since the Stokes regime is not applicable for such a case.

Since for incompressible flows pressure does only have a meaning when taking into account its variations, in the solutions of such a problem it is only defined up to constants. For this reason pressure is set to zero in the top right corner of the domain so that pressure fields can be evaluated uniquely.

For the offline phase we selected 50 random viscosity values so that $\dim(\mathbb{P}_h) = N_{\boldsymbol{\mu}} = 50$. These solutions are then used for the POD procedure. The online phase is then performed for 50 new randomly selected parameter values.

In Figure 5 we reported the trends for the eigenvalues both in their cumulated and single forms. As one may notice the decay reported on the right is pretty fast. This means that a few modes are sufficient in order to obtain a good reconstruction of the solution. In particular, since the cumulate of the eigenvalues is somehow representing the retained amount of information, by analysing the plot on the left it is clear that the use of 10 modal basis functions is more than enough for both velocity and pressure. For this reason, the solutions we are reporting in this section have been obtained by the employment of 10 basis functions for both the variables.

Figure 6, Figure 7 and Figure 8 refers to the solution obtained for $\nu = 0.0196$ which means that the system is described by $Re \simeq 357$.

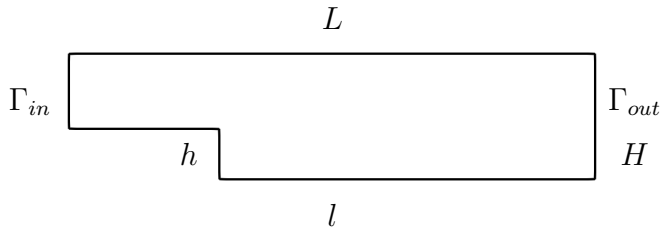


Figure 4: Representation of the geometrical configuration for the domain of interest.

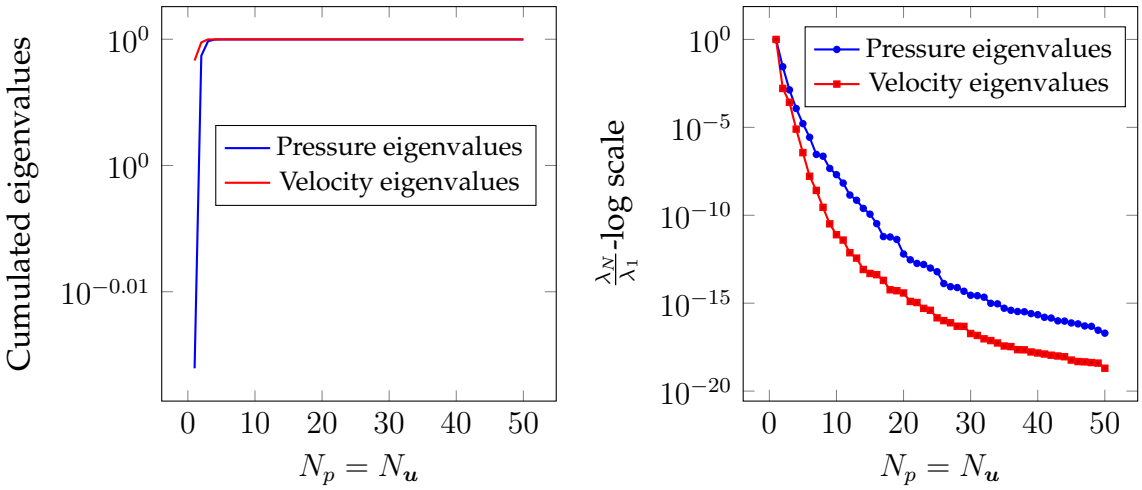


Figure 5: Eigenvalues trends for both pressure and velocity: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

Figure 6: Comparison between pressure solutions: high fidelity on the left, reduced order on the right.

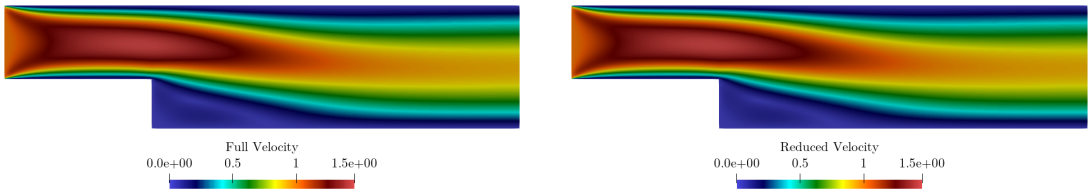


Figure 7: Comparison between velocity solutions: high fidelity on the left, reduced order on the right.

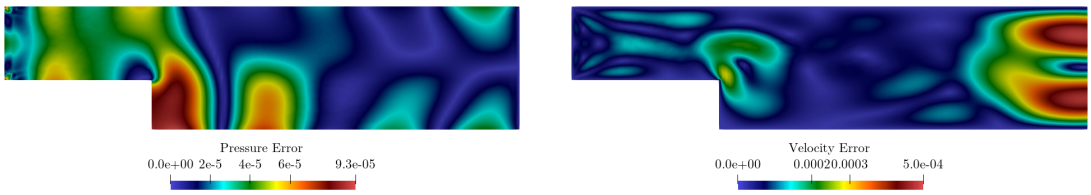


Figure 8: Pointwise error fields comparing full order and reduced order solutions for pressure on the left and velocity on the right.

Algorithm 2 shows to be reliable and efficient as it can be noticed comparing the online and the offline solutions. Figure 8 shows that for every point into the domain Ω the difference between the two solutions is very low while solutions provided in Figure 6 and Figure 7 shows good stability properties for both velocity and pressure.

Figure 9 provides here a comparison between two different resolutions strategies for the velocity field: what is here indicated by "coupled solution" refers to the resolution by the use of a block reduced order problem while "ROM SIMPLE" refers to the results obtained by the use of Algorithm 2.

The reduced block system strategy consists on assembling the whole coupled Finite Volume system which has to be projected over both velocity and pressure modal basis functions and solved through a Newton method. This is a very common choice for these problems as explained in subsection 3.2. Some examples can be found in [61, 128, 52].

The first aspect to be noticed is that the solutions obtained with the block system approach are always less accurate with respect to the solutions obtained through

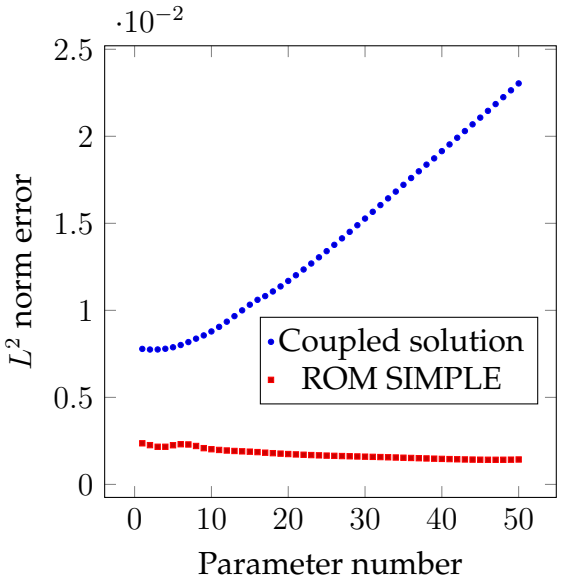


Figure 9: L^2 relative velocity error for all the parametric online solutions obtained by the use of a block approach compared with the segregated one we propose in this work.

Algorithm 2 and this can be addressed to the fact that the solutions obtained by utilizing a segregated approach are more compliant with the full order fields obtained by a SIMPLE algorithm.

A second aspect which may result of high relevance is the behaviour shown by the solutions obtained with a coupled approach: the error norm seems to increase when the *Reynolds* number gets increased. On the contrary, the reduced order segregated approach does not show any degradation from an accuracy point of view.

3.4 Incompressible laminar Navier-Stokes problems: geometrical parametrization

Geometrical parametrization test cases are nowadays playing an important role in the Model Order Reduction field. In fact the possibility to modify the geometry of the domain for a problem at a low cost is attractive for many different applications. In particular Reduced Order Models techniques can result very efficient for shape optimization processes as well as patient specific studies in both circulatory and respiratory systems or fluid structure interaction cases. A brief overview for this kind of problems may be found in [119].

Different techniques have been developed for the resolution of these applications both in a Finite Element environment (see [50, 117]) and in a Finite Volume framework (see [130, 70]).

In this section we will present the method we developed for the resolution of geometrical parametrization test cases when the fluid flow may be considered steady state, laminar and incompressible. Clearly this is quite an inceptive application but more difficult and intriguing studies will be presented in subsection 4.4 and subsection 5.4.

3.4.1 Mesh motion

When dealing with geometrical parametrization problems, the first aspect one has to take into consideration is the motion of the mesh. This is of course a necessity, no matter what the discretization technique is. In any case, when considering a Finite Volume scheme, some peculiar points have to be underlined to better performing in the motion of the grid:

- in a Finite Volume perspective, the possibility to use grid elements characterized by a random shape and a variable number of edged is a powerful tool. The drawback related to this choice is that it becomes very difficult to map the elements of the grid to a reference mesh;
- the mapping back to reference elements procedure would require the construction of a very complex non-linear map but the real issue entailed by this technique is that it would also lead to a change in the behaviour of the dynamical system (see [45]);
- Finite Volume algorithms do not rely on reference elements for the evaluation of the integrals. Thus the implementation of such a technique would require a computational cost that could be easily avoided in case it was possible to take into consideration the problem directly into its physical domain.

The previous arguments may suggest that a remeshing of the entire domain could be the right choice for our purpose. Of course this is not the case for three main reasons:

- the grid construction is a very costly procedure. Moreover a well designed mesh is not easy to be put together; many different requirements have to be fulfilled at the same time to obtain satisfactory results;
- the comparison between different solutions needed for a Proper Orthogonal Decomposition (see subsection 2.3 and subsubsection 3.4.2) can be performed only for meshes with a common topology so that the different solutions have all the same dimensions and, in addition, points characterized by the same indexes refer to the same cells;
- for what we have previously said, a mapping of the real domain to a reference one does not fit our requirements. For this reason we need to evaluate all the functionals in the real physical domain at both the offline and the online phase. This is a crucial point since a very efficient mesh motion strategy is required to obtain some kind of gain during the online phase. A bad choice for the reshaping of the grid would lead to highly increased reduced order computational time.

To fulfil these requirements two different strategies have been tested:

1. resolution of a Laplace problem on displacements with variable diffusivity;
2. application of a Radial Basis Functions (RBF) interpolation.

Laplace problem for displacements

In this case the mesh is moved thanks to the resolution of a Laplace problem over the whole grid by using the displacements δ of the points of the tessellation as the variable of the problem and the known fixed displacements $\overline{\delta}^b$ of the boundaries as boundary conditions:

$$\begin{cases} div(\gamma \nabla \delta) = 0 & \text{in } \Omega \\ \delta = \overline{\delta}^b & \text{on } \Gamma_D \end{cases},$$

where Γ_D refers to the moving boundaries while γ is the variable diffusivity. Its value can be selected in many different ways, for example as a constant field over the whole domain Ω. For our scope we decided to fix

$$\gamma = \frac{1}{\boldsymbol{r}^2},$$

being $\boldsymbol{r}$ the distance from the moving boundary, so that the points into the grid positioned far away from the moving boundaries are not affected that much from the changing in the shape of the domain.

This approach is valuable in the sense that it provides a continuous smooth solution able to preserve the original properties of the tessellation. Moreover, grids modified in this way have all the same topology so that the meshing process has only to be performed once.

Clearly the Laplace problem has to be discretized and the resolution of its discrete counterpart has to be performed. After the discretization we end up with

$$\begin{cases} \boldsymbol{A}_\delta \boldsymbol{\delta} = \boldsymbol{0} & \text{in } \Omega \\ \boldsymbol{\delta} = \overline{\boldsymbol{\delta}}^b & \text{on } \Gamma_D \end{cases} ,$$

where $\boldsymbol{A}_\delta$ represents the matrix obtained by discretizing the Laplace operator while $\boldsymbol{\delta}$ and $\overline{\boldsymbol{\delta}}^b$ are the discrete counterparts of δ and $\overline{\delta}^b$ respectively. Matrix $\boldsymbol{A}_\delta \in \mathbb{R}^{d\,N_h \times d\,N_h}$ results to be sparse; many different efficient algorithms are present in the literature for these class of matrices.

Radial Basis Functions interpolation of the displacements

The second approach rely on the idea of extracting the displacements for the internal points of the grid as the evaluation of an interpolant of the moving boundaries displacements based on Radial Basis Functions (see [41]). The general formula for the evaluation of the displacements of the grid reads:

$$\delta(\boldsymbol{x}) = \sum_{i=0}^{N_b} \omega_i \phi\left(\|\boldsymbol{x} - \boldsymbol{x}_i^b\|\right) + q(\boldsymbol{x}) \,, \tag{13}$$

where $\delta(\boldsymbol{x})$ is the displacement of the grid node positioned in $\boldsymbol{x}$, N_b is the number of selected control points on the moving boundary, ω_i are some calculated weights, ϕ is a fixed function whose support is a round area of predetermined radius r, $\boldsymbol{x}_i^b$ are the coordinates of the control points and $q(\boldsymbol{x})$ is a polynomial.

Equation 13 is scalar valued and provides only distances, no matter what the direction is. For this reason this scheme has to be applied d times for every point, considering the $\boldsymbol{x}$ components one by one.

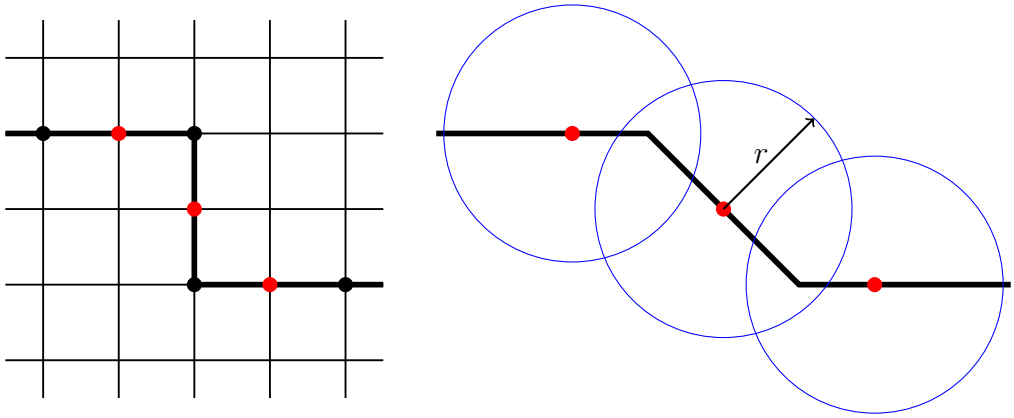


Figure 10: Scheme of the RBF mesh motion procedure: original mesh on the left, deformed boundary on the right where red dots are representing the control points while blue circles show the support of the function φ.

Under certain assumptions (see [13]), $q(\boldsymbol{x})$ can be chosen to be linear so that rigid translations are preserved. Thus we have:

$$q(\boldsymbol{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 \ ,$$

where $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ are scalar coefficients while $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$.

The procedure can be summarized in the following steps:

1. select the control points into the boundaries to be moved and shift their position obeying the fixed motion rule selected for the geometry modification, accordingly with the parameter dependent displace law: they can be either all the points into the boundary or just a fraction of their total amount if the dimension of the mesh is big enough (see Figure 10), since the higher is the number of control points, the bigger (and then expensive) is the resulting RBF linear problem to be solved;
2. calculate all the parameters for the RBF to ensure the interpolation capability of the scheme:

$$||\boldsymbol{\delta}(\boldsymbol{x}_i^b)|| = ||\overline{\boldsymbol{\delta}}_i^b|| = \overline{\delta}_i^b \ ,$$
$$\sum_{i=0}^{N_b} \omega_i q(\boldsymbol{x}_i^b) = 0 \ ,$$

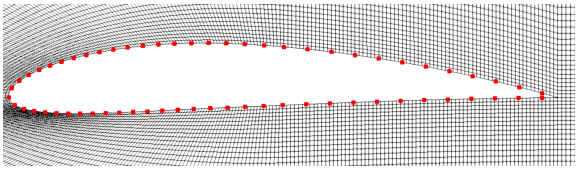


Figure 11: Control points distribution over an airfoil surface: not all the points of the surface are selected as control points and thus rigidly moved.

resulting on the solution of the following linear problem:

$$\begin{bmatrix} \boldsymbol{\Phi}_{RBF} & \boldsymbol{P}_q \\ \boldsymbol{P}_q^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \overline{\delta}^b \\ \mathbf{0} \end{bmatrix} , \tag{14}$$

where $\boldsymbol{\Phi}_{RBF} \in \mathbb{R}^{N_b \times N_b}$ contains the evaluations $\phi\left(\|\boldsymbol{x}_i^b - \boldsymbol{x}_j^b\|\right)$, $\boldsymbol{P} \in \mathbb{R}^{N_b \times (d+1)}$, with spacial dimension d, is filled as $\begin{bmatrix} 1 & \boldsymbol{x}_i^b \end{bmatrix}$ for each row, $\boldsymbol{\alpha}$ contains the coefficients for the polynomial $q(\boldsymbol{x})$ and $\overline{\delta}^b$ are the displacements for the control points, known a priori (see [24]);

3. evaluate all the remaining points of the grid by applying Equation 13.

Few aspects have to be underlined about the procedure above:

- Equation 13 is used not just to move the internal points of the grid but also the points located on the moving boundaries that are not selected as control points: even if their displacement could be calculated exactly, changing their position by rigid translation while all the points of the internal mesh are shifted by the use of the RBF may lead to a corrupted grid (see Figure 11);
- Equation 14 requires the resolution of a dense linear problem whose dimension is equal to $N_b + d + 1$. Thus, the number of control points have to be carefully selected. Fortunately the resolution of Equation 14 has to be carried out just once, storing all the necessary parameters to be used in the following mesh motions;
- also in this case one ends up with meshes having all the same topology which is, as previously highlighted, an essential feature when different geometries have to be compared.

To summarize the different aspects related to the two mesh motion strategies proposed in this section we can say:

- the resolution of a Laplace problem over the whole domain leads to the resolution of a very big system represented by a sparse matrix. This can be

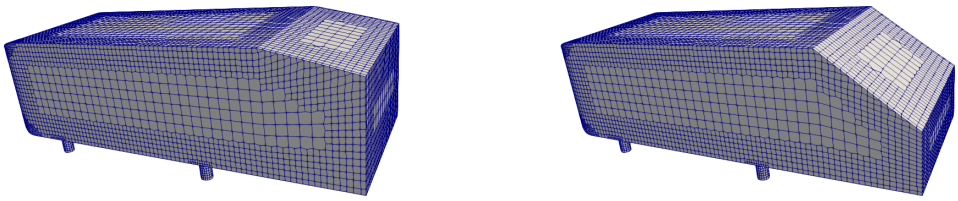


Figure 12: Mesh covering for an Ahmed body test case (see [1, 90]): reference undeformed configuration on the left, RBF interpolation modified grid on the right.

exploited by selecting an efficient algorithm devoted to the specific resolution of sparse matrices. Moreover this problem can be itself reduced by a POD approach so that no full dimension resolutions are required;

- the application of a Radial Basis Functions interpolation method gives birth to a much smaller matrix but it is dense. For this reason a less efficient technique has to be employed, as for example an LU decomposition in order to solve it properly.

It is then difficult to certainly select the best method for our applications basing the selection on the previous points. In our work we decided to rely on the RBF strategy due to the fact that the modified meshes show a lower non-orthogonality factor with respect to the ones obtained by the Laplace method. Since the good properties of the mesh are very relevant, especially for a Finite Volume scheme, the non-orthogonality factor played a primary role on the choice.

3.4.2 Proper Orthogonal Decomposition for moving domains

In subsection 2.3 we introduced a strategy to obtain modal basis functions by the use of a Proper Orthogonal Decomposition. Equation 4 is well defined whenever the norm $||\bullet||_{\mathbb{V}}$ is well defined. When dealing with geometrical parametrization problems this is clearly not the case: while changing the domain of the problem consequently to a change in the parameter value since $\Omega = \Omega(\boldsymbol{\mu})$ in this case, the definition of the norm varies itself so that the comparison of solutions defined over different geometries is not possible. In addition the resolution of the problem over different domains would lead to a difficulty on the definition of the correlation matrix $\boldsymbol{C}$: since the snapshots $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ have been calculated for two different parameter values and, thus, for different domains, the choice to be taken for the mass matrix $\boldsymbol{\mathcal{M}}$ is not straightforward.

In this work we try to overtake this problem by exploiting the fact that all the meshes were forced to have the same topology and connectivity. It is then possible to define a mid-configuration by the exploitation of the mesh motion obtained through a specific parameter $\boldsymbol{\mu}_{mid}$ resulting from:

$$\boldsymbol{\mu}_{mid} = \frac{1}{N_{\boldsymbol{\mu}}} \sum_{i=1}^{N_{\boldsymbol{\mu}}} \boldsymbol{\mu}_i \text{ for } \boldsymbol{\mu}_i \in \mathbb{P}_h \ .$$

In our case we use equispaced offline parameters to compose $\mathbb{P}_h$ leading to just $\boldsymbol{\mu}_{mid} = \dfrac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_{N_t}}{2}$.

The correlation matrix can then be easily assembled for this configuration as:

$$\overline{\boldsymbol{C}}_{ij} = \boldsymbol{s}_i^T \boldsymbol{\mathcal{M}}_{mid} \boldsymbol{s}_j \ ,$$

being $\boldsymbol{\mathcal{M}}_{mid} \in \mathbb{R}^{d\,N_h \times d\,N_h}$ the mass matrix defined for $\Omega(\boldsymbol{\mu}_{mid})$ where $\boldsymbol{s} \in \mathbb{R}^{d\,N_h}$ and d equal to the spacial dimension for vector valued quantities and equal to one for scalar valued variables. A caveat has to be here pointed out: by the use of this strategy the provided modal basis functions are not orthogonal in the current configuration, this only works for $\boldsymbol{\mu} = \boldsymbol{\mu}_{mid}$. In any case, as we will see, this will not be an issue and accuracy will not be affected.

Finally the POD basis functions are obtained as a linear combination of the training solutions as follows:

$$\boldsymbol{\xi}_i(\boldsymbol{x}) = \frac{1}{N_p} \sum_{j=1}^{N_{\boldsymbol{\mu}}} \overline{\boldsymbol{V}}_{ji} \boldsymbol{s}_j(\boldsymbol{x}) \ ,$$

being $\overline{\boldsymbol{V}}$ the matrix containing the eigenvectors related to $\overline{\boldsymbol{C}}$.

3.4.3 Geometrically parametrized problems

In this section we will present two different test cases, both related to geometrical parametrization. In particular we will focus on external flows for aerofoils where in the first problem the angle of attack is parametrized while the second one tackles a shape deformation study.

Variable angle of attack for a NACA4412 aerofoil

This paragraph will be devoted to the presentation of a geometrical parametrization test case. More in details we will report here the results related to the flow around an aerofoil where the angle of attack can be varied by rotating the aerofoil itself into the mesh. A classical choice for these problems is to change the direction of the incoming flow instead of modifying the geometry of the problem. Of course this would be much easier since no mesh motion strategies would be required. Anyway at the FOM level, the handling of the boundary conditions, especially on the upper and lower boundaries would become more challenging. The inflow velocity is in fact no more parallel with respect to the sides boundaries and the enforcement of symmetry or slip boundary conditions becomes more complex. Moreover, as it can be noticed analysing Figure 13, for this problem we decided to use a C-type domain instead of a classical rectangular mesh so that at the same time it was possible to get an inflow boundary capable of better following the leading edge of the aerofoil and to avoid critical points at the connection between inlet boundary and top-bottom ones. In fact, when dealing with straight inflow boundaries where a constant velocity is imposed, this may cause some issues at the connection with the consecutive boundaries where different conditions are selected. In case the angle of attack was varied by rotating the inflow velocity, the advantages given by such a choice for the domain would have been partially lost.

The domain is composed by a quadrilateral whose size is equal to 16.5 along the x direction, 16 along the y direction, and by a semicircle attached to the inlet side of the quadrilateral. The wing chord is equal to 1 and the foil is positioned on the center of the semicircle. The mesh counts 58000 hexahedral cells.

The motion of the mesh is carried out through the RBF strategy explained in subsubsection 3.4.1. The control points for the interpolation have been placed both on the moving boundaries and on the static patches of the domain Ω. The kernel of the RBF interpolation is given by Gaussian functions with a radius $r = 0.1$.

The problem we want to solve is the following one:

$$\begin{cases} \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot [\nu (\nabla \boldsymbol{u})] + \nabla p = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \boldsymbol{u} = [1, 0]^T & \text{on } \Gamma_{in} \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \Gamma_D \\ \nu \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} - p\boldsymbol{n} = 0 & \text{on } \Gamma_{out} \end{cases}, \tag{15}$$

where Γ_D stands for the surface of the aerofoil. It is here to be pointed out that the equations are independent from the parameters since the kinematic viscosity is fixed for this problem and equal to $3 \times 10^{-3} \frac{m^2}{s}$. The only way the problem is related to the parameter $\boldsymbol{\mu}$ is through the geometry since no physical parameters are taken into consideration. This aspect makes the relation parameter-solutions strongly non-linear. The *Reynolds* number for this test case can be evaluated as $Re = \frac{\boldsymbol{u}_\infty}{\nu} = 3 \times 10^3$ being $\boldsymbol{u}_\infty$ the magnitude of the inlet velocity. We can safely consider this test case as an incompressible laminar one so no turbulence closure models are required.

The offline phase is performed by solving the high fidelity problem for 100 randomly selected different angles of attack α so that $\alpha \in [-10°, 10°] \equiv \mathbb{P}$. The online parameter range has been slightly restricted to avoid very rare α values where the solution manifold spanned by the modal basis functions could have not enough information. For the online phase we solved the reduced problem for 50 different parameter values where $\alpha \in [-9.5°, 9.5°]$.

The full order SIMPLE algorithm runs with under relaxation for both velocity and pressure using the relaxation factors $\alpha_u = 0.7$ and $\alpha_p = 0.3$. As mentioned in subsection 3.2 the reduced order model has been constructed to be fully consistent with the SIMPLE procedure employed at the full order level. Therefore the same under-relaxation factors are used also at the reduced order level.

In Figure 14 we reported the decay of the eigenvalues for both velocity and pressure on the left while the table on the right shows the trend of the cumulated eigenvalues. We can notice that the decay is much slower if compared to the one reported in Figure 5. This id due to the fact that a geometrical parametrization problem is much more difficult if compared with the physical one reported in the previous section. For this reason many more modes are needed to obtain a reliable solution.

Online solutions are calculated as a linear combination of 40 modal basis functions for both pressure and velocity. In Figure 16 and Figure 17 are depicted full order and reduced order solutions for two different angles of attack, selected as the most extreme ones in the online parameter set since usually they are the worst reconstructed ones. It can be noticed by looking at the pointwise errors that also in this case the method is working pretty fine and the reduced approximations are very close to the high fidelity solutions.

Figure 15 shows the trends of the averaged L^2 norm errors in comparison with the projection ones for different amounts of modal basis functions used for both

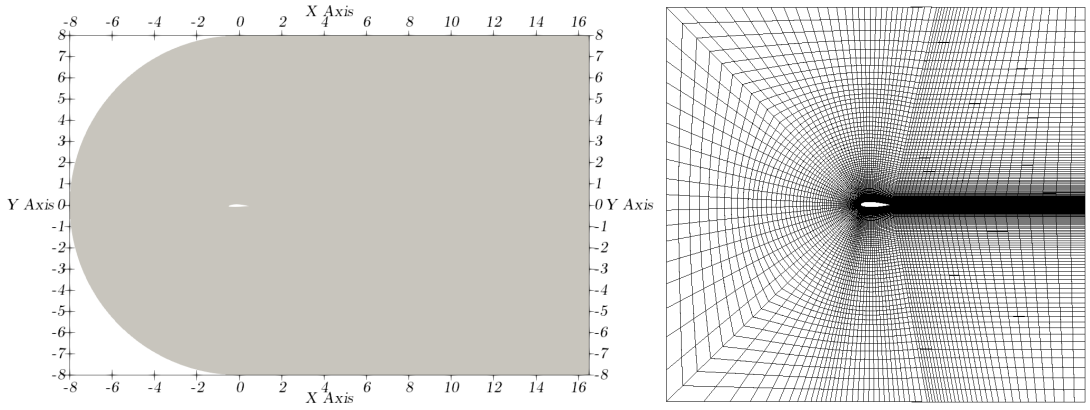


Figure 13: Comparison between the employed domain, left picture, and a classical rectangular mesh, right picture, employed for the resolution of the flow around a NACA4412 aerofoil.

velocity and pressure. The average is calculated over all the 50 online solutions. Two main aspects are here to be underlined:

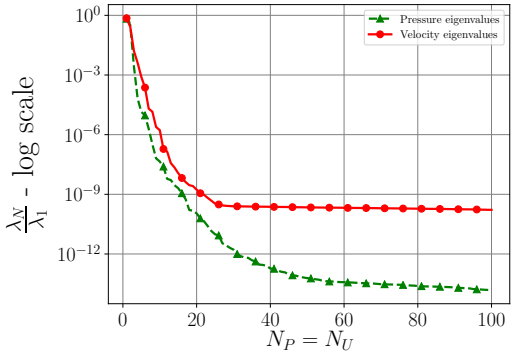
- projection errors and reduced errors show at large the same behaviour;
- reduced errors are pretty stable and present a quite good monotonicity.

Shape deformation for a NACA4412 aerofoil by the use of bump functions

This last incompressible laminar test case will treat the resolution of shape deformed aerofoil NACA4412 by the superposition of bump functions for both the top and the bottom surface (see [78, 30]). In particular we selected 5 bump functions as shown in Figure 18. These functions are defined over the chord of the aerofoil and they are designed so that they give a surface displacement on the whole surface along the aerofoil. They are added to the top surface and subtracted to the bottom one so that the possibility to have intersecting boundaries is avoided. We have:

$$f^{top} = \sum_{i=1}^{5} \mu_i^t f_i \qquad f^{bot} = \sum_{i=1}^{5} \mu_i^b f_i \ ,$$

where $\boldsymbol{\mu} = [\mu_1^t, \dots, \mu_5^t, \mu_1^b, \dots, \mu_5^b]$ and $\mu_i^{\square}$ are some amplitude factors for the bump functions. f^{top} is then added to the upper surface while f^{bot} is subtracted to the lower one.



	Pressure	Velocity
N = 1	0.674404466672	0.717187625760
N = 2	0.997304641239	0.979072177457
N = 5	0.999988310322	0.999728547289
N = 7	0.999999521373	0.999981203705
N = 10	0.999999956844	0.999999559123
N = 20	0.999999999818	0.999999980819
N = 40	0.999999999997	0.999999988151

Figure 14: Plot reporting the eigenvalue decay relative to the POD procedure used to compute the modes for both pressure and velocity on the left, while on the right the cumulates of the eigenvalues are reported.

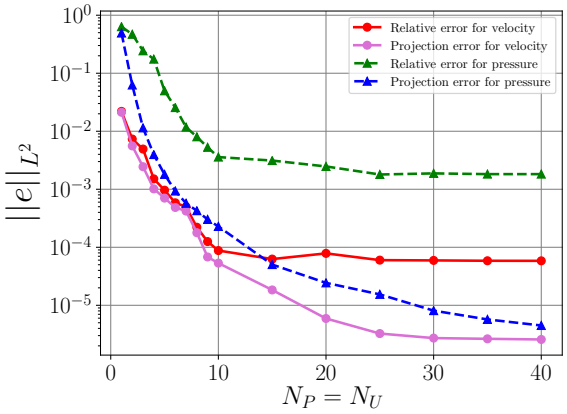


Figure 15: L^2 norm error and projection error trends for both velocity and pressure for different numbers of modal basis functions.

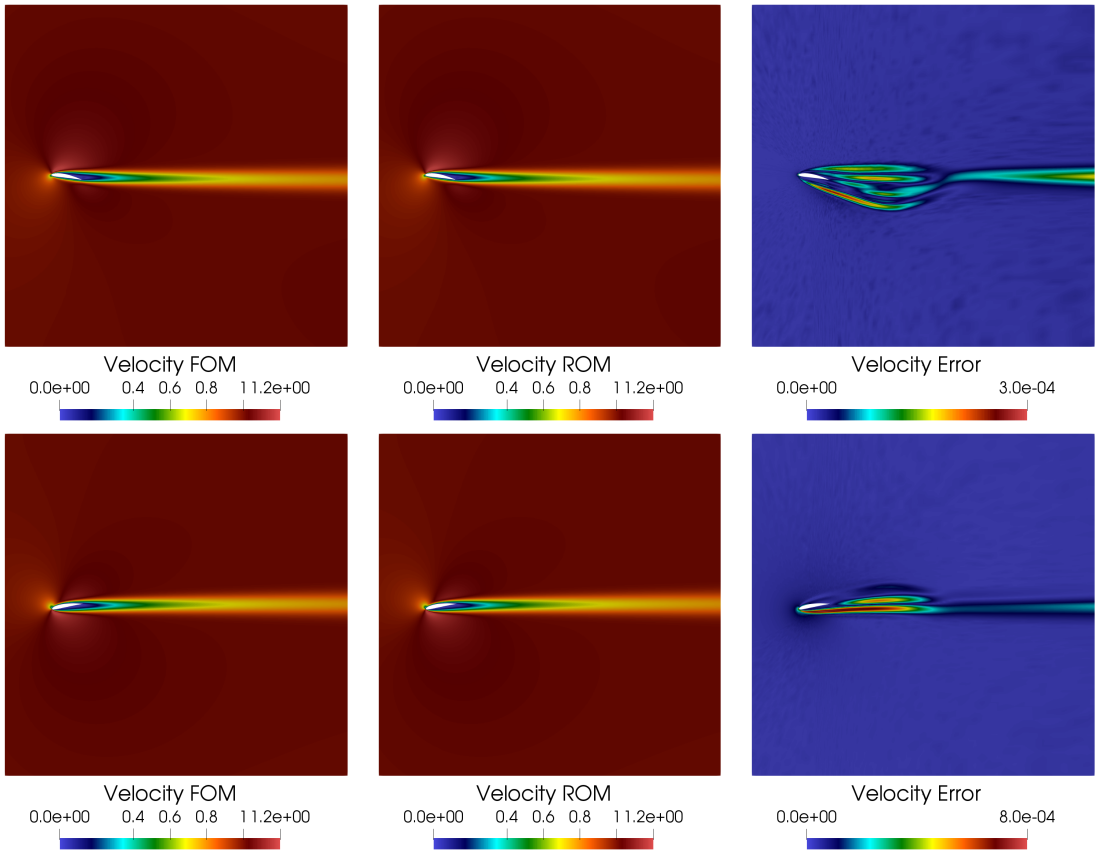


Figure 16: Comparison between high fidelity and reduced order velocity solutions for two different angles of attack $\alpha = 9.15°$ and $\alpha = -9.43°$. From left to right we can see the full order solution, the reduced order approximation and the pointwise error between them.

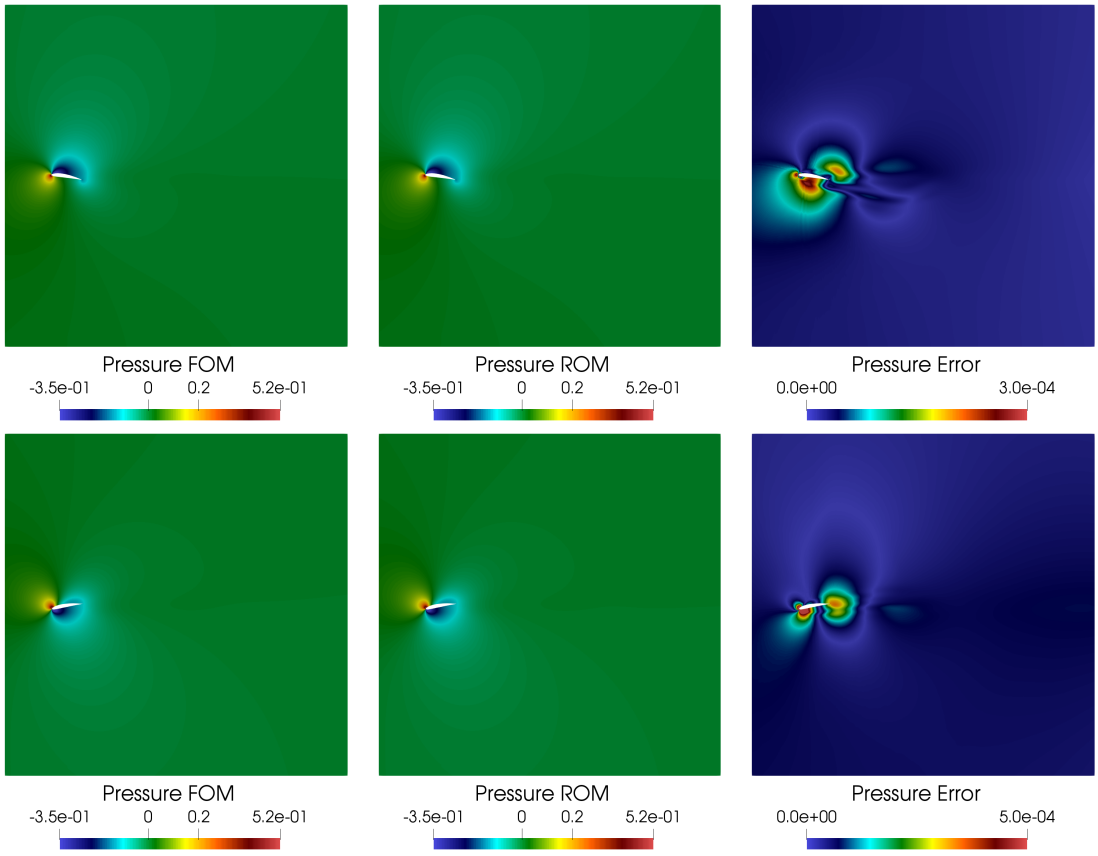


Figure 17: Comparison between high fidelity and reduced order pressure solutions for two different angles of attack $\alpha = 9.15°$ and $\alpha = -9.43°$. From left to right we can see the full order solution, the reduced order approximation and the pointwise error between them.

The problem we want to solve is the one reported in Equation 15 so, once again, we have $Re = 3 \times 10^3$. For the resolution of the full order problem the same mesh used in the previous test case has been employed and the same strategy for the mesh motion is implemented.

The multidimensional space where we are selecting the parameters is clearly much wider with respect to the one-dimensional set we were using in the previous case. For this reason 1000 offline parameter vectors have been selected where $\mu_i^{\square} \in [0, 0.02]$. It does not make any sense to select higher amplitudes for the bump functions since they would produce shapes without any kind of aerodynamic meaning. Even though we increased by one order of magnitude the number of offline solutions, they are in any case much less dense if compared to the previous test case since we are now selecting parameters into a ten-dimensional parameter space. This is the main reason for the much slower decay of the eigenvalues in Figure 19 with respect to the ones in Figure 14: since we are dealing with 10 parameters at a time, the information coverage given by the same amount of modal basis functions is much lower.

Online parameters have been selected in a slightly smaller range ($[0, 0.018]$) for what has already been explicated in the previous paragraph. 20 different online parameter vectors have been selected to test the resolution properties of the method. Online solutions are provided by the employment of 40 modal basis functions for both velocity and pressure fields. This choice has been dictated by the analysis of the eigenvalues decay shown in Figure 19 while trying to keep a low dimension of the problem as a goal for the resolution machinery.

Figure 20 shows the trends of both L^2 norm reduced and projection errors while changing the number of modal basis functions used for velocity and pressure fields at the same time. Even though the number of offline solutions is quite small if compared to the dimension of the parameter space, as explained above, the method shows good reliability properties together with an appreciable stable behaviour. The error given by the reduced resolution of the problem appears to be just one order of magnitude worse then the best possible one provided by the projection of the corresponding high fidelity solutions over the modal basis functions.

In Figure 21 are depicted offline and online solutions for the velocity field together with the pointwise error between the two for a random online selection of the parameters, while Figure 22 is dedicated to the pressure solutions for the same geometrical configuration. It can be noticed that the pointwise errors are pretty low even if the eigenvalues truncation shows a noticeable discard of information.

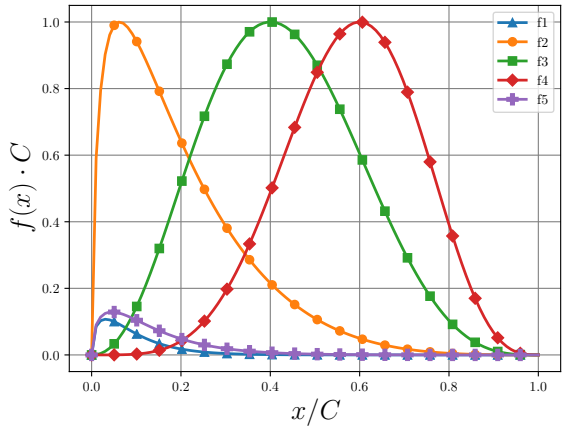


Figure 18: L^2 norm error and projection error trends for both velocity and pressure for different numbers of modal basis functions.

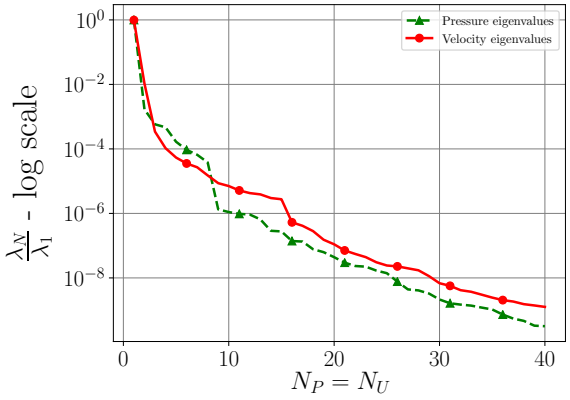


Figure 19: L^2 norm error and projection error trends for both velocity and pressure for different numbers of modal basis functions.

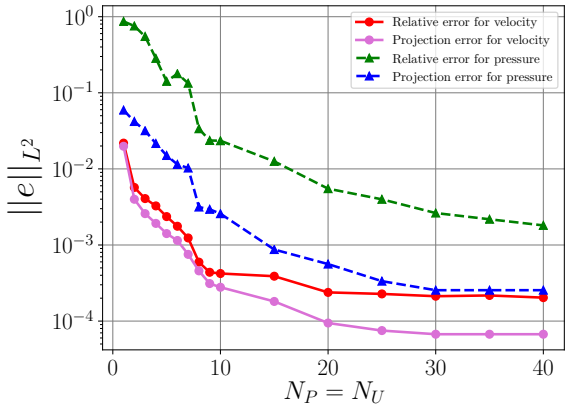


Figure 20: L^2 norm error and projection error trends for both velocity and pressure for different numbers of modal basis functions.

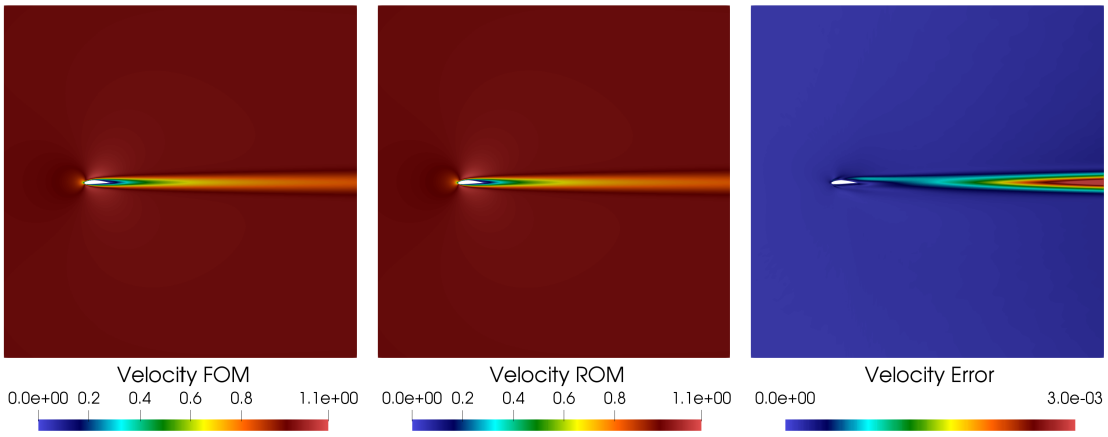


Figure 21: Comparison between high fidelity and reduced order velocity solutions for a random value of $\boldsymbol{\mu}$ in the online set. From left to right we can see the full order solution, the reduced order approximation and the pointwise error between them.

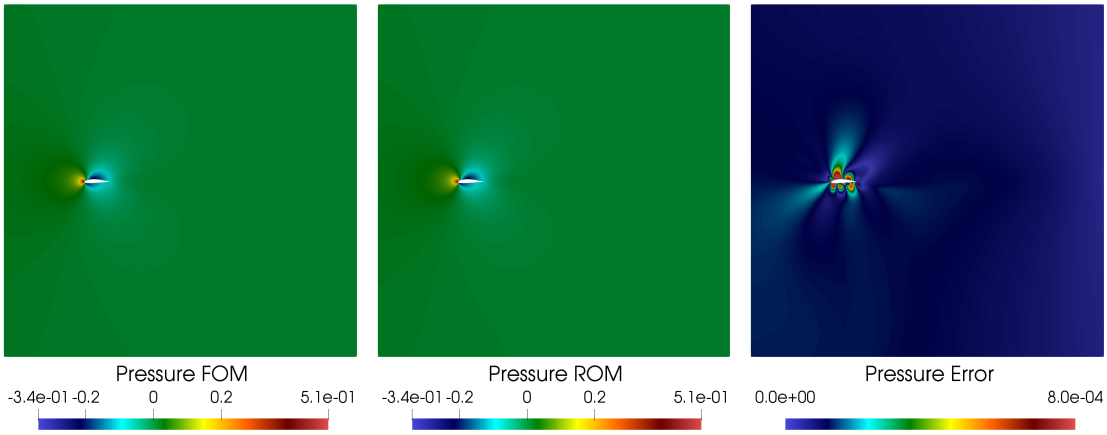


Figure 22: Comparison between high fidelity and reduced order velocity solutions for a random value of μ in the online set. From left to right we can see the full order solution, the reduced order approximation and the pointwise error between them.

3.5 Context and overview

In this chapter we presented a strategy for the resolution of laminar incompressible flow problems.

We firstly focused on the equations concerning the problem and their FV discretization in subsection 3.1. We devoted subsection 3.2 to the presentation of a segregated reduced order approach for the resolution of parametrized incompressible laminar flows in an efficient way.

In the last part of the chapter we focused on the resolution of physical and geometrical parametrization. A back step problem with a parametrized kinematic viscosity has been proposed in subsection 3.3 while subsection 3.4 completed the methods overview for what concerns laminar flows by showing possible different strategies for the mesh motion in geometrical parametrization applications and by presenting a more general architecture for the use of the POD even for different snapshots domains.

The techniques presented so far show a good reliability. They will be expanded and enriched in the following chapters.

4 Incompressible turbulent Navier-Stokes equations

In most fluid dynamics cases, a laminar treatment of the problem is not possible. This is due to the fact that, when the ratio of inertial forces to viscous forces is sufficiently high, the damping provided by the viscosity present in the flow is not able to take under control the whole amount of kinetic energy scattered in the surroundings. This is typically estimated by the evaluation of the *Reynolds* number $Re = \frac{\boldsymbol{u}_\infty L}{\nu}$, where $\boldsymbol{u}_\infty$ is a reference velocity, L a reference length while ν is the kinematic viscosity. When the *Reynolds* number is high enough, let us say $Re \sim 4 \times 10^3$, the flow moves from a state where streamlines are distributed in a parallel way to a state where chaotic motion of the fluid particles is present. For this reason the Kolmogorov energy cascade takes place: large eddies caused by the free kinetic energy are not stable and decay into smaller and smaller vortexes till Kolmogorov scale dimensions are reached. At that point the vortexes are small, weak and isotropically distributed enough to be dissipated by the viscosity effects. To simulate this behaviour a micro-scale mesh would be required: when employing a coarse mesh, sub-grid scales are not resolved and kinetic energy is no more correctly dissipated. This aspect leads to an energy surplus in the system causing a complete misleading of physical dynamics. On the contrary it is not possible to create a grid refined enough: it would lead to a huge system, long too big to be solvable (see the estimate on the number of cells in the grid dependent on the *Reynolds* number reported in subsection 4.1).

In the early 40s Andrey Kolmogorov started developing possible techniques to overcome this issue. From that moment, many different strategies have been proposed ([3, 77]) trying to model the dissipation: additional dissipating terms are added to better decrease the amount of kinetic energy accumulating into the system.

4.1 Equations, Finite Volume discretization and turbulence closure models

Since the dynamics of the fluid at the Kolmogorov scales are not relevant for what concern the macro phenomena, the most common strategy to attack the problem is an average on time of the equations so that it is possible to get rid of all the fluctuating parts. For incompressible fluids the Reynolds averaging is usually employed and the process is reported here in the following passages.

Averaging rules

Given two generic flow variables Φ and Ψ, the Reynolds averaging rule reads:

$$\overline{\Phi} = \frac{1}{T}\int_T \Phi(t)dt\ , \qquad \Phi = \overline{\Phi} + \Phi'\ .$$

From the above rule it follows:

$$\overline{\Phi'} = 0\ , \qquad \overline{\overline{\Phi}\Phi'} = 0\ , \qquad \overline{\overline{\Phi}\Psi'} = 0\ , \qquad \overline{\frac{\partial\Phi}{\partial t}} = \frac{\partial\overline{\Phi}}{\partial t}\ , \qquad \overline{\frac{\partial\Phi}{\partial x_i}} = \frac{\partial\overline{\Phi}}{\partial x_i}\ .$$

We can now use a Reynolds decomposition to separate the mean flow component $\overline{\square}$ from the fluctuating ones $\square'$ for both pressure and velocity:

$$\begin{aligned} p &= \overline{p} + p'\ , \\ \boldsymbol{u} &= \overline{\boldsymbol{u}} + \boldsymbol{u}'\ . \end{aligned}$$

Since we are considering parameters dependent problems, we have:

$$\begin{aligned} \overline{p} &= \overline{p}(t, \boldsymbol{x}, \boldsymbol{\mu})\ , & p' &= p'(t, \boldsymbol{x}, \boldsymbol{\mu})\ , \\ \overline{\boldsymbol{u}} &= \overline{\boldsymbol{u}}(t, \boldsymbol{x}, \boldsymbol{\mu})\ , & \boldsymbol{u}' &= \boldsymbol{u}'(t, \boldsymbol{x}, \boldsymbol{\mu}). \end{aligned}$$

Reynolds decomposition and averaging can now be applied to the Navier Stokes system to obtain the Reynolds Averaged Navier Stokes (RANS) equations.

CONTINUITY EQUATION:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot[\rho\boldsymbol{u}] = 0$$

In case of incompressible flows the continuity constraint reduces to:

$$\nabla\cdot\boldsymbol{u} = 0\ .$$

Reynolds decomposition:

$$\nabla\cdot(\overline{\boldsymbol{u}} + \boldsymbol{u}') = 0\ .$$

Reynolds averaging:

$$\overline{\nabla\cdot(\tilde{\boldsymbol{u}} + \boldsymbol{u}'')} = \overline{\nabla\cdot(\overline{\boldsymbol{u}})} + \overline{\nabla\cdot(\boldsymbol{u}')} = \nabla\cdot\overline{\boldsymbol{u}} + \nabla\cdot\underbrace{\overline{\boldsymbol{u}'}}_{0} = 0\ .$$

Final equation:

$$\nabla \cdot \overline{\boldsymbol{u}} = 0 \, .$$

MOMENTUM EQUATION:

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \left[\mu \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right) \right] - \frac{2}{3} \nabla \left(\mu \nabla \cdot \boldsymbol{u} \right) - \nabla p$$

In case of incompressible flows the momentum equation reduces to:

$$\underbrace{\frac{\partial \boldsymbol{u}}{\partial t}}_{ⓐ} + \underbrace{\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})}_{ⓑ} = \underbrace{\nu \nabla^2 \boldsymbol{u}}_{ⓒ} - \underbrace{\frac{\nabla p}{\rho}}_{ⓓ} \, .$$

Reynolds decomposition:

$$ⓐ : \frac{\partial \left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right)}{\partial t} \, ,$$
$$ⓑ : \nabla \cdot \left[\left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right) \otimes \left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right) \right] \, ,$$
$$ⓒ : \nu \nabla^2 \left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right) \, ,$$
$$ⓓ : \frac{\nabla \left(\overline{p} + p' \right)}{\rho} \, .$$

From now on in this chapter we will assume $p = \frac{p}{\rho}$ for sake of simplicity since the density is constant:

$$ⓓ : \nabla \left(\overline{p} + p' \right) \, .$$

Reynolds averaging:

$$ⓐ : \overline{\frac{\partial \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}' \right)}{\partial t}} = \frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \underbrace{\frac{\partial \overline{\boldsymbol{u}'}}{\partial t}}_{0} \, ,$$

$$ⓑ : \overline{\nabla \cdot \left[\left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right) \otimes \left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right) \right]} = \nabla \cdot \left[\overline{\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}} + \underbrace{\overline{\overline{\boldsymbol{u}} \otimes \boldsymbol{u}'}}_{0} + \underbrace{\overline{\boldsymbol{u}' \otimes \overline{\boldsymbol{u}}}}_{0} + \overline{\boldsymbol{u}' \otimes \boldsymbol{u}'} \right] =$$
$$= \nabla \cdot \left[\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}} + \overline{\boldsymbol{u}' \otimes \boldsymbol{u}'} \right] \, ,$$

$$ⓒ : \overline{\nu \nabla^2 \left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right)} = \nu \nabla^2 \overline{\left(\overline{\boldsymbol{u}} + \boldsymbol{u}' \right)} = \nu \nabla^2 \overline{\boldsymbol{u}} + \underbrace{\nu \nabla^2 \overline{\boldsymbol{u}'}}_{0} \, ,$$

$$ⓓ : \overline{\nabla \left(\overline{p} + p' \right)} = \nabla \overline{p} + \underbrace{\nabla \overline{p'}}_{0} \, .$$

Resulting equation:

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \nabla \cdot \left[\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}} + \overline{\boldsymbol{u}' \otimes \boldsymbol{u}'} \right] - \nu \nabla^2 \overline{\boldsymbol{u}} + \nabla \overline{p} = 0 \, .$$

Let us define $\tau_f = \overline{\boldsymbol{u}' \otimes \boldsymbol{u}'}$. This term is not known and has to be modeled in terms of main flow variables. In the Boussinesq hypothesis it is approximated as:

$$\tau_f = \overline{\boldsymbol{u}' \otimes \boldsymbol{u}'} = -\nu_t \nabla \overline{\boldsymbol{u}} + \frac{2}{3} k \boldsymbol{I} \, .$$

where k is the turbulent kinetic energy, defined as $k = \frac{1}{2}\overline{\boldsymbol{u}'^2}$, and ν_t is an additional viscosity due to turbulence, referred to as eddy viscosity.

Final equation:

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}) - (\nu + \nu_t) \nabla^2 \overline{\boldsymbol{u}} + \nabla \overline{p} + \frac{2}{3} \nabla k = 0 \, .$$

The RANS system in the Boussinesq hypothesis then reads:

$$\begin{cases} \dfrac{\partial \overline{\boldsymbol{u}}}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}) - (\nu + \nu_t) \nabla^2 \overline{\boldsymbol{u}} + \nabla \overline{p} + \dfrac{2}{3} \nabla k = 0 \\ \nabla \cdot \overline{\boldsymbol{u}} = 0 \end{cases} \, . \tag{16}$$

The RANS system is clearly not closed since now both ν_t and k have somehow to be evaluated. In the literature there are nowadays many possible strategy to obtain an efficient approximation for those terms. All these methods can be classified into three main groups:

- zero equations model;
- one equation models;
- two equations models.

In the zero equations models, the isotropic part in the Boussinesq hypothesis, i.e. $\frac{2}{3} k \boldsymbol{I}$, is incorporated into the main pressure term so that there is no need to evaluate the turbulent kinetic energy explicitly. Then the eddy viscosity is approximated once and for all using reference values. This approach is really simple but it clearly results inaccurate since it is not able to distinguish between different turbulence cases.

For the one equation models, typically an equation for the turbulent kinetic energy is used. The eddy viscosity is then evaluated as a function of k so that a closed set of equations is obtained.

What distinguishes two-equations models from the previous is the fact that two different transport equations are employed to estimate two independent turbulent quantities related to turbulence length on one side and to time scales on another. This is long the most used group and includes the famous $k-\epsilon$ and $k-\omega$ models, where ϵ is the rate of turbulent dissipation while ω stands for the turbulent time scale $\left(\omega = \frac{\epsilon}{k}\right)$.

For the $k-\epsilon$ model we have:

$$\begin{cases} \dfrac{\partial k}{\partial t} + \overline{\boldsymbol{u}}_i \dfrac{\partial k}{\partial x_i} = -\tau_f \dfrac{\partial \overline{\boldsymbol{u}}_i}{\partial x_j} - \epsilon + \dfrac{\partial}{\partial x_i}\left(\dfrac{\nu_t}{\sigma_k}\dfrac{\partial k}{\partial x_i}\right) + \nu \dfrac{\partial^2 k}{\partial x_i \partial x_i} \\ \dfrac{\partial \epsilon}{\partial t} + \overline{\boldsymbol{u}}_i \dfrac{\partial \epsilon}{\partial x_i} = -C_{\epsilon 1}\dfrac{\epsilon}{k}\tau_f \dfrac{\partial \overline{\boldsymbol{u}}_i}{\partial x_j} + \dfrac{\partial}{\partial x_i}\left(\dfrac{\nu_t}{\sigma_\epsilon}\dfrac{\partial \epsilon}{\partial x_i}\right) - C_{\epsilon 2}\dfrac{\epsilon^2}{k} + \nu \dfrac{\partial^2 \epsilon}{\partial x_i \partial x_i} \\ \nu_t = C_\mu \dfrac{k^2}{\epsilon} \end{cases} , \tag{17}$$

where $C_{\epsilon 1}$, $C_{\epsilon 2}$, C_μ, σ_k and σ_ϵ are constant parameters. These last constants have to be tuned to fit experimental data so that the model is consistent with real test cases. Components expressions and Einstein summation have been here employed. The expression for ν_t has been firstly proposed by Launder and Spalding [77].

Since k is here representing the turbulent kinetic energy, it is clear that for physical boundaries we have to impose $k = 0$ near the wall. We also have to impose $\dfrac{\partial k}{\partial \hat{n}} = 0$ to avoid any kind of discontinuity at the boundary. This is not trivial at all for the turbulent dissipation. In fact no boundary condition is present for ϵ near the walls. This causes in most cases very bad results close to the surfaces with diffusion terms in the equation for ϵ possibly approaching infinity values. Thus $k-\epsilon$ model is commonly avoided when the region close to the physical boundary is of interest and is, at the contrary, mostly used for free shear flows.

For this reason the $k-\omega$ turbulence model has been introduced [141, 140]. This model is much more accurate near the walls even if it becomes less efficient far from the boundaries. The turbulence model to be employed has to be then carefully selected depending on the test case it has to be applied to.

For the $k-\omega$ model we have:

$$\begin{cases} \dfrac{\partial k}{\partial t} + \overline{\boldsymbol{u}}_i \dfrac{\partial k}{\partial x_i} = -\tau_f \dfrac{\partial \overline{\boldsymbol{u}}_i}{\partial x_j} - C^* \dfrac{k^{3/2}}{l_0} + \dfrac{\partial}{\partial x_i}\left(\dfrac{\nu_t}{\sigma_k}\dfrac{\partial k}{\partial x_i}\right) + \nu \dfrac{\partial^2 k}{\partial x_i \partial x_i} \\ \dfrac{\partial \omega}{\partial t} + \overline{\boldsymbol{u}}_i \dfrac{\partial \omega}{\partial x_i} = -\gamma_1 \dfrac{\omega}{k}\tau_f \dfrac{\partial \overline{\boldsymbol{u}}_i}{\partial x_j} + \dfrac{\partial}{\partial x_i}\left(\dfrac{\nu_t}{\sigma_\omega}\dfrac{\partial \omega}{\partial x_i}\right) - \gamma_2 \omega^2 + \nu \dfrac{\partial^2 \omega}{\partial x_i \partial x_i} \\ \nu_t = \gamma^* \dfrac{k}{\omega} \end{cases}, \tag{18}$$

where C^*, σ_k, σ_ω, γ_1, γ_2 and γ^* are constant values to be properly tuned to fit experimental data.

If we now take into consideration Equation 16 together with Equation 17 or Equation 18 and complete this set with boundary and initial conditions for $\overline{\boldsymbol{u}}$ we obtain a closed problem that can be solved to obtain averaged solutions for both velocity and pressure.

The term $\dfrac{2}{3}\nabla k$ into Equation 16 can be neglected because of the fact that the variation of the turbulent kinetic energy, is in any case, very small since turbulence is, in practice, isotropic into the domain. For this reason this term can be dropped out without any further consideration (see [64]).

The Kolmogorov dissipation scales, denoted by η, can be estimated as $\eta = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}$. We also can approximate the turbulent dissipation rate as $\epsilon \approx \frac{\boldsymbol{u}_\infty^3}{L}$ being L a reference length of the problem, e.g. the total length of the domain. By combining the two of them we get $\eta = \left(\frac{\nu^3 L}{\boldsymbol{u}_\infty^3}\right)^{\frac{1}{4}}$. The resolution of the Kolmogorov scales would then require a size of every grid cell smaller or equal then η. Thus the number of segments to be employed to divide every side of the domain would result $N = \frac{L}{\eta} = \left(\frac{\boldsymbol{u}_\infty^3 L^3}{\nu^3}\right)^{\frac{1}{4}} = Re^{\frac{3}{4}}$. If we suppose to be dealing with a cubic domain divided into hexahedral cells, we can estimate the number of required grid units as $N_{tot} = N^3 = Re^{\frac{9}{4}}$. This means that the dimension of the problem for the resolution of a Direct Numerical Simulation (DNS) would dramatically increase for an increasing *Reynolds* number.

From now on we will consider just steady state problems. For this reason the time derivative into the momentum equation will be neglected. Moreover we get $\overline{\boldsymbol{u}}(t, \boldsymbol{x}, \boldsymbol{\mu}) = \overline{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\mu})$, $\overline{p}(t, \boldsymbol{x}, \boldsymbol{\mu}) = \overline{p}(\boldsymbol{x}, \boldsymbol{\mu})$ and we will refer to them as just $\overline{\boldsymbol{u}}$ and $\overline{p}$ for sake of simplicity.

The steady-state momentum equation written in its integral form for every cell of the tessellation $\mathcal{T}$, reads as follows:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}})\, dV + \int_{\Omega_i} \boldsymbol{\nabla} \overline{p}\, dV - \int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[(\nu + \nu_t) \left(\boldsymbol{\nabla} \overline{\boldsymbol{u}} + (\boldsymbol{\nabla} \overline{\boldsymbol{u}})^T \right) \right] dV = 0 \, .$$

Let us analyze this last equation, term by term. The convective term can be treated by the use of the Gauss' theorem:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}})\, dV = \int_{S_i} \overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}} \cdot d\boldsymbol{S} \simeq \sum_f \boldsymbol{S}_f \cdot \overline{\boldsymbol{u}}_f \otimes \overline{\boldsymbol{u}}_f = \sum_f \boldsymbol{F}_f \overline{\boldsymbol{u}}_f \, ,$$

where S_i is the total surface related to the cell i, $\boldsymbol{S}_f$ is the oriented surface dividing the two neighbor cells i and f, $\overline{\boldsymbol{u}}_f$ is the velocity evaluated at the center of the face S_f and $\boldsymbol{F}_f$ is the flux of the velocity through the face S_f (see Figure 2). Similarly to what has been explained in subsection 3.1, $\overline{\boldsymbol{u}}_f$ is not straight available since all the variables of the problem are evaluated at the center of the cells while a face-centered evaluations is required. Face values are obtained through an interpolation of the values at the center of the cells. Again fluxes are calculated by the use of the velocity obtained at previous iteration during the resolution of the problem so that the non-linearity is easily resolved.

To deal with the pressure term we exploit the gradient theorem:

$$\int_{\Omega_i} \boldsymbol{\nabla} \overline{p}\, dV = \int_{S_i} \overline{p}\, d\boldsymbol{S} \simeq \sum_f \boldsymbol{S}_f \overline{p}_f \, ,$$

where p_f is the pressure evaluated at the center of the face S_f.

The last term to be considered is the diffusive one:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[(\nu + \nu_t) \left(\boldsymbol{\nabla} \overline{\boldsymbol{u}} + (\boldsymbol{\nabla} \overline{\boldsymbol{u}})^T \right) \right] dV \simeq (\nu + \nu_t)_i \int_{\Omega_i} \boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla} \overline{\boldsymbol{u}} + (\boldsymbol{\nabla} \overline{\boldsymbol{u}})^T \right) dV =$$
$$= (\nu + \nu_t)_i \int_{\Omega_i} \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \overline{\boldsymbol{u}}\, dV = (\nu + \nu_t)_i \int_{S_i} \boldsymbol{\nabla} \overline{\boldsymbol{u}} \cdot d\boldsymbol{S} \simeq \sum_f \left[(\nu + \nu_t)_f \left(\boldsymbol{\nabla} \overline{\boldsymbol{u}} \right)_f \right] \cdot \boldsymbol{S}_f \, ,$$

where $(\nu + \nu_t)_i$ is the viscosity in the i-th cell, $(\nu + \nu_t)_f$ stands for the viscosity evaluated at the center of the face S_f and $(\boldsymbol{\nabla} \overline{\boldsymbol{u}})_f$ refers to the gradient of the velocity evaluated at the center of the face S_f. Notice that the gradient of the velocity is

not known at the face of the cell. If the mesh is orthogonal, the approximation of its flux is straightforward:

$$\boldsymbol{S}_f \cdot (\boldsymbol{\nabla}\overline{\boldsymbol{u}})_f \simeq |\boldsymbol{S}_f| \frac{\overline{\boldsymbol{u}}_i - \overline{\boldsymbol{u}}_j^f}{|\boldsymbol{d}_f|} \,,$$

where $\square_j^f$ terms stand for evaluations of a certain quantity at the center of the cell j being f the face dividing the two cells. If the mesh is not orthogonal (see Figure 2), as for the laminar case, a correction has to be added:

$$\boldsymbol{S}_f \cdot (\boldsymbol{\nabla}\overline{\boldsymbol{u}})_f \simeq |\boldsymbol{\pi}_f| \frac{\overline{\boldsymbol{u}}_i - \overline{\boldsymbol{u}}_j^f}{|\boldsymbol{d}_f|} + \boldsymbol{\omega}_f \cdot (\boldsymbol{\nabla}\overline{\boldsymbol{u}})_f \,,$$

where $\boldsymbol{S}_f$ has been decomposed into a component parallel to $\boldsymbol{d}_f$, namely $\boldsymbol{\pi}_f$, and another one orthogonal to $\boldsymbol{d}_f$, namely $\boldsymbol{\omega}_f$. The term $(\boldsymbol{\nabla}\overline{\boldsymbol{u}})_f$ is finally evaluated by interpolation starting from the values $(\boldsymbol{\nabla}\overline{\boldsymbol{u}})_i$ and $(\boldsymbol{\nabla}\overline{\boldsymbol{u}})_j^f$ at the centers of the neighbor cells.

Now the complete discrete momentum equation can be written:

$$\sum_i^{N_h} \left[\sum_f^{N_h} \boldsymbol{F}_f \overline{\boldsymbol{u}}_f + \sum_f^{N_h} \boldsymbol{S}_f \overline{p}_f - \sum_f^{N_h} (\nu + \nu_t)_f \, |\boldsymbol{\pi}_f| \frac{\overline{\boldsymbol{u}}_i - \overline{\boldsymbol{u}}_j^f}{|\boldsymbol{d}_f|} + \boldsymbol{\omega}_f \cdot (\boldsymbol{\nabla}\overline{\boldsymbol{u}})_f \right] = 0 \,,$$

After having applied the necessary interpolation for face centers quantities evaluation, the whole system can be rewritten into its matrix form as follow:

$$\begin{bmatrix} \boldsymbol{A}_u & \boldsymbol{B}_p \\ \boldsymbol{\nabla}(\cdot) & 0 \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{u}}_h \\ \overline{\boldsymbol{p}}_h \end{bmatrix} = \boldsymbol{0} \,, \tag{19}$$

where $\boldsymbol{A}_u$ is the matrix containing all the terms related to velocity into the discretized momentum equation, $\boldsymbol{B}_p$ is the matrix containing the terms related to pressure into the same equation, $\boldsymbol{\nabla}(\cdot)$ is the matrix representing the incompressibility constraint, $\overline{\boldsymbol{u}}_h$ is the vector where all the $\overline{\boldsymbol{u}}_i$ variables are collected and the same applies for $\overline{\boldsymbol{p}}_h$ with respect to $\overline{p}_i$ having $\overline{\boldsymbol{u}}_h \in \mathbb{U}_h \subset \mathbb{R}^{d\,N_h}$ and $\overline{\boldsymbol{p}}_h \in \mathbb{Q}_h \subset \mathbb{R}^{N_h}$ with d spacial dimension of the problem. The interested reader can find deeper explanations on the Finite Volume discretization technique in [68, 63, 94].

In this section the offline phase is performed by the use of a segregated pressure-based approach, the *Semi-Implicit Method for Pressure-Linked Equations* (SIMPLE) algorithm in its turbulent form, ensuing what has been pointed out in subsection 3.1. The pressure equation is obtained in the same way (see subsubsection 3.1.1) just considering $\overline{\boldsymbol{u}}_h$, $\overline{\boldsymbol{u}}^*$, $\overline{\boldsymbol{u}}'$, $\overline{p}_h$, $\overline{p}^*$ and $\overline{p}'$ instead of $\boldsymbol{u}_h$, $\boldsymbol{u}^*$, $\boldsymbol{u}'$, p_h, p^* and p'.

We only report here the final pressure equation for incompressible turbulent SIMPLE algorithm for sake of completeness:

$$[\boldsymbol{\nabla}(\cdot)]\,\overline{\boldsymbol{u}}_h = [\boldsymbol{\nabla}(\cdot)]\left\{\boldsymbol{A}^{-1}\left[\boldsymbol{H}(\overline{\boldsymbol{u}}^*) - \boldsymbol{B}_p\overline{\boldsymbol{p}}_h\right]\right\} \Rightarrow [\boldsymbol{\nabla}(\cdot)]\,\boldsymbol{A}^{-1}\boldsymbol{B}_p\overline{\boldsymbol{p}}_h = [\boldsymbol{\nabla}(\cdot)]\,\boldsymbol{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{u}}^*)\,.$$

4.2 Reduced order algorithm for incompressible turbulent flows

As introduced in subsection 3.2, we would like to rely on a method capable of being as coherent as possible with respect to the high fidelity algorithm. This idea will be applied in this work for all the different fluid conditions under examination: the same approach used in subsection 3.2 will also be employed in subsection 5.2.

Also in this case an algorithm that emulates the steps of the full order scheme has been developed. The main steps for the reduced methods related to incompressible laminar flows are reported in Algorithm 3.

For this scope we introduce here the reduced expansions for both velocity and pressure fields respectively:

$$\overline{\boldsymbol{u}}_r = \sum_{i=1}^{N_u} \overline{a}_i(\boldsymbol{\mu})\overline{\boldsymbol{\psi}}_i(\boldsymbol{x}) = \overline{\boldsymbol{\Psi}}\overline{\boldsymbol{a}}\,, \qquad \overline{p}_r = \sum_{i=1}^{N_p} \overline{b}_i(\boldsymbol{\mu})\overline{\varphi}_i(\boldsymbol{x}) = \overline{\boldsymbol{\Phi}}\overline{\boldsymbol{b}}\,,$$

where N_u and N_p are the numbers of basis functions selected for the reconstruction of time averaged velocity and pressure solutions respectively, $\overline{\boldsymbol{a}} \in \mathbb{R}^{N_u}$ is the vector containing the coefficients for the velocity expansion while the same reads for pressure with respect to $\overline{\boldsymbol{b}} \in \mathbb{R}^{N_p}$ being $\overline{\boldsymbol{\psi}}_i$ and $\overline{\varphi}_i$ the corresponding vector and scalar basis functions collected into the matrices $\overline{\boldsymbol{\Psi}} \in \mathbb{R}^{d\,N_h \times N_u}$ and $\overline{\boldsymbol{\Phi}} \in \mathbb{R}^{N_h \times N_p}$ respectively.

For turbulent flows clearly a way has to be found also for the reconstruction of the eddy viscosity field ν_t as shown in subsection 4.1. Since it has to be evaluated also during the online phase, a suited strategy has to be applied to optimize the performances of the reduced solver. Also for what regards the reduced eddy viscosity

ν_{tr}, then, we introduce a proper expansion:

$$\nu_{tr} = \sum_{i=0}^{N_{\nu_t}} m_i(\boldsymbol{\mu})\zeta_i(\boldsymbol{x}) = \boldsymbol{Z}\boldsymbol{m}, \tag{20}$$

indicating by N_{ν_t} the number of modal basis functions selected for the reconstruction of the eddy viscosity field, by $\boldsymbol{m} \in \mathbb{R}^{N_{\nu_t}}$ the vector containing the coefficients of the linear combination and by ζ_i the basis functions collected into the matrix $\boldsymbol{Z} \in \mathbb{R}^{N_h \times N_{\nu_t}}$ where $\nu_{tr} \in span\{\zeta_i | i = 1, \cdots, N_{\nu_t}\}$.

Basis functions ζ_i are obtained, as usual, by the application of a Proper Orthogonal Decomposition to a set of solutions for the eddy viscosity calculated during the offline phase (see subsection 2.3):

$$\boldsymbol{S}_{\nu_t} = \left[\nu_{t1}, \ldots, \nu_{tN_{\boldsymbol{\mu}}}\right] \in \mathbb{R}^{N_h \times N_{\boldsymbol{\mu}}} .$$

It can be easily shown, as it can be seen in subsection 4.1, that the eddy viscosity field is only dependent on the velocity field since it is related to the dissipation of the kinetic energy. For this reason we may assume the coefficients related to the additional dissipation term to be a function of the velocity. We have:

$$\boldsymbol{m} = \boldsymbol{m}(\boldsymbol{\mu}, \boldsymbol{u}) .$$

Since in our context we are not aware of the dynamics of the fluctuations and we are not even interested on them, we can just rely on the *Reynolds* averaged velocity $\overline{\boldsymbol{u}}$:

$$\boldsymbol{m} = \boldsymbol{c}(\boldsymbol{\mu}, \overline{\boldsymbol{u}}) .$$

During the online stage, velocity is just expressed as $\overline{\boldsymbol{u}}_r = \sum_{i=1}^{N_u} \overline{a}_i(\boldsymbol{\mu})\overline{\boldsymbol{\psi}}_i(\boldsymbol{x})$. Thus, since the modal basis functions are fixed, the only varying part for the averaged velocity is the one related to the expansion coefficients. This aspect leads to an evaluation of the reduced coefficients of Equation 20 only depending on the parameters and on the reduced coefficients for velocity:

$$\boldsymbol{m} = \boldsymbol{m}\left(\boldsymbol{\mu}, \boldsymbol{a}(\boldsymbol{\mu})\right) .$$

The easiest approximation we may think about is to just rely on the parameters of the problem as a dependency for the eddy viscosity:

$$\begin{aligned} \boldsymbol{a} &= \boldsymbol{a}(\boldsymbol{\mu}) \\ \boldsymbol{m} &= \boldsymbol{m}\left(\boldsymbol{\mu}, \boldsymbol{a}(\boldsymbol{\mu})\right) \end{aligned} \quad \Rightarrow \quad \boldsymbol{m} = \boldsymbol{m}\left(\boldsymbol{\mu}\right) . \tag{21}$$

This choice is obviously a strong assumption and simplification but it works accurately for cases very sensitive to the parametrization and when velocity dynamics do not exhibit a too wide variability.

To reach the stability, the same principles employed and explained in subsection 3.2 are used for both velocity and pressure so that pressure gets under relaxed directly while velocity gets treated by the under relaxation of the momentum equation. The results included in this chapter refers to the studies in [149].

Algorithm 3 The Reduced Order turbulent SIMPLE algorithm

Input: first attempt reduced pressure and velocity coefficients $\overline{\boldsymbol{b}}^{\star}$ and $\overline{\boldsymbol{a}}^{\star}$; modal basis functions matrices for pressure and velocity Θ and Ψ

Output: reduced pressure and velocity fields $\overline{p}_r$ and $\overline{\boldsymbol{u}}_r$

1: From $\overline{\boldsymbol{b}}^{\star}$ and $\overline{\boldsymbol{a}}^{\star}$, reconstruct reduced fields $\overline{p}^{\star}$ and $\overline{\boldsymbol{u}}^{\star}$:

$$\overline{p}^{\star} = \Phi^T \overline{\boldsymbol{b}}^{\star}, \qquad \overline{\boldsymbol{u}}^{\star} = \Psi^T \overline{\boldsymbol{a}}^{\star};$$

2: Evaluate the eddy viscosity field ν_{tr};

3: Momentum predictor step : assemble the momentum equation, project and solve it to obtain a new reduced velocity coefficients $\boldsymbol{a}^{\star\star}$:

$$(\boldsymbol{\psi}_i, \boldsymbol{A}\overline{\boldsymbol{u}}^{\star} - \boldsymbol{H}(\overline{\boldsymbol{u}}^{\star}) + \boldsymbol{\nabla}\overline{p}^{\star})_{L^2(\Omega)} = 0;$$

4: Reconstruct the new reduced velocity $\overline{\boldsymbol{u}}^{\star\star}$ and calculate the off-diagonal component $\boldsymbol{H}(\overline{\boldsymbol{u}}^{\star\star})$;

5: Pressure correction step: project pressure equation to get new reduced pressure coefficients $\boldsymbol{b}^{\star\star}$:

$$(\varphi_i, \boldsymbol{\nabla} \cdot [\boldsymbol{A}^{-1}\boldsymbol{\nabla}\overline{p}] - \boldsymbol{\nabla} \cdot [\boldsymbol{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{u}}^{\star\star})])_{L^2(\Omega)} = 0;$$

Then correct the velocity explicitly after having reconstructed the new pressure $\overline{p}^{\star\star}$;

6: Relax the pressure field and the velocity equation with the prescribed under-relaxation factors α_p and α_u, respectively. The under-relaxed fields are called p^{ur} and $\boldsymbol{u}^{ur}$;

7: **if** convergence **then**

8: $\overline{\boldsymbol{u}}_r = \overline{\boldsymbol{u}}^{ur}$ and $\overline{p}^{\star} = \overline{p}^{ur}$;

9: **else**

10: Assemble the conservative face fluxes F_f:

$$F_f = \overline{\boldsymbol{u}}_f \cdot \boldsymbol{S}_f;$$

11: set $\overline{\boldsymbol{u}}^{\star} = \overline{\boldsymbol{u}}^{ur}$ and $\overline{p}^{\star} = \overline{p}^{ur}$;

12: iterate from step 1.

13: **end if**

4.3 Incompressible turbulent Navier-Stokes problems: physical parametrization

4.3.1 Radial Basis Function interpolation for the eddy viscosity

In subsection 4.2 and in particular into Algorithm 3, the necessity for a reconstruction strategy related to the eddy viscosity has been exposed. In this paragraph we will show how it is possible to exploit a Radial Basis Function interpolation also for what concerns the ν_t expansion coefficients.

This approach will be here employed only for simple physical parametrization problems so that we can assume Equation 21 to be a reasonable assumption. What we are trying to approximate is then the map capable of linking parameter values to eddy viscosity coefficients values. It reads:

$$\boldsymbol{\mathcal{F}} : \mathbb{R}^{dim(\boldsymbol{\mu})} \rightarrow \mathbb{R}^{N_{\nu_t}} \ .$$

In other words we want to approximate the coefficients for the modal expansion of ν_t so that the field can be reconstructed as:

$$\nu_{tr} = \sum_{i=0}^{N_{\nu_t}} m_i(\boldsymbol{\mu}) \zeta_i(\boldsymbol{x}) \ .$$

At this point it has to be recalled that RBF interpolations are only able to provide scalar valued approximations. For this reason what can actually be done is to approximate the mapping for the coefficients one at a time:

$$\mathcal{F}_i = \mathbb{R}^{dim(\boldsymbol{\mu})} \rightarrow \mathbb{R} \quad \text{for } i = 1, \dots, N_{\nu_t} \ .$$

The first step is to project all the offline solutions for the eddy viscosity over the modal basis functions obtained trough a POD:

$$\boldsymbol{S}_{\nu_t} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_{\boldsymbol{\mu}}}] \quad \rightarrow \quad \boldsymbol{M}_{\nu_t} = [\boldsymbol{m}_1, \dots, \boldsymbol{m}_{N_{\boldsymbol{\mu}}}] \ .$$

All the couples $(\overline{\boldsymbol{\mu}}_i, \boldsymbol{m}_{ij})$ for $i = 1, \dots, N_{\boldsymbol{\mu}}$ and $j = 1, \dots, N_{\nu_t}$ can then be used as interpolation points to set the RBF.

The complete procedure is exposed in Algorithm 4.

What is good in this procedure is the fact that it is not bounded in any way by the choice of the turbulence model. This is a great advantage since the whole reduced

Algorithm 4 The Radial Basis Function interpolation for ν_t

Input: eddy viscosity snapshots matrix $\boldsymbol{S}_{\nu_t}$, eddy viscosity modal basis functions $\boldsymbol{Z}$, set of training parameters $\mathbb{P}_h$, new parameter to be evaluated $\boldsymbol{\mu}$

Output: reduced eddy viscosity field $\nu_{tr}(\boldsymbol{\mu})$

1: Project all the snapshots over the modal basis functions:

$$\boldsymbol{S}_{\nu_t} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_{N_{\boldsymbol{\mu}}}] \quad \rightarrow \quad \boldsymbol{M}_{\nu_t} = [\boldsymbol{m}_1, \dots, \boldsymbol{c}_{N_{\boldsymbol{\mu}}}] \; ;$$

2: Construct the connectivity matrix $\boldsymbol{\Phi}^{\nu_t} \in \mathbb{R}^{N_p \times N_p}$ as $\boldsymbol{\Phi}^{\nu_t}_{ij} = ||\overline{\boldsymbol{\mu}}_i - \overline{\boldsymbol{\mu}}_j||$ where $\overline{\boldsymbol{\mu}}_i, \overline{\boldsymbol{\mu}}_j \in \mathbb{P}_h$;

3: Construct the parameter matrix $\boldsymbol{P}^{\nu_t} \in \mathbb{R}^{N_p \times (1+dim(\boldsymbol{\mu}))}$ where each row is filled as $[1 \; \boldsymbol{\mu}]$;

4: Assemble the RBF matrix as:

$$\begin{bmatrix} \boldsymbol{\Phi}^{\nu_t} & \boldsymbol{P}^{\nu_t} \\ \boldsymbol{P}^{\nu_t T} & \boldsymbol{0} \end{bmatrix} \; ; \tag{22}$$

5: Solve the RBF system for every single coefficient in the ν_t expansion:

$$\begin{bmatrix} \boldsymbol{\Phi}^{\nu_t} & \boldsymbol{P}^{\nu_t} \\ \boldsymbol{P}^{\nu_t T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j^{\nu_t} \\ \boldsymbol{\alpha}_j^{\nu_t} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{m}}_j^T \\ \boldsymbol{0} \end{bmatrix} \; , \tag{23}$$

where $\overline{\boldsymbol{c}}_j$ stands for the j-th row of the matrix $\boldsymbol{C}_{\nu_t}$;

6: Approximate every single eddy viscosity coefficient as:

$$\boldsymbol{m}_j(\boldsymbol{\mu}) = \sum_{i=0}^{N_{\boldsymbol{\mu}}} \left(\boldsymbol{\omega}_j^{\nu_t}\right)_i \phi\left(\|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}_i\|\right) + q_j(\boldsymbol{\mu}) \; , \tag{24}$$

for $j = 1, \dots, N_{\nu_t}$, being $q_j(\boldsymbol{\mu})$ a linear polynomial that can be expressed as $q_j(\boldsymbol{\mu}) = \left(\boldsymbol{\alpha}_j^{\nu_t}\right)_0 + \sum_{i=0}^{dim(\boldsymbol{\mu})} \left(\boldsymbol{\alpha}_j^{\nu_t}\right)_i (\boldsymbol{\mu})_i$ where $(\bullet)_i$ stands for the i-th component of a vectorial quantity;

7: Reconstruct the approximated eddy viscosity field by its modal expansion:

$$\nu_{tr}(\boldsymbol{\mu}) = \sum_{i=0}^{N_{\nu_t}} m_i(\boldsymbol{\mu}) \zeta_i(\boldsymbol{x}) \; .$$

procedure is not depending on the turbulence equations. For this reason it is not necessary to modify the online phase in any way whenever the offline strategy for the evaluation of the eddy viscosity varies. For example the employment of a $k-\omega$ model instead of a $k-\epsilon$ one does not affect the online scheme. In principle even mixed solutions may be used for the construction of the snapshots matrix $\boldsymbol{S}_{\nu_t}$ since we are only interested in adding some dissipation by the introduction of the eddy viscosity term without paying any attention to the method used to obtain it.

What, on the contrary, may result inefficient in Algorithm 4 is the necessity, in practice, to apply the RBF procedure N_{ν_t} times, one for every expansion coefficient. In any case the bad impact of such an approach in the whole scheme is mitigated by the fact that the matrix into Equation 22 has to be assembled just once while Equation 23 has to be solved just N_{ν_t} times. These evaluations are performed during the offline phase while the online stage just requires N_{ν_t} evaluations of Equation 24. For this reason the whole machinery retains good efficiency properties even for cases requiring a not negligible number of modal basis for the reconstruction of the eddy viscosity field.

4.3.2 Turbulent back step problem with a variable viscosity

In this paragraph we aim at presenting a first turbulent incompressible test case. For this scope we decided to use the same geometry used in Figure 4. The domain is also discretized by the use of the same mesh. The problem to be solved is the following one:

$$\begin{cases} \nabla \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}) - \nabla \cdot [(\nu + \nu_t)(\nabla \overline{\boldsymbol{u}})] + \nabla p = 0 & \text{in } \Omega \\ \nabla \cdot \overline{\boldsymbol{u}} = 0 & \text{in } \Omega \\ \overline{\boldsymbol{u}} = [1, 0]^T & \text{on } \Gamma_{in} \\ \nu \dfrac{\partial \overline{\boldsymbol{u}}}{\partial \boldsymbol{n}} - \overline{p}\boldsymbol{n} = 0 & \text{on } \Gamma_{out} \end{cases} .$$

The offline eddy viscosity solution is obtained through a $k-\epsilon$ turbulence model since we are dealing with an inner flow problem where we are not interested on stresses near the wall.

The kinematic viscosity ν can vary in the manyfold $\mathbb{P} \equiv [10^{-4}, 10^{-5}]$. This means that the *Reynolds* number for this test case can be expressed as

$$Re = \frac{\boldsymbol{u}_\infty L}{\nu} \in [7 \times 10^4, 7 \times 10^5] .$$

Clearly, even though the test case looks pretty similar to the one reported in the previous chapter, the physical context is completely different. In fact we are dealing here with a turbulent problem where the techniques reported in Algorithm 2 would not succeed in providing reliable solutions. Since the *Reynolds* number is abundantly higher with respect to any possible value giving rise to a laminar flow, turbulence is here playing an important role.

The offline phase has been carried out for 50 equispaced ν values in $\mathbb{P}$.

Looking at Figure 23, we can see that the decay of the eigenvalues is performing well in the sense that, by the use of just 15 modal basis functions, it is possible to retain almost the whole information content described by the snapshots. The same counts for the eddy viscosity reconstruction with respect to Figure 24. The reconstruction of the eddy viscosity for this problem is performed by following what has been exposed in subsubsection 4.3.1. It is important to underline here that the turbulence model employed for the offline high fidelity snapshots collection is not relevant in any way for what concerns the online approximation for the ν_t field. Once again 15 modal basis functions have been selected to span the eddy viscosity solutions manifold.

20 kinematic viscosity values have been used for the online phase. They have been selected randomly in the same range used for the offline stage $\mathbb{P}$.

In Figure 25, Figure 26 and Figure 27 are reported pressure, velocity and eddy viscosity solutions respectively obtained through the use of Algorithm 3 for $\nu = 5.12 \times 10^{-05}$ which means $Re = 1.37 \times 10^5$.

Figure 28 provides a detailed image on the accuracy of the method: pointwise error fields are reported for both pressure, on the left, and velocity, on the right, in the top row while the errors between offline and online solutions for the eddy viscosity is reported in the bottom part of the figure.

Figure 29, on the contrary, shows the L^2 norm errors for pressure, velocity and eddy viscosity for every different ν in the online parameter set. The established method shows a good accuracy and reliability for all the different fields providing very good results also for what concerns interpolated eddy viscosity solutions.

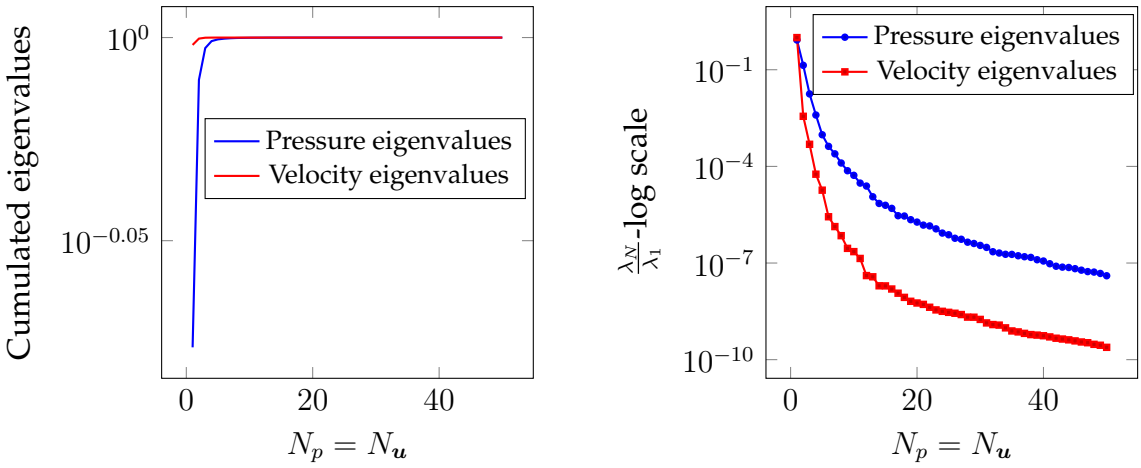


Figure 23: Eigenvalues trends for both pressure and velocity: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

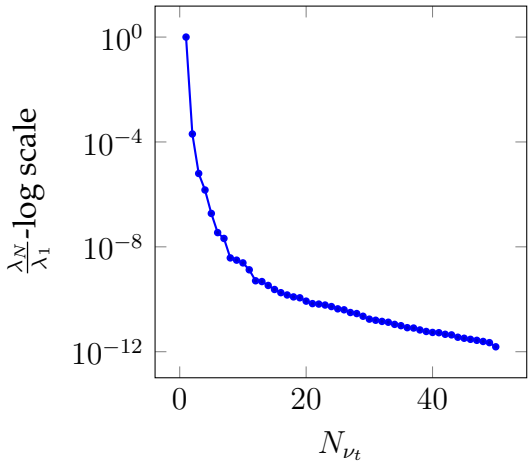


Figure 24: Eigenvalues trend for the eddy viscosity.

Figure 25: Comparison between pressure solutions: high fidelity on the left, reduced order on the right.

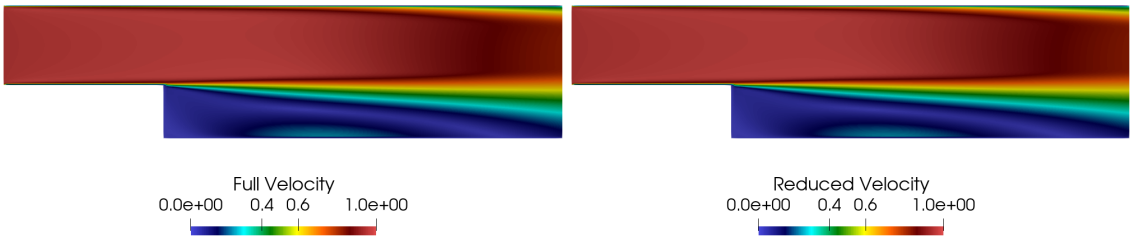


Figure 26: Comparison between velocity solutions: high fidelity on the left, reduced order on the right.

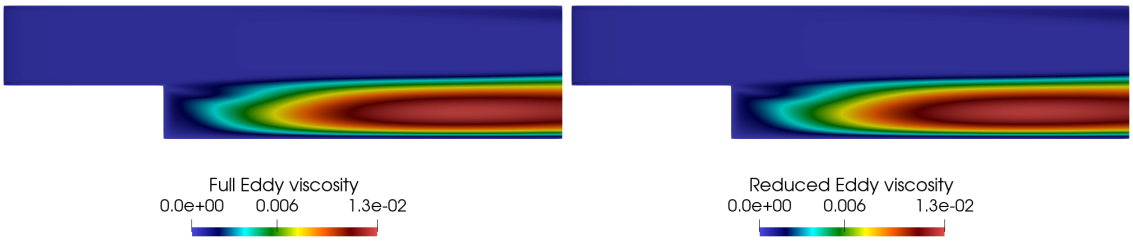


Figure 27: Comparison between eddy viscosity solutions: high fidelity on the left, reduced order on the right.

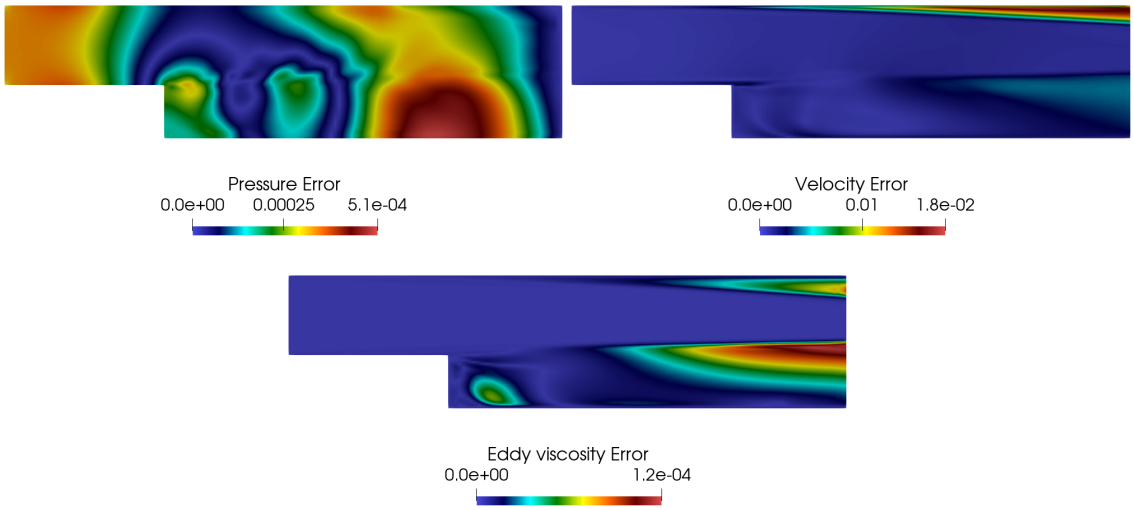


Figure 28: Pointwise error fields comparing full order and reduced order solutions for pressure on the left, velocity on the right and eddy viscosity on bottom.

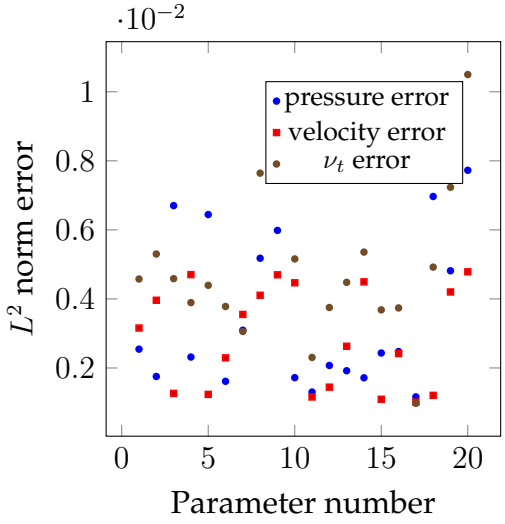


Figure 29: L^2 norm error for every different parameter value.

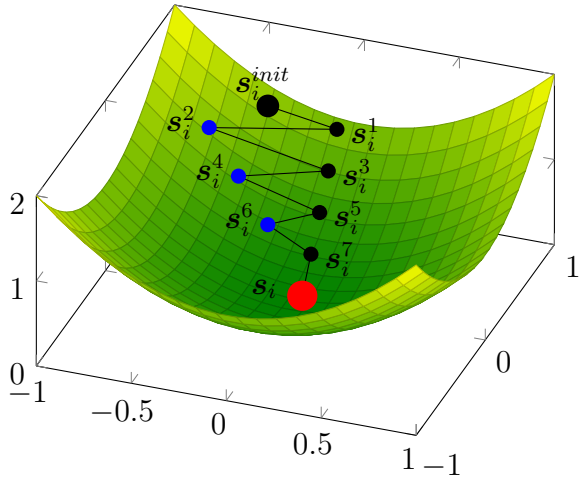


Figure 30: Scheme of the snapshots selection for $\Delta = 2$: black dots are discarded intermediate solutions, blue dots are saved intermediate solutions while the red dot represents the final solution.

4.4 Incompressible turbulent Navier-Stokes problems: geometrical parametrization

4.4.1 Snapshots selection for the Proper Orthogonal Decomposition

As we have introduced in subsection 3.2, subsection 4.2 and as we will see in subsection 5.2, in this work we deal with a segregated approach for both the online and the offline phase. This means that the set of equations, also at the reduced level, are iterated until convergence is reached. Since the solution fields during these iterations may vary a lot, from the first attempt for the variables to last resolution, the information contained into the converged snapshots is not sufficient to ensure the correct reduced reconstruction of the path to the global minimum of the system,i.e. the correct solution.

To improve convergence performances, then, we decided to enrich the set of snapshots by saving a certain amount of intermediate solutions s_i^j obtained during the offline iterations. The distance between exported intermediate solutions is set to Δ (see Figure 30). By adding some non-physical solutions to the snapshots matrix, which is what is happening by inserting non-converged fields, we are somehow polluting the physical content but the convergence propertied of the algorithm are improved a lot. To reach a balance between convergence and reliability, Δ can be varied and the total amount N_{int} of selected intermediate solutions can be

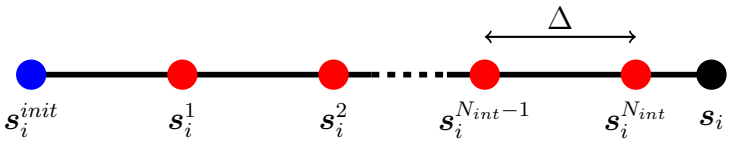


Figure 31: Scheme of the snapshots selection for every parameter $\boldsymbol{\mu}_i$: all red and black dots are collected together to compose the train set. Here $\boldsymbol{s}_i^{init}$ is the first attempt solution, $\boldsymbol{s}_i^j$ is the j-th iteration solution while $\boldsymbol{s}_i$ is the final converged snapshot.

modified. The new snapshots matrix then reads:

$$\boldsymbol{S} = \left[\boldsymbol{s}_1^1, \boldsymbol{s}_1^2, \ldots, \boldsymbol{s}_1^{N_{int}}, \boldsymbol{s}_1, \boldsymbol{s}_2^1, \boldsymbol{s}_2^2, \ldots, \boldsymbol{s}_2^{N_{int}}, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{N_\pi}^1, \boldsymbol{s}_{N_{\boldsymbol{\mu}}}^2, \cdots, \boldsymbol{s}_{N_{\boldsymbol{\mu}}}^{N_{int}}, \boldsymbol{s}_{N_{\boldsymbol{\mu}}}\right]$$

where $\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{j}}$ is the solution obtained at the $(j \cdot \Delta)$-th iteration for the i-th offline parameter (see Figure 31).

4.4.2 Neural Networks for the eddy viscosity

As introduced in subsection 4.2, turbulence effects are always depending on the velocity and in particular on its fluctuations. The method exposed in subsubsection 4.3.1 is simple to be constructed but presents some leaks for what concerns the physical correlation between turbulent viscosity and velocity. In fact these two quantities are just related between each others through the parameter:

$$\boldsymbol{m}(\boldsymbol{\mu}) \overset{\boldsymbol{\mu}}{\longleftrightarrow} \overline{\boldsymbol{a}}(\boldsymbol{\mu}) \ .$$

For more complex test cases, where the eddy viscosity is not just playing a role as an additional dissipation term, useful to keep the amount of kinetic energy into the system under control, but is actually modifying velocity behaviours and thus velocity solutions shapes, this very mild connection between the two is no more sufficient. Moreover we would like to find out a strategy capable of approximating the reduced eddy viscosity coefficients by the use of a single evaluation so that it is possible to rely on a more efficient technique with respect to the RBF one, requiring N_{ν_t} evaluations for every new provided $\boldsymbol{\mu}$.

To achieve all the aforementioned requirements for the online phase to evaluate the ν_t field, we decided to rely on a data driven scheme.

Let us recall the modal expansion used for the turbulent viscosity:

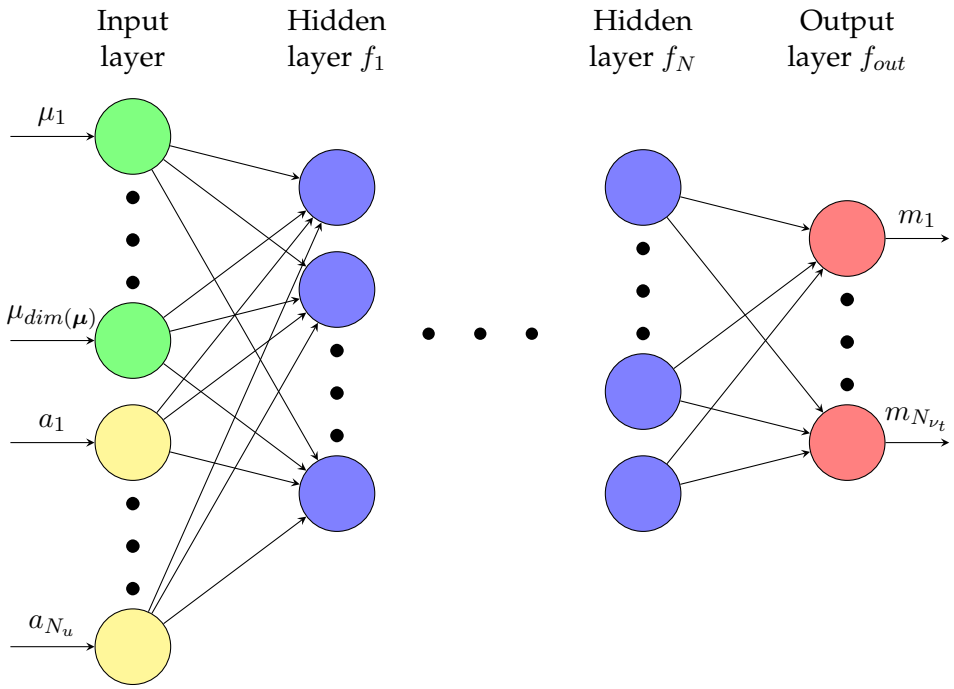


Figure 32: Schematic perspective of a fully connected neural network composed by an input layer, an undefined number N of hidden layers and an output layer, linking parameters μ_i and reduced velocity coefficients a_i to reduced eddy viscosity coefficients c_i, being $dim(\boldsymbol{\mu})$ the number of parameters possibly existing in the problem.

$$\nu_{tr} = \sum_{i=1}^{N_{\nu_t}} m_i(\boldsymbol{x})\zeta_i(\boldsymbol{\mu}) \ ,$$

where N_{ν_t} stands for the number of basis functions selected to reconstruct the eddy viscosity field, m_i are the coefficients depending only on the position $\boldsymbol{x}$ while ζ_i are the ν_t basis functions depending only on the parameter. Also in this case the first step consists on calculating and storing a certain number N_p of ν_t solutions at the offline stage. These snapshots are then again collected into the $\boldsymbol{S}_{\nu_t}$ matrix and used, as explained in subsection 2.3, to obtain the requested basis functions $\zeta_i(\boldsymbol{\mu})$.

What differs from the previous approach is the approximation of the spacial coefficients: they are evaluated through a Neural Network (NN) scheme, linking the parameters of the problem $\boldsymbol{\mu}_i$ and the reduced velocity coefficients a_i to the eddy viscosity ones c_i.

The scope of the Neural Network is the approximation of the map:

$$\boldsymbol{\mathcal{F}} : \mathbb{R}^{dim(\boldsymbol{\mu})+N_u} \rightarrow \mathbb{R}^{N_{\nu_t}} \ ,$$

so that $\boldsymbol{\mathcal{F}}(\boldsymbol{\mu}, \overline{\boldsymbol{a}}\boldsymbol{\mu}) = \boldsymbol{m}$.

A reduced problem based on such a construction is thus completely independent on the choice of the turbulence model and point 2 into Algorithm 3 can be performed in an efficient way. Let us recall here that this would not have been the case if turbulence equations were projected: in case there was the necessity of changing the adopted turbulence model, all the architecture had to be modified.

In this work we decided to only select fully connected Neural Networks composed by an input layer, a variable number of hidden layers and an output layer. The input vector $\boldsymbol{z}$ and output vector $\boldsymbol{m}$ are defined as mentioned before:

$$\boldsymbol{z} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{dim(\boldsymbol{\mu})} \\ a_1 \\ \vdots \\ a_{N_u} \end{bmatrix} \ , \qquad \boldsymbol{m} = \begin{bmatrix} m_1 \\ \vdots \\ m_{N_{\nu_t}} \end{bmatrix} \ .$$

Neural Networks have always to be firstly trained and then evaluated. The training phase starts by the projection of the snapshots contained into $\boldsymbol{S}_{\nu_t}$, over their

own basis functions ζ_i, to obtain the set of real coefficients $\{\boldsymbol{m}_i\}_{i=1}^{N_p}$. They can be compared with the NN estimated coefficients $\{\tilde{\boldsymbol{m}}_i\}_{i=1}^{N_p}$ into a loss function to target the training procedure. We adopted one of the most commonly used quadratic loss function ℓ:

$$\ell = ||\boldsymbol{m} - \tilde{\boldsymbol{m}}||_{L^2} \ .$$

The quantity $\mathcal{L}$ to be minimized during the training of the network is the sum of the loss function evaluated for all the different snapshots:

$$\mathcal{L} = \sum_{i=1}^{N_p} ||\boldsymbol{m}_i - \tilde{\boldsymbol{m}}_i||_{L^2} \ .$$

The coefficients estimated by the network can be written as:

$$\tilde{\boldsymbol{m}} = f_{out}\left(\boldsymbol{W}_{out} f_N\left(\cdots\left(\boldsymbol{W_2} f_1\left(\boldsymbol{W_1}\boldsymbol{z} + \boldsymbol{b}_1\right) + \boldsymbol{b}_2\right)\cdots\right) + \boldsymbol{b}_{out}\right) \ , \tag{25}$$

where f_1, f_2 and f_{out} are the activation functions used into the neurons, $\boldsymbol{W_1}$, $\boldsymbol{W_2}$ and $\boldsymbol{W_{out}}$ are the corresponding weights while $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ and $\boldsymbol{b}_{out}$ are the biases, related to the first and the second hidden layer and to the output layer respectively. For the hidden layers the best performing activation function is hard to be guessed. Different applications may require different activation functions to the point that a bad selection on the activation function may cause the whole scheme not working. For this reason the employed activation functions will be declared later on for every specific application.

The training procedure is resumed into Figure 33.

For what concerns the training procedure, two different parameters sets have to be provided:

- $\mathbb{P}_{train}$: the set of parameters used for the backward feeding training loop of the network;
- $\mathbb{P}_{test}$: a set of independent parameters that follows the same probability distribution as the training dataset, used to provide information on the generalizing capability of the network.

Both of them are necessary. The first one is the one providing the real information on how to approximate the desired data. Since there is not a fixed rule on how

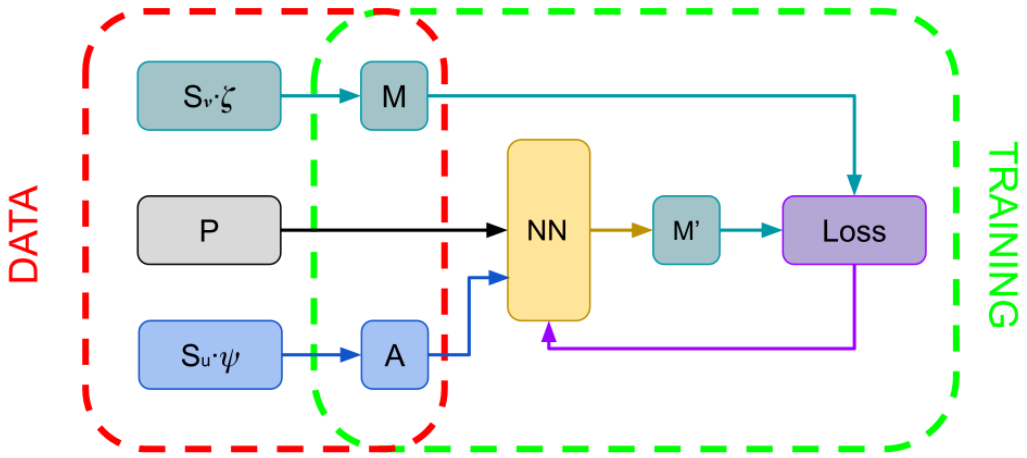


Figure 33: Schematic diagram for the training of the neural network.

many training epochs are required for a good approximation, an over estimation on this quantity would lead to a network which can perform in a very accurate way when dealing with parameters included into the training dataset but it would perform really badly when trying to approximate the solution for a parameter $\boldsymbol{\mu} \notin \mathbb{P}_{train}$.

This issue can be overtaken by comparing the loss functions evaluated for training and testing sets. While both of them are decreasing, this means that the generalization capability of the network is still increasing while the training procedure has to be stopped in case the training loss function is still decreasing while the test one changes trend.

The training phase for a Neural Network can be computationally very expensive, depending on how large and how deep the architecture we have put in place is. In any case this is not affecting the scope of this work since it is performed during the offline phase. Moreover it has to be underlined that the training procedure has just to be executed once: from that point on, the network has just to be evaluated by the use of Equation 25.

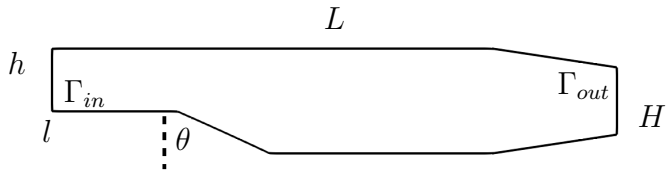


Figure 34: Representation of the geometrical configuration for the domain of interest.

4.4.3 Turbulent back step problem with a variable slope angle

This section will be concerning a turbulent geometrical parametrization test case. More in detail we focused on a back step problem where the slope of the step can be varied. We also added a convergent section at the outflow of the domain to improve convergence properties: this solution gives rise to an out pointing negative pressure gradient so that we get rid of possible reverse flows at the outlet due to an increasing pressure capable of hampering the out flowing in some cases.

The domain for this test case is presented in Figure 34 where we have $h = 1m$, $L = 7m$, $H = 1.7m$, $l = 2m$ and the length of the convergent outflow section equal to $2m$. The slope angle for what regards the step can be varied in the range $\mathbb{P} \equiv [0°, 75°]$. This means that the geometry is varying a lot together with the behaviour of the fluid: when the angle is close to zero a big recirculation bubble is supposed to be there while this phenomenon is no more taking place when the domain becomes smoother. This test case is also a good way to prove the efficiency of the mesh motion strategy we are using, as explained in subsubsection 3.4.1, since we are dealing with big deformations of the bottom part of the channel.

The problem is described by the following equations:

$$\begin{cases} \nabla \cdot (\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}) - \nabla \cdot [(\nu + \nu_t)(\nabla \overline{\boldsymbol{u}})] + \nabla p = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla \cdot \overline{\boldsymbol{u}} = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \overline{\boldsymbol{u}} = [1, 0]^T & \text{on } \Gamma_{in} \\ \nu \dfrac{\partial \overline{\boldsymbol{u}}}{\partial \boldsymbol{n}} - \overline{p}\boldsymbol{n} = 0 & \text{on } \Gamma_{out} \end{cases},$$

where $\nu = 0.001$.

As it has been pointed out in subsubsection 3.4.2, all the norms needed for the

application of the architecture provided in this chapter are based on a reference grid characterized by

$$\boldsymbol{\mu}_{mid} = \frac{\boldsymbol{\mu}_{min} + \boldsymbol{\mu}_{max}}{2} = 37.5^{\circ} \ .$$

The *Reynolds* number characterizing the dynamics of the problem can be evaluated taking into account both the fluid properties together with geometrical aspects as:

$$Re = \frac{\boldsymbol{u}_{\infty} L_t}{\nu} = 9 \times 10^3 \ ,$$

where L_t stands for the total length of the channel.

Since the range for the *Reynolds* number we are working at is on the border line between laminar and turbulent flows, we are forced to consider a turbulence closure model.

For the offline phase we selected 50 equispaced values of the parameter $\boldsymbol{\mu} \in \mathbb{P}$. Those values of the angle of the step are used to solve 50 different full order problems in order to construct the snapshots matrix. Both converged and intermediate snapshots are collected together with $\Delta = 50$, so that one intermediate step gets stored every 50 iterations.

By applying a POD procedure, we can obtain the modal basis functions we need to project the equations.

By analyzing Figure 35 we can notice that at least 25 modes have to be selected for ν_t in order to catch the main part of the information contained into the offline snapshots. For what regards pressure and velocity manifolds, they are here projected and then reconstructed using 35 basis functions.

Thus, a neural network has been constructed for the eddy viscosity approximation at every reduced SIMPLE algorithm step as explained in subsubsection 4.4.2.

The neural network employed here is composed by:

- an input layer, whose dimension is equal to the dimension of the reduced velocity, i.e. 35, plus one for the parameter;
- two hidden layers of dimension 256 and 64 respectively;
- an output layer of dimension 25 for the reduced eddy viscosity coefficients.

The net is a fully connected one. Moreover the neurons of the hidden layers are characterized by the employment of ReLU activation functions. For the training

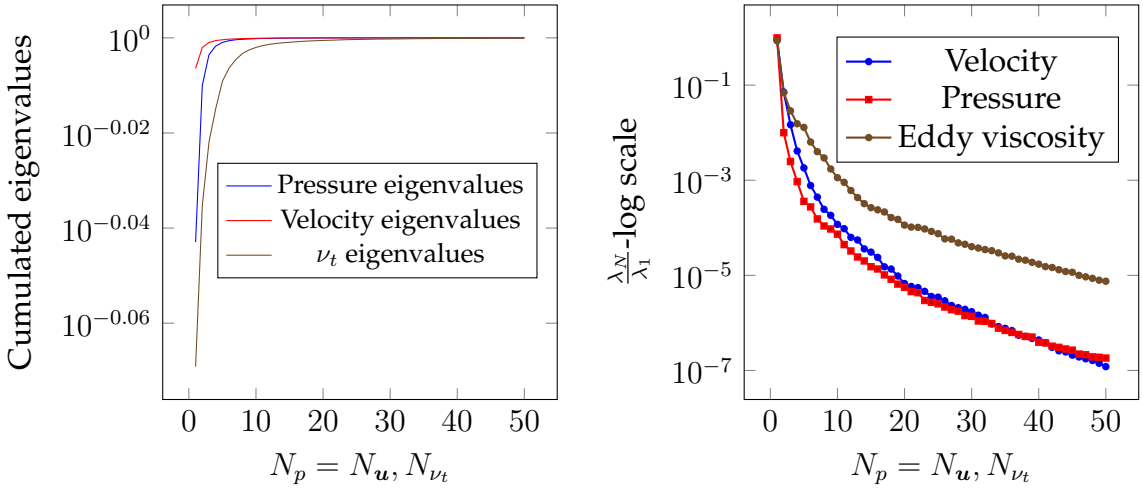


Figure 35: Eigenvalues trends for pressure, velocity and eddy viscosity: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

procedure, the Adam optimizer has been selected and 10^4 epochs have been fixed.

The training set is composed by both the intermediate and final solutions obtained during the offline phase, randomly selected. To control the training procedure, a test set has been selected too: 10 totally random new parameter values have been chosen and their related full solutions have been calculated, saving both final and intermediate steps, coherently with the offline snapshots used for training. Loss functions for both training and testing sets are reported in Figure 36.

Looking at Figure 36, it can be noticed that there is a nice agreement between train and test loss functions. This is a good indicator for the extrapolation capability of the net since the loss function evaluated for test set solutions is not far away from the one obtained for training solutions.

For the online phase 10 different angles have been randomly selected in the range $[5°, 65°]$.

Figures 37, 38, 39, 40, 41 and 42 show a comparison between high fidelity versus reduced order solutions for velocity, pressure and eddy viscosity. The results are reported for two different parameter values: a very small one, where the deformation of the mesh is almost negligible, and a high one characterized by a strong mesh deformation. In both cases the pointwise errors show a high reliability of the method even though higher discrepancies can be noticed in the zones of the grid affected by a high non-orthogonality of the cells.

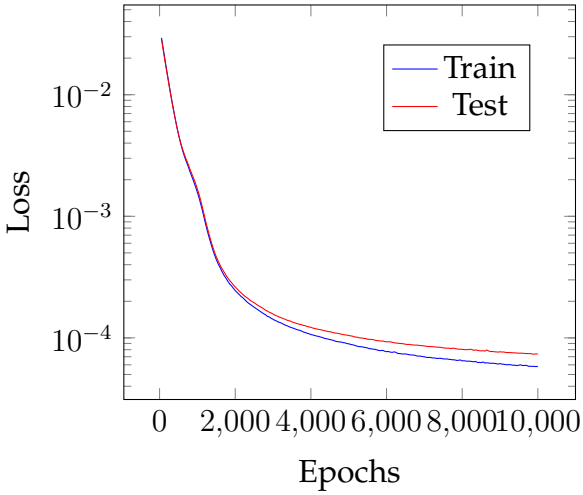


Figure 36: Loss function decay for both train and test sets.

Figure 43 present the behaviour of the L^2 norm relative error average, over all the 10 online solutions, for both velocity and pressure, for different numbers of modal basis functions used to approximate them, while keeping $N_{\nu_t} = 25$. This plot clearly shows that an amount of modes lower than 25 provides results that are not accurate enough. On the contrary, even the usage of more than 40 modal basis functions is not advisable since the numerical error is capable of overwhelming the additional information provided by the last added modes.

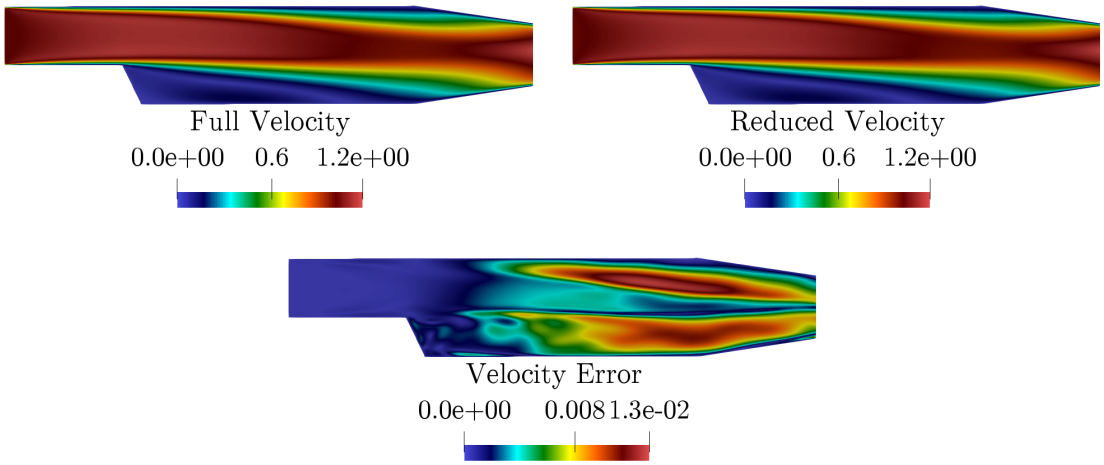


Figure 37: Comparison between full order and reduced order velocity solutions on top and pointwise error between them on bottom for $\theta = 10.7°$.

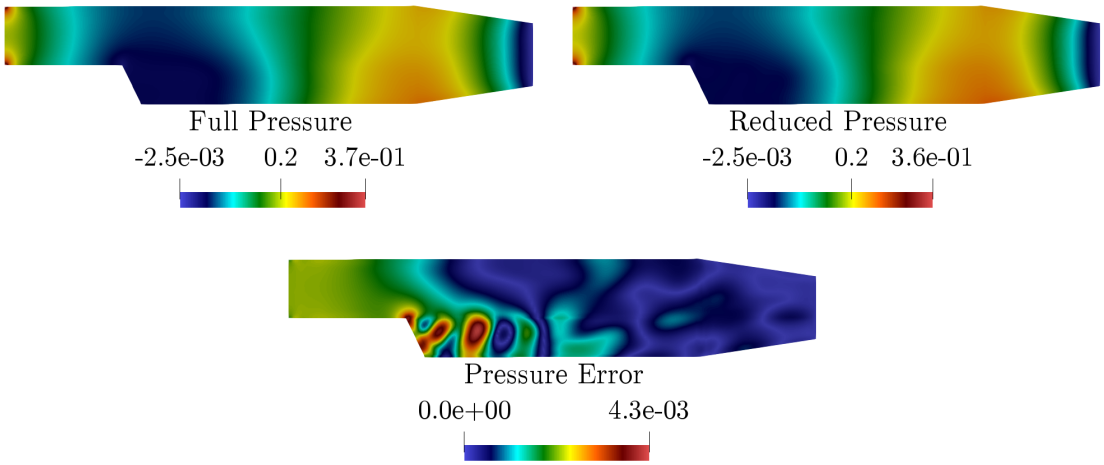


Figure 38: Comparison between full order and reduced order pressure solutions on top and pointwise error between them on bottom for $\theta = 10.7°$.

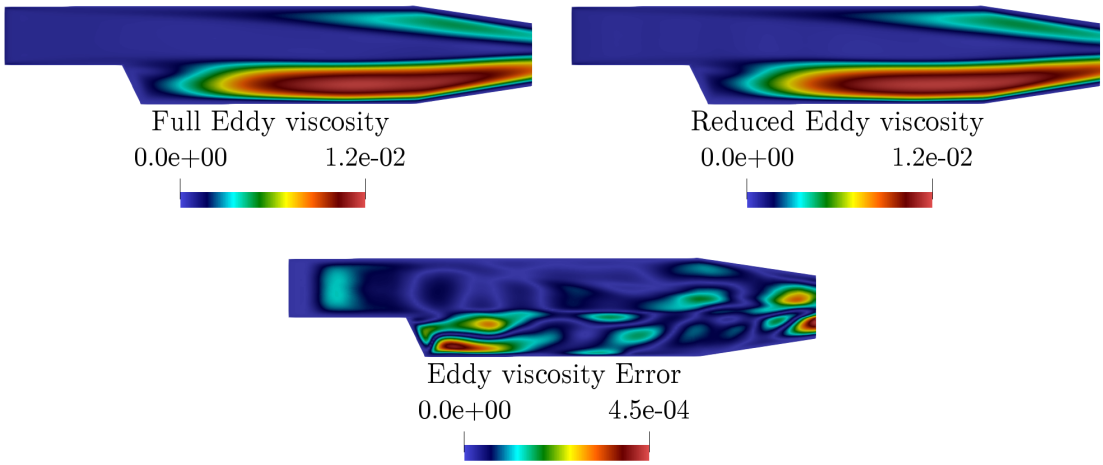


Figure 39: Comparison between full order and reduced order eddy viscosity solutions on top and pointwise error between them on bottom for $\theta = 10.7°$.

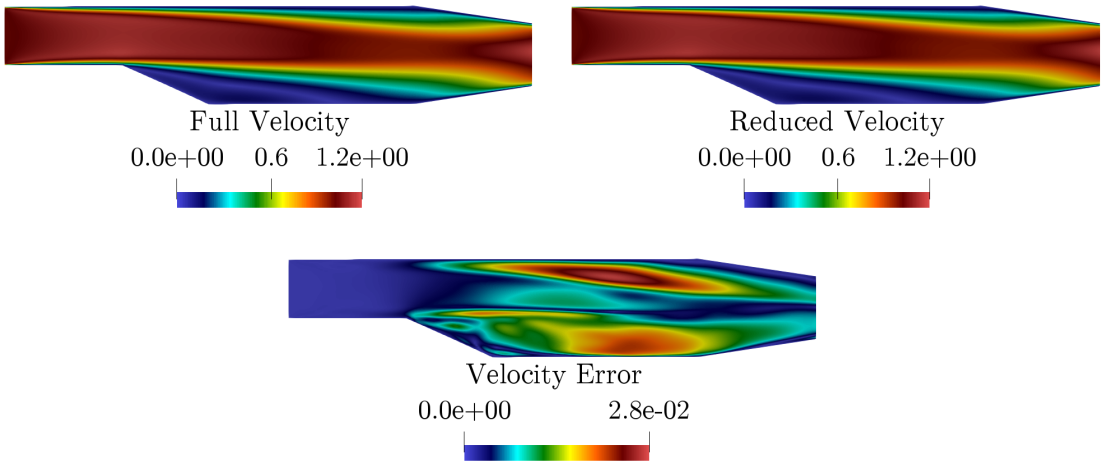


Figure 40: Comparison between full order and reduced order velocity solutions on top and pointwise error between them on bottom for $\theta = 63.2°$.

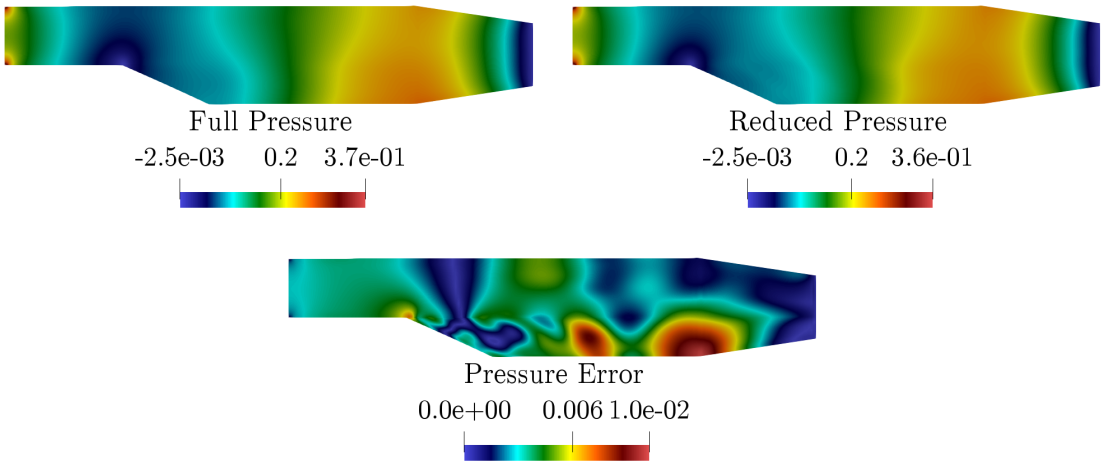


Figure 41: Comparison between full order and reduced order pressure solutions on top and pointwise error between them on bottom for $\theta = 63.2°$.

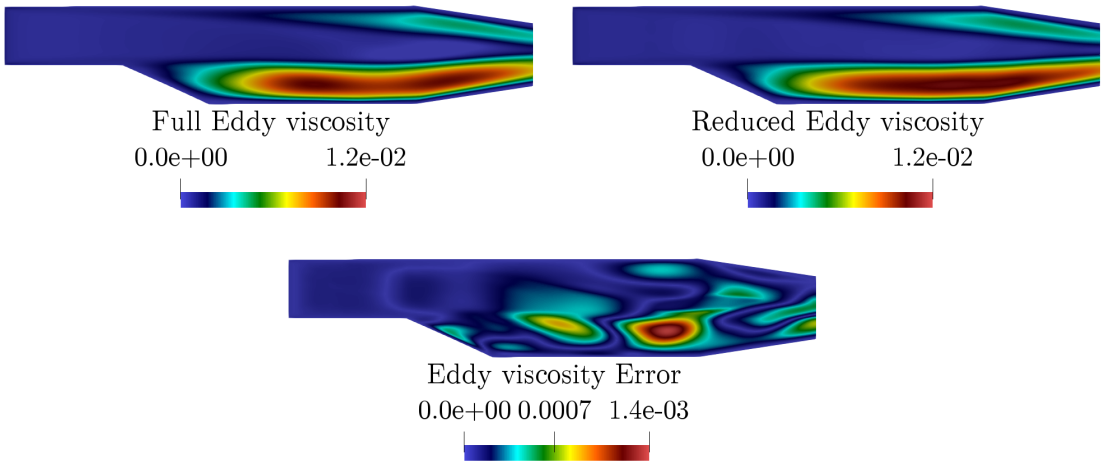


Figure 42: Comparison between full order and reduced order eddy viscosity solutions on top and pointwise error between them on bottom for $\theta = 63.2°$.

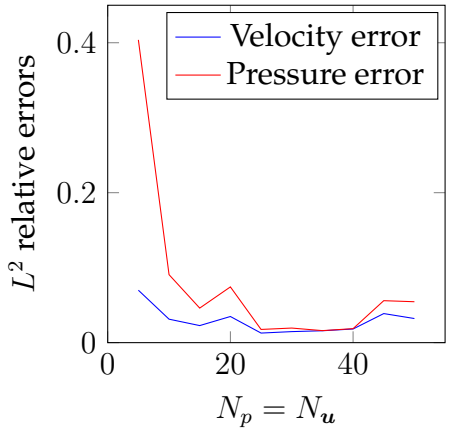


Figure 43: L^2 norm relative error for both velocity and pressure for $N_{\nu_t} = 25$.

4.5 Context and overview

This chapter has been devoted to incompressible turbulent flows and possible efficient methods for both physical and geometrical parametrization problems.

In particular the aim of this work was to find reliable methods capable of providing good reduced order approximations independent from the turbulence model employed during the offline phase. This is a goal dictated by the fact that there are many different approaches for turbulence resolution based on eddy viscosity models. This variety may lead to the necessity of changing the reduced order model whenever the turbulence model was changed, in case turbulence equations were used also at the reduced level to close the set of equations describing the phenomena. A first approach for such a purpose has been introduced in subsubsection 4.3.1 and tested in subsection 4.3 showing a good reliability. This approach, based on an RBF interpolation, is a good compromise between low computational cost and high accuracy. It provides accurate results for physical parametrization problems where the usage of converged solutions for the eddy viscosity, also for the first iterations of the algorithm, do not cause instabilities in the method.

This is not the case for geometrical parameter applications where the relation between the parameter and the solutions is much more complex. In these cases a step by step evaluation of the eddy viscosity field is needed to avoid a possible blow-up of the reduced SIMPLE algorithm introduced in subsection 4.2. For this reason an artificial intelligence approach has been proposed in subsubsection 4.4.2, where the eddy viscosity field gets reconstructed starting from velocity reduced coefficients together with the parameter by the use of a neural network. Moreover a different way to collect the snapshots to be used for the POD has been pointed out in subsubsection 4.4.1 so that the behaviour of the full order SIMPLE algorithm could have been emulated in a more accurate way by providing, to the reduced algorithm, also some information about intermediate steps to be taken to reach the right solution. This last architecture for the resolution of geometry parametrization test cases has been proved to be accurate and stable even for large grid deformation, as shown in subsection 4.4.

5 Compressible Navier-Stokes equations

The compressible Navier-Stokes equations are the most complete and widely applicable set of equations representing the conservation laws for a fluid in motion. The interested reader may refer to [133, 132, 110, 107, 40] for a complete description of the physical aspects related to compressibility. The equations presented in section 4 and even more the ones used in section 3 can be employed for the study of some specific fluid flows. In particular, incompressible Navier-Stokes equations can be employed only to describe fluid flows characterized by a ratio between the velocity of the flow and the speed of sound, namely the *Mach* number, in every point of the domain lower than 0.3. If this is the case, the interactions between the fluid particles can be almost neglected and all the thermodynamic effects are ignored. If this is not the case, an equation for the energy conservation has to be added to the momentum and continuity ones, together with a state equation needed to close the system.

Compressible laminar problems can exist for some specific conditions but in this chapter we will only focus on turbulent compressible problems since we will overtake the limits imposed for the use of incompressible Navier-Stokes equations because of a velocity increase. Usually higher velocities also imply a higher "disorder" into the domain of interest and, thus, the arise of turbulent phenomena.

The following sections are referred to smooth fluid flows where no discontinuities in the solutions are present. For this reason we will suppose the flow at every point of the domain not to exceed $Mach = 1$. The techniques we will present cannot be used, then, for supersonic flows; only subsonic or low transonic problems are suitable for the following approaches.

All the considerations reported in subsection 5.1 are general and can be used for every different kind of fluid. Anyway in this work, for sake of simplicity, we will consider only perfect gasses. This choice will not compromise the contents of this chapter since the state equation, as explicated in Algorithm 5, is used explicitly.

Many different approaches are possible for compressible parametrized fluid flows. Some are reported in [82, 12, 5].

The interested reader may, instead, find some reduced order model examples for supersonic flows in [39, 143, 80].

5.1 Equations and Finite Volume discretization

As it has been pointed out in subsection 4.1, we are not interested in fluid fluctuations, due to the presence of a chaotic motion, since they are not playing an important role from a macroscopic point of view. For this reason some averaging techniques have to be applied in order to retain only the most relevant part for all the involved variables.

More in details, for compressible flows it is not possible to take into consideration only the Reynolds averaging as we did in the previous chapter because it would lead to very unhandy terms containing mixed products between density fluctuations and other variables fluctuations.

To solve this issue, in this work, we will combine Favre averaging, for what concerns temperature, velocity and energy, together with Reynolds averaging, for what concerns pressure and density. The main averaging rules for both of aforementioned strategies are reported in the following box.

Averaging rules

Given a generic flow variable Φ and and the density variable ρ we can define both the Favre and the Reynolds averaging procedures.
Favre averaging rules:

$$\tilde{\Phi} = \frac{\overline{\rho\Phi}}{\overline{\rho}} , \qquad \Phi = \tilde{\Phi} + \Phi'' .$$

Reynolds averaging rules:

$$\overline{\Phi} = \frac{1}{T}\int_T \Phi(t)dt , \qquad \Phi = \overline{\Phi} + \Phi' .$$

From the above rules it follows:

$$\overline{\Phi'} = 0 , \qquad \tilde{\Phi'} \neq 0 , \qquad \overline{\rho\Phi''} = 0 , \qquad \overline{\rho\tilde{\Phi}} = \overline{\rho}\tilde{\Phi} .$$

We can now decompose all the variables into an averaged part and a fluctuating one, recalling all the different averaging techniques selected for all the different

variables:

$$p = \overline{p} + p' \,, \quad \rho = \overline{\rho} + \rho' \,,$$
$$\boldsymbol{u} = \tilde{\boldsymbol{u}} + \boldsymbol{u}'' \,, \quad e = \tilde{e} + e'' \,,$$
$$e_0 = \tilde{e_0} + e_0'' \,, \quad T = \tilde{T} + T'' \,,$$

where e stands for the internal energy, e_0 represents the total energy while T indicates the temperature.

For all the equations we will substitute the Favre decomposition in place of every Favre-decomposed variable. After that we will apply the Reynolds average to the whole equation.

CONTINUITY EQUATION:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \boldsymbol{u}] = 0 \,.$$

Favre decomposition:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)] = 0 \,.$$

Reynolds averaging:

$$\overline{\frac{\partial \rho}{\partial t}} + \overline{\nabla \cdot [\rho \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)]} = \frac{\partial \overline{\rho}}{\partial t} + \overline{\nabla \cdot (\rho \tilde{\boldsymbol{u}})} + \overline{\nabla \cdot (\rho \boldsymbol{u}'')} = \frac{\partial \overline{\rho}}{\partial t} + \nabla \cdot (\overline{\rho} \tilde{\boldsymbol{u}}) + \nabla \cdot \underbrace{\overline{(\rho \boldsymbol{u}'')}}_{0} = 0 \,.$$

Final equation:

$$\frac{\partial \overline{\rho}}{\partial t} + \nabla \cdot (\overline{\rho} \tilde{\boldsymbol{u}}) = 0 \,.$$

MOMENTUM EQUATION:

$$\underbrace{\frac{\partial \rho \boldsymbol{u}}{\partial t}}_{ⓐ} + \underbrace{\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u})}_{ⓑ} = \underbrace{\nabla \cdot \left[\mu \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right]}_{ⓒ} - \underbrace{\frac{2}{3} \nabla \left(\mu \nabla \cdot \boldsymbol{u}\right)}_{ⓓ} - \underbrace{\nabla p}_{ⓔ} \,.$$

Favre decomposition:

$$ⓐ : \frac{\partial \rho \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)}{\partial t} \,;$$
$$ⓑ : \nabla \cdot [\rho \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right) \otimes \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)] \,;$$
$$ⓒ : \nabla \cdot \left[\mu \left(\nabla \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)\right) + \mu \left(\nabla \left(\tilde{\boldsymbol{u}} + \boldsymbol{u}''\right)^T\right)\right] \,;$$

$$\textcircled{d} : \frac{2}{3}\nabla\left[\mu\nabla\cdot(\tilde{\boldsymbol{u}}+\boldsymbol{u}'')\right] \ ;$$
$$\textcircled{e} : \nabla p \ ;$$

Reynolds averaging:

$$\textcircled{a} : \overline{\frac{\partial\rho\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)}{\partial t}} = \frac{\partial\overline{\rho}\tilde{\boldsymbol{u}}}{\partial t} + \underbrace{\frac{\partial\overline{\rho\boldsymbol{u}''}}{\partial t}}_{0} \ ;$$

$$\textcircled{b} : \overline{\nabla\cdot\left[\rho\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)\otimes\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)\right]} = \nabla\cdot\left[\overline{\rho\tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}}} + \underbrace{\overline{\rho\tilde{\boldsymbol{u}}\otimes\boldsymbol{u}''}}_{0} + \underbrace{\overline{\rho\boldsymbol{u}''\otimes\tilde{\boldsymbol{u}}}}_{0} + \overline{\rho\boldsymbol{u}''\otimes\boldsymbol{u}''}\right] \ ;$$

$$\textcircled{c} : \overline{\nabla\cdot\left[\mu\left(\nabla\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)\right)+\mu\left(\nabla\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)^{T}\right)\right]} = \nabla\cdot\overline{\left[\mu\left(\nabla\tilde{\boldsymbol{u}}+\nabla\boldsymbol{u}''+\nabla\tilde{\boldsymbol{u}}^{T}+\nabla\boldsymbol{u}''^{T}\right)\right]} =$$
$$= \nabla\cdot\left[\mu\left(\nabla\tilde{\boldsymbol{u}}+\nabla\tilde{\boldsymbol{u}}^{T}\right)+\mu\overline{\left(\nabla\boldsymbol{u}''+\nabla\boldsymbol{u}''^{T}\right)}\right] \ ;$$

$$\textcircled{d} : \overline{\frac{2}{3}\nabla\left[\mu\nabla\cdot\left(\tilde{\boldsymbol{u}}+\boldsymbol{u}''\right)\right]} = \frac{2}{3}\nabla\left[\mu\nabla\cdot\tilde{\boldsymbol{u}}+\mu\nabla\cdot\overline{\boldsymbol{u}''}\right] \ ;$$
$$\textcircled{e} : \overline{\nabla p} = \nabla\overline{p} \ .$$

We can define:

$$\boldsymbol{\tau}(\boldsymbol{u}) = 2\mu\left[\frac{\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^{T}}{2}-\frac{1}{3}\nabla\cdot\boldsymbol{u}\boldsymbol{I}\right] \ .$$

It follows:

$$\textcircled{c}+\textcircled{d} = \nabla\cdot\left(\tilde{\boldsymbol{\tau}}+\overline{\boldsymbol{\tau}''}\right) \ ,$$

where $\tilde{\boldsymbol{\tau}} = \boldsymbol{\tau}(\tilde{\boldsymbol{u}})$ and $\boldsymbol{\tau}'' = \boldsymbol{\tau}(\boldsymbol{u}'')$.

Resulting equation:

$$\frac{\partial\overline{\rho}\tilde{\boldsymbol{u}}}{\partial t} + \nabla\cdot\left[\overline{\rho\tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}}} + \overline{\rho\boldsymbol{u}''\otimes\boldsymbol{u}''} - \tilde{\boldsymbol{\tau}} - \underbrace{\overline{\boldsymbol{\tau}''}}_{\star}\right] + \nabla\overline{p} = 0 \ .$$

Terms denoted by $\star$ can be neglected under reasonable assumptions:

$$\frac{\partial\overline{\rho}\tilde{\boldsymbol{u}}}{\partial t} + \nabla\cdot\left[\overline{\rho\tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}}} + \overline{\rho\boldsymbol{u}''\otimes\boldsymbol{u}''} - \tilde{\boldsymbol{\tau}}\right] + \nabla\overline{p} = 0 \ .$$

Moreover the term $\overline{\rho\boldsymbol{u}''\otimes\boldsymbol{u}''}$ is not known and has to be modeled:

$$\overline{\rho\boldsymbol{u}''\otimes\boldsymbol{u}''} = -\tilde{\boldsymbol{\tau}}_{turb} \ .$$

Final equation:

$$\frac{\partial \overline{\rho}\tilde{\boldsymbol{u}}}{\partial t} + \nabla \cdot [\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}} + \overline{p}\boldsymbol{I}] = 0 \, .$$

TOTAL ENERGY EQUATION:

$$\underbrace{\frac{\partial \rho e_0}{\partial t}}_{ⓐ} + \underbrace{\nabla \cdot (\rho \boldsymbol{u} e_0)}_{ⓑ} = -\underbrace{\nabla \cdot q}_{ⓒ} - \underbrace{\nabla \cdot (p\boldsymbol{u})}_{ⓓ} + \underbrace{\nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u})}_{ⓔ} \, .$$

Favre decomposition:

$$ⓐ : \frac{\partial \rho e_0}{\partial t} = \frac{\partial \rho \tilde{e}_0}{\partial t} + \frac{\partial \rho e_0''}{\partial t} \, ;$$

$$ⓑ : \nabla \cdot [\rho \, (\tilde{\boldsymbol{u}} + \boldsymbol{u}'') \, (\tilde{e}_0 + e_0'')] = \nabla \cdot [\rho \tilde{\boldsymbol{u}} \tilde{e}_0 + \rho \tilde{\boldsymbol{u}} e_0'' + \rho \boldsymbol{u}'' \tilde{e}_0 + \rho \boldsymbol{u}'' e_0''] \, ;$$

$$ⓒ : \nabla \cdot q \, ;$$

$$ⓓ : \nabla \cdot (p\boldsymbol{u}) = \nabla \cdot (p\tilde{\boldsymbol{u}}) + \nabla \cdot (p\boldsymbol{u}'') \, ;$$

$$ⓔ : \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}) = \nabla \cdot [(\, \tilde{\boldsymbol{\tau}} + \boldsymbol{\tau}'' \,) \cdot (\tilde{\boldsymbol{u}} + \boldsymbol{u}'')] =$$

$$= \nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}}) + \nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \boldsymbol{u}'') + \nabla \cdot (\boldsymbol{\tau}'' \cdot \tilde{\boldsymbol{u}}) + \nabla \cdot (\boldsymbol{\tau}'' \cdot \boldsymbol{u}'') \, .$$

Reynolds averaging:

$$ⓐ : \overline{\frac{\partial \rho \tilde{e}_0}{\partial t}} + \overline{\frac{\partial \rho e_0''}{\partial t}} = \frac{\partial \overline{\rho} \tilde{e}_0}{\partial t} + \underbrace{\frac{\partial \overline{\rho e_0''}}{\partial t}}_{0} \, ;$$

$$ⓑ : \overline{\nabla \cdot [\rho \tilde{\boldsymbol{u}} \tilde{e}_0 + \rho \tilde{\boldsymbol{u}} e_0'' + \rho \boldsymbol{u}'' \tilde{e}_0 + \rho \boldsymbol{u}'' e_0'']} = \nabla \cdot \left[\overline{\rho \tilde{\boldsymbol{u}} \tilde{e}_0} + \overline{\rho \tilde{\boldsymbol{u}} e_0''} + \overline{\rho \boldsymbol{u}'' \tilde{e}_0} + \overline{\rho \boldsymbol{u}'' e_0''} \right] =$$

$$= \nabla \cdot \left[\overline{\rho} \tilde{\boldsymbol{u}} \tilde{e}_0 + \underbrace{\tilde{\boldsymbol{u}} \overline{\rho e_0''}}_{0} + \underbrace{\tilde{e}_0 \overline{\rho \boldsymbol{u}''}}_{0} + \overline{\rho \boldsymbol{u}'' e_0''} \right] \, ;$$

$$ⓒ : \overline{\nabla \cdot q} = \nabla \cdot \overline{q} \, ;$$

$$ⓓ : \overline{\nabla \cdot (p\tilde{\boldsymbol{u}})} + \overline{\nabla \cdot (p\boldsymbol{u}'')} = \nabla \cdot \overline{(p\tilde{\boldsymbol{u}})} + \nabla \cdot \overline{(p\boldsymbol{u}'')} = \nabla \cdot (\overline{p}\tilde{\boldsymbol{u}}) + \nabla \cdot \left(\overline{p\boldsymbol{u}''} \right) \, ;$$

$$ⓔ : \overline{\nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}})} + \overline{\nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \boldsymbol{u}'')} + \overline{\nabla \cdot (\boldsymbol{\tau}'' \cdot \tilde{\boldsymbol{u}})} + \overline{\nabla \cdot (\boldsymbol{\tau}'' \cdot \boldsymbol{u}'')} =$$

$$= \nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}}) + \nabla \cdot (\tilde{\boldsymbol{\tau}} \cdot \overline{\boldsymbol{u}}'') + \nabla \cdot \left(\overline{\boldsymbol{\tau}''} \cdot \tilde{\boldsymbol{u}} \right) + \nabla \cdot \left(\overline{\boldsymbol{\tau}'' \cdot \boldsymbol{u}''} \right) \, .$$

By putting all together:

$$\frac{\partial \overline{\rho} \tilde{e}_0}{\partial t} + \nabla \cdot \left[\overline{\rho} \tilde{\boldsymbol{u}} \tilde{e}_0 + \overline{\rho \boldsymbol{u}'' e_0''} + \overline{q} + \overline{p} \tilde{\boldsymbol{u}} + \overline{p \boldsymbol{u}''} - \tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}} \cdot \overline{\boldsymbol{u}}'' - \overline{\boldsymbol{\tau}''} \cdot \tilde{\boldsymbol{u}} - \overline{\boldsymbol{\tau}'' \cdot \boldsymbol{u}''} \right] = 0 \, .$$

We can here exploit the state equation together with some expressions for what concerns the internal energy and specific heat capacities to reshape the expressions above:

$$p = \rho R T \qquad e = C_v T \qquad C_p - C_v = R$$

- $\overline{\boldsymbol{u}'' p} = \overline{C_p \boldsymbol{u}'' \rho T} - \overline{C_v \boldsymbol{u}'' \rho T}$;
- $\dfrac{\boldsymbol{u}\cdot\boldsymbol{u}}{2} = \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} + \tilde{\boldsymbol{u}}\cdot\boldsymbol{u}'' + \dfrac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2}$;
- $\tilde{e_0} = \left(e + \widetilde{\dfrac{\boldsymbol{u}\cdot\boldsymbol{u}}{2}}\right) = \tilde{e} + \dfrac{\widetilde{\boldsymbol{u}\cdot\boldsymbol{u}}}{2} = \tilde{e} + \dfrac{\widetilde{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}}{2} + \widetilde{\tilde{\boldsymbol{u}}\cdot\boldsymbol{u}''} + \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2} =$
  $= \tilde{e} + \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} + \tilde{\boldsymbol{u}}\cdot\underbrace{\tilde{\boldsymbol{u}''}}_{\frac{\overline{\rho\boldsymbol{u}''}}{\overline{\rho}}=0} + \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2} = \tilde{e} + \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} + \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2}$;
- $e_0'' = e_0 - \tilde{e_0} = e + \dfrac{\boldsymbol{u}\cdot\boldsymbol{u}}{2} - \tilde{e} - \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} - \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2} =$
  $= e + \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} + \tilde{\boldsymbol{u}}\cdot\boldsymbol{u}'' + \dfrac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2} - \tilde{e} - \dfrac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2} - \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2} = e + \tilde{\boldsymbol{u}}\cdot\boldsymbol{u}'' + \dfrac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2} -$
  $\tilde{e} - \dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2}$;
- $\overline{\rho\boldsymbol{u}''e_0''} = \overline{\rho\boldsymbol{u}''e} + \overline{\rho\boldsymbol{u}''\left(\tilde{\boldsymbol{u}}\cdot\boldsymbol{u}''\right)} + \overline{\rho\boldsymbol{u}''\left(\dfrac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2}\right)} - \overline{\rho\boldsymbol{u}''\tilde{e}} - \overline{\rho\boldsymbol{u}''\left(\dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2}\right)} =$
  $= C_v\overline{\rho\boldsymbol{u}''T} + \overline{\rho\boldsymbol{u}''\left(\tilde{\boldsymbol{u}}\cdot\boldsymbol{u}''\right)} + \overline{\rho\boldsymbol{u}''\left(\dfrac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2}\right)} - \tilde{e}\underbrace{\overline{\rho\boldsymbol{u}''}}_{0} - \left(\dfrac{\widetilde{\boldsymbol{u}''\cdot\boldsymbol{u}''}}{2}\right)\underbrace{\overline{\rho\boldsymbol{u}''}}_{0}$;
- $q = -k\nabla T \Rightarrow \overline{q} = -C_p\dfrac{\mu}{Pr}\nabla\left(\overline{\tilde{T} + T''}\right) = -C_p\dfrac{\mu}{Pr}\nabla\tilde{T} - C_p\dfrac{\mu}{Pr}\nabla\overline{T''}$.

We can then substitute those terms into the energy equation:

$$\frac{\partial\overline{\rho}\tilde{e}_0}{\partial t} + \nabla\cdot\Bigg[\overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}_0 + C_v\overline{\rho\boldsymbol{u}''T} + \overline{\rho\boldsymbol{u}''\left(\tilde{\boldsymbol{u}}\cdot\boldsymbol{u}''\right)} + \overline{\rho\boldsymbol{u}''\left(\frac{\boldsymbol{u}''\cdot\boldsymbol{u}''}{2}\right)} - C_p\frac{\mu}{Pr}\nabla\tilde{T} +$$
$$-C_p\frac{\mu}{Pr}\nabla\overline{T''} + \overline{p}\tilde{\boldsymbol{u}} + C_p\overline{\boldsymbol{u}''\rho T} - C_v\overline{\boldsymbol{u}''\rho T} - \tilde{\boldsymbol{\tau}}\cdot\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}\cdot\overline{\boldsymbol{u}}'' - \overline{\boldsymbol{\tau}''}\cdot\tilde{\boldsymbol{u}} - \overline{\boldsymbol{\tau}''\cdot\boldsymbol{u}''}\Bigg] = 0\,.$$

By simplifying we obtain:

$$\frac{\partial \overline{\rho}\tilde{e}_0}{\partial t} + \nabla \cdot \Bigg[\overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}_0 + \overline{\rho \boldsymbol{u}'' \left(\tilde{\boldsymbol{u}} \cdot \boldsymbol{u}'' \right)} + \underbrace{\overline{\rho \boldsymbol{u}'' \left(\frac{\boldsymbol{u}'' \cdot \boldsymbol{u}''}{2} \right)}}_{\star} - C_p \frac{\mu}{Pr} \nabla \tilde{T} +$$

$$- \underbrace{C_p \frac{\mu}{Pr} \nabla \overline{T''}}_{\star} + \overline{p}\tilde{\boldsymbol{u}} + C_p \overline{\boldsymbol{u}'' \rho T} - \tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}} - \underbrace{\tilde{\boldsymbol{\tau}} \cdot \overline{\boldsymbol{u}}'' - \overline{\boldsymbol{\tau}''} \cdot \tilde{\boldsymbol{u}} - \overline{\boldsymbol{\tau}'' \cdot \boldsymbol{u}''}}_{\star} \Bigg] = 0 \, .$$

Terms denoted by $\star$ can be neglected under reasonable assumptions resulting in the equation:

$$\frac{\partial \overline{\rho}\tilde{e}_0}{\partial t} + \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}_0 + \overline{\rho \boldsymbol{u}'' \left(\tilde{\boldsymbol{u}} \cdot \boldsymbol{u}'' \right)} - C_p \frac{\mu}{Pr} \nabla \tilde{T} + \overline{p}\tilde{\boldsymbol{u}} + C_p \overline{\boldsymbol{u}'' \rho T} - \tilde{\boldsymbol{\tau}} \cdot \tilde{\boldsymbol{u}} \right] = 0 \, .$$

Two terms into previous equation are not known and have to be modeled:

$$C_p \overline{\boldsymbol{u}'' \rho T} = -C_p \frac{\mu_t}{Pr_t} \nabla \tilde{T} \, ,$$

$$\overline{\rho \boldsymbol{u}'' \left(\tilde{\boldsymbol{u}} \cdot \boldsymbol{u}'' \right)} = -\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}}_{turb} \, ,$$

where Pr_t is a constant value (~ 0.9) representing the turbulent Prandtl number.

The final equation we get reads:

$$\frac{\partial \overline{\rho}\tilde{e}_0}{\partial t} + \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}_0 - C_p \frac{\mu}{Pr} \nabla \tilde{T} - C_p \frac{\mu_t}{Pr_t} \nabla \tilde{T} + \overline{p}\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}}_{turb} \right] = 0 \, ,$$

or

$$\frac{\partial \overline{\rho}\tilde{e}_0}{\partial t} + \nabla \cdot \Bigg[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} + \underbrace{\frac{\widetilde{\boldsymbol{u}'' \cdot \boldsymbol{u}''}}{2}}_{\star} \right) - \underbrace{\frac{C_p}{C_v} \frac{\mu}{Pr}}_{\alpha} \nabla \tilde{e} +$$

$$- \underbrace{\frac{C_p}{C_v} \frac{\mu_t}{Pr_t}}_{\alpha_t} \nabla \tilde{e} + \overline{p}\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}}_{turb} \Bigg] = 0 \, .$$

The term denoted here by $\star$ represents the turbulent kinetic energy and can be neglected. Moreover for steady-state cases we end up with a pure internal energy equation:

$$\nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} \right) - \underbrace{\frac{C_p}{C_v}\frac{\mu}{Pr}}_{\alpha} \nabla \tilde{e} - \underbrace{\frac{C_p}{C_v}\frac{\mu_t}{Pr_t}}_{\alpha_t} \nabla \tilde{e} + \overline{p}\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}}_{turb} \right] = 0 \, .$$

The FANS steady-state equations can then be collected into a single system. They reads:

$$\begin{cases} \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}) = 0 \\ \nabla \cdot [\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}} + \overline{p}\boldsymbol{I}] = 0 \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} \right) - \frac{C_p}{C_v}\frac{\mu}{Pr} \nabla \tilde{e} - \frac{C_p}{C_v}\frac{\mu_t}{Pr_t} \nabla \tilde{e} + \overline{p}\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}} - \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{\tau}}_{turb} \right] = 0 \end{cases} .$$

Most of the time, for sake of simplicity, into the solvers the viscous components for the energy equations are not implemented, ending up with the final system:

$$\begin{cases} \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}) = 0 \\ \nabla \cdot [\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}} + \overline{p}\boldsymbol{I}] = 0 \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} \right) - \frac{C_p}{C_v}\frac{\mu}{Pr} \nabla \tilde{e} - \frac{C_p}{C_v}\frac{\mu_t}{Pr_t} \nabla \tilde{e} + \overline{p}\tilde{\boldsymbol{u}} \right] = 0 \end{cases} .$$

5.1.1 Compressible pressure equation

Let us take into consideration the compressible momentum equation:

$$\nabla \cdot [\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}}] = -\nabla \overline{p} \, .$$

This equation can be rewritten by substituting the forms related to velocity into their Finite Volume matrix counterpart as follows:

$$\boldsymbol{A}_u(\tilde{\boldsymbol{u}}) = -\nabla \overline{p} \Rightarrow \boldsymbol{A}\tilde{\boldsymbol{u}} = \boldsymbol{H}(\tilde{\boldsymbol{u}}) - \nabla \overline{p} \, ,$$

where $\boldsymbol{A}_u(\tilde{\boldsymbol{u}})$ is the Finite Volume discretized form of $\nabla \cdot [\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{\tau}}_{turb} - \tilde{\boldsymbol{\tau}}]$, $\boldsymbol{A}\tilde{\boldsymbol{u}}$ is the diagonal part of $\boldsymbol{A}_u(\tilde{\boldsymbol{u}})$ while $-\boldsymbol{H}(\overline{\boldsymbol{u}})$ is its extra diagonal part so that $\boldsymbol{A}_u(\tilde{\boldsymbol{u}}) = \boldsymbol{A}\tilde{\boldsymbol{u}} - \boldsymbol{H}(\tilde{\boldsymbol{u}})$.

If we localize that equation at a generic point P, we get:

$$\tilde{\boldsymbol{u}}_P = \frac{\boldsymbol{H}(\tilde{\boldsymbol{u}})}{a_P} - \frac{\nabla \overline{p}_P}{a_P} \, .$$

Let us rename $\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{u}}^*$ and $\overline{\rho} = \overline{\rho}^*$ both velocity and density we have at this point, after having solved the momentum equation, for a reason that will be clarified in a moment.

The mass flux, at the generic point P, can be obtained as:

$$\overline{\rho}^*_P \tilde{\boldsymbol{u}}^*_P = \overline{\rho}^*_P \frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}^*)}{a_P} - \overline{\rho}^*_P \frac{\nabla \overline{p}^{n-1}_P}{a_P} \, .$$

Since the pressure gradient has to be calculated explicitly, we indicate it as $\nabla \overline{p}^{n-1}$ meaning that the pressure field has to be previously calculated.

It is easy to realize that the set $\overline{\rho}^*, \tilde{\boldsymbol{u}}^*, \overline{p}^{n-1}$ will not satisfy the mass conservation constrain since velocity field has been evaluated by the use of the pressure gradient at time step $n-1$. We can then imagine to introduce some corrections to all the terms so that $\overline{\rho} = \overline{\rho}^* + \overline{\rho}', \tilde{\boldsymbol{u}} = \tilde{\boldsymbol{u}}^* + \tilde{\boldsymbol{u}}', \overline{p} = \overline{p}^{n-1} + \overline{p}'$.

It is now possible to rewrite the mass flux as:

$$(\overline{\rho}^*_P + \overline{\rho}'_P)\,(\tilde{\boldsymbol{u}}^*_P + \tilde{\boldsymbol{u}}'_P) = (\overline{\rho}^*_P + \overline{\rho}'_P) \left[\frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}^*)}{a_P} + \frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}')}{a_P} \right] - (\overline{\rho}^*_P + \overline{\rho}'_P) \left[\frac{\nabla \overline{p}^{n-1}_P}{a_P} + \frac{\nabla \overline{p}'_P}{a_P} \right] \, .$$

By the definition of compressibility Ψ, we can write $\overline{\rho} = \Psi \overline{p}$ and then $\rho = \overline{\rho}^* + \overline{\rho}' = \Psi \overline{p}^{n-1} + \Psi \overline{p}' \Rightarrow \overline{\rho}' = \Psi p - \Psi \overline{p}^{n-1} = \Psi \overline{p}'$.

We can then interpolate that expression to obtain the variables evaluations at the faces and finally sum over all the faces surrounding the point P to get the mass conservation equation in its pressure correction shape:

$$\sum_f (\overline{\rho}^*_P + \Psi \overline{p}'_P) \left[\frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}^*)}{a_P} + \frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}')}{a_P} \right] \Bigg|_f - \sum_f \left(\overline{\rho}^*_P + \underbrace{\overline{\rho}'_P}_{\star} \right) \left[\frac{\nabla \overline{p}^{n-1}_P}{a_P} + \frac{\nabla \overline{p}'_P}{a_P} \right] \Bigg|_f = 0 \, .$$

The $\star$ term can be neglected obtaining the correction equation for pressure. The only term that has to be modeled in some way is $\boldsymbol{H}(\tilde{\boldsymbol{u}}')$. In the SIMPLE algorithm the correction extra diagonal velocity term is neglected leading to the following

final pressure correction equation:

$$\sum_f (\overline{\rho}_P^* + \Psi \overline{p}'_P) \left[\frac{\boldsymbol{H}(\tilde{\boldsymbol{u}}^*)}{a_P} \right] \Bigg|_f - \sum_f (\overline{\rho}_P^*) \left[\frac{\nabla \overline{p}_P^{n-1}}{a_P} + \frac{\nabla \overline{p}'_P}{a_P} \right] \Bigg|_f = 0 \, . \tag{26}$$

The Finite Volume variables can be here introduced: $\overline{p}_h \in \mathbb{Q}_h$, $\tilde{\boldsymbol{u}}_h \in \mathbb{Q}_h$ and $\tilde{e}_h \in \mathbb{E}_h$. They are not continuous and they are constant in the interior part of each cell assuming everywhere the value at the center of the cell. For sake of simplicity in this section we will keep on referring to this variables without the $\square_h$ subscript to not make the formulas too heavy.

In this chapter we aim at employing a segregated approach based on a compressible formulation of the SIMPLE algorithm. This aspect has to be kept in mind for the Finite Volume discretization strategy.

We can then take into consideration the equations one by one, starting from the continuity constrain. The equation can be written into its integral form over each cell as follows:

$$\int_{\Omega_i} \nabla \cdot (\overline{\rho} \tilde{\boldsymbol{u}}) \, dV = 0 \, .$$

By exploiting the divergence theorem, the equation above leads to:

$$\int_{\delta\Omega_i} \overline{\rho} \tilde{\boldsymbol{u}} \cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} (\overline{\rho} \tilde{\boldsymbol{u}})|_f \cdot \boldsymbol{S}_f = \sum_{f=1}^{Nf_i} F_f \, ,$$

where the subscript $\square|_f$ indicates that those variables are evaluated at the center of the face f and $\boldsymbol{S}_f$ is the oriented surface of the same face while Nf_i is the total number of faces surrounding the i-th cell while F_f stands for the mass flux crossing the face f (see Figure 2).

The discretized version of the continuity equation then reads:

$$\sum_{f=1}^{Nf_i} F_f = 0 \, . \tag{27}$$

Let us now take into consideration the momentum equation. It has to be integrated over the volume of every cell and it can then be analyzed term by term, starting from the convective one:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot (\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}})\, dV = \int_{\delta\Omega_i} \overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} \cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \boldsymbol{S}_f \cdot (\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}})|_f = \sum_{f=1}^{Nf_i} \boldsymbol{F}_f \tilde{\boldsymbol{u}}_f \ .$$

The first part of the diffusion term is discretized as follows:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}\right]\, dV = \int_{\delta\Omega_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}\right] \cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}\right]|_f \cdot \boldsymbol{S}_f \ .$$

For orthogonal meshes we can approximate this term as:

$$\sum_{f=1}^{Nf_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}\right]|_f \cdot \boldsymbol{S}_f \simeq (\mu + \mu_t)|_f\ |\boldsymbol{S}_f| \frac{\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_j}{|\boldsymbol{d}_f|} \ ,$$

being $\boldsymbol{d}_f$ the oriented vector bridging the cell centers of two neighbor cells. When this is not the case, a non-orthogonal correction is added:

$$\sum_{f=1}^{Nf_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}\right]|_f \cdot \boldsymbol{S}_f \simeq \sum_{f=1}^{Nf_i} (\mu + \mu_t)|_f \left[|\boldsymbol{\mathcal{P}}_f| \frac{\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_j}{|\boldsymbol{d}_f|} + \boldsymbol{\mathcal{O}}_f \cdot \nabla\tilde{\boldsymbol{u}}|_f\right] \ ,$$

where we have $\boldsymbol{\mathcal{P}}_f \parallel \boldsymbol{d}_f$, $\boldsymbol{\mathcal{O}}_f \perp \boldsymbol{d}_f$ and $\boldsymbol{\mathcal{P}}_f + \boldsymbol{\mathcal{O}}_f = \boldsymbol{S}_f$ while $\nabla\tilde{\boldsymbol{u}}|_f$ is evaluated starting from its value at the cell centers $\nabla\tilde{\boldsymbol{u}}_i$ and $\nabla\tilde{\boldsymbol{u}}_j$ by interpolation.

The second part of the diffusion term is treated following the previous steps:

$$\int_{\Omega_i} \boldsymbol{\nabla} \cdot \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}^T\right]\, dV = \int_{\delta\Omega_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}^T\right] \cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \left[(\mu + \mu_t)\, \nabla\tilde{\boldsymbol{u}}^T\right]|_f \cdot \boldsymbol{S}_f \ .$$

In this case, the face center evaluation is treated explicitly so that this term is considered to be a forcing term:

$$\nabla\tilde{\boldsymbol{u}}^T|_f \cdot \boldsymbol{S}_f = \begin{bmatrix} \frac{\partial\tilde{u}_x}{\partial x} & \frac{\partial\tilde{u}_x}{\partial y} & \frac{\partial\tilde{u}_x}{\partial z} \\ \frac{\partial\tilde{u}_y}{\partial x} & \frac{\partial\tilde{u}_y}{\partial y} & \frac{\partial\tilde{u}_y}{\partial z} \\ \frac{\partial\tilde{u}_z}{\partial x} & \frac{\partial\tilde{u}_z}{\partial y} & \frac{\partial\tilde{u}_z}{\partial z} \end{bmatrix}_f^T \cdot \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix}_f = \begin{bmatrix} \frac{\partial\tilde{u}_x}{\partial x}S_x + \frac{\partial\tilde{u}_y}{\partial x}S_y + \frac{\partial\tilde{u}_z}{\partial x}S_z \\ \frac{\partial\tilde{u}_x}{\partial y}S_x + \frac{\partial\tilde{u}_y}{\partial y}S_y + \frac{\partial\tilde{u}_z}{\partial y}S_z \\ \frac{\partial\tilde{u}_x}{\partial z}S_x + \frac{\partial\tilde{u}_y}{\partial z}S_y + \frac{\partial\tilde{u}_z}{\partial z}S_z \end{bmatrix} .$$

The same applies for the last part of the diffusive term:

$$\begin{aligned} \int_{\Omega_i} \boldsymbol{\nabla}\left[(\mu+\mu_t)\,\frac{2}{3}\nabla\cdot\tilde{\boldsymbol{u}}\right] dV &= \int_{\delta\Omega_i}\left[(\mu+\mu_t)\,\frac{2}{3}\nabla\cdot\tilde{\boldsymbol{u}}\right] d\boldsymbol{S} \simeq \\ &\simeq \sum_{f=1}^{Nf_i}\left[(\mu+\mu_t)\,\frac{2}{3}\nabla\cdot\tilde{\boldsymbol{u}}\right]\Big|_f \boldsymbol{S}_f \,. \end{aligned}$$

where once again the divergence of the velocity is interpolated to the surface and treated explicitly leading to an additional forcing term.

The last term to be considered is the pressure gradient:

$$\int_{\Omega_i} \boldsymbol{\nabla}\overline{p}\, dV = \int_{\delta\Omega_i} \overline{p}\; d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \overline{p}_f \boldsymbol{S}_f \,.$$

In the momentum equation, pressure is interpolated to the faces and then treated explicitly as a source term. The final momentum equation reads:

$$\begin{aligned} &\sum_{f=1}^{Nf_i}\left[\boldsymbol{F}_f\tilde{\boldsymbol{u}}_f - (\mu+\mu_t)|_f\left(|\boldsymbol{\mathcal{P}}_f|\frac{\tilde{\boldsymbol{u}}_i-\tilde{\boldsymbol{u}}_j}{|\boldsymbol{d}_f|} + \boldsymbol{\mathcal{O}}_f\cdot\nabla\tilde{\boldsymbol{u}}|_f\right)\right] = \\ &\quad = \sum_{f=1}^{Nf_i}\left[(\mu+\mu_t)|_f\left(\nabla\tilde{\boldsymbol{u}}_f^T\cdot\boldsymbol{S}_f - \frac{2}{3}\nabla\cdot\tilde{\boldsymbol{u}}_f\boldsymbol{S}_f\right) - \overline{p}_f\boldsymbol{S}_f\right] , \end{aligned} \tag{28}$$

where all the terms composing the right-hand side of the equation are treated explicitly as source terms.

Equation 28 can be rewritten into its Finite Volume matrix form as follows:

$$\boldsymbol{A}_u(\tilde{\boldsymbol{u}}) = -\nabla\overline{p} \Rightarrow \boldsymbol{A}\tilde{\boldsymbol{u}} = \boldsymbol{H}(\tilde{\boldsymbol{u}}) - \nabla\overline{p}\,, \tag{29}$$

where $\boldsymbol{A}_u(\tilde{\boldsymbol{u}})$ is the Finite Volume discretized form containing all the terms related to velocity of both left-hand and right-hand sides of Equation 28, $\boldsymbol{A}\tilde{\boldsymbol{u}}$ is the diagonal part of $\boldsymbol{A}_u(\tilde{\boldsymbol{u}})$ while $-\boldsymbol{H}(\overline{\boldsymbol{u}})$ is its extra diagonal part so that $\boldsymbol{A}_u(\tilde{\boldsymbol{u}}) = \boldsymbol{A}\tilde{\boldsymbol{u}} - \boldsymbol{H}(\tilde{\boldsymbol{u}})$.

The last equation to be analysed regards the energy conservation:

$$\int_{\Omega_i} \nabla\cdot[\overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}]\;dV = \int_{\delta\Omega_i} \overline{\rho}\tilde{\boldsymbol{u}}\tilde{e}\cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \tilde{e}_f\overline{\rho}_f\tilde{\boldsymbol{u}}_f\cdot\boldsymbol{S}_f = \sum_{f=1}^{Nf_i}\tilde{e}_f F_f\,.$$

The kinetic part of the total energy is treated explicitly and leads to:

$$\int_{\Omega_i} \nabla\cdot\left[\overline{\rho}\tilde{\boldsymbol{u}}\frac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2}\right]\;dV = \int_{\delta\Omega_i} \overline{\rho}\tilde{\boldsymbol{u}}\frac{\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}}{2}\cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i} \frac{\tilde{\boldsymbol{u}}_f\cdot\tilde{\boldsymbol{u}}_f}{2}\overline{\rho}_f\tilde{\boldsymbol{u}}_f\cdot\boldsymbol{S}_f = \sum_{f=1}^{Nf_i}\frac{\tilde{\boldsymbol{u}}_f\cdot\tilde{\boldsymbol{u}}_f}{2}F_f\,.$$

The diffusive term reads:

$$\int_{\Omega_i} \nabla\cdot\left[\frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\nabla\tilde{e}\right]\;dV =$$

$$= \int_{\delta\Omega_i} \frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\nabla\tilde{e}\cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i}\frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\Big|_f\nabla\tilde{e}_f\cdot\boldsymbol{S}_f\,.$$

Once again the energy gradient is not available at the center of the faces but it can be approximated:

$$\sum_{f=1}^{Nf_i}\frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\Big|_f\nabla\tilde{e}_f\cdot\boldsymbol{S}_f \simeq \sum_{f=1}^{Nf_i}\frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\Big|_f\left[|\boldsymbol{\mathcal{P}}_f|\frac{\tilde{e}_i-\tilde{e}_j}{|\boldsymbol{d}_f|}+\boldsymbol{\mathcal{O}}_f\cdot\nabla\tilde{e}|_f\right]\,.$$

Finally also the pressure term is discretized and treated explicitly:

$$\int_{\Omega_i}\nabla\cdot[\overline{p}\tilde{\boldsymbol{u}}]\;dV = \int_{\delta\Omega_i}\overline{p}\tilde{\boldsymbol{u}}\cdot d\boldsymbol{S} \simeq \sum_{f=1}^{Nf_i}\overline{p}_f\tilde{\boldsymbol{u}}_f\cdot\boldsymbol{S}_f = \sum_{f=1}^{Nf_i}\frac{\overline{p}_f}{\overline{\rho}_f}F_f\,.$$

The resulting equation reads:

$$\sum_{f=1}^{Nf_i}\left[\tilde{e}_f F_f - \frac{C_p}{C_v}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\Big|_f\left(|\boldsymbol{\mathcal{P}}_f|\frac{\tilde{e}_i-\tilde{e}_j}{|\boldsymbol{d}_f|}+\boldsymbol{\mathcal{O}}_f\cdot\nabla\tilde{e}|_f\right)\right] = \\ = -\sum_{f=1}^{Nf_i}\left(\frac{\tilde{\boldsymbol{u}}_f\cdot\tilde{\boldsymbol{u}}_f}{2}+\frac{\overline{p}_f}{\overline{\rho}_f}\right)F_f\,. \tag{30}$$

Also Equation 30 can be written into its matrix form as follow:

$$\boldsymbol{E}(\tilde{e}) = \boldsymbol{F}(\overline{p},\tilde{\boldsymbol{u}})\,. \tag{31}$$

5.2 Reduced order algorithm for compressible flows

In this paragraph we will introduce the reduced order algorithm developed for the resolution of compressible flows where no discontinuities are present. This means it is only suited for subsonic cases where the *Mach* number is lower then 1 in all the points of the domain. The resolution of supersonic flows require a specific treatment of the discontinuities that can not be circumvented (see e.g. [143]).

As it has been done in subsection 3.2 and subsection 4.2, also for compressible flows we would like to develop a segregated algorithm based on a SIMPLE approach.

When dealing with compressibility, the thermodynamics of the problem can no longer be neglected and energy evolution becomes of primary importance for the behaviour of the system. For this reason, four different modal expansions have to be taken into consideration for compressible cases:

$$\tilde{\boldsymbol{u}}_r = \sum_{i=1}^{N_u}\tilde{\boldsymbol{a}}_i(\boldsymbol{\mu})\tilde{\boldsymbol{\psi}}_i(\boldsymbol{x}) = \tilde{\boldsymbol{\Psi}}\tilde{\boldsymbol{a}}\,,\quad \overline{p}_r = \sum_{i=1}^{N_p}\tilde{\boldsymbol{b}}_i(\boldsymbol{\mu})\tilde{\varphi}_i(\boldsymbol{x}) = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{b}}\,,\quad \tilde{e}_r = \sum_{i=1}^{N_e}\tilde{\boldsymbol{c}}_i(\boldsymbol{\mu})\tilde{\theta}_i(\boldsymbol{x}) = \tilde{\boldsymbol{\Theta}}\tilde{\boldsymbol{c}}\,,$$

$$\nu_{tr} = \sum_{i=0}^{N_{\nu_t}} m_i(\boldsymbol{x})\zeta_i(\boldsymbol{\mu})\,,$$

where N_u, N_p and N_e are the numbers of basis functions selected for velocity, pressure and energy solutions respectively, $\tilde{\boldsymbol{a}}\in\mathbb{R}^{N_u}$ is the vector containing the coefficients for the velocity expansion while the same reads for pressure with respect to

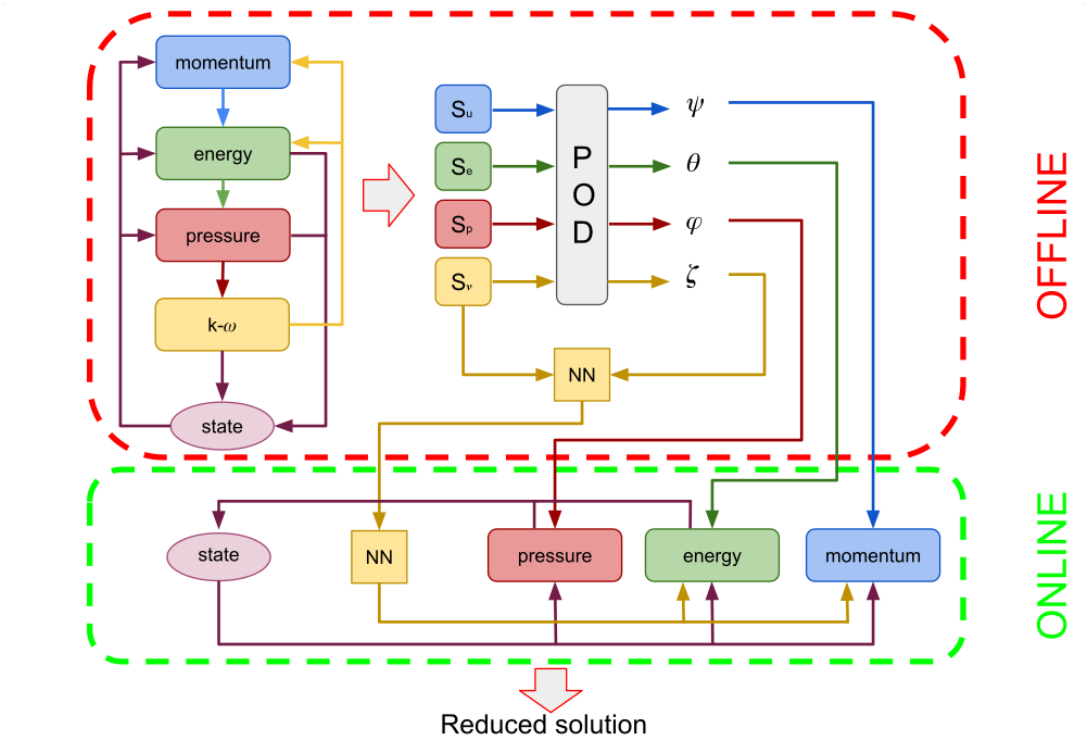


Figure 44: Schematic diagram for the compressible reduced SIMPLE algorithm.

$\tilde{\boldsymbol{b}} \in \mathbb{R}^{N_p}$ and for energy with respect to $\tilde{\boldsymbol{c}} \in \mathbb{R}^{N_e}$, being $\tilde{\psi}_i$, $\tilde{\varphi}_i$ and $\tilde{\theta}$ the corresponding vector and scalar basis functions collected into the matrices $\tilde{\boldsymbol{\Psi}} \in \mathbb{R}^{d N_h \times N_u}$, $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{N_h \times N_p}$ and $\tilde{\boldsymbol{\Theta}} \in \mathbb{R}^{N_h \times N_e}$ respectively.

As introduced in subsection 5.1, an additional state equation is required to obtain a closed set of equations. Also at the reduced level we use the perfect gasses relation without any need to discretize it since it is used for reconstructed solution fields.

The procedure employed in this chapter for the resolution of the reduced problem is explicated in Algorithm 5 and resumed in Figure 44.

Algorithm 5 The Reduced Order compressible SIMPLE algorithm

Input: first attempt reduced velocity, pressure and energy coefficients $\tilde{\boldsymbol{a}}^{\star}$, $\tilde{\boldsymbol{b}}^{\star}$ and $\tilde{\boldsymbol{c}}^{\star}$; modal basis functions matrices for pressure, velocity and energy $\tilde{\Phi}$, $\tilde{\Psi}$ and $\tilde{\Theta}$

Output: reduced pressure, velocity and energy fields $\overline{p}_r$, $\tilde{\boldsymbol{u}}_r$ and $\tilde{e}_r$

1: From $\tilde{\boldsymbol{a}}^{\star}$, $\tilde{\boldsymbol{b}}^{\star}$ and $\tilde{\boldsymbol{c}}^{\star}$, reconstruct reduced fields $\overline{p}^{\star}$, $\tilde{\boldsymbol{u}}^{\star}$ and $\tilde{e}^{\star}$:

$$\overline{p}^{\star} = \tilde{\Phi}\tilde{\boldsymbol{a}}^{\star}, \qquad \tilde{\boldsymbol{u}}^{\star} = \tilde{\Psi}\tilde{\boldsymbol{b}}^{\star}, \qquad \tilde{e}^{\star} = \tilde{\Theta}\tilde{\boldsymbol{c}}^{\star};$$

2: Evaluate the eddy viscosity field μ_t;
3: Momentum predictor step : assemble Equation 29, relax it employing prescribed under-relaxation factor α_u, project it over the velocity basis functions $\boldsymbol{\psi}_i$ and solve it to obtain new reduced velocity coefficients vector $\tilde{\boldsymbol{b}}^{\star\star}$;
4: Reconstruct the new reduced velocity $\tilde{\boldsymbol{u}}^{\star\star}$ and calculate the off-diagonal component $\boldsymbol{H}(\tilde{\boldsymbol{u}}^{\star\star})$;
5: Energy equation step : assemble Equation 31, relax it employing prescribed under-relaxation factor α_e, project it over the energy basis functions θ_i and solve it to obtain new reduced energy coefficients vector $\tilde{\boldsymbol{c}}^{\star\star}$;
6: Reconstruct the new reduced energy $\tilde{e}^{\star\star}$;
7: Calculate both density $\overline{\rho}^{\star\star}$ and temperature $\tilde{T}^{\star\star}$ fields starting from $\overline{p}^{\star}$, $\tilde{\boldsymbol{u}}^{\star\star}$ and $\tilde{e}^{\star\star}$ by the use of the state equation;
8: Pressure correction step: assemble Equation 26, project it over the pressure basis functions φ_i to get new reduced pressure coefficients $\tilde{\boldsymbol{a}}^{\star\star}$; then correct the velocity explicitly after having reconstructed the new pressure $\overline{p}^{\star\star}$;
9: Relax the pressure field with the prescribed under-relaxation factor α_p. The under-relaxed field is called $\overline{p}^{ur}$;
10: **if** convergence **then**
11: $\overline{p}_r = \overline{p}^{ur}$, $\tilde{\boldsymbol{u}}_r = \tilde{\boldsymbol{u}}^{\star\star}$ and $\tilde{e}_r = \tilde{e}^{\star\star}$
12: **else**
13: Assemble the conservative face fluxes F_f:

$$F_f = \tilde{\boldsymbol{u}}_f \cdot \boldsymbol{S}_f;$$

14: set $\overline{p}^{\star} = \overline{p}^{ur}$, $\tilde{\boldsymbol{u}}^{\star} = \tilde{\boldsymbol{u}}^{\star\star}$ and $\tilde{e}^{\star} = \tilde{e}^{\star\star}$;
15: iterate from step 1.
16: **end if**

5.3 Compressible Navier-Stokes problems: physical parametrization

5.3.1 Compressible aerofoil problem with a variable viscosity

The first compressible flows test case we present in this work is a physically parametrized external flow: a NACA0012 aerofoil is immersed into a fluid with variable dynamic viscosity μ. The problem can be formulated as:

$$\begin{cases} \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}) = 0 & \text{in } \Omega \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - (\mu + \mu_t) \left(\nabla \tilde{\boldsymbol{u}} + \nabla \tilde{\boldsymbol{u}}^T - \frac{2}{3} \nabla \cdot \tilde{\boldsymbol{u}} \boldsymbol{I} \right) + \overline{p} \boldsymbol{I} \right] = 0 & \text{in } \Omega \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} \right) - \frac{C_p}{C_v} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \nabla \tilde{e} + \overline{p}\tilde{\boldsymbol{u}} \right] = 0 & \text{in } \Omega \\ \tilde{\boldsymbol{u}} = [250, 0, 0]^T & \text{in } \Gamma_{in} \\ \nu \dfrac{\partial \tilde{\boldsymbol{u}}}{\partial \boldsymbol{n}} - \overline{p}\boldsymbol{n} = 0 & \text{in } \Gamma_{out} \end{cases} .$$

The chord of the aerofoil is equal to one. As already said, the viscosity can vary so that $\mu \in [10^{-5}, 10^{-2}]$. The domain is characterized by the same shape showed in Figure 13, left, where the radius of the inlet boundary is equal to $2m$ while the length of the rectangular rear part is equal to $4m$ so that the total length is equal to $6m$. The tessellation is composed by 16000 hexahedral cells.

The speed of sound at the inlet can easily be evaluated by taking into consideration the thermophysical properties of the gas we are working with. We consider perfect gasses. Thus the specific heat transfer at constant pressure is sufficient to evaluate $\gamma = \frac{C_p}{C_v} = \frac{C_p}{C_p - R}$ where $C_p = 1005 \frac{J}{Kg\ K}$ while $R = 8,314 \frac{J}{mol\ K}$ is the constant for perfect gasses. We suppose our aerofoil to move into air so that $M = 28,9 \frac{g}{mol}$ where M stands for the molar weight. Temperature is fixed at $T = 298K$. Collecting all these data together, we end up with

$$C = \sqrt{\frac{\gamma RT}{M}} = 341.17 \frac{m}{s}.$$

This means that at the inlet the *Mach* number can be calculated as

$$Mach = \frac{\tilde{\boldsymbol{u}}_\infty}{C} \simeq 0.73.$$

For this test case, consequently, a compressible treatment for the flow is needed since we are approaching the transonic regime and compressible effects are of significant relevance. At the inlet, pressure is fixed to 10^5 Pa. Then also the *Reynolds*

number can be evaluated as

$$Re = \frac{\rho L \tilde{\boldsymbol{u}}_{inlet}}{\mu} = \frac{pL\tilde{\boldsymbol{u}}_{inlet}M}{\mu RT}.$$

The resulting *Reynolds* number is then $Re \in 2.92 \times [10^4, 10^7]$, which clearly requires a treatment for turbulence since the system is operating in fully turbulent regime.

For the offline phase, 50 random values have been selected: $\boldsymbol{\mu} \in [10^{-5}, 10^{-2}] \equiv \mathbb{P}$. Full order eddy viscosity is calculated by the resolution of a $k-\omega$ turbulence model. This last choice is justified by the fact that the physical dynamics we expect to observe in such a test case are different from the oneS exhibited in both subsection 4.4 and subsection 4.3: as pointed out in section 4, we are here dealing with a body surrounded by the fluid which is a classical test case where $k-\omega$ models outperform $k-\epsilon$ models.

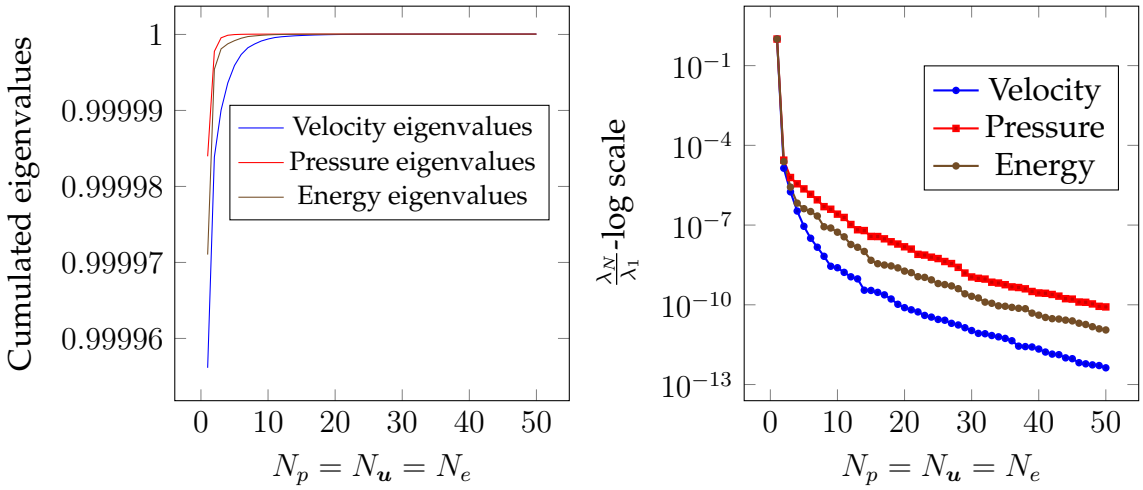


Figure 45: Eigenvalues trends for pressure, velocity and energy: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

Figure 45 shows the trends of the eigenvalues for velocity, pressure and energy. The same is reporteD in Figure 46 for the eddy viscosity. As we may notice, by just considering a few modes for every variable, the amount of discarded information is almost negligible. For this reason, just the first 20 modal basis functions have been selected for velocity, pressure and energy while 30 modal basis functions are used to reconstruct the eddy viscosity field. This is due to the fact that, analyzing Figure 46, left, it is clear that a higher number of basis functions are needed in order to approach the unity for what concerns the eddy viscosity behaviour.

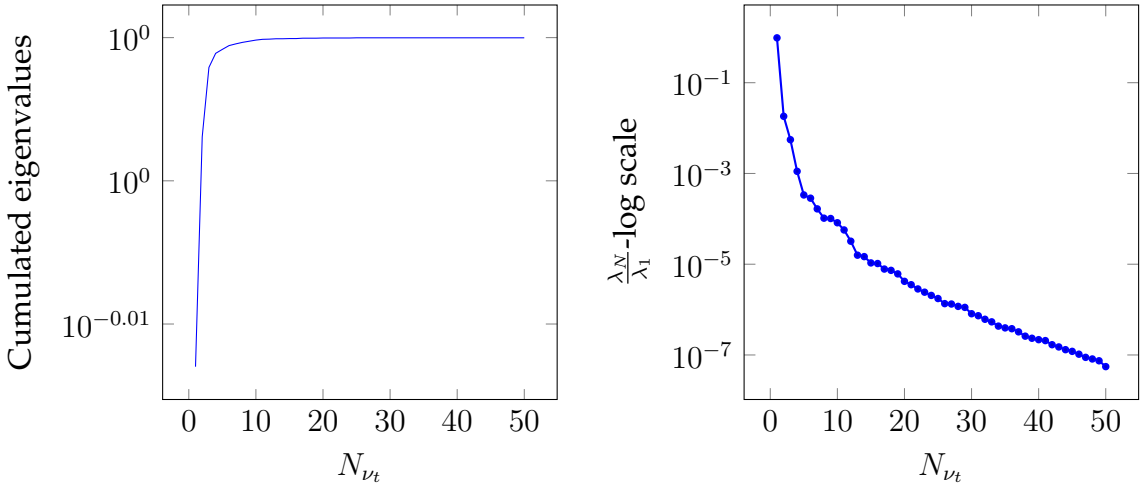


Figure 46: Eigenvalues trends for the eddy viscosity: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

Once again we construct a neural network for the ν_t coefficients estimation, as explained in subsubsection 4.4.2: two hidden layer are present, the first one composed by 256 neurons and the second one composed by 64 neurons, resulting in a fully connected network where only *Tanh* activation functions are used. Offline solutions, including the intermediate steps, are retained to train the network.

The training procedure is carried out in 2×10^3 epochs. 20 new random offline solutions have been performed to obtain a testing set which was not correlated to the solutions used for the training stage. The Adam optimizer has been selected for this procedure. A mean squared error loss function is used to evaluate the reconstruction capability of the network for both training and testing sets. The decay behaviour of both losses are depicted in Figure 47. The training stage has been stopped after 2×10^4 epochs since test loss was no more decreasing and the discrepancy between test and train losses was starting to increase significantly.

Figure 48, left, shows the L^2 norm relative errors for all the different parameters in the online set concerning velocity, pressure and internal energy. Figure 48, right, shows the L^2 norm relative error for the eddy viscosity, between full order and reduced order fields, for the whole online parameter set. As we may notice, even if the order of magnitude of the ν_t error is equal to 10^{-2}, it is sufficient to ensure a lower error for the quantities of interest, i.e. velocity, pressure and energy. By this observation we are allowed to employ such a small neural network which is not

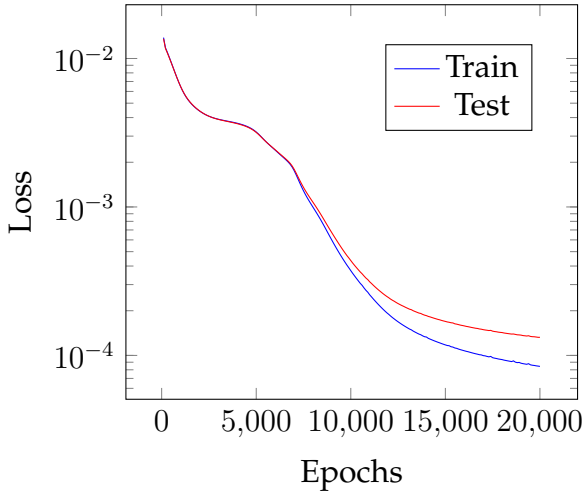


Figure 47: Loss function decay for both train and test sets.

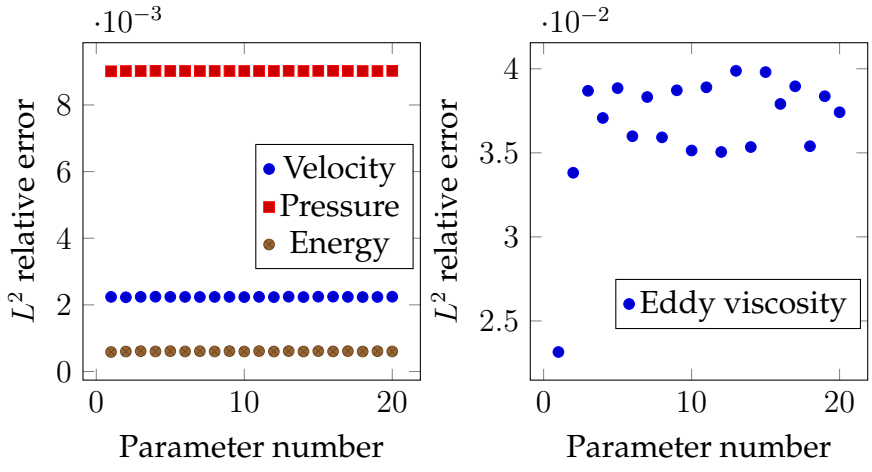


Figure 48: L^2 norm relative errors.

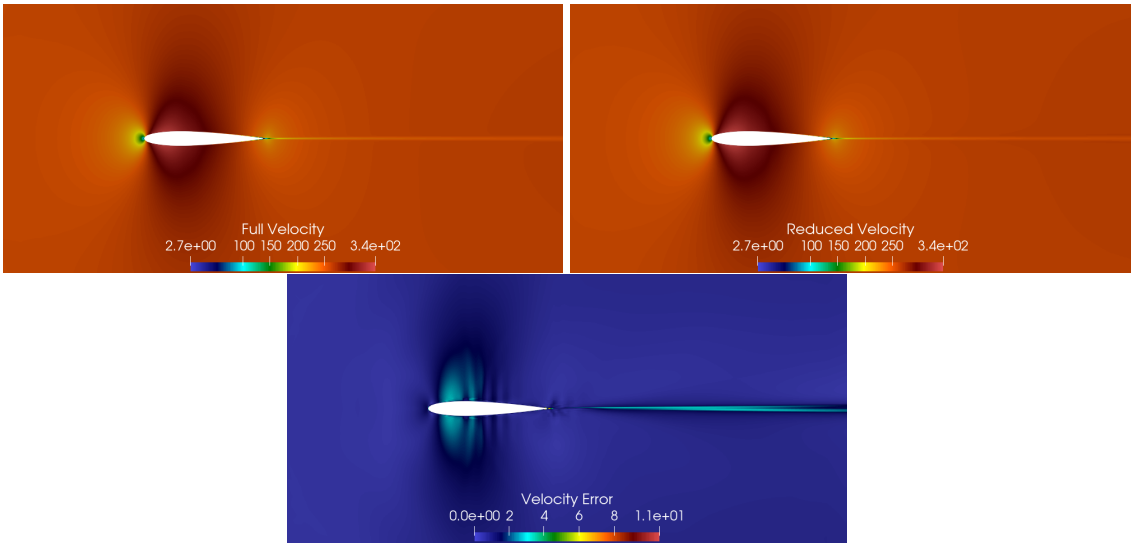


Figure 49: Comparison between full order (left) and reduced order (right) solutions for the velocity field. These fields refer to the resolution of the problem for $\mu = 1.2 \times 10^{-3}$ which has been selected as a random value in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

compromising the computational cost, still ensuring good performances.

In Figure 49, Figure 50, Figure 51 and Figure 52 a comparison between full order and reduced order solutions is depicted, for a random value of the parameter, included in the online set. By analysing the reported fields, full order and reduced order solutions appear to be very similar and most important areas into the domain, i.e. the zones surrounding the aerofoil together with the wake created by the body, are well reconstructed.

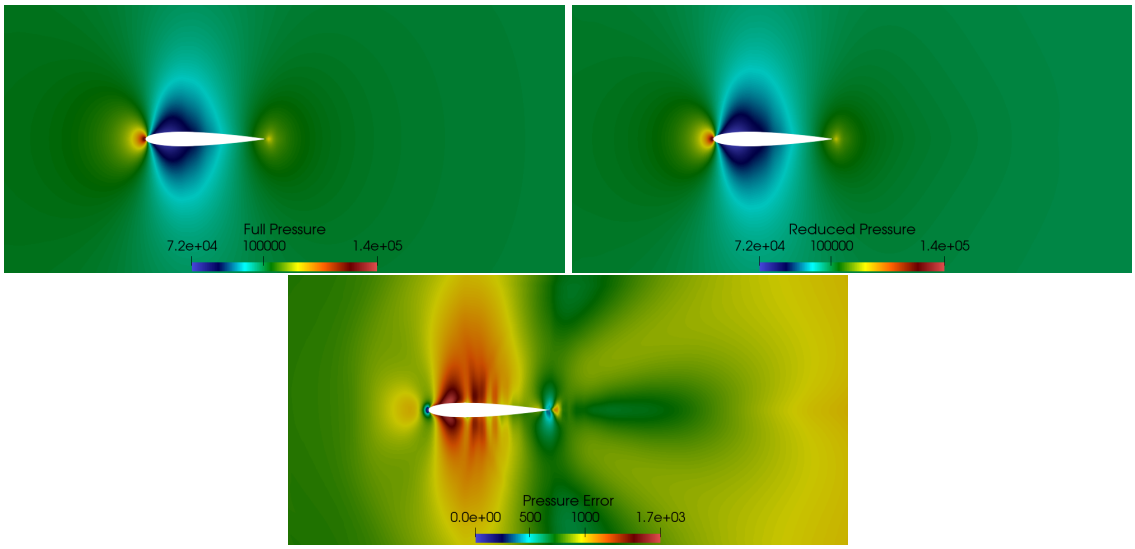


Figure 50: Comparison between full order (left) and reduced order (right) solutions for the pressure field. These fields refer to the resolution of the problem for $\mu = 1.2 \times 10^{-3}$ which has been selected as a random value in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

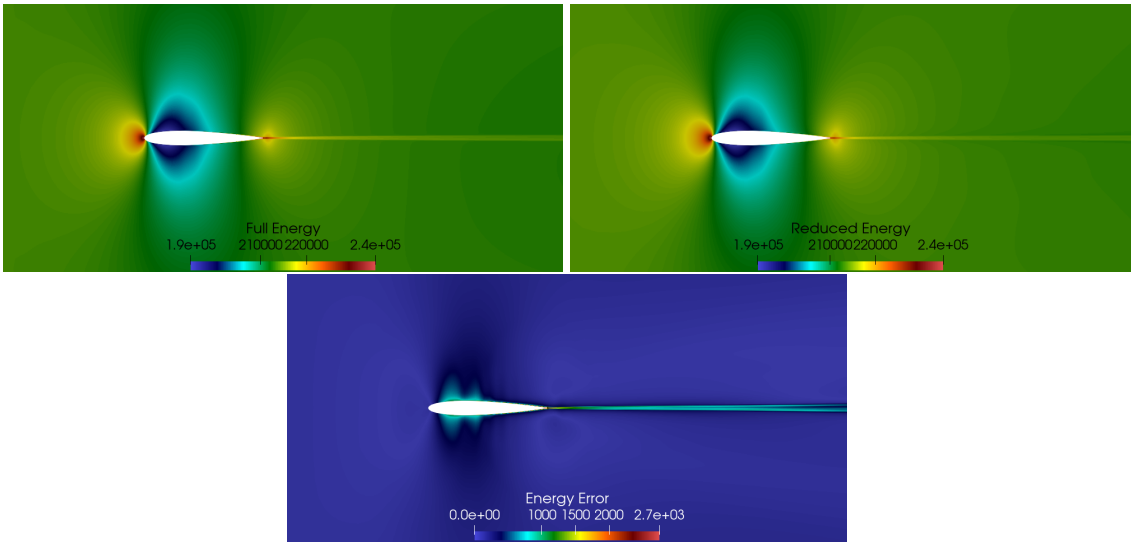


Figure 51: Comparison between full order (left) and reduced order (right) solutions for the energy field. These fields refer to the resolution of the problem for $\mu = 1.2 \times 10^{-3}$ which has been selected as a random value in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

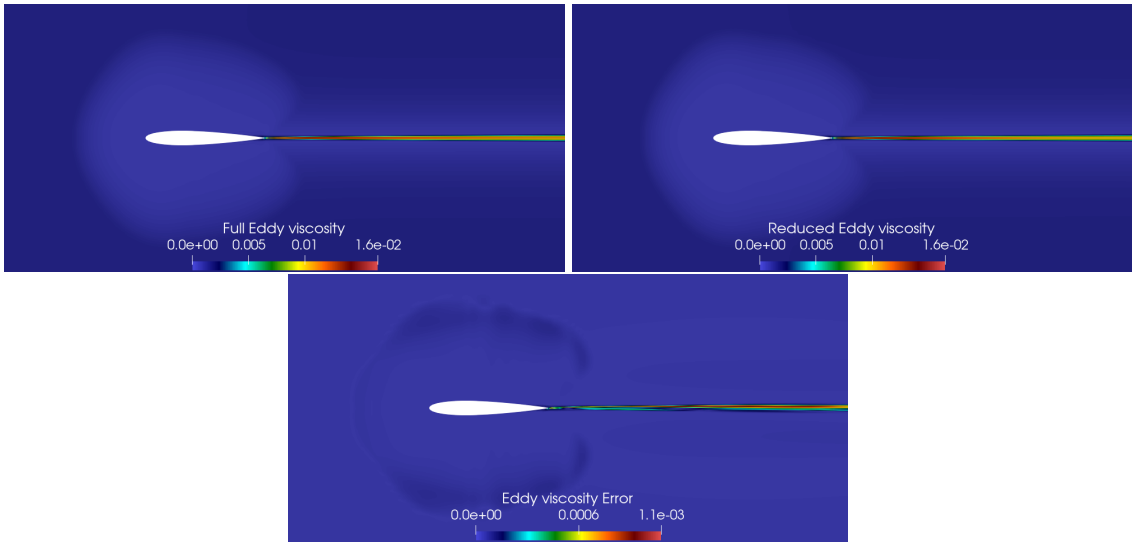


Figure 52: Comparison between full order (left) and reduced order (right) solutions for the eddy viscosity field. These fields refer to the resolution of the problem for $\mu = 1.2 \times 10^{-3}$ which has been selected as a random value in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

5.4 Compressible Navier-Stokes problems: geometrical parametrization

5.4.1 Shape deformation for a NACA0012 aerofoil by the use of bump functions in a compressible flow

In this section we present a second test case, focused on a geometry parametric problem: the shape of the aerofoil used for the previous test case is modified by the use of a bump function. In particular the foil is divided into a top and a bottom part by the chord. The bump function depicted in Figure 53 is added to the top and subtracted to the bottom surface, premultiplied by two different amplitude scalar factors: every solution is parametrized uniquely by two different scalar values. We use the same thermophysical properties used for physical parametrization case previously described, but the dynamic viscosity is fixed and equal to $1.74 \times 10^{-5} Pa\, s$. The equations to be solved for this test case are the following:

$$\begin{cases} \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}) = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} - (\mu + \mu_t) \left(\nabla\tilde{\boldsymbol{u}} + \nabla\tilde{\boldsymbol{u}}^T - \frac{2}{3}\nabla \cdot \tilde{\boldsymbol{u}}\boldsymbol{I} \right) + \overline{p}\boldsymbol{I} \right] = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla \cdot \left[\overline{\rho}\tilde{\boldsymbol{u}} \left(\tilde{e} + \frac{\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}}{2} \right) - \frac{C_p}{C_v} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \nabla\tilde{e} + \overline{p}\tilde{\boldsymbol{u}} \right] = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \tilde{\boldsymbol{u}} = [170, 0, 0]^T & \text{in } \Gamma_{in} \\ \nu \dfrac{\partial \tilde{\boldsymbol{u}}}{\partial \boldsymbol{n}} - \overline{p}\boldsymbol{n} = 0 & \text{in } \Gamma_{out} \end{cases} .$$

The inlet velocity has been slightly decreased since the random modification of the geometry may lead to high curvature areas where the flow could eventually become supersonic. This means that the *Mach* number at the inlet is now around 0.5.

For the offline phase, 50 random values have been selected:$\mu_{top_i}, \mu_{bottom_i} \in [0, 0.1]$ for $i = 1, ..., 50$ where $\begin{bmatrix} \mu_{top_1}, & \mu_{bottom_1} \\ \vdots & \vdots \\ \mu_{top_{50}}, & \mu_{bottom_{50}} \end{bmatrix} \equiv \mathbb{P}_h$. Full order eddy viscosity is calculated by the resolution of a $k - \omega$ turbulence model.

As pointed out in subsubsection 3.4.2, a general POD approach is not directly applicable to a geometrical parametrization problem since the L^2 norm used for the inner products is not well defined in case of multiple different domains. The mesh

in our case is once again moved thanks to an RBF algorithm by exploiting the displacements of the control points located over the surface of the aerofoil. In this case the reference configuration used for the norms evaluation is the unperturbed one.

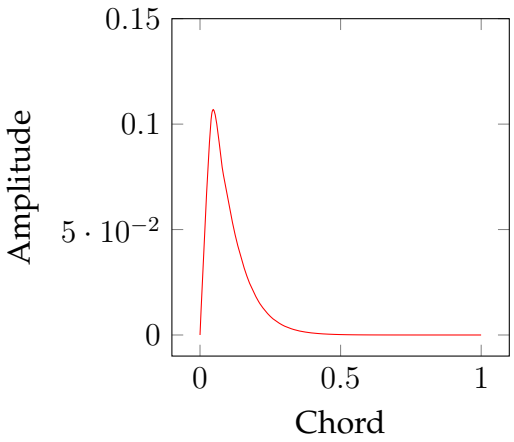


Figure 53: Shape of the employed bump function.

To test the online performances, 20 new scalar amplitude couples have been randomly selected. $N_u = N_p = N_e = 30$ has been fixed for the reconstruction of velocity, pressure and internal energy fields while 15 modal basis functions have been employed for ν_t. This choice is supported by what is shown in Figure 55: the increasing trend of the cumulated eigenvalues is appreciably fast and this fact allows the discarding of the higher modes. For every new parameter couple the mesh motion has to be performed but the procedure is very efficient since the coefficients for the RBF have to be evaluated and stored just once.

The same neural network architecture used for the previous application is employed here for what concerns the eddy viscosity. Again, looking at Figure 56, it can be noticed that the learning of the net seems to stabilize after 2×10^4 epochs which is the threshold we fixed for the training procedure.

The resulting L^2 norm errors for all the parameter couples in the online set are shown in Figure 57. Once again a discrepancy of about one order of magnitude can be noticed between the relative errors for the quantities of interest and the one calculated for the eddy viscosity. This is due to the fact that we are using a very simple and small network but it reveals to be reliable enough to make the online algorithm work in an accurate way.

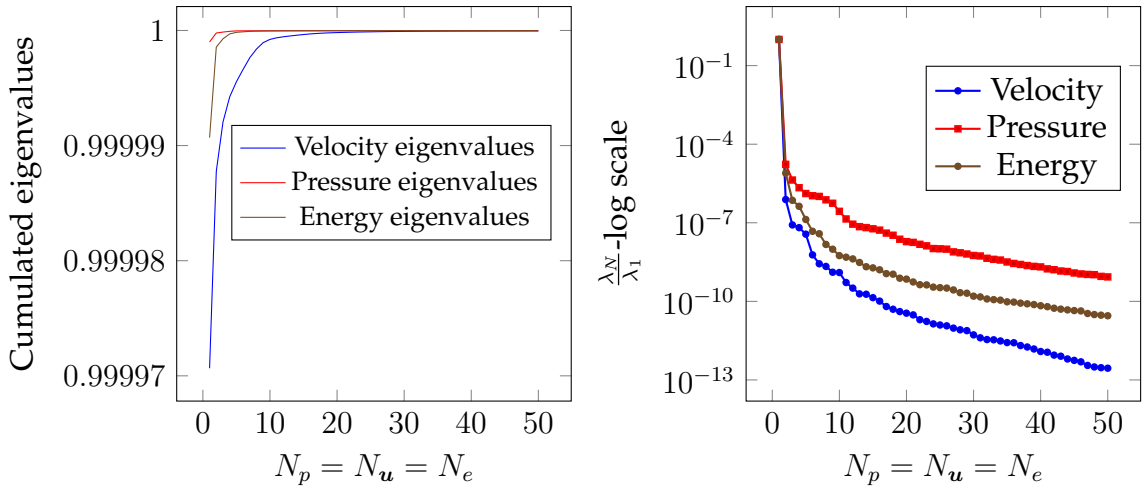


Figure 54: Eigenvalues trends for pressure, velocity and energy: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

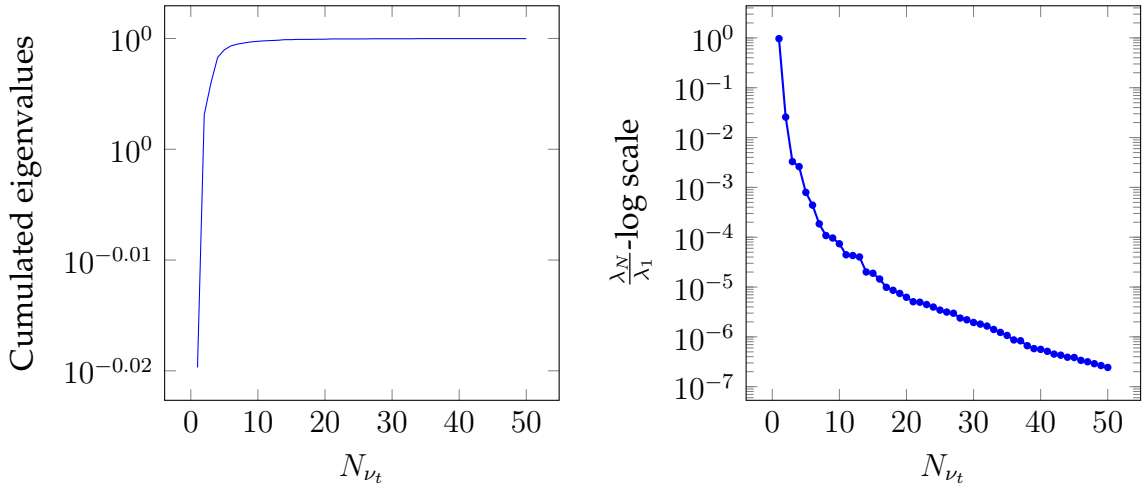


Figure 55: Eigenvalues trends for the eddy viscosity: cumulated eigenvalues are reported on the left, eigenvalues decays on the right.

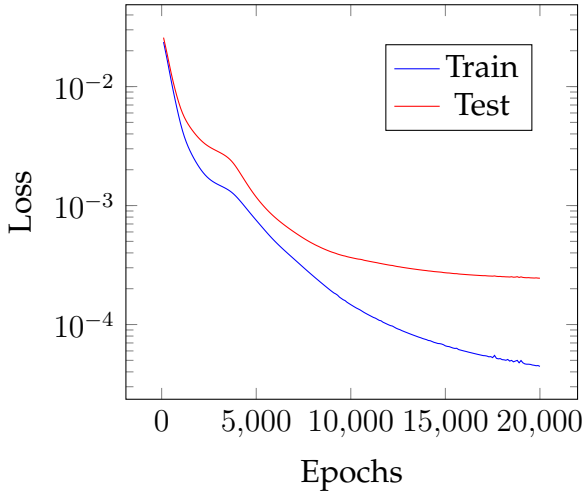


Figure 56: Loss function decay for both train and test sets.

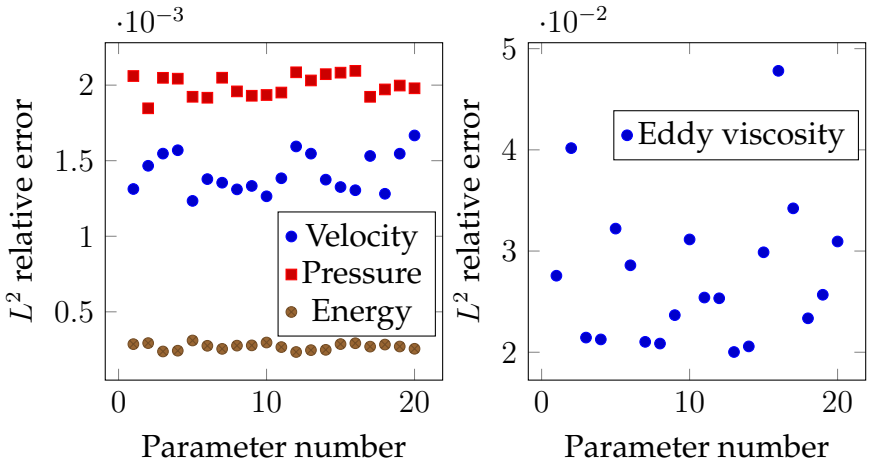


Figure 57: L^2 norm relative errors.

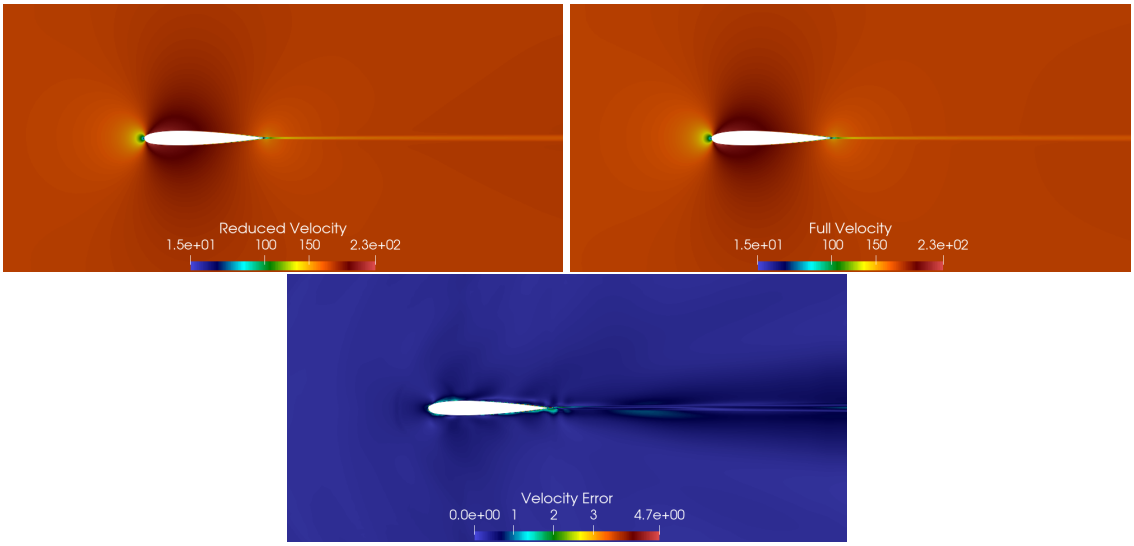


Figure 58: Comparison between full order (left) and reduced order (right) solutions for the velocity field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.004$ and $\mu_{bottom} \simeq 0.086$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

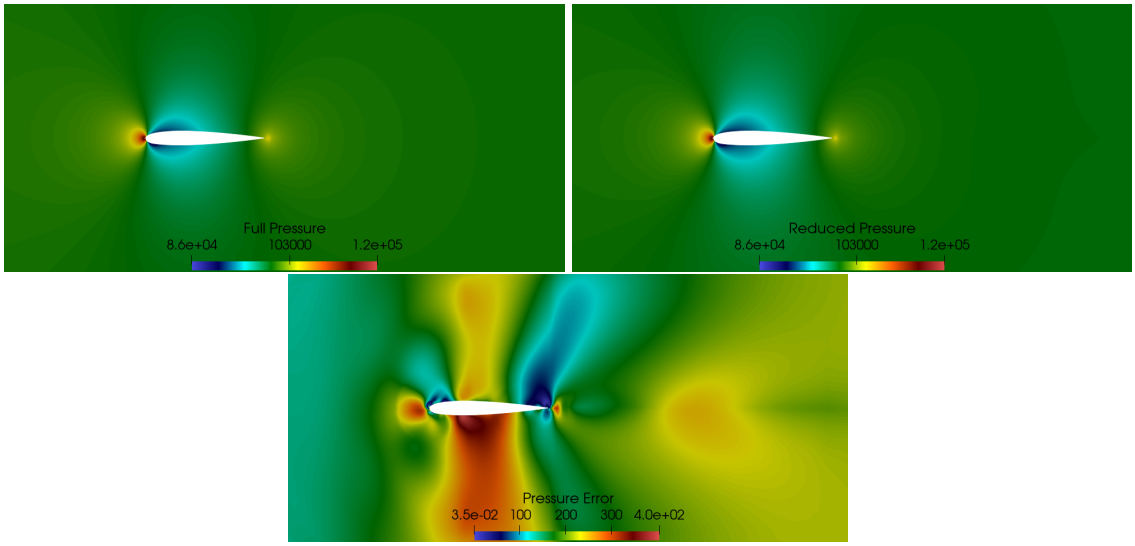


Figure 59: Comparison between full order (left) and reduced order (right) solutions for the pressure field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.004$ and $\mu_{bottom} \simeq 0.086$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

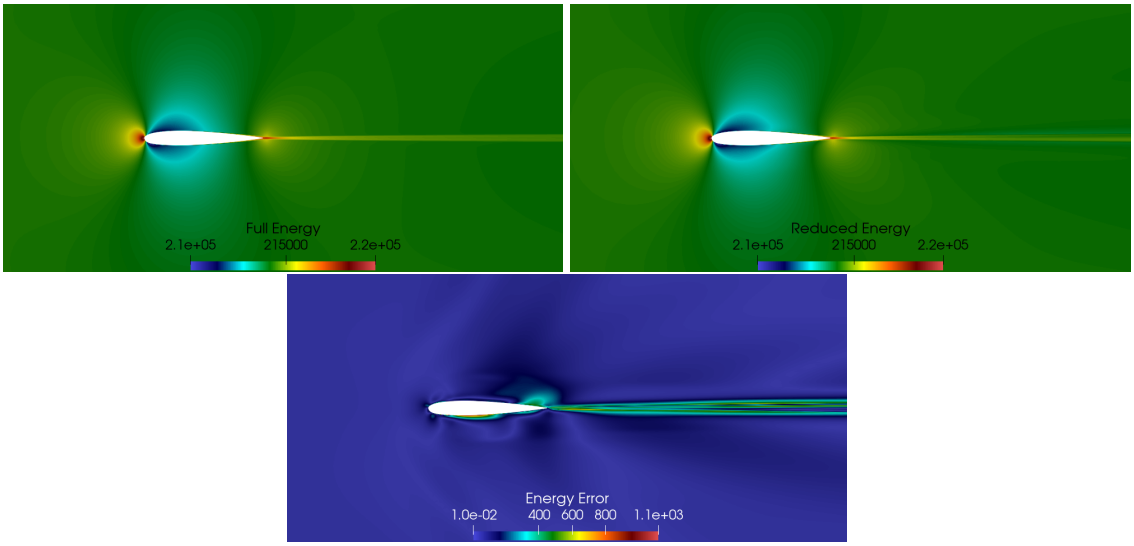


Figure 60: Comparison between full order (left) and reduced order (right) solutions for the energy field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.004$ and $\mu_{bottom} \simeq 0.086$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

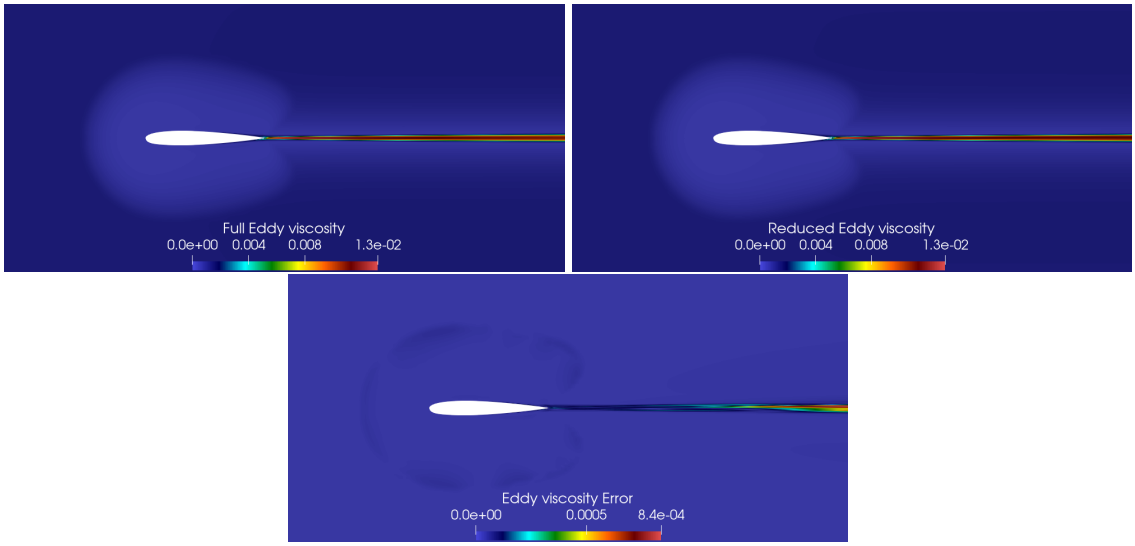


Figure 61: Comparison between full order (left) and reduced order (right) solutions for the eddy viscosity field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.004$ and $\mu_{bottom} \simeq 0.086$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

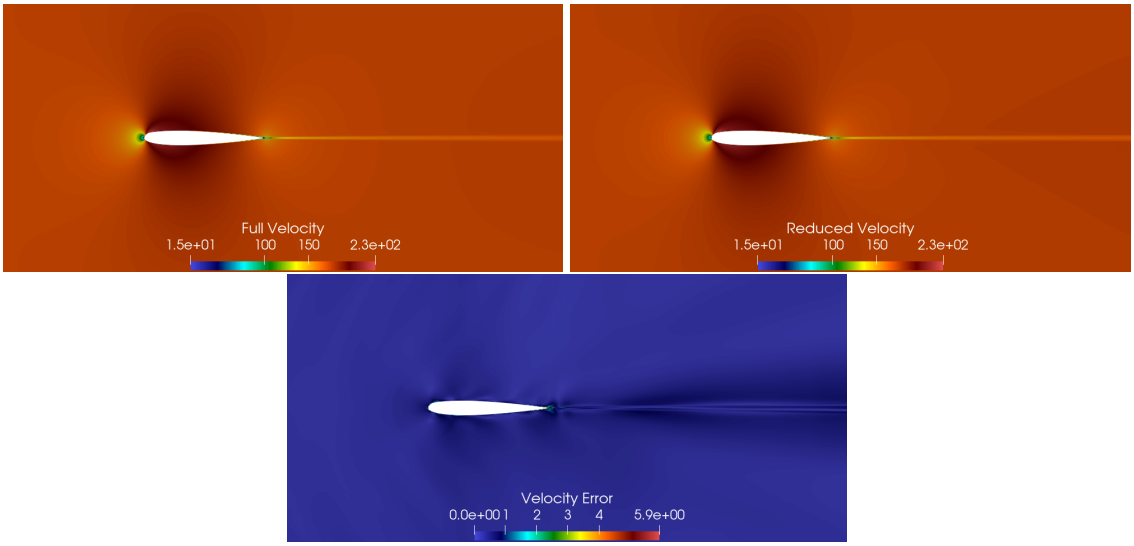


Figure 62: Comparison between full order (left) and reduced order (right) solutions for the velocity field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.095$ and $\mu_{bottom} \simeq 0.003$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

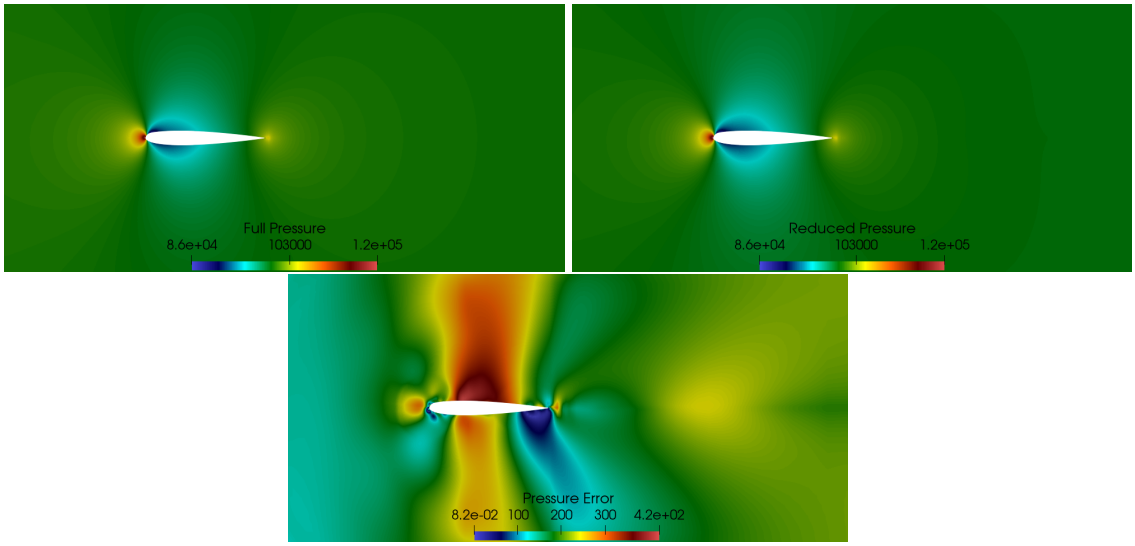


Figure 63: Comparison between full order (left) and reduced order (right) solutions for the pressure field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.095$ and $\mu_{bottom} \simeq 0.003$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

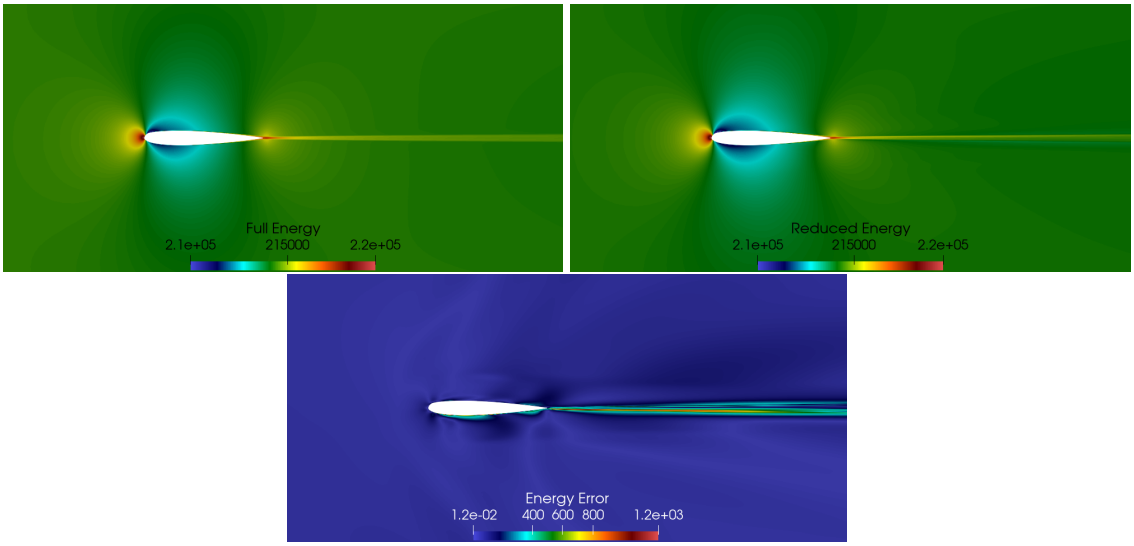


Figure 64: Comparison between full order (left) and reduced order (right) solutions for the energy field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.095$ and $\mu_{bottom} \simeq 0.003$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

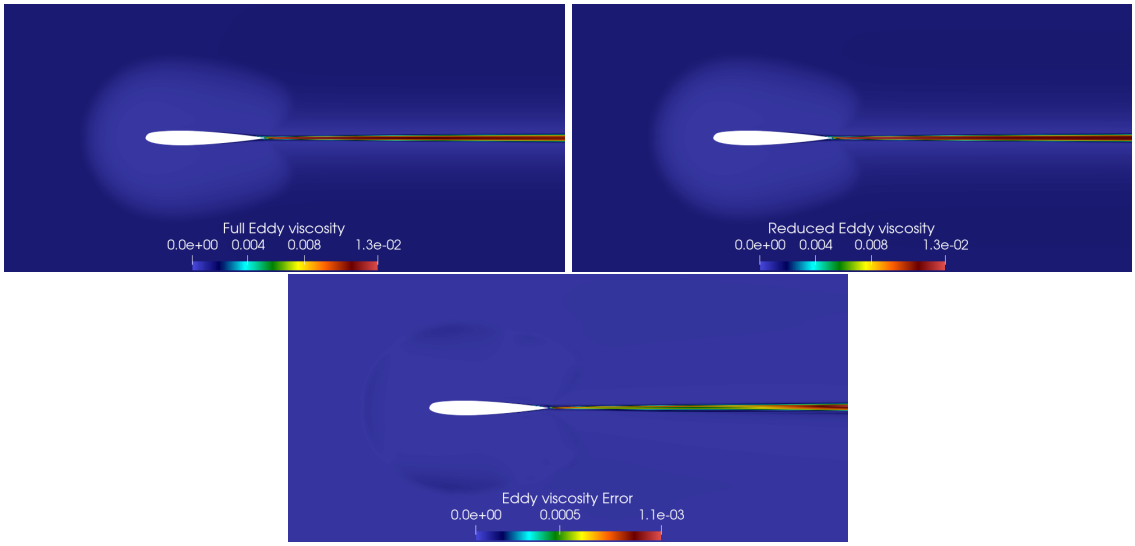


Figure 65: Comparison between full order (left) and reduced order (right) solutions for the eddy viscosity field. These fields refer to the resolution of the problem for $\mu_{top} \simeq 0.095$ and $\mu_{bottom} \simeq 0.003$ both selected as random values in the online parameter set. On bottom the pointwise error between full order and reduced order solutions is reported.

In figures 58-65 a comparison between offline and online solutions is depicted for two different parameter couples selected from the online set. Even if the two solutions are obtained for airfoil geometries that are perturbed in opposite directions, in both cases the method exhibits good reliability properties.

5.5 Context and overview

In this chapter we focused our attention on compressible flows devoting our efforts to the development of efficient and reliable methods for the resolution of different problems dependent on a parametrization. In particular we tried to combine all the different methodologies exposed in section 3 and section 4 and cast them to obtain a reliable architecture for compressible turbulent parametrized problems for both physical dependencies as well as geometrical ones.

The subsection 5.1 provides a wide description of the involved conservation equations together with possible simplifications, justified by physical considerations, and necessary restrictions, e.g. on *Mach* number application ranges.

By following what has been described in subsection 3.2 and subsection 4.2, subsection 5.2 introduces a segregated reduced order method for the resolution of the aforementioned parametrized test cases.

The efficiency of the method is supported by the presentation of two different test cases: a physical parametrization for the resolution of the flow dynamics around a NACA0012 aerofoil with a variable viscosity is shown in subsection 5.3 while a geometrical parametrization regarding the behaviour of an aerofoil with a variable shape is applied to the test case introduced in subsection 5.4.

6 Conclusions and future perspectives

This manuscript has been devoted to the presentation of some developed techniques for the resolution of fluid dynamics problems. In particular the novelties introduced by this thesis are:

- a fully consistent algorithm with respect to the full order scheme for the resolution of incompressible laminar flows. It has been introduced in section 3.2 and performances have been tested in sections 3.3 and 3.4. It shows good convergence properties without any necessity of additional stabilization for what concerns pressure solutions. This is due to the fact that we are utilizing a segregated scheme which tries to circumvent the saddle-point stability issue. In figure 9 we demonstrated that consistency between full order and reduced order solutions is actually paying off in terms of errors;
- a definition for the proper orthogonal decomposition in moving domains. Since norms are well posed only when the domain of definition is provided and fixed, it is in general not possible to compare solutions calculated over different geometries. To overcome this hurdle, a simple architecture based on mid configurations is proposed in 3.4.2;
- a hybrid approach for the resolution of incompressible turbulent flows. In section 4.2 we introduced an approach based on Galerkin projection for what concerns conservation laws while the eddy viscosity is reconstructed by a data-driven procedure. In particular in section 4.3 an RBF interpolation is utilized to reconstruct the contribution due to turbulence in the viscosity field for physical parametrization problems. On the contrary section 4.4 is focused on the resolution of geometrically parametrized problems by exploiting machine learning methods. In both cases good performances are ensured by the architectures;
- a new strategy for the selection of snapshots. The paragraph 4.4.1 includes the explanation on how to select the needed snapshots for the POD procedure so that enough information about the path followed by the full order algorithm is captured. This ploy improves a lot the performances of the ROM SIMPLE algorithm at the point that, for complex geometrically parametrized problems, a global minimum is not reached in case it is not used;
- a hybrid algorithm for treating compressible subsonic flows. Section 5.2 has been dedicated to the presentation of a reduced segregated algorithm for the resolution of fluid dynamics problems characterized by a medium-high *Mach* number. Once again neural networks are employed for the approximation of the eddy viscosity while state equation is used explicitly. Even for

very small and simple networks, leading to rough reconstructions of the viscous term, the method shows high robustness and reliability. Good performances are confirmed by physical parametric problems, as shown in section 5.3, as well as by geometrically parametrized test cases, as demonstrated in section 5.4.

The introduced methodologies are very promising since they cover a wide set of possible problems and provide accurate results while reducing, at the same time, the computational cost for their resolution.

Some proposals can be pointed out as future developments and enhancements of this work:

- one of the main weaknesses for the reduced algorithms we proposed is the necessity to reconstruct all the variables back in the full order solution manifold at each iteration so that these fields can be used to assemble the other equations involved in the iterative procedure. This setup obviously penalize the performances of the complete scheme. The best solution would be to use appropriate strategies as, e.g., a discrete empirical interpolation method, to obtain a fully reduced problem. In that case all the equations would be assembled by the only usage of the POD expansion coefficients;
- a possible alternative choice for the complete reduction of the SIMPLE algorithm would be the application of a dedicated neural network for the approximation of the reduced functionals, once the POD coefficients are provided. This would be a very cheap evaluation procedure and possibly an accurate architecture, avoiding a continuous throwback to the full order space;
- a better construction of the neural network affording the reconstruction of the eddy viscosity, for both the incompressible and the compressible models, may eventually lead to even lower errors with respect to the high fidelity solutions for what concerns the viscosity field. A deeper tuned network could improve the accuracy while preserving computations performances;
- all the proposed techniques have been developed to deal with steady state fluid flows. A big step forward for what concerns these methods would be their extension to unsteady problems. Since steady state cases just occupy a tiny fraction of possible applications, the implementation of a suited process for the resolution of time dependent Navier-Stokes equations would sensibly increase the applicability;
- for what regards the construction of the basis functions for the solution manifold covering, a good improvement could eventually lie in the application of a greedy algorithm, in place of the POD we selected for this work, exploit-

ing the evaluation of a suited error estimator. This choice would possibly decrease the computational cost related to the offline stage in such a way that the online accuracy is not affected by the variation;

- in this thesis *Reynolds* averaged Navier-Stokes equations have been entrusted with the treatment of turbulence. This aspect could be improved by relying on large eddy simulations instead of using averaging techniques so that a more accurate and detailed solution could be reached.

References

[1] Syed R. Ahmed, Alexander G. Ramm, and Gunter Faltin. "Some salient features of the time-averaged ground vehicle wake". In: *SAE Transactions* (1984), pp. 473–503.

[2] Imran Akhtar, Ali H. Nayfeh, and Calvin J. Ribbens. "On the stability and extension of reduced-order Galerkin models in incompressible flows". In: *Theoretical and Computational Fluid Dynamics* 23.3 (2009), pp. 213–237.

[3] Giancarlo Alfonsi. "Reynolds-averaged Navier-Stokes equations for turbulence modeling". In: *Applied Mechanics Reviews* 62.4 (2009).

[4] Antoni Alomar, Aurélie Nicole, Denis Sipp, Valérie Rialland, and François Vuillot. "Reduced-order model of a reacting, turbulent supersonic jet based on proper orthogonal decomposition". In: *Theoretical and Computational Fluid Dynamics* (2020), pp. 1–29.

[5] David Amsallem and Charbel Farhat. "Interpolation method for adapting reduced-order models and application to aeroelasticity". In: *AIAA journal* 46.7 (2008), pp. 1803–1813.

[6] John D. Anderson and John F. Wendt. *Computational fluid dynamics*. Vol. 206. Springer, 1995.

[7] John D. Anderson Jr. *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.

[8] Mejdi Azaiez, Tomás Chacón Rebollo, and Samuele Rubino. "A cure for instabilities due to advection-dominance in POD solution to advection-diffusion-reaction equations". In: *Journal of Computational Physics* 425 (2021), p. 109916.

[9] Joan Baiges, Ramon Codina, and Sergio R. Idelsohn. "Reduced-order modelling strategies for the finite element approximation of the incompressible Navier-Stokes equations". In: *Numerical Simulations of Coupled Problems in Engineering*. Springer, 2014, pp. 189–216.

[10] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. "Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations". In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1136–1161.

[11] Matthew F. Barone, Irina Kalashnikova, Daniel J. Segalman, and Heidi K. Thornquist. "Stable Galerkin reduced order models for linearized compressible flow". In: *Journal of Computational Physics* 228.6 (2009), pp. 1932–1946.

[12] Matthew F. Barone, Daniel Segalman, Heidi Thornquist, and Irina Kalashnikova. "Galerkin reduced order models for compressible flow with structural interaction". In: *46th AIAA Aerospace Sciences Meeting and Exhibit*. 2008, p. 612.

[13] Armin Beckert and Holger Wendland. "Multivariate interpolation for fluid–struc- ture interaction problems using radial basis functions". In: *Aerospace Science and Technology* 5.2 (2001), pp. 125–134.

[14] Peter Benner, Serkan Gugercin, and Karen Willcox. "A survey of projection-based model reduction methods for parametric dynamical systems". In: *SIAM review* 57.4 (2015), pp. 483–531.

[15] Peter Benner, Mario Ohlberger, Anthony T. Patera, Gianluigi Rozza, and Karsten Urban. *Model reduction of parametrized systems*. Springer, 2017.

[16] Peter Benner, Wil Schilders, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, and Luis M. Silveira. *Model Order Reduction, vol. 1-3*. De Gruyter, 2020.

[17] Michel Bergmann, Charles-Henri Bruneau, and Angelo Iollo. "Enablers for robust POD models". In: *Journal of Computational Physics* 228.2 (2009), pp. 516–538.

[18] Luigi C. Berselli, Traian Iliescu, and William J. Layton. "Mathematics of Large Eddy Simulation of Turbulent Flows." In: *Journal of Fluid Mechanics* 582 (2007), pp. 473–475.

[19] Marco Evangelos Biancolini, Katia Capellini, Emiliano Costa, Corrado Groth, and Simona Celi. "Fast interactive CFD evaluation of hemodynamics assisted by RBF mesh morphing and reduced order models: the case of aTAA modelling". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14.4 (2020), pp. 1227–1238.

[20] Jiri Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[21] Daniele Boffi, Franco Brezzi, Michel Fortin, et al. *Mixed finite element methods and applications*. Vol. 44. Springer, 2013.

[22] Sylvain Boivin, Florent Cayré, and Jean-Marc Herard. "A finite volume method to solve the Navier–Stokes equations for incompressible flows on unstructured meshes". In: *International journal of thermal sciences* 39.8 (2000), pp. 806–825.

[23] Jay P. Boris, Fernando F. Grinstein, Elaine S. Oran, and Ronald L. Kolbe. "New insights into large eddy simulation". In: *Fluid dynamics research* 10.4-6 (1992), p. 199.

[24] Frank M. Bos, Bas W. van Oudheusden, and Hester Bijl. "Radial basis function based mesh deformation applied to simulation of flow around flapping wings". In: *Computers & Fluids* 79 (2013), pp. 167–177.

[25] Steven L. Brunton and Nathan J. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[26] Sandeep R. Bukka, Rachit Gupta, Allan R. Magee, and Rajeev K. Jaiman. "Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models". In: *Physics of Fluids* 33.1 (2021), p. 013601.

[27] Guido Buresti. "A note on Stokes' hypothesis". In: *Acta Mechanica* 226 (Oct. 2015).

[28] John Burkardt, Max Gunzburger, and Hyung-Chun Lee. "POD and CVT-based reduced-order modeling of Navier–Stokes flows". In: *Computer methods in applied mechanics and engineering* 196.1-3 (2006), pp. 337–355.

[29] Kevin Carlberg, Youngsoo Choi, and Syuzanna Sargsyan. "Conservative model reduction for finite-volume models". In: *Journal of Computational Physics* 371 (2018), pp. 280–314.

[30] Patrice Castonguay and Siva Nadarajah. "Effect of shape parameterization on aerodynamic shape optimization". In: *45th AIAA Aerospace Sciences Meeting and Exhibit*. 2007, p. 59.

[31] Saifon Chaturantabut and Danny C Sorensen. "Nonlinear model reduction via discrete empirical interpolation". In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[32] Hamn C. Chen, Viredra C. Patel, and Shihao Ju. "Solutions of Reynolds-averaged Navier-Stokes equations for three-dimensional incompressible flows". In: *Journal of Computational Physics* 88.2 (1990), pp. 305–336.

[33] Wang Chen, Jan S. Hesthaven, Bai Junqiang, Yasong Qiu, Zhang Yang, and Yang Tihao. "Greedy nonintrusive reduced order model for fluid dynamics". In: *AIAA Journal* 56.12 (2018), pp. 4927–4943.

[34] Francisco Chinesta, Pierre Ladeveze, and Elias Cueto. "A short review on model order reduction based on proper generalized decomposition". In: *Archives of Computational Methods in Engineering* 18.4 (2011), pp. 395–404.

[35] Alexandre J. Chorin. "Numerical solution of the Navier-Stokes equations". In: *Mathematics of computation* 22.104 (1968), pp. 745–762.

[36] Stéphane Clain, Steven Diot, and Raphaël Loubère. "A high-order finite volume method for systems of conservation laws—Multi-dimensional Optimal Order Detection (MOOD)". In: *Journal of computational Physics* 230.10 (2011), pp. 4028–4050.

[37] Ray W. Clough. "The finite element method in plane stress analysis". In: *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960*. 1960.

[38] Ramon Codina, Ricardo Reyes, and Joan Baiges. "A posteriori error estimates in a finite element VMS-based reduced order model for the incompressible Navier-Stokes equations". In: *Mechanics Research Communications* 112 (2021), p. 103599.

[39] Derek J. Dalle, Matt L. Fotia, and James F. Driscoll. "Reduced-order modeling of two-dimensional supersonic flows with applications to scramjet inlets". In: *Journal of Propulsion and Power* 26.3 (2010), pp. 545–555.

[40] Raphaël Danchin. "Global existence in critical spaces for compressible Navier-Stokes equations". In: *Inventiones Mathematicae* 141.3 (2000), pp. 579–614.

[41] Aukje De Boer, Martijn S. Van der Schoot, and Hester Bijl. "Mesh deformation based on radial basis function interpolation". In: *Computers & structures* 85.11-14 (2007), pp. 784–795.

[42] Castro M. J. Díaz, Enrique D. Fernández-Nieto, and Ana M. Ferreiro. "Sediment transport models in shallow water equations and numerical approach by high order finite volume methods". In: *Computers & Fluids* 37.3 (2008), pp. 299–316.

[43] Manfred Dobrowolski. "On the LBB condition in the numerical analysis of the Stokes equations". In: *Applied numerical mathematics* 54.3-4 (2005), pp. 314–323.

[44] Zlatko Drmac and Serkan Gugercin. "A new selection operator for the discrete empirical interpolation method–improved a priori error bound and extensions". In: *SIAM Journal on Scientific Computing* 38.2 (2016), A631–A648.

[45] Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger. "Reduced basis method for finite volume approximation of evolution equations on parametrized geometries". In: *Proceedings of ALGORITMY*. Vol. 2008. 2009, pp. 111–120.

[46] Antoine Dumon, Cyrille Allery, and Amine Ammar. "Proper Generalized Decomposition method for incompressible Navier–Stokes equations with a spectral discretization". In: *Applied Mathematics and Computation* 219.15 (2013), pp. 8145–8162.

[47] Jack G. M. Eggels, Friedemann Unger, MH Weiss, Jerry Westerweel, Ronald J. Adrian, Rainer Friedrich, and Frans T. M. Nieuwstadt. "Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment". In: *Journal of Fluid Mechanics* 268 (1994), pp. 175–210.

[48] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. "Finite volume methods". In: *Handbook of numerical analysis* 7 (2000), pp. 713–1018.

[49] Michalis Frangos, Youssef Marzouk, Karen Willcox, and Bart van Bloemen Waanders. "Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems [Chapter 7]". In: (2010).

[50] Benjamin Fröhlich, Jan Gade, Florian Geiger, Manfred Bischoff, and Peter Eberhard. "Geometric element parameterization and parametric model order reduction in finite element based shape optimization". In: *Computational Mechanics* 63.5 (2019), pp. 853–868.

[51] Thomas B. Gatski and Mark N. Glauser. "Proper orthogonal decomposition based turbulence modeling". In: *Instability, Transition, and Turbulence*. Springer, 1992, pp. 498–510.

[52] Sokratia Georgaka, Giovanni Stabile, Kelbij Star, Gianluigi Rozza, and Michael J. Bluck. "A hybrid reduced order method for modelling turbulent heat transfer problems". In: *Computers & Fluids* 208 (2020), p. 104615.

[53] Fernando F. Grinstein, Len G. Margolin, and William J. Rider. *Implicit large eddy simulation*. Vol. 10. Cambridge university press Cambridge, 2007.

[54] Bernard Haasdonk and Mario Ohlberger. "Adaptive basis enrichment for the reduced basis method applied to finite volume schemes". In: *Proc. 5th International Symposium on Finite Volumes for Complex Applications*. 2008, pp. 471–478.

[55] Bernard Haasdonk and Mario Ohlberger. "Reduced basis method for explicit finite volume approximations of nonlinear conservation laws". In: *Proc. 12th International Conference on Hyperbolic Problems: Theory, Numerics, Application*. Citeseer. 2008.

[56] Bernard Haasdonk and Mario Ohlberger. "Reduced basis method for finite volume approximations of parametrized linear evolution equations". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42.2 (2008), pp. 277–302.

[57] Bernard Haasdonk, Mario Ohlberger, and Gianluigi Rozza. "A reduced basis method for evolution schemes with parameter-dependent explicit operators". In: *ETNA, Electronic Transactions on Numerical Analysis* 32.ARTICLE (2008), pp. 145–168.

[58] Martin W. Hess, Annalisa Quaini, and Gianluigi Rozza. "A comparison of reduced-order modeling approaches for PDEs with bifurcating solutions". In: *arXiv preprint arXiv:2010.07370* (2020).

[59] Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Vol. 590. Springer, 2016.

[60] Jan S. Hesthaven and Stefano Ubbiali. "Non-intrusive reduced order modeling of nonlinear problems using neural networks". In: *Journal of Computational Physics* 363 (2018), pp. 55–78.

[61] Saddam Hijazi, Shafqat Ali, Giovanni Stabile, Francesco Ballarin, and Gianluigi Rozza. "The effort of increasing Reynolds number in projection-based reduced order methods: from laminar to turbulent flows". In: *Numerical Methods for Flows*. Springer, 2020, pp. 245–264.

[62] Saddam Hijazi, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. "Data-driven POD-Galerkin reduced order model for turbulent flows". In: *Journal of Computational Physics* 416 (2020), p. 109513.

[63] Charles Hirsch. *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier, 2007.

[64] Tobias Holzmann. “Mathematics, numerics, derivations and OpenFOAM®”. In: *Loeben, Germany: Holzmann CFD* (2016).

[65] Traian Iliescu and Zhu Wang. “Variational multiscale proper orthogonal decomposition: Navier-stokes equations”. In: *Numerical Methods for Partial Differential Equations* 30.2 (2014), pp. 641–663.

[66] Lucian Itu, Puneet Sharma, Vwrel Mihalef, Ali Kamen, Constantin Suciu, and Dorm Lomaniciu. “A patient-specific reduced-order model for coronary circulation”. In: *2012 9th IEEE international symposium on biomedical imaging (ISBI)*. IEEE. 2012, pp. 832–835.

[67] Niclas Jansson, Johan Hoffman, and Murtazo Nazarov. “Adaptive simulation of turbulent flow past a full car model”. In: *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2011, pp. 1–8.

[68] Hrvoje Jasak. “Error analysis and estimation for the finite volume method with applications to fluid flows.” In: (1996).

[69] Irina Kalashnikova and Matthew F. Barone. “Efficient non-linear proper orthogonal decomposition/Galerkin reduced order models with stable penalty enforcement of boundary conditions”. In: *International Journal for Numerical Methods in Engineering* 90.11 (2012), pp. 1337–1362.

[70] Efthymios N. Karatzas, Francesco Ballarin, and Gianluigi Rozza. “Projection-based reduced order models for a cut finite element method in parametrized domains”. In: *Computers & Mathematics with Applications* 79.3 (2020), pp. 833–851.

[71] Youngkyu Kim, Youngsoo Choi, David Widemann, and Tarek Zohdi. “Efficient nonlinear manifold reduced order model”. In: *arXiv preprint arXiv:2011.07727* (2020).

[72] Dmitry Kolmogorov. “Finite Volume Methods for Incompressible Navier-Stokes Equations on Collocated Grids with Nonconformal Interfaces”. In: *DTU Wind Energy* (2014).

[73] Jiaqing Kou, Soledad Le Clainche, and Weiwei Zhang. “A reduced-order model for compressible flows with buffeting condition using higher order dynamic mode decomposition with a mode selection criterion”. In: *Physics of Fluids* 30.1 (2018), p. 016103.

[74] Karl Kunisch and Stefan Volkwein. "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics". In: *SIAM Journal on Numerical analysis* 40.2 (2002), pp. 492–515.

[75] Nathan J. Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[76] Nathan J. Kutz, Xing Fu, and Steven L. Brunton. "Multiresolution dynamic mode decomposition". In: *SIAM Journal on Applied Dynamical Systems* 15.2 (2016), pp. 713–735.

[77] Brian Edward Launder and Dudley Brian Spalding. "The numerical computation of turbulent flows". In: *Numerical prediction of flow, heat transfer, turbulence and combustion*. Elsevier, 1983, pp. 96–116.

[78] Patrick LeGresley and Juan Alonso. "Investigation of non-linear projection for pod based reduced order models for aerodynamics". In: *39th Aerospace Sciences Meeting and Exhibit*. 2001, p. 926.

[79] Randall J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[80] Thuan Lieu and Charbel Farhat. "Adaptation of aeroelastic reduced-order models and application to an F-16 configuration". In: *AIAA journal* 45.6 (2007), pp. 1244–1257.

[81] Stefano Lorenzi, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. "POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations". In: *Computer Methods in Applied Mechanics and Engineering* 311 (2016), pp. 151–179.

[82] David J. Lucia and Philip S. Beran. "Projection methods for reduced order models of compressible flows". In: *Journal of Computational Physics* 188.1 (2003), pp. 252–280.

[83] Hugo F. S. Lui and William R. Wolf. "Construction of reduced-order models for fluid flows using deep feedforward neural networks". In: *Journal of Fluid Mechanics* 872 (2019), pp. 963–994.

[84] Hugo F. S. Lui and William R. Wolf. "Flow modal decomposition and deep neural networks for the construction of reduced order models of compressible flows". In: *AIAA Scitech 2019 Forum*. 2019, p. 1407.

[85] John L. Lumley. "The structure of inhomogeneous turbulent flows". In: *Atmospheric turbulence and radio wave propagation* (1967).

[86] Robert MacCormack and Alvin Paullay. "Computational efficiency achieved by time splitting of finite difference operators". In: *10th Aerospace Sciences Meeting*. 1972, p. 154.

[87] Ernst Mach. *Die Mechanik in ihrer Entwickelung historisch-kritisch dargestellt: Mit einen Anhang," Das Verhältnis der Machschen Gadankenwelt zur Relativitätstheorie"*. FA Brockhaus, 1921.

[88] Andrea Mannarino and Paolo Mantegazza. "Nonlinear aeroelastic reduced order modeling by recurrent neural networks". In: *Journal of Fluids and Structures* 48 (2014), pp. 103–121.

[89] Patrick W. McDonald. "The computation of transonic flow through two-dimensional gas turbine cascades". In: *ASME 1971 International Gas Turbine Conference and Products Show*. American Society of Mechanical Engineers Digital Collection. 1971.

[90] Matthieu Minguez, Richard Pasquetti, and Eric Serre. "High-order large-eddy simulation of flow over the "Ahmed body" car model". In: *Physics of fluids* 20.9 (2008), p. 095101.

[91] Arvind T. Mohan and Datta V. Gaitonde. "A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks". In: *arXiv preprint arXiv:1804.09269* (2018).

[92] Parviz Moin and Krishnan Mahesh. "Direct numerical simulation: a tool in turbulence research". In: *Annual review of fluid mechanics* 30.1 (1998), pp. 539–578.

[93] Robert D. Moser, John Kim, and Nagi N. Mansour. "Direct numerical simulation of turbulent channel flow up to Re τ= 590". In: *Physics of fluids* 11.4 (1999), pp. 943–945.

[94] Fadl Moukalled, Luca Mangani, and Marwan Darwish. *The finite volume method in computational fluid dynamics*. Vol. 113. Springer, 2016.

[95] Markus Mrosek, Carsten Othmer, and Rolf Radespiel. "Reduced-Order Modeling of Vehicle Aerodynamics via Proper Orthogonal Decomposition". In: *SAE International Journal of Passenger Cars-Mechanical Systems* 12.3 (2019), pp. 225–237.

[96] Ngoc C. Nguyen and Jaime Peraire. "An efficient reduced-order modeling approach for non-linear parametrized partial differential equations". In: *International Journal for Numerical Methods in Engineering* 76.1 (2008), pp. 27–55.

[97] Nikolay Nikitin. "Finite-difference method for incompressible Navier–Stokes equations in arbitrary orthogonal curvilinear coordinates". In: *Journal of Computational Physics* 217.2 (2006), pp. 759–781.

[98] Anthony Nouy. "A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations". In: *Computer Methods in Applied Mechanics and Engineering* 199.23-24 (2010), pp. 1603–1626.

[99] *OpenFOAM documentation website*. Accessed: 29/07/2021. URL: https://openfoam.org/.

[100] Ronald L. Panton. *Incompressible flow*. John Wiley & Sons, 2006.

[101] Suhas V. Patankar and D. Brian Spalding. "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows". In: *Numerical prediction of flow, heat transfer, turbulence and combustion*. Elsevier, 1983, pp. 54–73.

[102] Anthony T. Patera. "A spectral element method for fluid dynamics: laminar flow in a channel expansion". In: *Journal of computational Physics* 54.3 (1984), pp. 468–488.

[103] Anthony T. Patera and Gianluigi Rozza. *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*. 2007.

[104] Arthur Paul-Dubois-Taine and David Amsallem. "An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models". In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1262–1292.

[105] Benjamin Peherstorfer, Daniel Butnaru, Karen Willcox, and Hans-Joachim Bungartz. "Localized discrete empirical interpolation method". In: *SIAM Journal on Scientific Computing* 36.1 (2014), A168–A192.

[106] Ugo Piomelli. "Large-eddy simulation: achievements and challenges". In: *Progress in aerospace sciences* 35.4 (1999), pp. 335–362.

[107] Pavel Plotnikov and Jan Sokołowski. *Compressible Navier-Stokes equations: theory and shape optimization*. Vol. 73. Springer Science & Business Media, 2012.

[108] Joshua L. Proctor, Steven L. Brunton, and Nathan J. Kutz. "Dynamic mode decomposition with control". In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161.

[109] Christophe Prud'Homme, Dimitrios V. Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T. Patera, and Gabriel Turinici. "Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods". In: *J. Fluids Eng.* 124.1 (2002), pp. 70–80.

[110] Luigi Quartapelle and Franco Auteri. *Fluidodinamica comprimibile*. Casa Editrice Ambrosiana, 2013.

[111] Luigi Quartapelle and Franco Auteri. *Fluidodinamica incomprimibile*. Casa Editrice Ambrosiana, 2013.

[112] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*. Vol. 92. Springer, 2015.

[113] Sk Rahman, Adil Rasheed, and Omer San. "A hybrid analytics paradigm combining physics-based modeling and data-driven modeling to accelerate incompressible flow solvers". In: *Fluids* 3.3 (2018), p. 50.

[114] Reetesh Ranjan and Suresh Menon. "Nonlinear Reduced Order Modeling for Large Eddy Simulation of Turbulent Reacting Flows". In: *AIAA Scitech 2020 Forum*. 2020, p. 2140.

[115] Tomás Chacón Rebollo, Enrique Delgado Avila, Macarena Gómez Mármol, Fran- cesco Ballarin, and Gianluigi Rozza. "On a certified Smagorinsky reduced basis turbulence model". In: *SIAM Journal on Numerical Analysis* 55.6 (2017), pp. 3047–3067.

[116] Osborne Reynolds. "XXIX. 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". In: *Philosophical Transactions of the Royal society of London* 174 (1883), pp. 935–982.

[117] Gianluigi Rozza and Andrea Manzoni. "Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics". In: *Proceedings of the ECCOMAS CFD 2010, V European Conference on Computational Fluid Dynamics*. CONF. 2010.

[118] Gianluigi Rozza and Karen Veroy. "On the stability of the reduced basis method for Stokes equations in parametrized domains". In: *Computer Methods in Applied Mechanics and Engineering* 196.7 (2007), pp. 1244–1260.

[119] Filippo Salmoiraghi, Francesco Ballarin, Giovanni Corsi, Andrea Mola, Marco Tezzele, and Gianluigi Rozza. "Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: overview and perspectives". In: *VII European Congress on Computational Methods in Applied Sciences and Engineering*. Vol. 1. Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece. 2016, pp. 1013–1031.

[120] Madhusudanan K. Sampath, Anestis Dounavis, and Roni Khazaka. "Generation of geometrically parameterized reduced order models for full wave problems". In: *2006 IEEE Electrical Performane of Electronic Packaging*. IEEE. 2006, pp. 343–346.

[121] Omer San and Traian Iliescu. "Proper orthogonal decomposition closure models for fluid flows: Burgers equation". In: *International Journal of Numerical Analysis & Modeling* (2014).

[122] Peter J. Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.

[123] Salman M. Siddiqui, Adil Rasheed, and Trond Kvamsdal. "Numerical assessment of rans turbulence models for the development of data driven reduced order models". In: *Ocean Engineering* 196 (2020), p. 106799.

[124] John R. Singler. "Optimality of balanced proper orthogonal decomposition for data reconstruction". In: *Numerical functional analysis and optimization* 31.7 (2010), pp. 852–869.

[125] Lawrence Sirovich. "Turbulence and the dynamics of coherent structures, Parts I-III". In: *Quarterly of applied mathematics* (1987), pp. 561–590.

[126] G. Stabile and G. Rozza. *ITHACA-FV - In real Time Highly Advanced Computational Applications for Finite Volumes*. Accessed: 29/07/2021. URL: http://www.mathlab.sissa.it/ithaca-fv.

[127] Giovanni Stabile, Francesco Ballarin, Giacomo Zuccarino, and Gianluigi Rozza. "A reduced order variational multiscale approach for turbulent flows". In: *Advances in Computational Mathematics* 45.5 (2019), pp. 2349–2368.

[128] Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, and Gianluigi Rozza. "POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder". In: *Communications in Applied and Industrial Mathematics* 8.1 (2017), pp. 210–236.

[129] Giovanni Stabile and Gianluigi Rozza. "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations". In: *Computers & Fluids* 173 (2018), pp. 273–284.

[130] Giovanni Stabile, Matteo Zancanaro, and Gianluigi Rozza. "Efficient geometrical parametrization for finite-volume-based reduced order methods". In: *International Journal for Numerical Methods in Engineering* 121.12 (2020), pp. 2655–2682.

[131] Roger Temam. *Navier-Stokes equations: theory and numerical analysis*. Vol. 343. American Mathematical Soc., 2001.

[132] Philip A. Thompson. *Compressible-fluid Dynamics*. Advanced engineering series. 1988.

[133] Philip A. Thompson and Gordon S. Beavers. "Compressible-fluid dynamics". In: *Journal of Applied Mechanics* 39.2 (1972), p. 366.

[134] Sebastian Ullmann, Stefan Löbig, and Jens Lang. "Adaptive large eddy simulation and reduced-order modeling". In: *Flow and Combustion in Advanced Gas Turbine Combustors*. Springer, 2013, pp. 349–378.

[135] Karsten Urban, Stefan Volkwein, and Oliver Zeeb. "Greedy sampling using nonlinear optimization". In: *Reduced Order Methods for modeling and computational reduction*. Springer, 2014, pp. 137–157.

[136] Milton Van Dyke. *An album of fluid motion*. Vol. 176. Parabolic press Stanford, 1982.

[137] Karen Veroy and Anthony T. Patera. "Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds". In: *International Journal for Numerical Methods in Fluids* 47.8-9 (2005), pp. 773–788.

[138] Qian Wang, Jan S. Hesthaven, and Deep Ray. "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem". In: *Journal of computational physics* 384 (2019), pp. 289–307.

[139] John F. Wendt. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2008.

[140] David C Wilcox. "Reassessment of the scale-determining equation for advanced turbulence models". In: *AIAA journal* 26.11 (1988), pp. 1299–1310.

[141] David C. Wilcox and R. Traci. "A complete model of turbulence". In: *9th Fluid and PlasmaDynamics Conference*. 1976, p. 351.

[142] Karen Willcox and Guillaume Lassaux. "Model reduction of an actively controlled supersonic diffuser". In: *Dimension Reduction of Large-Scale Systems*. Springer, 2005, pp. 357–361.

[143] Kyle J. Woolwine, Kenneth E. Jansen, George Kopasakis, and Joseph W. Connolly. "Reduced order modeling of a supersonic flow field". In: *Journal of Propulsion and Power* 35.2 (2019), pp. 277–288.

[144] Dunhui Xiao, Fangxin Fang, Andrew G. Buchan, Christopher C. Pain, Ionel M. Navon, and Ann Muggeridge. "Non-intrusive reduced order modelling of the Navier–Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 293 (2015), pp. 522–541.

[145] Dunhui Xiao, Fangxin Fang, Christopher Pain, and Guangwei Hu. "Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation". In: *International Journal for Numerical Methods in Fluids* 79.11 (2015), pp. 580–595.

[146] Dunhui Xiao, Pan Yang, Fangxin Fang, Jiansheng Xiang, Christopher C. Pain, Ionel M. Navon, and Min Chen. "A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting". In: *Journal of Computational Physics* 330 (2017), pp. 221–244.

[147] Jian Yu, Chao Yan, and Mengwu Guo. "Non-intrusive reduced-order modeling for fluid problems: A brief review". In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 233.16 (2019), pp. 5896–5912.

[148] Jian Yu, Chao Yan, Zhenhua Jiang, Wu Yuan, and Shusheng Chen. "Adaptive non-intrusive reduced order modeling for compressible flows". In: *Journal of Computational Physics* 397 (2019), p. 108855.

[149] Matteo Zancanaro, Markus Mrosek, Giovanni Stabile, Carsten Othmer, and Gianluigi Rozza. "Hybrid neural network reduced order modelling for turbulent flows with geometric parameters". In: *arXiv preprint arXiv:2107.09591* (2021).

[150] Olgierd C. Zienkiewicz, Robert L. Taylor, Perumal Nithiarasu, and Jizhong Z. Zhu. *The finite element method*. Vol. 3. McGraw-hill London, 1977.

[151] Ralf Zimmermann and Stefan Görtz. "Non-linear reduced order models for steady aerodynamics". In: *Procedia Computer Science* 1.1 (2010), pp. 165–174.

[152] Victor Zucatti, William Wolf, and Michel Bergmann. “Calibration of projection-based reduced-order models for unsteady compressible flows”. In: *Journal of Computational Physics* 433 (2021), p. 110196.

"But if you are stuck in Svalbard and want to make the sunset last a little longer, you can always try spinning counterclockwise.
It's true that it will add only an immeasurably small fraction of a nanosecond to the Earth's clock. But depending on who you're with...
it might be worth it."
R.M.

Acknowledgements

I am firstly grateful to Professor Traian Iliescu and Professor Angelo Iollo for accepting to revise this work, spending their time for no reason but kindness and dedication to the research world.

Alla fine di questo percorso ringrazio innanzitutto il professor Gianluigi Rozza per aver creduto nelle mie capacità e per avermi sospinto fino al termine, incoraggiando le mie inclinazioni. Egualmente sono grato al dottor Giovanni Stabile per avermi supportato in questa avventura trovando sempre per me uno spazio, anche nei momenti più bui. Mi piace pensare che in qualche modo le generazioni di ricercatori possano tramandarsi qualcosa e il sentirmi parte di questo succedersi ha reso la strada maggiormente significativa.

Voglio qui menzionare anche il professor Giovanni Noselli per aver mantenuto vivo in me l'interesse riguardo ambiti non prettamente legati al mio percorso ma a me innegabilmente affini. Grazie per la dedizione e per la voglia infaticabile nel rispondere a ogni mia più disparata curiosità.

Un posto speciale in questi ringraziamenti lo occupa la mia famiglia nel senso più lato possibile del termine. I miei genitori, Sara e Marjola sono stati i primi e maggiori sostenitori di questo traguardo raggiunto in un susseguirsi di alti e bassi, dove le difficoltà hanno reso il risultato più ricco di signifiato. Grazie a loro e con loro molte altre persone sono entrate a far parte di una famiglia più grande ma non meno stretta che ha coadiuvato in maniera insostituibile al raggiungimento di questa meta.

Compagni per me ineguagliabili sono stati i miei colleghi più prossimi tra i quali Federico e Saddam hanno ricoperto il ruolo di primi e più grandi fraterni: senza il vostro incipit non sarei a questo epilogo. Grazie a Monica e Luca che in tante occasioni hanno fatto della loro presenza un sostegno. Grazie in particolare a Maria, amica, collega e grande donna, che ha condiviso con me dalla prima all'ultima orma di questo sentiero: ancora tu.

Non posso non menzionare i volti che tutti i giorni hanno contribuito alla mia stabilità, strappando un sorriso anche nei momenti più scossi. Un grazie speciale a tutti i compagni d'ufficio passati ma soprattutto ai presenti, Sara, Ivan, Emanuele e Laura.

Sono grato ad Agnese per avermi dimostrato che le conoscenze non si misurano a giorni ma a sensazioni.

Tra le persone più care che Trieste mi abbia dato l'occasione di conoscere ringrazio di cuore Simone e Laura. Mi siete stati d'aiuto e d'appoggio, amici e anche più.

Il più folto tra i grazie lo spendo per gli amici. Grazie Silvia, Lisa e Nicolò che da tanto ormai mi accompagnate ma che ancora mi incoraggiate. Grazie Giacomo e Daniele, grazie Barbara ed Elena, grazie Claudia e Mirko, grazie Matteo, grazie ai compagni del Politecnico. Un forte grazie alla mia numerosa compagnia che mi ha accolto sempre al meglio come una parte valevole del gruppo. A tutti voi va il merito di avermi reso possibile l'essere la persona che sono, accettandomi pur nei difetti o forse per quelli.

Ancora grazie ai miei coinquilini tutti per avermi fatto sentire sempre a casa e al mio posto.

Voglio ringraziare tutti voi perchè, se sono giunto fin qui, è stato soprattutto merito delle innumerevoli pietre che avete posato a costruire il mio percorso.

"In certi momenti il vecchio e il nuovo,
il dolore e il piacere,
l'apprensione e la gioia
erano stranamente mescolati assieme."
H.H.

"Il grande paradosso di tutto questo
è che piú migliorano i nostri strumenti,
piú lontano riusciamo a vedere.
Ma non nel futuro, nel passato."
A.

"Se ti è nato il gusto di scoprire
non potrai che sentire il bisogno
di andare piú in là."
W.B.