Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

TREBALL FINAL DE MÀSTER

Development of a fractional step scheme for a compressible flow finite element solver

Treball realitzat per:
Samuel Parada Bustelo

Dirigit per:
Ramón Codina Rovira
Joan Baiges Aznar

Màster en:
Mètodes Numèrics en Enginyeria

Barcelona, 15/06/2018

Departament d'Enginyeria Civil i Ambiental

I would like to dedicate this master thesis to my beloved nephews. *Izan* and *Oliver*, the future is yours.

Acknowledgements

Una vez leí que no había mayor deber que aquel de ser agradecido. Me gustaría dedicar esta página a todas aquellas personas, que de una manera u otra, han contribuido a que haya terminado mi tesina de máster con éxito.

Quiero empezar agradeciendo a mis dos tutores, Ramón Codina y Joan Baiges. Sinceramente creo que no podría haber encontrado mejores directores de tesis. Agredecer a Ramón todo el tiempo dedicado. Ha sido una auténtica suerte poder tenerte como tutor. Gracias por tener siempre la puerta abierta y la paciencia necesaria para contestar mis preguntas y proponer soluciones. Joan, a ti darte las gracias por haber sido tan cercano desde el principio. Por todas las veces que me has ayudado con el código. De verdad que los momentos que has pasado conmigo frente a la pantalla para revisar *FEMUSS* tienen un tremendo valor para mi. A los dos, *gràcies per haver-me donat l'oportunitat de fer feina al CIMNE.*

También me gustaría incluír un apartado especial para mis amigos y compañeros. En primer lugar agradecer a *Javi*. Por la confianza, la paciencia y por ser mi compañero de aventuras ciclistas. El viaje de este año a Bélgica para participar en *De Ronde* no se me olvidará jamás. Una combinación perfecta de risas y de sufrimiento. Estoy seguro de que están muchos más por venir. Gracias por todas las llamadas nocturnas durante los últimos meses. Gracias a *Gonzalo*, por haber sido una referencia durante todos estos años, no solo como estudiante o compañero de piso, sino como ser humano. También gracias a *Carlos*, por haberme demostrado que aventura y pasión siempre pueden (y deben) ir de la mano. Igual que yo y Javi también respiras bici, y espero que en el futuro podamos compartir alguna que otra aventurilla todos juntos. Gracias a *Alex*, por las innumerables charlas que tenemos cada vez que vuelvo *a mi casa, a Galicia*. Creo que con pocas personas se puede hablar de tantas cosas como contigo. Agradecer también a *Ino*, por todo lo vivido en el máster. Todas las horas en la sala de estudio haciendo los trabajos, las noches de videojuegos en su casa, y más recientemente, las tardes que pasamos en la oficina C1-102 "debugueando" *FEMUSS* para adivinar por qué fallaban los tests, o las paradas para ir a buscar café a la segunda planta.

También a mis compañeros de piso, Arnau y Jordi, por hacerlo todo fácil desde el primer día.

Lo más importante, agradecer a mis padres que siempre me permitieron estudiar lo que quería. Gracias por todos los sacrificios que habéis hecho a lo largo de estos años. A mi hermana, *Fani*, que siempre fue el espejo perfecto en el que mirarme y que me ha regalado dos sobrinos fantásticos. Gracias por apoyarme incondicionalmente. A mis abuelos, sin los cuales no estaría aquí. Y finalmente, a *Claudia*, por todos los viajes a Barcelona para estar conmigo y por animarme siempre que era preciso.

Abstract

In this work we propose a second order stabilized pressure-correction scheme for the solution of the isentropic compressible Navier-Stokes equations.

Our algorithm consists in a compressible formulation with primitive variables without solving for the energy equation, which is mathematically uncoupled. This is due to the fact that the flow is considered to be isentropic (of constant entropy), which after properly relating density and pressure fields, becomes a system of equations in terms of velocity and pressure. Thus, the formulation can be seen as an extension of the incompressible case. As a consequence, the implementation is way less laborious when one departs from an already implemented segregated-incompressible flow solver.

Similarly to other compressible formulations, the prescription of boundary conditions will have to deal with the backscattering of acoustic waves. In this sense, we review a method already proposed in the literature for the monolithic solution of the problem, which is based on non-reflecting boundary conditions and we adapt it for the fractional step version by means of several extrapolations of the required order.

The approach we chose in this work is to present a pressure-correction technique at the pure algebraic level, departing from the matrix problem of the monolitich case. Finally, the statibilization is performed within the Variational MustiScale (VMS) framework and, in particular, we suggest a term-by-term orthogonal stabilization.

Key words: Stabilized finite element methods, term-by-term stabilization, isentropic flow, non-reflecting boundary conditions.

Contents

List of Figures

Chapter 1

Introduction

1.1 State of the art

The finite element method is a powerful computational technique very often employed to deal with the numerical simulation of flow problems. One of such problems, which has captured the attention of researchers for several decades, corresponds to the Navier-Stokes equations, especially in its incompressible form. This incompressibility feature, in practice, translates into a coupling of velocity and pressure, being its numerical solution demanding in the finite element computational context. In addition, the velocity and pressure interpolation functional spaces must satisfy a compatibility condition for standard discretizations, referred in the literature as *inf-sup* or LBB condition, being the latter honoring Ladyženskaja, Babuška and Brezzi, the matemathicians who proposed this constraint. Originally, in the framework of the finite element method, the LBB condition was tackled by defining special elements. Still, those finite elements satisfying the *inf-sup* condition are complicated and computationally expensive in practice (see for instace [10]). Another difficulty related to this problem is the instability associated to the advective term, whose nature is completely different from the previous one. Spurious node-to-node oscillations may show up for convection dominated flows even though they can be avoided for a specific mesh size, which commonly is not computationally affordable.

Those two kinds of instabilities are at the base of the development of works on the design of numerical methods stabilizing the convective term, in order to permit equal combinations of velocity-pressure interpolations. In these methods, the weak form of the problem obtained from the standard Galerkin approach is modified by adding some mesh-dependent terms weighted by the residuals (or even part of them) of the differential equations.

Initially, stabilized finite element methods were developed in the context of convective

flows and, as a first approach, numerical diffusion was introduced to counteract the associated numerical oscillations, but the proposed schemes were found to be extremely overdiffusive. Later, trying to reduce this overdiffusive character, the original techniques evolved to the introduction of such artificial diffusion only along the streamlines,[37]. Within this context, it was when the widely known SUPG -*Streamline Upwind Petrov-Galerkin*- method was introduced by Brooks and Hughes in [11]. In this method, the streamline diffusion idea was presented together with the concept of weighted residual. Next, the also popular GLS -*Galerkin/Least-squares*- method would be suggested by Hughes et al. in [38]. Other examples of similar stabilized methods are the Characteristic Galerkin method [27], or the Taylor-Galerkin method [28].

But it would be in 1995 (see [36]) when Hughes introduced a general technique for developing a new family of numerical methods capable of dealing with the socalled *multiscale phenomena*, later named the Variational Multi-Scale (VMS) method [34]. The key idea is to approximate the effect of the scales that cannot be resolved by the finite element mesh on the discrete finite element solution. This technique has helped the community to decipher the origins of stabilized finite element methods and provide a variational framework for subgrid scale models [35]. For a complete exposition of the VMS framework together with examples of application, the reader is encouraged to review [26], published by the advisors of this master thesis and co-workers.

While the nuances of stabilization captured the researchers attention for decades, there were others focused on the development of a family of methods with the objective of decoupling the velocity and pressure approximation. Such methods were originally labeled as *projection methods* (indistinctly called fractional step methods herein) as they were based on the splitting of the differential operator, by means of a Helmholtz decomposition. The pioneering works of Chorin and Temam [14, 42] in the late 1960's established the foundations and, since then, these methods have enjoyed a widespread popularity as they allow an important reduction of computational time. Furthermore, fractional step schemes present an inherent stability providing control over the pressure grandient, thus allowing in many situations the use of space interpolations which do not satisfy the compatibility condition.

A quite more recent mathematical problem is the one considering the complete set of the Navier-Stokes equations, that is to say, the coupled problem arising from mass, momentum and energy conservation equations. Usually, this problem is referred in the literature as the *compressible* Navier-Stokes problem. Its approximation via the finite element method together with VMS stabilization, has been recently proposed (see [4, 5]) in order to develop a compressible flow model to simulate the flow and the acoustic scales (aeroacoustic). This problem, among other features, is highly non-linear, it may need to deal with the formation

of shocks (discontinuities) and it is shown to be really demanding in terms of computational power, an important drawback. But for low Mach number flows, with neither shocks nor thermal coupling, both flow dynamics and wave propagation can be considered *isentropic*, that is to say, of *constant entropy*. In this master thesis, we focus our attention in this isentropic case. Under these assumptions, and as we shall see later, density and pressure fluctuations are directly proportional, and *a two-field velocity-pressure compressible formulation can be derived as an extension of the incompressible problem*. The energy conservation equation still remains mathematically uncoupled.

The prescription of boundary conditions for the fully compressible case is not so straightforward. As a matter of fact, Dirichlet boundary conditions are imposed depending on the compressibility of the medium, which can range from subsonic, transonic, supersonic and hypersonic flow, for different values of the Mach number. Although such laborious prescription of boundary conditions is avoided in the isentropic case, new challenges arise which need to be taken into account. In principle, an appropriate boundary condition for the pressure field must be enforced, since we are solving for flow and acoustic scales together. The main purpose of such condition is to allow the sound waves to smoothly leave the domain external boundaries [40]. In other words, we want to avoid those sound waves to be backscattered by the external boundaries into the computational domain, a fact that would pollute the solution of the problem. This distinguishing feature has already been studied in the past and as a result, there are several numerical techniques which deal with this backscattering problem. See for instace the review in [30, 31] and references therein. The treatment of the waves must be compatible with the flow velocity boundary conditions. This fact is of most importance on Dirichlet boundaries where the velocity is to be prescribed. This need has motivated the development of a method for a unified prescription of flow and non-reflecting boundary conditions applied to a monolithic scheme [41], which we will review later in this work.

1.2 Outline

This master thesis is organized in different chapters, in order to make it easier for the reader to follow the contents. The distribution of the work is,

- Chapter 1: *Introduction*: in what remains of the present chapter, we first include a short section on FEMUSS, which is the in-house parallel code developed at Prof. Ramón Codina´s research group. In addition, we include some basic concepts on functional spaces and finite element method.

- Chapter 2: *Isentropic equations*: this chapter aims at presenting the isentropic com-

pressible problem, as a modification of the incompressible Navier-Stokes equations.

- Chapter 3: *Stabilization within the VMS framework*: this chapter is entirely devoted to the Variational MultiScale framework. We will start by introducing this technique and its key idea, to end up by stating the stabilized isentropic compressible Navier-Stokes problem.
- Chapter 4: *Special treatment of boundary conditions*: as we shall see in this chapter, special attention has to be paid to the imposition of boundary conditions, as we aim at solving for the flow and acoustic scales together.
- Chapter 5: *Design of fractional step schemes*: this is the key chapter of this work, where a detailed description of the fractional step algorithm to solve the problem in hand is presented.
- Chapter 6: *Numerical results*. Here we present a benchmark in order to asses the correctness of the implementation. It consists in the aerodynamic sound radiated by flow past a cylinder, for $Re = 1000$ and $Mach = 0.0583$.
- Chapter 7: *Conclusions*. In this final chapter, we include the final conclusions of the work together with some open lines of research that could be eventually considered.

1.3 FEMUSS

FEMUSS is an acronym which stands for *Finite Element Method Using Subgrid Scales*. This is the name of the in-house multiphysics parallel code developed at this research group of CIMNE and which arises as a result of the numerous research projects carried out in the preceeding years. It is a large scale, parallel and object-oriented code written using the Fortran 2003 standard. The parallelization is completely done through MPI standards and, as its name suggests, the finite element equations are stabilized via the Variational MultiScale framework.

The code entails several modules, each of them fully dedicated to compute the solution of a certain problem. Still, the process of transfering information among them is possible, what makes actually feasible to solve coupled problems (via communication channels). The included modules in FEMUSS are, among others, incompressible Navier-Stokes equations (both monolithich and fractional step schemes), fully compressible Navier-Stokes problem (monolithic approach), optics and wave propagation problems, temperature and solids problems, etc. In addition, the code is flexible in the sense that it can handle different type of elements for a single mesh, allowing remeshing strategies and various implicit-explicit time integration schemes.

For the parallelization process, the computational domain is to be divided into subdomains, being each of these subdomains solved by a single processor containing all the required information. The domain partition is done through a identification process of the nodes in the mesh to be used. As a result, the nodes are labeled either as local or ghost. Local points are those beloging to a single subdomain, whereas the boundary nodes which belong to elements with at least a local point, are denoted as ghost points. See Figure 1.1 for details.

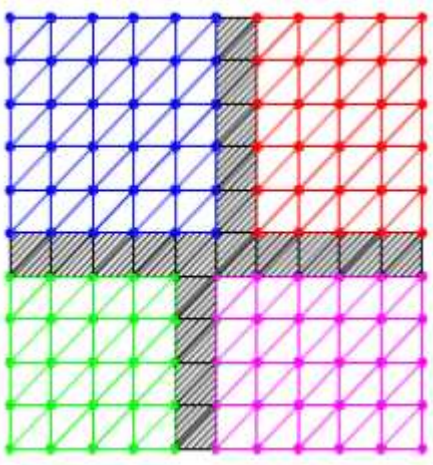


Figure 1.1: Mesh Partition. Nodes are assigned to a subdomain. Red, blue, green and pink nodes are local. Gray elements are duplicated in several subdomains containing the ghost nodes.

Moreover, the ghost points are solved as local in other subdomain, a fact that allows in practice to decouple the system into independent problems. This basically means that, at the pure computational level, the matrix assembly is a local process. The initial problem data as well as boundary conditions are broadcasted to all subdomains before any calculation process actually begins.

Apart from this, FEMUSS has its own defined *problem type*, what makes it compatible with the *GiD* pre-post processor developed here at CIMNE. Finally, the code is also coupled with other external libraries, being among then, *VTK* (allowing *Paraview* support), *Zoltan* or *PETSc*.

1.4 Some useful functional spaces

In this section we introduce some mathematical concepts and notation that is to be used hereafter in this work. Let $\Omega \in \mathbb{R}^d$, $d = 2,3$ be a bounded domain. All the functions defined over the domain Ω are real. We denote as $\mathscr{C}_0^\infty(\Omega)$ the set of infinitely differentiable real functions

with compact support on Ω. In addition, we denote as $L^p(\Omega)$, $1 \leq p \leq \infty$ the space of real functions defined on Ω with the p-*th* power absolutely integrable. Let $f : \Omega \to \mathbb{R}$, so we define

$$||f||_p \equiv ||f||_{L^p(\Omega)} = \left(\int_\Omega |f|^p \, d\Omega \right)^{\frac{1}{p}}, \; 1 \leq p < \infty \tag{1.1}$$

$$||f||_\infty \equiv ||f||_{L^\infty(\Omega)} = \underset{x \in \Omega}{ess\ sup} \, |f(x)|, \; p = \infty \tag{1.2}$$

and

$$L^p(\Omega) \coloneqq \left\{ f : \Omega \to \mathbb{R} \mid ||f||_{L^p(\Omega)} < \infty \right\} \tag{1.3}$$

where the definition (1.1) has to be understood with respect to the classical Lebesgue measure. The case of $p = 2$ is of special interest in finite element analysis. The space $L^2(\Omega)$ is a Hilbert space defined as,

$$L^2(\Omega) \coloneqq \left\{ f : \Omega \to \mathbb{R} \mid \left(\int_\Omega |f|^2 \, d\Omega \right)^{\frac{1}{2}} < \infty \right\} \tag{1.4}$$

that is to say, the space of functions which are square integrable, and which is endowed with the scalar product

$$(f, g)_\Omega \coloneqq \int_\Omega f(x) g(x) \, d\Omega \tag{1.5}$$

and its induced norm

$$||f||_{L^2(\Omega)} \coloneqq (f, f)^{\frac{1}{2}} \tag{1.6}$$

For bounded domains, as we consider in this work, it can de demonstrated that,

$$L^\infty(\Omega) \subset \ldots \subset L^2(\Omega) \subset L^1(\Omega) \tag{1.7}$$

The Sobolev space $W^{m,p}(\Omega)$ is the space of functions in $L^p(\Omega)$ whose weak derivatives, i.e. derivatives understood in the sense of a distribution, of order less than or equal to m belong to $L^p(\Omega)$, being m a non-negative integer and $1 \leq p \leq \infty$. By using multi-index notation, let now the d-tuple $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{N}^d$, and the non-negative integer $|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_d$. We also define $\mathscr{D}^\alpha f = \frac{\partial^{|\alpha|} f}{\partial^{\alpha_1} x_1 \ldots \partial^{\alpha_d} x_n}$. Then, we introduce the following notation $\forall \, m = 1, 2, \ldots,$ and $p \geq 1$,

$$||f||_{m,p} \equiv ||f||_{W^{m,p}(\Omega)} \coloneqq \sum_{|\alpha|=0}^{m} ||\mathscr{D}^\alpha f||_{L^p(\Omega)} \tag{1.8}$$

and thus, the Sobolev spaces are formally defined as,

$$W^{m,p}(\Omega) \coloneqq \left\{ f : \Omega \to \mathbb{R} \mid ||f||_{m,p} < \infty \right\} \tag{1.9}$$

In addition, we have that $W^{m,p}(\Omega) = H^m(\Omega)$ which is a Hilbert space endowed with a scalar product and its associated norm $||\cdot||_{H^m(\Omega)}$ (also denoted as $||\cdot||_m$). Therefore, $H^m(\Omega)$ consists of square integrable functions all of whose derivatives or order up to m are also square integrable. Again, the case $p = 2$ is of interest, and an important space in finite element analysis arises when considering also $m = 1$. Then, it yields,

$$W^{1,2}(\Omega) = H^1(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid \sum_{|\alpha|=0}^{1} ||\mathscr{D}^\alpha f||_{L^2(\Omega)} < \infty \right\} \tag{1.10}$$

which corresponds to the space of functions whose gradient is square integrable. This space is equipped with the inner product,

$$((f,g))_\Omega := (f,g)_\Omega + \sum_{i=1}^{d} (\partial_i f, \partial_i g) \tag{1.11}$$

where we have used the short notation for the partial derivative $\frac{\partial(\cdot)}{\partial x_i} \equiv \partial_i$, and also its induced norm,

$$||f|| := ((f,f))_\Omega^{1/2} \tag{1.12}$$

Remark: *Extension to vector-valued functions*. In the finite element analysis of flow problems, consideration needs to be given not only to scalar functions (such as pressure or temperature) but also to vector-valued functions (such as fluid velocity). For vector-valued functions with d components, that is $\boldsymbol{u} : \Omega \to \mathbb{R}^d$, the procedure is in fact the same as for scalar functions. Considering again a domain $\Omega \subset \mathbb{R}^d, d \geq 1$, we denote by $[H^m(\Omega)]^d$ or $\boldsymbol{H}^m(\Omega)$ (in bold character) the space of vector functions with d components, for which each component $u_i \in H^m(\Omega), 1 \leq i \leq d$. For the particular case of functions belonging to $\boldsymbol{L}^2(\Omega) = [H^0(\Omega)]^d$, the inner product is given by,

$$(\boldsymbol{u}, \boldsymbol{v}) := \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v} \, d\Omega \tag{1.13}$$

where there should be no ambiguity in using the same notation to represent the inner product of both scalar and vector functions.

Remark: In general, the integral of two functions g_1 and g_2 over a domain ω will be denoted as $\langle g_1, g_2 \rangle_\omega$. For the specific case of the L^2 product we use the notation $(\cdot,\cdot)_\Omega \equiv (\cdot,\cdot)$.

When dealing with time-dependent problems, as it will be the case through this work, it is interesting to also consider the set of spaces of the form $L^p(0,T;X)$, defined as,

$$L^p(0,T;X) := \left\{ f : (0,T) \to X \mid \int_0^T ||f||_X^p \, dt < \infty \right\} \tag{1.14}$$

that is to say, the spaces of functions such that their X norm in the spatial argument is an $L^p(0,T)$ function in time, i.e., its p-th power is integrable if $1 \leq p < \infty$ or bounded if $p = \infty$. In principle, we will be interested in the spaces $\boldsymbol{L}^2(0,T;[H^1(\Omega)]^d)$ and $L^2(0,T;H^1(\Omega))$ as we shall see later.

1.5 Finite Element Method

In the previous section we introduced the functional spaces to be used throughout this work, all at the continuous level. Now, we aim at introducing the approximation of the previous spaces by means of finite-dimensional subspaces (conforming approximation) that can be handled numerically. The reader is referred to, for instance [9], for a deep exposition on the finite element theory .

At this point, we have to view the domain of the problem Ω as discretized into element subdomains. Let $\mathscr{T}_h(\Omega)$ be a regular partition, also referred in the literature as *triangulation*, of Ω into n_{el} subdomains (elements), such that,

$$\Omega^e \neq \emptyset, \qquad \bar{\Omega} = \bigcup_{e=1}^{n_{el}} \bar{\Omega}^e \qquad \text{and} \qquad \Omega^e \cap \Omega^f = \emptyset \qquad \forall\, e \neq f.$$

Each subdomain Ω^e has a piecewise smooth boundary $\Gamma^e = \partial\Omega^e$, and we denote as h the characteristic mesh size, i.e. $h \leq diam(\Omega^e)$ for all elements. Formally, a finite element is defined as a triplet $\{\bar{\Omega}^e, \mathscr{P}_k(\Omega^e), \Sigma^e\}$ being $\bar{\Omega}^e$ a closed subdomain of Ω, $\mathscr{P}_k(\Omega^e)$ the finite dimensional interpolating space defined over Ω^e (commonly the set of polynomials in $x_1, \ldots, x_d$ of degree less than or equal to k) and Σ^e is the set of elemental degrees of freedom. The finite element spaces we will consider in the following are:

$$\mathscr{Q}_h = \{q_h \in \mathscr{C}^0(\Omega) \mid q_h|_{\Omega^e} \in \mathscr{P}_k(\Omega^e),\ \forall\, e \in \mathscr{T}_h\} \tag{1.15}$$

$$\mathscr{V}_h^d = \{\boldsymbol{v}_h \in [\mathscr{C}^0(\Omega)]^d \mid \boldsymbol{v}_h|_{\Omega^e} \in [\mathscr{P}_k(\Omega^e)]^d,\ \forall\, e \in \mathscr{T}_h\} \tag{1.16}$$

which are finite dimensional spaces approximating $H^1(\Omega)$, and $[H^1(\Omega)]^d$ respectively. From now on, finite element functions will be identified with a subscript $(\cdot)_h$. The space $\mathscr{Q}_h$ will be associated with the pressure (k being the degree of the approximation) and $\mathscr{V}_h^d$ with the velocity field, being d the number of space dimensions of the problem in hand. Both spaces are referred to the same partition and constructed by means of continuous functions.

1.6 Time discretization and extrapolation operators

In this section, let us introduce some notation that we will use with respect to time discretization. For the sake of conciseness we will restrict ourselves to the classical backward-difference (BDF) approximation. Let us consider a partition of the time interval $(0,T)$ into time steps of size δt, assumed to be constant, for simplicity. We also identify $t^n = n\delta t$, $\forall\, n = 0,1,2,\ldots$ as a time step for the problem. The approximation of a generic time dependent function $g(t)$ at a time t^n will be denoted as g^n. An approximation to the derivative of $g(t)$ of order $k = 1,2,\ldots$ is denoted as $\delta_k g^{n+1}/\delta t$ where the numerator is given by the following operator,

$$\delta_k g^{n+1} = \frac{1}{\gamma_k}\left(g^{n+1} - \sum_{i=0}^{k-1} \alpha_k^i g^{n-i}\right) \tag{1.17}$$

being γ_k and α_k^i numerical parameters depending on the order of the temporal approximation. In particular, for the first, second and third order schemes it is found that,

$$\delta_1 g^{n+1} = \delta g^{n+1} = g^{n+1} - g^n \tag{1.18a}$$

$$\delta_2 g^{n+1} = \frac{3}{2}\left(g^{n+1} - \frac{4}{3}g^n + \frac{1}{3}g^{n-1}\right) \tag{1.18b}$$

$$\delta_3 g^{n+1} = \frac{11}{6}\left(g^{n+1} - \frac{18}{11}g^n + \frac{9}{11}g^{n-1} - \frac{2}{11}g^{n-2}\right) \tag{1.18c}$$

The first order scheme, i.e. $k = 1$, is referred in the literature as the BDF1 scheme and it coincides with the so popular backward Euler method. BDF1 and the second order scheme, BDF2, are proven to be unconditionally stable, whereas high order schemes do not maintain this interesting property (limitation known as *second Dahlquist barrier*). Likewise, the BDF2 scheme is not self-starting, that is to say, we need to include a strategy to compute g^{n-1} for the first time step, e.g. using the backward Euler scheme previously mentioned.

In the design of fractional step schemes, we will be also making use of the extrapolation operators of order k, generally defined as $\hat{g}_k^{n+1} = g^{n+1} + \mathcal{O}(\delta t^k)$, which for $k = 1,2,3$ are given by,

$$\hat{g}_1^{n+1} = g^n \tag{1.19a}$$

$$\hat{g}_2^{n+1} = 2g^n - g^{n-1} \tag{1.19b}$$

$$\hat{g}_3^{n+1} = 3g^n - 3g^{n-1} + g^{n-2} \tag{1.19c}$$

In addition, the extrapolation of order $k = 0$ is defined as $\hat{g}_0^{n+1} = 0$.

Chapter 2

Isentropic compressible Navier–Stokes equations

2.1 Introduction

The numerical solution of the fully compressible Navier-Stokes equations, i.e. the coupled problem involving mass, momentum and energy equations, is known to be computationally demanding in the finite element context. Moreover, solvers for compressible flows in the low Mach regime have shown poor performance. These two facts have led to the necessity of developing more affordable numerical methods. In this chapter we will focus our attention on the behavior of an ideal gas which undergoes a reversible thermodynamical process. This approach, that sometimes could be considered naive, turns out to be quite realistic in engineering problems not involving heat transfer processes or shocks. This basic assumption allows a drastic simplification of the complete set of Navier-Stokes equations, since the energy equation is mathematically uncoupled from both mass and momentum conservation equations. Additionally, the primitive variables of the problem can be used, i.e. pressure and velocity, and a two-field compressible formulation can be derived as an extension of the incompressible case.

2.2 Basic theory of isentropic compressible flows

As introduced above, for low Mach number subsonic flows with neither heat transfer nor shocks, a quite realistic hypothesis is to assume that the gas undergoes a reversible thermodynamical process. We start by recalling some of the basic relations from the compressible flow theory. We refer to [39] for a deeper review on this topic. By definition for an isentropic flow, the *entropy remains constant*. Using this fact, it can be shown that pressure and density are

related through the following fundamental expression,

$$\frac{p}{\rho^{\gamma}} = C \qquad \text{(a constant)} \tag{2.1}$$

where γ is the socalled adiabatic coefficient (e.g. $\gamma = 1.4$ for air) or ratio of specific heats, since $\gamma := c_p/c_v$ being c_p and c_v the specific heat of the fluid at constant pressure and volume, respectively. Additionally, p denotes the total pressure and ρ is the total density. The word *total* is included so that to clarify that those variables include the possible perturbations due to the compressible nature of the medium.

Stagnation conditions are those that would exist if the flow at any points in a fluid stream was isentropically brought to rest. If the entire flow is essentially isentropic and if the velocity is essentially zero at some point in the flow, then the stagnation conditions will be those existing at the zero velocity point. All fluid properties at stagnation points are denoted with a subscript 0, $(\cdot)_0$. For two points, being one of them at the stagnation condition, it can be shown that density, pressure and temperature from both locations are related through the following expressions,

$$\frac{\rho}{\rho_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{1}{\gamma-1}} \tag{2.2}$$

$$\frac{p}{p_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{2.3}$$

$$\frac{T}{T_0} = \left(1 + \frac{\gamma-1}{2}M^2\right) \tag{2.4}$$

where ρ_0, p_0 and T_0 are the density, pressure and temperature at stagnation conditions, respectively. The symbol M refers to the Mach number, a non-dimensional relation defined as,

$$M := \frac{|\boldsymbol{u}|}{c_0} \tag{2.5}$$

being $|\boldsymbol{u}|$ the modulus of the flow velocity and c_0 the speed of the sound of an ideal gas, defined as $c_0 := \sqrt{\gamma R T_0/\mathscr{M}}$ where $R[J/K\cdot mol]$ is the ideal gas contant, i.e. $R = 8.314472\ \frac{J}{K\cdot mol}$ and $\mathscr{M}[kg/mol]$ is the molar mass of the gas, e.g. $\mathscr{M}$ =0.02897 kg/mol for air .

Now, we will derive two useful expressions relating pressure and density derivatives, what will allow later to decrease the global complexity of the finite element formulation of the isentropic problem. From Equations (2.2) and (2.3) one can easily get the following equality,

$$\frac{p_0}{p} = \left(\frac{\rho_0}{\rho}\right)^{\gamma}$$

which is in direct concordance with the fundamental expression in Equation (2.1). Then, taking time derivatives in both sides of previous relation and recalling the equation of state for an ideal gas, i.e., $p_0\mathscr{M} = \rho_0 R T_0$, an expression which relates pressure and density time derivatives can be found, yielding,

$$\partial_t p = \frac{p_0}{\rho_0}\gamma\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}\partial_t\rho = \frac{RT_0}{\mathscr{M}}\gamma\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}\partial_t\rho \tag{2.6}$$

where we have used the short notation $\partial_t(\cdot)$ for the time derivative, $\frac{\partial(\cdot)}{\partial t}$. If, instead of the time derivative, one takes the gradient, a similar expression arises,

$$\boldsymbol{\nabla} p = \frac{p_0}{\rho_0}\gamma\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}\boldsymbol{\nabla}\rho = \frac{RT_0}{\mathscr{M}}\gamma\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}\boldsymbol{\nabla}\rho \tag{2.7}$$

and both expressions can be simplified if we identify the speed of sound as,

$$c^2 = c_0^2\left(1+\frac{\gamma-1}{2}M^2\right)^{-1} \tag{2.8}$$

Hence, we have shown that density and pressure derivatives are directly proportional under the isentropic assumption. Now, Equations (2.6) and (2.7) can be rewritten as,

$$\partial_t p = c^2\partial_t\rho \tag{2.9}$$

$$\boldsymbol{\nabla} p = c^2\boldsymbol{\nabla}\rho \tag{2.10}$$

and this explicit connection between pressure and density variations allows one to greatly simplify the derivation of the final scheme.

2.3 The isentropic flow problem

The isentropic compressible equations are composed by the momentum and mass conservations equations, namely,

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \mu\Delta\boldsymbol{u} - \frac{\mu}{3}\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}) + \boldsymbol{\nabla} p = \boldsymbol{0} \quad \text{in } \Omega\times(0,T) \tag{2.11a}$$

$$\frac{\partial\rho}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho + \rho\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega\times(0,T) \tag{2.11b}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \ \text{ in } \Omega,\ t=0 \tag{2.11c}$$

$$p = p_0 \ \text{ in } \Omega,\ t=0 \tag{2.11d}$$

since, as we described above, the energy equation is mathematically uncoupled as we consider the constant entropy problem. In these equations, $\boldsymbol{u}(\boldsymbol{x},t)$ and μ stand for the flow velocity and fluid viscosity, respectively. As usual, bold characters refer to vector variables. The symbol $\boldsymbol{\Delta}$ refers to the laplacian operator. The problem still needs to be provided with boundary conditions, a topic that we will discuss in detail in Chapter 4.

By using Equation (2.9), the continuity equation can be reformulated in terms of the sound velocity and pressure. It yields,

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t}+\frac{1}{\rho c^2}(\boldsymbol{u}\cdot\boldsymbol{\nabla})p+\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \quad \text{in } \Omega\times(0,T) \tag{2.12}$$

Despite all the simplications introduced, the equations of the problem still depend on the sound velocity and density. Calculating these two fields as implicit functions of the pair $[\boldsymbol{u},p]$ would definitely increase the complexity of the final scheme with new non-linearities. In order to avoid this fact, Equations (2.2) and (2.8) are used for closing the problem so that the proposed formulation only depends on velocity and pressure fields, what in practice reduces the difficulty at the programming stage too. With respect to the incompressible Navier-Stokes equations, we note that mainly the mass conservation equation is modified, as now we have a temporal derivative and a convective term involving the pressure field. For the momentum equation, just one additional viscous term needs to be taken into account, since the incompressibility constraint no longer holds. Thus, the strong form of the isentropic compressible Navier-Stokes problem is,

$$\rho\frac{\partial \boldsymbol{u}}{\partial t}+\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}-\mu\Delta\boldsymbol{u}-\frac{\mu}{3}\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u})+\boldsymbol{\nabla}p=\boldsymbol{0} \quad \text{in } \Omega\times(0,T) \tag{2.13a}$$

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t}+\frac{1}{\rho c^2}\boldsymbol{u}\cdot\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \quad \text{in } \Omega\times(0,T) \tag{2.13b}$$

$$\boldsymbol{u}=\boldsymbol{u}_0 \ \text{ in } \Omega,\ t=0 \tag{2.13c}$$

$$p=p_0 \ \text{ in } \Omega,\ t=0 \tag{2.13d}$$

together with a suitable set of boundary conditions. This is the problem we will analyze throughout this work, which can be seen as a direct extension of the incompressible case.

2.3.1 Galerkin variational formulation

Let us now start the derivation of the method by obtaining the Galerkin variational or weak form of the problem in hand, Equations (2.13a)–(2.13d) . We start by taking a vector test

function $\boldsymbol{v} \in \mathscr{V}^d$ for the momentum equation and integrating it over the whole domain so that,

$$(\rho \boldsymbol{v}, \partial_t \boldsymbol{u}) + (\rho \boldsymbol{v}, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) - \mu(\boldsymbol{v}, \Delta \boldsymbol{u}) - \frac{\mu}{3}(\boldsymbol{v}, \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{u})) + (\boldsymbol{v}, \boldsymbol{\nabla} p) = 0 \tag{2.14}$$

As usual, the last three terms are integrated by parts in order to reduce the continuity requirements. Hence, one gets,

$$\begin{aligned} &(\rho \boldsymbol{v}, \partial_t \boldsymbol{u}) + (\rho \boldsymbol{v}, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) + \mu(\boldsymbol{\nabla} \boldsymbol{v}, \boldsymbol{\nabla} \boldsymbol{u}) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}, \boldsymbol{\nabla} \cdot \boldsymbol{u}) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}, p) \\ &= \left\langle \boldsymbol{v}, \boldsymbol{n} \left[-p\boldsymbol{I} + \mu \boldsymbol{\nabla} \boldsymbol{u} + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{u}) \boldsymbol{I} \right] \right\rangle_\Gamma = \langle \boldsymbol{v}, \boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) \rangle_\Gamma \end{aligned} \tag{2.15}$$

where we have identified the definition of the stress tensor, $\boldsymbol{\sigma}(\boldsymbol{u}, p)$.

Taking now a scalar test function $q \in \mathscr{Q}$ for the continuity equation, one can directly write,

$$\left(\frac{1}{\rho c^2} q, \partial_t p \right) + \left(\frac{1}{\rho c^2} q, \boldsymbol{u} \cdot \boldsymbol{\nabla} p \right) + (q, \boldsymbol{\nabla} \cdot \boldsymbol{u}) = 0 \tag{2.16}$$

and therefore the variational problem consists now in finding, $\boldsymbol{u}(t) \in \mathscr{X}^d \equiv \boldsymbol{L}^2(0, T; \mathscr{V}^d)$ and $p(t) \in \mathscr{Y} \equiv L^2(0, T; H^1(\Omega))$ such that

$$\begin{aligned} (\rho \boldsymbol{v}, \partial_t \boldsymbol{u}) + (\rho \boldsymbol{v}, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) + \mu(\boldsymbol{\nabla} \boldsymbol{v}, \boldsymbol{\nabla} \boldsymbol{u}) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}, \boldsymbol{\nabla} \cdot \boldsymbol{u}) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}, p) \\ = \langle \boldsymbol{v}, \boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) \rangle_\Gamma \end{aligned} \tag{2.17}$$

$$\left(\frac{1}{\rho c^2} q, \partial_t p \right) + \left(\frac{1}{\rho c^2} q, \boldsymbol{u} \cdot \boldsymbol{\nabla} p \right) + (q, \boldsymbol{\nabla} \cdot \boldsymbol{u}) = 0 \tag{2.18}$$

$\forall\, \boldsymbol{v} \in \mathscr{V}^d \equiv \boldsymbol{H}^1(\Omega)$ and $q \in \mathscr{Q} \equiv L^2(\Omega)$.

Remark: Strictly speaking, the test function for the continuity equation lives in the space $L^2(\Omega)$, since no spatial derivative for this function appears in the problem stated in Equation (2.18). As of now, the regularity requirements are stronger for the pressure p than for its associated test function q. As we shall see later in Chapter 3, once we introduce the stabilization technique for this problem, we will actually require $q \in H^1(\Omega)$, which is the same functional space as for the pressure, in space.

The problem above, can be reformulated in a more compact manner after introducing the following form,

$$\begin{aligned} a : (\mathscr{X}^d, \mathscr{Y}) \quad &\times \quad (\mathscr{V}^d, \mathscr{Q}) \longrightarrow \mathbb{R} \\ [\boldsymbol{u}, p] \quad &, \quad [\boldsymbol{v}, q] \quad \longmapsto a([\boldsymbol{u}, p], [\boldsymbol{v}, q]) \end{aligned} \tag{2.19}$$

where, $a([\boldsymbol{u},p],[\boldsymbol{v},q]) := (\rho\boldsymbol{v},\partial_t\boldsymbol{u}) + (\rho\boldsymbol{v},\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}) + \mu(\boldsymbol{\nabla}\boldsymbol{v},\boldsymbol{\nabla}\boldsymbol{u}) + \frac{\mu}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v},\boldsymbol{\nabla}\cdot\boldsymbol{u}) - (\boldsymbol{\nabla}\cdot\boldsymbol{v},p) + \left(\frac{1}{\rho c^2}q,\partial_t p\right) + \left(\frac{1}{\rho c^2}q,\boldsymbol{u}\cdot\boldsymbol{\nabla}p\right) + (q,\boldsymbol{\nabla}\cdot\boldsymbol{u})$.

Therefore the semidiscrete problem reads now as,

Find $\boldsymbol{u}(t)\in\boldsymbol{L}^2(0,T;\mathscr{V}^d)$ *and* $p(t)\in L^2(0,T;H^1(\Omega)$ *such that*,

$$a([\boldsymbol{u},p],[\boldsymbol{v},q]) = \langle\boldsymbol{v},\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_\Gamma \tag{2.20}$$

$\forall\ \boldsymbol{v}\in\mathscr{V}^d$ *and* $q\in\mathscr{Q}$.

Remark: Later we will provide a technique to impose the boundary conditions for the isentropic compressible problem. Then, the right hand side in previous problem will be modified so that to include the proper terms.

Chapter 3

Stabilization within the Variational MultiScale (VMS) framework

3.1 Overview

The numerical approximation of the Navier-Stokes equations can be easily obtained using the Galerkin approximation. Unfortunately, the standard Galerkin approach might not be the best approximation to compute the solution of the problem. As we already pointed out in the introductory chapter, this is mainly due to two reasons: first a compatibility is required between the pressure and velocity finite element spaces, and second the effect of the convective term when this one dominates the problem, which can lead to numerical instabilities. A similar case scenario needs to be considered for the isentropic Navier-Stokes problem, which is nothing but an extension of the incompressible case. In this chapter we present an stabilization technique based on the Orthogonal Subgrid Scale concept, widely analyzed in different situations [20, 22] . We aim at proposing an extension of this concept to the isentropic compressible case. We will start from the roots of the VMS framework, to end up stating the stabilized problem to be considered to the end of the work. We refer to [26] for a complete review of the VMS framework together with examples of application and being this chapter based on such publication.

3.2 Key idea of VMS: scale splitting

Let for this section the time variable to be continuous and let us also forget about the boundary conditions, e.g. taking $\boldsymbol{n}\cdot\sigma([\boldsymbol{u},p]) = \boldsymbol{t} = \boldsymbol{0}$ for the moment in (2.20). Additionally, let also $\mathcal{V}_h^d \subset \mathcal{V}^d \equiv [H^1(\Omega)]^d$ and $\mathcal{Q}_h \subset \mathcal{Q} \equiv H^1(\Omega)$ be the finite element spaces to approximate

the velocity and the pressure, respectively, *in space*. The starting and key idea of the VMS framework is to split the spaces $\mathcal{V}^d$ and $\mathcal{Q}$ at the continous level as,

$$\mathcal{V}^d = \mathcal{V}_h^d \oplus \hat{\mathcal{V}}^d \tag{3.1a}$$

$$\mathcal{Q} = \mathcal{Q}_h \oplus \hat{\mathcal{Q}} \tag{3.1b}$$

where $\hat{\mathcal{V}}^d$ and $\hat{\mathcal{Q}}$ are any complement to the finite element spaces in $\mathcal{V}^d$ and $\mathcal{Q}$, respectively.

The complementary spaces are commonly referred as *subgrid spaces*. The splitting in (3.1a) and (3.1b) induces the *two scale* decomposition of unknows and test functions. Thus, $\boldsymbol{u} = \boldsymbol{u}_h + \hat{\boldsymbol{u}}$, $p = p_h + \hat{p}$ and $\boldsymbol{v} = \boldsymbol{v}_h + \hat{\boldsymbol{v}}$, $q = q_h + \hat{q}$, with $\boldsymbol{u}_h, \boldsymbol{v}_h \in \mathcal{V}_h^d$, $p_h, q_h \in \mathcal{Q}_h$ and $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}} \in \hat{\mathcal{V}}^d$, $\hat{p}, \hat{q} \in \hat{\mathcal{Q}}$. Therefore we may write now the continuous problem as follows,

Find $\boldsymbol{u}_h \in \mathcal{V}_h^d$, $p_h \in \mathcal{Q}_h$,$\hat{\boldsymbol{u}} \in \hat{\mathcal{V}}^d$ *and* $\hat{p} \in \hat{\mathcal{Q}}$ *such that,*

$$a([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) + a([\hat{\boldsymbol{u}}, \hat{p}], [\boldsymbol{v}_h, q_h]) = \ell(\boldsymbol{v}_h, q_h) \quad \forall \quad [\boldsymbol{v}_h, q_h] \in \mathcal{V}_h^d \times \mathcal{Q}_h \tag{3.2a}$$

$$a([\boldsymbol{u}_h, p_h], [\hat{\boldsymbol{v}}, \hat{q}]) + a([\hat{\boldsymbol{u}}, \hat{p}], [\hat{\boldsymbol{v}}, \hat{q}]) = \ell(\hat{\boldsymbol{v}}, \hat{q}) \quad \forall \quad [\hat{\boldsymbol{v}}, \hat{q}] \in \hat{\mathcal{V}}^d \times \hat{\mathcal{Q}} \tag{3.2b}$$

The reader has to note that adding up both problems we recover the initial formulation stated in Equation (2.20) and that no approximation has been done yet. This is an splitting at the pure continous level. In fact, if one takes $\hat{\mathcal{V}}^d = \boldsymbol{0}$ and $\hat{\mathcal{Q}} = \emptyset$ (empty sets) then it would yield the standard Galerkin method. But our goal in this section is to find a better definition for such spaces when the standard method shows undesired numerical instabilities. Likewise, the approximation to the subgrid spaces $\hat{\mathcal{V}}^d$, $\hat{\mathcal{Q}}$ will be a consequence of the approximation to $\hat{\boldsymbol{u}}$ and $\hat{p}$. The way these variables are modeled defines the particular numerical approximation to the problem. In the literature, it is common to identify the finite element components of the solution as the *resolved scales*, whereas the subscales are the *unresolved scales*. Similarly, Equation (3.2a) is referred as *equation for the finite element scales*, and Equation (3.2b) is named *equation for the subgrid scales*.

3.3 Stabilization of the isentropic Navier-Stokes equations

In this section our objective is to propose a stabilized formulation which can circumvent the problem associated to the numerical instabilities of the standard Galerkin formulation. Generally speaking, we will proceed as follows: first we will isolate the terms involving the subscales in the finite element scales equation. Later, we will actually define the approximation chosen to compute the subgrid scales so that, when we introduce such definition in the previous isolated terms, we will be able to finally obtain a closed form for the stabilized problem.

Differently as it was originally proposed for the incompressible Navier-Stokes problem, we aim at providing a closed formulation considering both velocity and pressure subscales.

3.3.1 Finite element scales equation

We will work first with the equation for the finite element scales, i.e., Equation (3.2a). Let us write such equation in its complete form, that is to say, separating the momentun and mass conservation equations. Thus,

$$(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \partial_t \hat{\boldsymbol{u}}) + (\rho \boldsymbol{v}_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{\boldsymbol{u}}) + \mu(\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \boldsymbol{u}_h) + \mu(\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \hat{\boldsymbol{u}}) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, p_h) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \hat{p}) = 0 \tag{3.3a}$$

$$\left(\frac{1}{c^2 \rho} q_h, \partial_t p_h\right) + \left(\frac{1}{c^2 \rho} q_h, \partial_t \hat{p}\right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} p_h\right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{p}\right) + (q_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) + (q_h, \boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}) = 0 \tag{3.3b}$$

Now, let us integrate by parts those terms involving spatial derivatives of the subgrid scales, i.e., where there are varibles with the symbol ^. This trick is convenient as it allows to avoid modelling derivatives of the subscales, what would lead to a more involved formulation. Another initial approximation which is usually done is to assume that the subscales vanish on the element boundaries. This is in fact equivalent to considering that the subcales actually behave as *bubble functions* [26], and then $\hat{\boldsymbol{u}} = \boldsymbol{0},\ \hat{p} = 0$ on $\partial K,\ \forall K \in \mathscr{T}_h$. Hence, it yields,

$$(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \partial_t \hat{\boldsymbol{u}}) + (\rho \boldsymbol{v}_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}_h) - (\rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{v}_h, \hat{\boldsymbol{u}}) + \mu(\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \boldsymbol{u}_h) - (\Delta \boldsymbol{v}, \hat{\boldsymbol{u}}) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - \frac{\mu}{3}(\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h), \hat{\boldsymbol{u}}) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, p_h) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \hat{p}) = 0 \tag{3.4a}$$

$$\left(\frac{1}{c^2 \rho} q_h, \delta_t p_h\right) + \left(\frac{1}{c^2 \rho} q_h, \partial_t \hat{p}\right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} p_h\right) - \left(\frac{1}{c^2 \rho} \boldsymbol{u} \cdot \boldsymbol{\nabla} q_h, \hat{p}\right) + (q_h \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - (\boldsymbol{\nabla} q_h, \hat{\boldsymbol{u}}) = 0 \tag{3.4b}$$

where we have neglected the terms that could involve derivatives on the velocity or density when integrating by parts, since they might complicate the behavior of the formulation and sometimes even lead to other kind of instabilities.

Second order derivatives of finite element functions within element interiors will be ne-

glected too. Those terms are exactly zero for linear elements and for higher order interpolations disregarding them leads to a method which is still consistent (see Remarks in [17]). Therefore, the equations for the finite element scales turn out to be,

$$(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \partial_t \hat{\boldsymbol{u}}) + (\rho \boldsymbol{v}_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}_h) - (\rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{v}_h, \hat{\boldsymbol{u}}) + \mu (\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \boldsymbol{u}_h) + \frac{\mu}{3} (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, p_h) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \hat{p}) = 0 \tag{3.5a}$$

$$\left(\frac{1}{c^2 \rho} q_h, \partial_t p_h\right) + \left(\frac{1}{c^2 \rho} q_h, \partial_t \hat{p}\right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u} \cdot \boldsymbol{\nabla} p_h\right) - \left(\frac{1}{c^2 \rho} \boldsymbol{u} \cdot \boldsymbol{\nabla} q_h, \hat{p}\right) + (q_h \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - (\boldsymbol{\nabla} q_h, \hat{\boldsymbol{u}}) = 0 \tag{3.5b}$$

In the next subsection, we will work on the subgrid scales equation in order to provide a way to numerically compute $\hat{\boldsymbol{u}}$ and $\hat{p}$.

3.3.2 Subgrid scales equation

Similarly as the finite element equations described above can be understood as the projection of the original equations onto the finite element spaces (i.e. by multiplying the equation by a test function in the proper space and integrating over the whole domain), the equations for the subscales are obtained by projecting the original equations onto the corresponding spaces $\hat{\mathscr{V}}^d$ and $\hat{\mathscr{Q}}$. Let us denote by $\hat{\Pi}$ the projection operator onto any of these subgrid spaces. Hence, the subscales equations can also be written from Equations (2.13a)–(2.13b) as,

$$\hat{\Pi}\left[\rho \partial_t \hat{\boldsymbol{u}} + \rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{\boldsymbol{u}} - \mu \boldsymbol{\Delta} \hat{\boldsymbol{u}} - \frac{\mu}{3} \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}) + \boldsymbol{\nabla} \hat{p}\right] = \hat{\Pi}\left[\boldsymbol{r}_m(p_h, \boldsymbol{u}_h)\right] \tag{3.6a}$$

$$\hat{\Pi}\left[\frac{1}{c^2 \rho} \partial_t \hat{p} + \frac{1}{c^2 \rho} \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{p} + \nabla \cdot \hat{\boldsymbol{u}}\right] = \hat{\Pi}\left[r_c(p_h, \boldsymbol{u}_h)\right] \tag{3.6b}$$

where we have defined the finite element residuals of the momentum and mass conservation equations respectively as,

$$\boldsymbol{r}_m(p_h, \boldsymbol{u}_h) = -\rho \partial_t \boldsymbol{u}_h - \rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}_h + \mu \boldsymbol{\Delta} \boldsymbol{u}_h + \frac{\mu}{3} \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - \boldsymbol{\nabla} p_h \tag{3.7}$$

$$r_c(p_h, \boldsymbol{u}_h) = -\frac{1}{c^2 \rho} \partial_t p_h - \frac{1}{c^2 \rho} \boldsymbol{u} \cdot \boldsymbol{\nabla} p_h - \nabla \cdot \boldsymbol{u}_h \tag{3.8}$$

Note that the combination of these expressions is exactly equivalent to (3.2b). Let us now rewrite the previous Equations (3.6a)–(3.6b) upon the definition of the following spatial operators, $\mathscr{L}_m(\hat{\boldsymbol{u}}, \hat{p}) := \rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{\boldsymbol{u}} - \mu \boldsymbol{\Delta} \hat{\boldsymbol{u}} - \frac{\mu}{3} \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}) + \boldsymbol{\nabla} \hat{p}$ and $\mathscr{L}_c(\hat{\boldsymbol{u}}, \hat{p}) := \frac{1}{c^2 \rho} \boldsymbol{u} \cdot \boldsymbol{\nabla} \hat{p} + \nabla \cdot \hat{\boldsymbol{u}}$. It

yields,

$$\hat{\Pi}\left[\rho\,\partial_t\hat{\boldsymbol{u}}+\mathscr{L}_m(\hat{\boldsymbol{u}},\hat{p})\right]=\hat{\Pi}\left[\boldsymbol{r}_m(p_h,\boldsymbol{u}_h)\right] \tag{3.9a}$$

$$\hat{\Pi}\left[\frac{1}{c^2\rho}\partial_t\hat{p}+\mathscr{L}_c(\hat{\boldsymbol{u}},\hat{p})\right]=\hat{\Pi}\left[r_c(p_h,\boldsymbol{u}_h)\right] \tag{3.9b}$$

Up to this point, the only approximation introduced for the subscales was to assume that they vanish on element boundaries. However, this approximation is not sufficient to obtain a numerical scheme since the spaces of subscales, i.e. $\hat{\mathscr{V}}^d$ and $\hat{\mathscr{Q}}$, are still *infinite dimensional*. The additional assumption we make is based on replacing the spatial operator by an algebraic operator which can be easily computed and inverted. The proposed approximation is to introduce a diagonal matrix $\boldsymbol{\tau}=diag(\tau_1,\tau_2)$, usually referred as *matrix of stabilization parameters*, which is defined over each element, so that the projection equations (3.9a)–(3.9b) could become,

$$\rho\,\partial_t\hat{\boldsymbol{u}}+\tau_{1,K}^{-1}\hat{\boldsymbol{u}}=\hat{\Pi}\left[\boldsymbol{r}_m(p_h,\boldsymbol{u}_h)\right] \tag{3.10a}$$

$$\frac{1}{c^2\rho}\partial_t\hat{p}+\tau_{2,K}^{-1}\hat{p}=\hat{\Pi}\left[r_c(p_h,\boldsymbol{u}_h)\right] \tag{3.10b}$$

Remark: Several clarifying comments are in order when performing this step,

- Note that we have dropped the projection operator from the left hand side of the equations. We assume that,

$$\hat{\Pi}\left[\rho\,\partial_t\hat{\boldsymbol{u}}+\tau_{1,K}^{-1}\hat{\boldsymbol{u}}\right]\simeq\rho\,\partial_t\hat{\boldsymbol{u}}+\tau_{1,K}^{-1}\hat{\boldsymbol{u}}$$

$$\hat{\Pi}\left[\frac{1}{c^2\rho}\partial_t\hat{p}+\tau_{2,K}^{-1}\hat{p}\right]\simeq\frac{1}{c^2\rho}\partial_t\hat{p}+\tau_{2,K}^{-1}\hat{p}$$

  This might not be necessarily true since we cannot guarantee that $\rho\,\partial_t\hat{\boldsymbol{u}}\in\hat{\mathscr{V}}^d$, $\tau_1^{-1}\hat{\boldsymbol{u}}\in\hat{\mathscr{V}}^d$ and $\frac{1}{c^2\rho}\partial_t\hat{p}\in\hat{\mathscr{Q}}$, $\tau_2^{-1}\hat{p}\in\hat{\mathscr{Q}}$. This is due to the fact that, in principle, both c and ρ *are not constants of the problem*. In fact, they were unknowns, initially. But it is important to note that the formulation proposed in Section 2.1 allows to drop the new non-linearities that both variables would bring to the problem. Therefore, if one assumed that c and ρ *behave as constants*, then the approximation introduced is valid.

- We have assumed that $\tau_{1,K}\hat{\boldsymbol{u}}\approx\mathscr{L}_m(\hat{\boldsymbol{u}},\hat{p})$ and $\tau_{2,K}\hat{p}\approx\mathscr{L}_c(\hat{\boldsymbol{u}},\hat{p})$. It is important to remark that this approximation has to be understood in such a way that there exist values of $\tau_{1,K}$ and $\tau_{2,K}$ for which those terms have (approximately) the same L^2-norm over an element $K\subset\mathscr{T}_h$ [17].

Another crucial point is how to treat the time-dependency of the subscales. A simple approach is to neglect the time derivatives of the subscales. In this situation, the subcales are named *quasi-static* in constrast to *dynamic* subscales, when they are considered to be time-dependent. We choose for simplicity the quasi-static approach, and therefore Equations (3.10a) – (3.10b) yield,

$$\hat{\boldsymbol{u}} = \tau_{1,K}\hat{\Pi}\left[\boldsymbol{r}_m(p_h, \boldsymbol{u}_h)\right] \tag{3.11a}$$

$$\hat{p} = \tau_{2,K}\hat{\Pi}\left[r_c(p_h, \boldsymbol{u}_h)\right] \tag{3.11b}$$

which finally provide a close expression for the computation of the subscales.

Remark: The approach of neglecting the time derivative of the subscales is not strictly consistent because it induces a stady-state solution which is time-step dependent. Such behaviour can be unstable (in time) for anisotropic space-time discretization [3, 8]. Generally speaking, quasi-static stabilized VMS methods may become unstable when the time step size is reduced without remeshing. However, for the sake of simplicity, we consider the quasi-static approach.

Within the VMS framework, different type of methods arise when we provide distinct definitions of the projection operator $\hat{\Pi}$. When one considers the space of subscales as that of the residuals, that is to say, to choose $\hat{\Pi} = I$ (the identity operator) when applied to the finite element residuals, the method which yields is termed Algebraic SubGrid-Scale (ASGS). Although this method is the simplest one, it is not really suitable for designing fractional step schemes, which is the real objective of this work. The ASGS approach combined with a segregation technique would involve to compute the inverse of matrices with a wide stencial, which is usually computationally unaffordable. Moreover, this technique would lead to a non-symmetric formulation [2].

When the spaces of subscales $\hat{\mathcal{V}}^d$ and $\hat{\mathcal{Q}}$, that is, the spaces where $\hat{\boldsymbol{u}}$ and $\hat{p}$ belong for t fixed, are enforced to be L^2-orthogonal to the finite element spaces $\mathcal{V}_h^d$ and $\mathcal{Q}_h$, the method is termed Orthogonal SubGrid-Scale (OSGS or simply OSS). It corresponds to taking $\hat{\Pi} = \Pi_h^{\perp} = I - \Pi_h$ where Π_h is the projection operator onto the finite element spaces. This definition makes the subscales active in regions which cannot be resolved by the finite element mesh. Generally speaking, compared to the ASGS method, the OSS approach is more computationally demanding and it is more vulnerable, in the sense that it needs to be properly tunned. On the contrary it provides a better accuracy and matrices with a narrower stencil. The OSS method is *consistent*, in the way that it is commonly understood in finite element theory, that is to say, the stabilization terms which modify the variational form of the problem tend to vanish as a finner discretization is considered.

The stabilization parameters $\tau_{1,K}$ and $\tau_{2,K}$ defined over each element $K \in \mathcal{T}_h$ contribute to

provide the stabilization for the weak forms of the momentum and mass conservation equations. Their definition is based on a Fourier analysis (see [17]), which is not part of the scope of this master thesis. They are computed as:

$$\tau_{1,K} = \left[c_1 \frac{\mu}{h^2} + c_2 \rho \frac{|\boldsymbol{u}_h|_K}{h} \right]^{-1} \tag{3.12a}$$

$$\tau_{2,K} = \frac{h^2}{c_1 \tau_{1,K}} \tag{3.12b}$$

where $|\boldsymbol{u}_h|_K$ is the mean Euclidean norm of the velocity in each element $K \in \mathscr{T}_h$. The algorithmic constants c_1 and c_2 depend on the polynomial order of the interpolation. For linear elements, it is set $c_1 = 4$ and $c_2 = 2$. At the programming stage, the reader should note that the values of the stabilization parameter would be needed at each integration point and, as a consequence, the values of the density and velocity too.

3.3.3 Term-by-term stabilized formulation

For the moment, we will restrict ourselves to the OSS approach, i.e. we take $\hat{\Pi} = \Pi_h^{\perp}$ in (3.11a) –(3.11b). Hence, the subscales are the result of computing,

$$\hat{\boldsymbol{u}} = \tau_{1,K} \Pi_h^{\perp} [\boldsymbol{r}_m(p_h, \boldsymbol{u}_h)] \tag{3.13a}$$

$$\hat{p} = \tau_{2,K} \Pi_h^{\perp} [r_c(p_h, \boldsymbol{u}_h)] \tag{3.13b}$$

and when those expressions are inserted into Equations (3.5a)–(3.5b), the following formulation arises,

$$\begin{aligned} &(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \boldsymbol{u} \cdot \nabla \boldsymbol{u}_h) + \mu (\nabla \boldsymbol{v}_h, \nabla \boldsymbol{u}_h) + \frac{\mu}{3} (\nabla \cdot \boldsymbol{v}_h, \nabla \cdot \boldsymbol{u}_h) - (\nabla \cdot \boldsymbol{v}_h, p_h) \\ &- \sum_{K \in \mathscr{T}_h} \left\langle \rho \boldsymbol{u} \cdot \nabla \boldsymbol{v}_h, \tau_{1,K} \Pi_h^{\perp} [\boldsymbol{r}_m(p_h, \boldsymbol{u}_h)] \right\rangle_K - \sum_{K \in \mathscr{T}_h} \left\langle \nabla \cdot \boldsymbol{v}_h, \tau_{2,K} \Pi_h^{\perp} [r_c(p_h, \boldsymbol{u}_h)] \right\rangle_K = 0 \end{aligned} \tag{3.14a}$$

$$\begin{aligned} &\left(\frac{1}{c^2 \rho} q_h, \partial_t p_h \right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u} \cdot \nabla p_h \right) + (q_h \nabla \cdot \boldsymbol{u}_h) \\ &- \sum_{K \in \mathscr{T}_h} \left\langle \frac{1}{c^2 \rho} \boldsymbol{u} \cdot \nabla q_h, \tau_{2,K} \Pi_h^{\perp} [r_c(p_h, \boldsymbol{u}_h)] \right\rangle_K - \sum_{K \in \mathscr{T}_h} \left\langle \nabla q_h, \tau_{1,K} \Pi_h^{\perp} [\boldsymbol{r}_m(p_h, \boldsymbol{u}_h)] \right\rangle_K = 0 \end{aligned} \tag{3.14b}$$

where we have introduced the notation $\langle \cdot, \cdot \rangle_K$ for the product over an element domain $K \in \mathscr{T}_h$.

Remark: For the sake of simplicity, when computing the orthogonal projection of the

residuals in both equations, we will assume that the orthogonal projection of the temporal terms vanishes. That is to say,

$$\Pi_h^{\perp}\left[-\rho\,\partial_t \boldsymbol{u}_h\right] = -\rho\,\partial_t \boldsymbol{u}_h - \Pi_h\left[-\rho\,\partial_t \boldsymbol{u}_h\right] = -\rho\,\partial_t \boldsymbol{u}_h + (\boldsymbol{v}_h, \rho\,\partial_t \boldsymbol{u}_h) \simeq 0 \tag{3.15a}$$

$$\Pi_h^{\perp}\left[-\frac{1}{c^2\rho}\partial_t p_h\right] = -\frac{1}{c^2\rho}\partial_t p_h - \Pi_h\left[-\frac{1}{c^2\rho}\partial_t p_h\right] = -\frac{1}{c^2\rho}\partial_t p_h + \left(q_h, \frac{1}{c^2\rho}\partial_t p_h\right) \simeq 0 \tag{3.15b}$$

in other words, we will suppose that both terms already belong to the finite element spaces $\mathscr{V}_h^d$ and $\mathscr{Q}_h$, and hence, their Π_h projections are precisely the functions themselves. This assumption gives place to a weakly consistent method.

Taking into account this information, expanding the finite element residuals and neglecting the terms involving second derivatives (for reasons already discussed), the final stabilized equations using the OSS approach yield,

$$\begin{aligned}
&(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h) + \mu(\boldsymbol{\nabla}\boldsymbol{v}_h, \boldsymbol{\nabla}\boldsymbol{u}_h) + \frac{\mu}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_h, \boldsymbol{\nabla}\cdot\boldsymbol{u}_h) - (\boldsymbol{\nabla}\cdot\boldsymbol{v}_h, p_h) \\
&+ \sum_{K\in\mathscr{T}_h}\left\langle \rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{v}_h, \tau_{1,K}\Pi_h^{\perp}\left[\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h + \boldsymbol{\nabla}p_h\right]\right\rangle_K + \sum_{K\in\mathscr{T}_h}\left\langle \boldsymbol{\nabla}\cdot\boldsymbol{v}_h, \tau_{2,K}\Pi_h^{\perp}\left[\frac{1}{c^2\rho}\boldsymbol{u}\cdot\boldsymbol{\nabla}p_h + \boldsymbol{\nabla}\cdot\boldsymbol{u}_h\right]\right\rangle_K = 0
\end{aligned} \tag{3.16a}$$

$$\begin{aligned}
&\left(\frac{1}{c^2\rho}q_h, \partial_t p_h\right) + \left(\frac{1}{c^2\rho}q_h, \boldsymbol{u}\cdot\boldsymbol{\nabla}p_h\right) + (q_h, \boldsymbol{\nabla}\cdot\boldsymbol{u}_h) \\
&+ \sum_{K\in\mathscr{T}_h}\left\langle \frac{1}{c^2\rho}\boldsymbol{u}\cdot\boldsymbol{\nabla}q_h, \tau_{2,K}\Pi_h^{\perp}\left[\frac{1}{c^2\rho}\boldsymbol{u}\cdot\boldsymbol{\nabla}p_h + \boldsymbol{\nabla}\cdot\boldsymbol{u}_h\right]\right\rangle_K + \sum_{K\in\mathscr{T}_h}\left\langle \boldsymbol{\nabla}q_h, \tau_{1,K}\Pi_h^{\perp}\left[\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h + \boldsymbol{\nabla}p_h\right]\right\rangle_K = 0
\end{aligned} \tag{3.16b}$$

Once arrived to this problem, we can make some further modifications provided the completeness of the method is maintained, that it to say, these modifications do not alter the fact that the exact solution is still a solution of the discrete problem. As the reader might have noticed, the advection velocity in the equations is still $\boldsymbol{u}$. The usual approach is to replace such advection velocity directly by the finite element one, i.e., $\boldsymbol{u}_h$, what physically means that the influence of the subscales in the transport of momentum is neglected [24].

Additionally, there is a slight modification of the previous scheme which consists in neglecting crossed terms in the stabilization integrals, since those terms do not enhance the stability of the formulation. The resultant method is usually referred as *split* OSS method or *term-by-term* stabilization (see for instace [13]). In fact, this possibility, when applied to the incompressible Navier-Stokes problem, was analyzed and turned out to have even (slightly) improved stability with respect to the original OSS scheme [7, 21]. This *split* OSS approach is

not completely residual-based, and hence, it is not consistent, being consistency understood in the finite element context. The key idea behind this technique is the space where the subscales are defined, which we recall that is the orthogonal complement to the finite element space. This fact is what in practice ensures that the term-by-term formulation below has an optimal consistency error [18]. Finally, we arrive at the following *split* OSS semi-discrete problem,

$$(\rho \boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + (\rho \boldsymbol{v}_h, \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{u}_h) + \mu(\boldsymbol{\nabla} \boldsymbol{v}_h, \boldsymbol{\nabla} \boldsymbol{u}_h) + \frac{\mu}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) - (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, p_h)$$
$$+ \sum_{K \in \mathscr{T}_h} \left\langle \rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{v}_h, \tau_{1,K} \Pi_h^{\perp} [\rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{u}_h] \right\rangle_K + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \tau_{2,K} \Pi_h^{\perp} [\boldsymbol{\nabla} \cdot \boldsymbol{u}_h] \right\rangle_K = 0 \tag{3.17a}$$

$$\left(\frac{1}{c^2 \rho} q_h, \partial_t p_h \right) + \left(\frac{1}{c^2 \rho} q_h, \boldsymbol{u}_h \cdot \boldsymbol{\nabla} p_h \right) + (q_h, \boldsymbol{\nabla} \cdot \boldsymbol{u}_h) + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} q_h, \tau_{1,K} \Pi_h^{\perp} [\boldsymbol{\nabla} p_h] \right\rangle_K$$
$$+ \sum_{K \in \mathscr{T}_h} \left\langle \frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} q_h, \tau_{2,K} \Pi_h^{\perp} \left[\frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} p_h \right] \right\rangle_K = 0 \tag{3.17b}$$

To conclude this section, let us be more formal and introduce the following stabilization forms for the left and right hand side of the problem in hand,

$$\begin{aligned} a_s : (\mathscr{V}_h^d, \mathscr{Q}_h) \quad &\times \quad (\mathscr{V}_h^d, \mathscr{Q}_h) \longrightarrow \mathbb{R} \\ [\boldsymbol{u}_h, p_h] \quad &, \quad [\boldsymbol{v}_h, q_h] \longmapsto a_s([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) := \sum_{K \in \mathscr{T}_h} \left\langle \rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{v}_h, \tau_{1,K} \rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{u}_h \right\rangle_K \\ &\qquad + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \tau_{2,K} \boldsymbol{\nabla} \cdot \boldsymbol{u}_h \right\rangle_K + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} q_h, \tau_{1,K} \boldsymbol{\nabla} p_h \right\rangle_K \\ &\qquad + \sum_{K \in \mathscr{T}_h} \left\langle \frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} q_h, \tau_{2,K} \frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} p_h \right\rangle_K \end{aligned} \tag{3.18}$$

$$\begin{aligned} \ell_s : \mathscr{V}_h^d \quad &\times \quad \mathscr{Q}_h \longrightarrow \mathbb{R} \\ \boldsymbol{v}_h \quad &, \quad q_h \longmapsto \ell_s([\boldsymbol{v}_h, q_h]) := \sum_{K \in \mathscr{T}_h} \left\langle \rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{v}_h, \tau_{1,K} \Pi_h [\rho \boldsymbol{u}_h \cdot \boldsymbol{\nabla} \boldsymbol{u}_h] \right\rangle \\ &\qquad + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \tau_{2,K} \Pi_h [\boldsymbol{\nabla} \cdot \boldsymbol{u}_h] \right\rangle_K + \sum_{K \in \mathscr{T}_h} \left\langle \boldsymbol{\nabla} q_h, \tau_{1,K} \Pi_h [\boldsymbol{\nabla} p_h] \right\rangle_K \\ &\qquad + \sum_{K \in \mathscr{T}_h} \left\langle \frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} q_h, \tau_{2,K} \Pi_h \left[\frac{1}{c^2 \rho} \boldsymbol{u}_h \cdot \boldsymbol{\nabla} p_h \right] \right\rangle_K \end{aligned} \tag{3.19}$$

so that the stabilized problem to be solved, including the boundary condition term yields,

Find $\boldsymbol{u}_h \in \boldsymbol{L}^2(0,T;\mathscr{V}_h^d)$ *and* $p_h \in L^2(0,T;\mathscr{Q}_h)$ *such that*,

$$a([\boldsymbol{u}_h,p_h],[\boldsymbol{v}_h,q_h]) + a_s([\boldsymbol{u}_h,p_h],[\boldsymbol{v}_h,q_h]) = \ell_s([\boldsymbol{v}_h,q_h]) + \langle \boldsymbol{v}_h, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u}_h,p_h)\rangle_\Gamma \tag{3.20}$$

$\forall\ \boldsymbol{v}_h \in \mathscr{V}_h^d \subset \mathscr{V}^d \equiv \boldsymbol{H}^1(\Omega)$ *and* $q_h \in \mathscr{Q}_h \subset \mathscr{Q} \equiv H^1(\Omega)$.

Remark: we have moved the terms involving the projection onto the finite element spaces to the right hand side of the problem in (3.20). This is because we will treat them explicitly in a fixed point manner, i.e., we will compute them based on the values of velocity and pressure from previous time steps within the non-linearity loop. This allows even a further reduction of the general complexity of the method.

Chapter 4

Imposition of boundary conditions for the isentropic Navier–Stokes equations

4.1 Introduction

Up to now, we have proposed the isentropic compressible formulation as an extesion of the incompressible Navier stokes case. This allows one to forget about solving the fully compressible problem. Although the elaborated presciption of boundary conditions of the complete compressible case is avoided with the present formulation (recall that Dirichlet boundary conditions need to be imposed according to flow regime), there are still new challenges to take into account as we want to solve for the flow and acoustic scales at once. As we will see, a special type of condition must be imposed for the pressure field. The main purpose of such condition is to allow the sound waves to smoothly leave the domain external boundaries. If this did not happen, the solution would be completely polluted by the backscattering of such waves.

In this chapter we review the method proposed in [41] for the monolithic solution of the problem. This technique allows one to prescribe flow and non-reflecting boundary conditions in an unified and compatible way.

4.2 Unknown splitting: mean and acoustic components

The starting idea of the method is the splitting of the two unknown fields of the problem, i.e. velocity and pressure, into mean and oscillatory components, in a similar fashion as it is done

in some turbulence models. Thus, we have,

$$\boldsymbol{u}(\boldsymbol{x},t) = \bar{\boldsymbol{u}}(\boldsymbol{x},t) + \boldsymbol{u}'(\boldsymbol{x},t) \tag{4.1a}$$

$$p(\boldsymbol{x},t) = \bar{p}(\boldsymbol{x},t) + p'(\boldsymbol{x},t) \tag{4.1b}$$

where the mean variables are mathematically described as,

$$\bar{\boldsymbol{u}}(\boldsymbol{x},t) = \frac{1}{T_w}\int_{t-T_w}^{t} \boldsymbol{u}(\boldsymbol{x},s)\, ds \tag{4.2a}$$

$$\bar{p}(\boldsymbol{x},t) = \frac{1}{T_w}\int_{t-T_w}^{t} \bar{p}(\boldsymbol{x},s)\, ds \tag{4.2b}$$

being T_w and appropiate time window. In the following, we will identify the oscillatory components with the acoustic fluctuations and the mean variables with the flow variables. Likewise, these mean flow quantities are allowed to evolve during the problem calculation and they do not need to be homogeneous along the boundary. On the contrary, high frequency variations of these variables are not allowed as they would interfere with $\boldsymbol{u}'(\boldsymbol{x},t)$ and $p'(\boldsymbol{x},t)$.

4.3 Description of the applied boundary conditions

4.3.1 Boundary decomposition

Let us introduce a boundary splitting at the continuous level, in order to treat the flow and acoustic boundary conditions in a suitable manner. For a generic domain Ω, we start by dividing its boundary $\Gamma \equiv \partial\Omega$ into three different disjoint subsets namely, Γ_S, Γ_L and Γ_O. These subsets are such that $\Gamma_S \cap \Gamma_L = \emptyset$, $\Gamma_L \cap \Gamma_O = \emptyset$, $\Gamma_O \cap \Gamma_S = \emptyset$ and $\Gamma_S \cup \Gamma_L \cup \Gamma_O = \Gamma$. The boundary Γ_S refers to the solid boundary, where usually the velocity is prescribed to zero, Γ_L is identified with the lateral boundaries plus inlet, and Γ_O stands for the outflow boundary. See Figure 4.1 down below for details.

Whereas the boundaries Γ_S and Γ_O have a clear physical meaning, i.e. the possible solid and outer boundaries, Γ_L is defined for numerical convenience. This boundary is composed of any frontier with at least one component of the velocity prescribed to a known value, meaning that it can also entail the inlet boundary. This artificial truncation of the domain, which is performed with the objective of limiting the computational cost, does not assume that the affected boundaries are part of the outflow because this may not properly represent the physics of the flow and could even lead to numerical instabilities [40]. On Γ_L and Γ_O, which are to be located sufficiently far, *it is assumed that the acoustic scales are dominant*. This basically means that, using the typical traction boundary condition applied to the whole variable on the

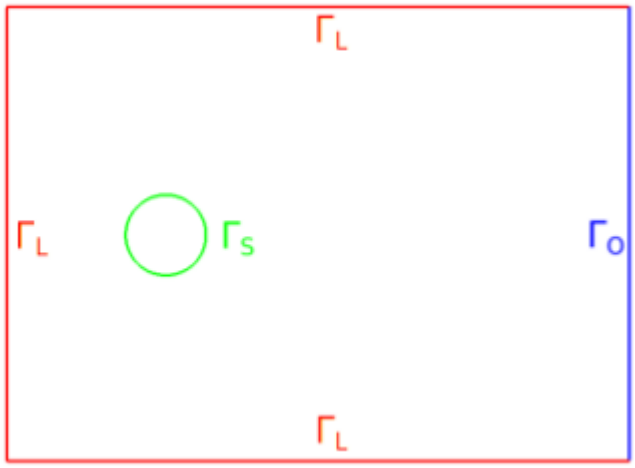


Figure 4.1: Schematic definition of the boundaries of the computational domain [40].

outlet frontier, would lead to the reflection of waves into the computational domain. Hence, a non-reflecting numerical technique will be considered, similarly as described in [29].

4.3.2 Non-reflecting and weak boundary conditions

Let us provide now a detailed description of the boundary conditions to be imposed for the problem in hand. We recall that the proposed methodology aims at providing a compatible technique for flow and acoustic components. This is to be made on Γ_L and Γ_O based on a weak prescription of the Dirichlet conditions plus a definition of a Sommerfled type non-reflecting boundary condition (NRBC).

First the reader should note that, having in mind the field decomposition introduced in (4.1a)–(4.1b) and the boundary splitting just explained above, the boundary term from the momentum conservation equation in (2.20) can be rewritten as,

$$\begin{aligned} -\langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_{\Gamma} &= -\langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_{\Gamma_S} - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_{\Gamma_L} - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_{\Gamma_O} \\ &= -\langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u},p)\rangle_{\Gamma_S} - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\bar{\boldsymbol{u}},\bar{p})\rangle_{\Gamma_L} - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u}',p')\rangle_{\Gamma_L} \\ &\quad - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\bar{\boldsymbol{u}},\bar{p})\rangle_{\Gamma_O} - \langle \boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u}',p')\rangle_{\Gamma_O} \end{aligned} \tag{4.3}$$

Let us now introduce the formulation of the boundary conditions for the problem in hand:

1. On the solid boundary, i.e. Γ_S, the velocity is assumed to be known and it will be prescribed in a strong form:

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{u}' = \boldsymbol{u}_s \quad \text{on} \quad \Gamma_S \tag{4.4}$$

2. On the frontiers belonging to Γ_L, distinct conditions are to be enforced:

   - The mean value of the velocity is prescribed to the flow inlet velocity,

   $$\bar{\boldsymbol{u}} = \boldsymbol{u}_L \quad \text{on} \quad \Gamma_L \tag{4.5}$$

   - A Sommerfeld-like non-reflecting boundary condition (NRBC) is used for the acoustic component of the velocity field. In the normal direction to the boundary we define,

   $$\boldsymbol{n}\cdot\boldsymbol{u}' = -\frac{1}{\rho c}\boldsymbol{n}\cdot[\boldsymbol{n}\cdot\sigma(\boldsymbol{u}',p')] \quad \text{on} \quad \Gamma_L \tag{4.6}$$

   being $\boldsymbol{n}$ the unit outward normal to Γ_L. For the tangential direction we write,

   $$\boldsymbol{m}\cdot[\boldsymbol{n}\cdot\sigma(\boldsymbol{u}',p')] = 0 \quad \text{on} \quad \Gamma_L \tag{4.7}$$

   for any vector $\boldsymbol{m}$ in the tangent direction to Γ_L.

3. Finally, on the outflow boundary Γ_O, the following conditions are going to be enforced.

   - For the mean component, the usual approach of prescribing the traction is used,

   $$\boldsymbol{n}\cdot\sigma(\bar{\boldsymbol{u}},\bar{p}) = \boldsymbol{t}_O \quad \text{on} \quad \Gamma_O \tag{4.8}$$

   - The same approach as in Γ_L is used now for the fluctuating component. A Sommerfeld-like condition is used in the normal direction and a zero traction is prescribed in the tangential direction. Then,

   $$\boldsymbol{n}\cdot\boldsymbol{u}' = -\frac{1}{\rho c}\boldsymbol{n}\cdot[\boldsymbol{n}\cdot\sigma(\boldsymbol{u}',p')] \quad \text{on} \quad \Gamma_O \tag{4.9}$$

   $$\boldsymbol{m}\cdot[\boldsymbol{n}\cdot\sigma(\boldsymbol{u}',p')] = 0 \quad \text{on} \quad \Gamma_O \tag{4.10}$$

Several remarks are now in order for the reader:

- The solid boundary Γ_S is a classical strong-Dirichlet type boundary. As we prescribe $\boldsymbol{u} = \boldsymbol{u}_s$ in a strong way on this boundary, the test functions will have a zero value on Γ_S and therefore the term $\langle\cdot,\cdot\rangle_{\Gamma_S}$ in Equation (4.3) vanishes.

- On Γ_L we have stated Dirichlet-type boundary conditions for the mean velocity $\bar{\boldsymbol{u}}$ and non-reflecting boundary conditions for the fluctuating part $\boldsymbol{u}'$. The prescription of $\bar{\boldsymbol{u}} = \boldsymbol{u}_L$

will be done weakly through the popular Nitche's method which provides a symmetric and better conditioned problem.

- Γ_O is a boundary with natural boundary conditions prescribed for $\bar{\boldsymbol{u}}$ and Sommerfeld condition for $\boldsymbol{u}'$. All these conditions will be prescribed in a weak sense.

- For inviscid flows, or when the viscous terms appearing in the definition of the stress tensor are neglected, the condition on the normal component of the fluctuating part of the velocity $\boldsymbol{u}'$ reduces to a Sommerfeld condition of the form, $p' = \rho c(\boldsymbol{u}' \cdot \boldsymbol{n})$.

Taking now into account these definitions and remarks, Equation (4.3) becomes,

$$\begin{aligned} -\langle \boldsymbol{v}, \boldsymbol{n} \cdot \sigma(\boldsymbol{u}, p) \rangle_\Gamma = &- \langle \boldsymbol{v}, \boldsymbol{n} \cdot \sigma(\bar{\boldsymbol{u}}, \bar{p}) \rangle_{\Gamma_L} + \langle \boldsymbol{v} \cdot \boldsymbol{n}, \rho c \boldsymbol{u}' \cdot \boldsymbol{n} \rangle_{\Gamma_L} \\ &- \langle \boldsymbol{v}, \boldsymbol{t}_O \rangle_{\Gamma_O} + \langle \boldsymbol{v} \cdot \boldsymbol{n}, \rho c \boldsymbol{u}' \cdot \boldsymbol{n} \rangle_{\Gamma_O} \end{aligned} \tag{4.11}$$

where the term over Γ_S is removed as we impose the velocity value in a strong manner. We still need to prescribe $\bar{\boldsymbol{u}} = \boldsymbol{u}_L$ weakly. As introduced above, we consider the Nitche's method for this purpose. Hence, the boundary term takes now the form,

$$\begin{aligned} -\langle \boldsymbol{v}, \boldsymbol{n} \cdot \sigma(\boldsymbol{u}, p) \rangle_\Gamma = &- \langle \boldsymbol{v}, \boldsymbol{n} \cdot \sigma(\bar{\boldsymbol{u}}, \bar{p}) \rangle_{\Gamma_L} + \langle \boldsymbol{v} \cdot \boldsymbol{n}, \rho c \boldsymbol{u}' \cdot \boldsymbol{n} \rangle_{\Gamma_L} + \langle \boldsymbol{v} \cdot \boldsymbol{n}, \rho c \boldsymbol{u}' \cdot \boldsymbol{n} \rangle_{\Gamma_O} \\ &- \langle \bar{\boldsymbol{u}} - \boldsymbol{u}_L, \boldsymbol{n} \cdot \sigma(\boldsymbol{v}, q) \rangle_{\Gamma_L} + \beta \frac{\mu_p}{l_p} \langle \boldsymbol{v}, \bar{\boldsymbol{u}} - \boldsymbol{u}_L \rangle_{\Gamma_L} + \langle \boldsymbol{v}, \boldsymbol{t}_O \rangle_{\Gamma_O} \end{aligned} \tag{4.12}$$

being β, μ_p and l_p numerical parameters. The first one is identified with the *penalty* dimensionaless parameter, to be chosen by the user. The second one has units of viscosity and the latter units of length. As discussed in [19], the parameters μ_p and l_p can be later computed as $\mu_p = \mu + |\boldsymbol{u}|h$ and $l_p = h$ once the finite element approximation is introduced, being h the element size of the mesh. As one can note, there are several terms in Equation (4.12) that are known and therefore can be taken to the right hand side of the problem. Thus, let us now group those boundary terms introducing the following forms for the finite element unknowns,

$$\begin{aligned} a_b([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) := &- \langle \boldsymbol{v}_h, \boldsymbol{n} \cdot \sigma(\bar{\boldsymbol{u}}_h, \bar{p}_h) \rangle_{\Gamma_L} + \langle \boldsymbol{v}_h \cdot \boldsymbol{n}, \rho c \boldsymbol{u}_h' \cdot \boldsymbol{n} \rangle_{\Gamma_L \cup \Gamma_O} \\ &+ \beta \frac{\mu_p}{l_p} \langle \boldsymbol{v}_h, \bar{\boldsymbol{u}}_h \rangle_{\Gamma_L} - \langle \bar{\boldsymbol{u}}_h, \boldsymbol{n} \cdot \sigma(\boldsymbol{v}_h, q_h) \rangle_{\Gamma_L} \end{aligned} \tag{4.13}$$

$$\ell_b([\boldsymbol{v}_h, q_h]) := \langle \boldsymbol{v}_h, \boldsymbol{t}_O \rangle_{\Gamma_O} - \langle \boldsymbol{u}_L, \boldsymbol{n} \cdot \sigma(\boldsymbol{v}_h, q_h) \rangle_{\Gamma_L} + \beta \frac{\mu_p}{l_p} \langle \boldsymbol{v}_h, \boldsymbol{u}_L \rangle_{\Gamma_L} \tag{4.14}$$

which will go to the left and right hand side of the problem, respectively.

As the reader might note, we have not provided yet a discrete expression to compute the mean and fluctuating components of the unknows, namely $[\bar{\boldsymbol{u}}, \bar{p}]$ and $[\boldsymbol{u}', p']$. This is precisely

the last ingredient of the proposed isentropic compressible formulation. In Equation (4.2a)–(4.2b) we introduced the concept of time window. At the discrete level, it is computed as $T_w = N_w \delta t$ being N_w a certain amount of time steps. It is proposed to use the trapezoidal rule for integration, and when a second order scheme is to be used in time, the expressions for the mean velocity and pressure are, respectively,

$$\bar{\boldsymbol{u}}_h^{n+1} = \frac{\delta t}{T_w}\left(\frac{1}{2}\boldsymbol{u}_h^{n+1} + \sum_{k=n-N_w+2}^{n} \boldsymbol{u}_h^k + \frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\right) \tag{4.15}$$

$$\bar{p}_h^{n+1} = \frac{\delta t}{T_w}\left(\frac{1}{2}p_h^{n+1} + \sum_{k=n-N_w+2}^{n} p_h^k + \frac{1}{2}p_h^{n-N_w+1}\right) \tag{4.16}$$

and these expressions keep the temporal integration implicit and second order accurate. However, it is important to run several time steps (N_w) prior to the application of the present formulation. This is due to the sharp nature of the pressure field at initial time steps together with the absence of minimally developed mean flow.

Those definitions for the mean variables, directly imply that the boundary forms from Equations (4.13) , (4.14) take now the form,

$$\begin{aligned} a_b^{n+1}([\boldsymbol{u}_h, p_h],[\boldsymbol{v}_h, q_h]) =& \frac{1}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}p_h^{n+1}\right\rangle_{\Gamma_L} - \frac{\mu}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}\cdot\nabla\boldsymbol{u}_h^{n+1}\right\rangle_{\Gamma_L} \\ &- \frac{\mu}{6N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}(\nabla\cdot\boldsymbol{u}_h^{n+1})\right\rangle_{\Gamma_L} + \frac{1}{2N_w}\left\langle \boldsymbol{u}_h^{n+1}, \boldsymbol{n}q_h\right\rangle_{\Gamma_L} \\ &- \frac{\mu}{2N_w}\left\langle \boldsymbol{u}_h^{n+1}, \boldsymbol{n}\cdot\nabla\boldsymbol{v}_h\right\rangle_{\Gamma_L} - \frac{\mu}{6N_w}\left\langle \boldsymbol{u}_h^{n+1}, \boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L} \\ &+ \left(1 - \frac{1}{2N_w}\right)\left\langle \rho c\boldsymbol{v}_h\cdot\boldsymbol{n}, \boldsymbol{u}_h^{n+1}\cdot\boldsymbol{n}\right\rangle_{\Gamma_L\cup\Gamma_O} \\ &+ \frac{\beta}{2N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}\right\rangle_{\Gamma_L} \end{aligned} \tag{4.17}$$

$$\begin{aligned} \ell_b^{n+1}([\boldsymbol{v}_h, q_h]) =& \left\langle \boldsymbol{v}_h, \boldsymbol{u}_L^{n+1}\right\rangle_{\Gamma_L} + \left\langle \boldsymbol{u}_L, \boldsymbol{n}q_h\right\rangle_{\Gamma_L} + \left\langle \boldsymbol{v}, \boldsymbol{t}_O\right\rangle_{\Gamma_O} \\ &- \mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}\cdot\nabla\boldsymbol{v}_h\right\rangle_{\Gamma_L} - \frac{\mu}{3}\mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L} \\ &- \frac{1}{N_w}\left\langle \boldsymbol{v}_h, \sum_{k=n-N_w+2}^{n}\left[\boldsymbol{n}p_h^k - \mu\boldsymbol{n}\cdot\nabla\boldsymbol{u}_h^k - \frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{u}_h^k)\right]\right\rangle_{\Gamma_L} \\ &- \frac{1}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}p_h^{n-N_w+1} - \mu\boldsymbol{n}\cdot\nabla\boldsymbol{u}_h^{n-N_w+1} - \frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{u}_h^{n-N_w+1})\right\rangle_{\Gamma_L} \\ &- \frac{1}{N_w}\left\langle \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k, \boldsymbol{n}q_h - \mu\boldsymbol{n}\cdot\nabla\boldsymbol{v}_h - \frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L} \end{aligned}$$

$$
\begin{aligned}
&-\frac{1}{2N_w}\left\langle \boldsymbol{u}_h^{n-N_w+1}, \boldsymbol{n}q_h - \mu\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{v}_h - \frac{\mu}{3}\boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L} \\
&+\frac{1}{N_w}\left\langle \rho c\boldsymbol{v}_h\cdot\boldsymbol{n}, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k\cdot\boldsymbol{n} + \frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\cdot\boldsymbol{n}\right\rangle_{\Gamma_L\cup\Gamma_O} \\
&-\frac{\beta}{N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{v}_h, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k + \frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\right\rangle_{\Gamma_L}
\end{aligned}
\tag{4.18}
$$

where we have also introduced the definition of the stress tensor, i.e. $\sigma_h(\boldsymbol{u}_h, p_h) = -p_h\boldsymbol{I} + \mu\boldsymbol{\nabla}\boldsymbol{u}_h + \frac{\mu}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_h)\boldsymbol{I}$ and $\sigma_h(\boldsymbol{v}_h, q_h) = -q_h\boldsymbol{I} + \mu\boldsymbol{\nabla}\boldsymbol{v}_h + \frac{\mu}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_h)\boldsymbol{I}$, the latter needed for the simetrization terms coming from the application of Nitche's method.

Chapter 5

Design of fractional step schemes

5.1 Introduction

In constrast to a typical monolithic algorithm, where the solution of a problem is computed all at once, the general idea of the fractional step technique is to segregate or split the calculation of the unknowns of the problem, in our case velocity and pressure. Our approach is to present a segregation technique at the pure algebraic level, in constrast to the continuous approach originally done for the incompressible Navier-Stokes equations. In this chapter we propose a $k = 1, 2$ order algorithm to solve the isentropic compressible problem in a segregate manner.

5.2 Monolithic matrix version of the problem

Let us start first by writting the monolithic matrix version of the problem in hand. Gathering all the information described in the previous chapters, the formulation of the isentropic compressible problem can be arranged with the forms $a([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$, $a_b([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$, $a_s([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$, $\ell_s([\boldsymbol{v}_h, q_h])$, $\ell_b([\boldsymbol{v}_h, q_h])$ previously defined. When the problem is discretized in time, as described in Section (1.6), the final discrete problem with all the ingredients will read as follows:

Given the inital values $\boldsymbol{u}_h^0$ and p_h^0, find the compressible velocity and pressure finite element unknowns at time step t^{n+1}, $\boldsymbol{u}_h^{n+1} \in \mathscr{V}_h^d$, $p_h^{n+1} \in \mathscr{Q}_h$, such that,

$$\begin{aligned} a^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) + a_s^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) + a_b^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) \\ = \ell_s^{n+1}([\boldsymbol{v}_h, q_h]) + \ell_b^{n+1}([\boldsymbol{v}_h, q_h]) \end{aligned} \tag{5.1}$$

$\forall\, \boldsymbol{v}_h \in \mathscr{V}_h^d \subset \mathscr{V}^d \equiv \boldsymbol{H}^1(\Omega),\ q_h \in \mathscr{Q}_h \subset \mathscr{Q} \equiv H^1(\Omega),$

where the expression for $a^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$ is given by Equation (2.19) (substituting the FE unknowns and the time derivative by its discrete version), the form $a_s^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$ can be computed from (3.18) and the boundary terms in $a_b^{n+1}([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h])$ are calculated as defined in (4.17). On the other hand, the right hand side forms $\ell_s([\boldsymbol{v}_h, q_h])$ and $\ell_b([\boldsymbol{v}_h, q_h])$ are computed as in Equations (3.19) and (4.18), from known values from previous time steps.

We assume that velocity and pressure are to be constructed using the standard finite element interpolation from the nodal values. Let us introduce the following Lagrange basis $\{\boldsymbol{\phi}_{i,j}\}$, $i \in \mathscr{N}_u$, $j = 1, \ldots, d$ and $\{\pi_m\}$, $m \in \mathscr{N}_p$ associated to the finite element spaces $\mathscr{V}_h^d$ and $\mathscr{Q}_h$. In these definitions, $\mathscr{N}_u$ and $\mathscr{N}_p$ are the set of free velocity and pressure nodes, respectively. On the other hand, $\boldsymbol{\phi}_{i,j}$ is the vector containing the standard shape functions ϕ_i of node i at position j and the rest of components are null. Likewise, π_m stands for the pressure shape function at a certain node, m. Hence, we write the finite element approximation of the unknowns in the following manner,

$$\boldsymbol{u}_h^{n+1}(\boldsymbol{x}) = \sum_{a \in \mathscr{N}_u}^{n_{nodes}} \phi_a(\boldsymbol{x}) \left(\mathsf{U}^{n+1}\right)_a \quad ; \quad p_h^{n+1}(\boldsymbol{x}) = \sum_{a \in \mathscr{N}_p} \pi_a(\boldsymbol{x}) \left(\mathsf{P}^{n+1}\right)_a \tag{5.2}$$

where U^{n+1} and P^{n+1} are the arrays of nodal values for velocity and pressure and n_{nodes} is the total number of nodes in the mesh. The symbol $(\cdot)_a$ denotes the sub-array with the values for node a, i.e., an array of d components for velocity and a single component for pressure. Equivalent finite element definitions apply for the associated test functions $\boldsymbol{v}_h$ and q_h. It is assumed that these shape functions are time-independent. The previous definitions lead to the following *non-linear* algebraic system for the problem in Equation (5.1),

$$\begin{aligned} \frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}\mathsf{U}^{n+1} + \mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{S}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{K}_{b,\boldsymbol{u}}\mathsf{U}^{n+1} \\ + \mathsf{G}\mathsf{P}^{n+1} + \mathsf{G}_b\mathsf{P}^{n+1} = \mathsf{F}_{\boldsymbol{u}} + \mathsf{F}_{s,\boldsymbol{u}} + \mathsf{F}_{b,\boldsymbol{u}} \end{aligned} \tag{5.3a}$$

$$\begin{aligned} \frac{1}{\gamma_k \delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{D}\mathsf{U}^{n+1} \\ + \mathsf{D}_b\mathsf{U}^{n+1} = \mathsf{F}_p + \mathsf{F}_{s,p} + \mathsf{F}_{b,p} \end{aligned} \tag{5.3b}$$

where subscripts $(\cdot)_{\boldsymbol{u}}$ and $(\cdot)_p$ refer to matrices of the momentum and continuity equation. In addition, $(\cdot)_s$ and $(\cdot)_b$ refer, respectively, to terms arising from the stabilization and from the special treatment of boundary conditions. Again, k refers to the time accuracy of the scheme.

The matrices appearing in the left hand side of the system are defined at the elemental level

for $1 \leq i, j \leq d$ as,

$$(\mathsf{M}_{\boldsymbol{u}})_{ij}^{ab} := \left\langle \boldsymbol{\phi}_{a,i}, \rho^{n+1}\boldsymbol{\phi}_{b,j} \right\rangle_K, \qquad a,b \in \mathscr{N}_u \tag{5.4a}$$

$$\begin{aligned}\mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})_{ij}^{ab} := &\left\langle \boldsymbol{\phi}_{a,i}, \rho^{n+1}\boldsymbol{u}^{n+1}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{b,j} \right\rangle_K + \mu \left\langle \boldsymbol{\nabla}\boldsymbol{\phi}_{a,i}, \boldsymbol{\nabla}\boldsymbol{\phi}_{b,j} \right\rangle_K \\ &+ \frac{\mu}{3}\left\langle \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i}, \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{b,j} \right\rangle_K, \quad a,b \in \mathscr{N}_u\end{aligned} \tag{5.4b}$$

$$\begin{aligned}\mathsf{S}_{\boldsymbol{u}}(\mathsf{U}^{n+1})_{ij}^{ab} := &\left\langle \rho^{n+1}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i}, \tau_{1,K}\rho^{n+1}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{b,j} \right\rangle_K \\ &+ \left\langle \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i}, \tau_{2,K}\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{\mathrm{b,j}} \right\rangle_K \quad a,b \in \mathscr{N}_u\end{aligned} \tag{5.4c}$$

$$\begin{aligned}(\mathsf{K}_{b,\boldsymbol{u}})_{ij}^{ab} := &-\frac{\mu}{2N_w}\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{n}(\boldsymbol{\nabla}\boldsymbol{\phi}_{b,j}) \right\rangle_{\partial K\cap\Gamma_L} - \frac{\mu}{6N_w}\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{b,j}) \right\rangle_{\partial K\cap\Gamma_L} \\ &-\frac{\mu}{2N_w}\left\langle \boldsymbol{\phi}_{b,j}, \boldsymbol{n}(\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i}) \right\rangle_{\partial K\cap\Gamma_L} - \frac{\mu}{6N_w}\left\langle \boldsymbol{\phi}_{b,j}, \boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i}) \right\rangle_{\partial K\cap\Gamma_L} \\ &+\left(1-\frac{1}{2N_w}\right)\left\langle \rho c\boldsymbol{\phi}_{a,i}\cdot\boldsymbol{n}, \boldsymbol{\phi}_{b,j}\cdot\boldsymbol{n} \right\rangle_{\partial K\cap\Gamma_L\cup\Gamma_O} \\ &+\frac{\beta}{2N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{\phi}_{b,j} \right\rangle_{\partial K\cap\Gamma_L}, \qquad a,b \in \mathscr{N}_u\end{aligned} \tag{5.4d}$$

$$(\mathsf{G}_i)^{ab} := -\left\langle \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i}, \pi_b \right\rangle_K, \qquad a \in \mathscr{N}_u,\ b \in \mathscr{N}_p \tag{5.4e}$$

$$(\mathsf{G}_b)_i^{ab} := \frac{1}{2N_w}\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{n}\pi_b \right\rangle_{\partial K\cap\Gamma_L}, \qquad a \in \mathscr{N}_u,\ b \in \mathscr{N}_p \tag{5.4f}$$

for the algebraic version of the momentum conservation equation and,

$$(\mathsf{M}_p)^{ab} := \left\langle \pi_a, \frac{1}{(c^2)^{n+1}\rho^{n+1}}\pi_b \right\rangle_K, \qquad a,b \in \mathscr{N}_p \tag{5.5a}$$

$$(\mathsf{K}_p(\mathsf{U}^{n+1})^{ab} := \left\langle \pi_a, \frac{1}{(c^2)^{n+1}\rho^{n+1}}\boldsymbol{u}^{n+1}\cdot\boldsymbol{\nabla}\pi_b \right\rangle_K, \quad a,b \in \mathscr{N}_p \tag{5.5b}$$

$$\begin{aligned}(\mathsf{S}_p(\mathsf{U}^{n+1})^{ab} := &\left\langle \frac{1}{(c^2)^{n+1}\rho^{n+1}}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\pi_a, \tau_{2,K}\frac{1}{(c^2)^{n+1}\rho^{n+1}}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\pi_b \right\rangle_K \\ &+\left\langle \boldsymbol{\nabla}\pi_a, \tau_{1,K}\boldsymbol{\nabla}\pi_b \right\rangle_K \quad a,b \in \mathscr{N}_p\end{aligned} \tag{5.5c}$$

$$(\mathsf{D}_b)_j^{ab} := \frac{1}{2N_w}\left\langle \boldsymbol{\phi}_{b,j}, \boldsymbol{n}\pi_a \right\rangle_{\partial K\cap\Gamma_L}, \qquad a \in \mathscr{N}_p,\ b \in \mathscr{N}_u \tag{5.5d}$$

$$\mathsf{D}_j^{ab} := \left\langle \pi_a, \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{b,j} \right\rangle_K, \qquad a \in \mathscr{N}_p,\ b \in \mathscr{N}_u \tag{5.5e}$$

for the algebraic mass conservation equation. In addition to this, the right hand side vectors,

containing known information from previous time steps, are computed as follows,

$$(\mathsf{F}_{\boldsymbol{u}})_i^a := \frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}\left(\sum_{m=0}^{k-1}\alpha_k^m \mathsf{U}_{a,i}^{n-m}\right) \tag{5.6}$$

$$\begin{aligned}(\mathsf{F}_{s,\boldsymbol{u}})_i^a := &\left\langle \rho^{n+1}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i}, \tau_{1,K}\Pi_h[\rho^{n+1}\boldsymbol{u}_h^{n+1}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h^n]\right\rangle_K \\ &+\left\langle \boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i}, \tau_{2,K}\Pi_h[\boldsymbol{\nabla}\cdot\boldsymbol{u}_h^n]\right\rangle_K, \qquad a\in\mathscr{N}_u\end{aligned} \tag{5.7}$$

$$\begin{aligned}(\mathsf{F}_{b,\boldsymbol{u}})_i^a := &\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{u}_L^{n+1}\right\rangle_{\partial K\cap\Gamma_L} - \mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i}\right\rangle_{\partial K\cap\Gamma_L} - \frac{\mu}{3}\mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i})\right\rangle_{\partial K\cap\Gamma_L} \\ &-\frac{1}{N_w}\left\langle \boldsymbol{\phi}_{a,i}, \sum_{k=n-N_w+2}^{n}\left[\boldsymbol{n}p_h^k - \mu\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h^k - \frac{\mu}{3}\boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_h^k)\right]\right\rangle_{\partial K\cap\Gamma_L} \\ &-\frac{1}{2N_w}\left\langle \boldsymbol{\phi}_{a,i}, \boldsymbol{n}p_h^{n-N_w+1} - \mu\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h^{n-N_w+1} - \frac{\mu}{3}\boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_h^{n-N_w+1})\right\rangle_{\partial K\cap\Gamma_L} \\ &+\frac{1}{N_w}\left\langle \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k, \mu\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i} + \frac{\mu}{3}\boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i})\right\rangle_{\partial K\cap\Gamma_L} \\ &+\frac{1}{2N_w}\left\langle \boldsymbol{u}_h^{n-N_w+1}, \mu\boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{\phi}_{a,i} + \frac{\mu}{3}\boldsymbol{n}(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{a,i})\right\rangle_{\partial K\cap\Gamma_L} \\ &+\frac{1}{N_w}\left\langle \rho c\boldsymbol{\phi}_{a,i}\cdot\boldsymbol{n}, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k\cdot\boldsymbol{n} + \frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\cdot\boldsymbol{n}\right\rangle_{\partial K\cap\Gamma_L\cup\Gamma_O} \\ &-\frac{\beta}{N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{\phi}_{a,i}, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k + \frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\right\rangle_{\partial K\cap\Gamma_L}, \qquad a\in\mathscr{N}_u\end{aligned} \tag{5.8}$$

$$\mathsf{F}_p^a := \frac{1}{\gamma_k\delta t}\mathsf{M}_p\left(\sum_{m=0}^{k-1}\alpha_k^m \mathsf{P}_a^{n-m}\right), \qquad a\in\mathscr{N}_p \tag{5.9}$$

$$\begin{aligned}(\mathsf{F}_{s,p})^a := &\left\langle \frac{1}{(c^2)^{n+1}\rho^{n+1}}\boldsymbol{u}^{n+1}\cdot\boldsymbol{\nabla}\pi_a, \tau_{2,K}\Pi_h\left[\frac{1}{(c)^{n+1}\rho^{n+1}}\boldsymbol{u}\cdot\boldsymbol{\nabla}p_h^n\right]\right\rangle_K \\ &+\left\langle \boldsymbol{\nabla}\pi_a, \tau_{1,K}\Pi_h[\boldsymbol{\nabla}p_h^n]\right\rangle_K, \qquad a\in\mathscr{N}_p\end{aligned} \tag{5.10}$$

$$\begin{aligned}(\mathsf{F}_{b,p})^a := &\left\langle \boldsymbol{u}_L, \boldsymbol{n}\pi_a\right\rangle_{\partial K\cap\Gamma_L} - \frac{1}{N_w}\left\langle \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k, \boldsymbol{n}\pi_a\right\rangle_{\partial K\cap\Gamma_L} \\ &-\frac{1}{N_w}\left\langle \boldsymbol{u}_h^{n-N_w+1}, \boldsymbol{n}\pi_a\right\rangle_{\partial K\cap\Gamma_L}, \qquad a\in\mathscr{N}_p.\end{aligned} \tag{5.11}$$

In all these definitions, the dependence on the vector of unknowns has been explicitely displayed in order to remark the non-linear character of the problem.

Remark: we recall that all the possible non-linearities that appear in the problem, mainly associated with the velocity $\boldsymbol{u}_h^{n+1}$, density ρ^{n+1} and speed of sound $(c^2)^{n+1}$, are computed by means of the Picard or fixed-point algorithm. Hence, we will be using the velocity from

the previous iteration in the non-linearity loop, i.e., $\boldsymbol{u}_h^{n+1,i}$ where i denotes now the iteration counter. Then, we will compute $\boldsymbol{u}_h^{n+1,i} \cdot \boldsymbol{\nabla}(\cdot)^{n+1,i+1}$ and,

$$\rho^{n+1} = \rho_0 \left(1 + \frac{\gamma - 1}{2} \frac{|\boldsymbol{u}_h^{n+1,i}|}{c_0^2} \right)^{\gamma - 1} \tag{5.12}$$

$$\left(c^2\right)^{n+1} = c_0^2 \left(1 + \frac{\gamma - 1}{2} \frac{|\boldsymbol{u}_h^{n+1,i}|}{c_0^2} \right)^{-1} \tag{5.13}$$

The monolitich approach of the non-linear algebraic system (5.3a)–(5.3b) can be rewritten in the usual form $\mathsf{A}\boldsymbol{X}^{n+1} = \boldsymbol{F}$, with the following structure,

$$\begin{bmatrix} \mathsf{A}_{11} & \mathsf{A}_{12} \\ \mathsf{A}_{21} & \mathsf{A}_{22} \end{bmatrix} \begin{Bmatrix} \mathsf{U}^{n+1} \\ \mathsf{P}^{n+1} \end{Bmatrix} = \begin{Bmatrix} \mathsf{F}_1 \\ \mathsf{F}_2 \end{Bmatrix} \tag{5.14}$$

where we now redefine,

$$\begin{aligned}
\mathsf{A}_{11} &= \frac{1}{\gamma_k \delta t} \mathsf{M}_{\boldsymbol{u}} + \mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1}) + \mathsf{S}_{\boldsymbol{u}}(\mathsf{U}^{n+1}) + \mathsf{K}_{b,\boldsymbol{u}} \\
\mathsf{A}_{12} &= \mathsf{G} + \mathsf{G}_b \\
\mathsf{A}_{21} &= \mathsf{D} + \mathsf{D}_b \\
\mathsf{A}_{22} &= \frac{1}{\gamma_k \delta t} \mathsf{M}_p + \mathsf{K}_p(\mathsf{U}^{n+1}) + \mathsf{S}_p(\mathsf{U}^{n+1}) \\
\mathsf{F}_1 &= \mathsf{F}_{\boldsymbol{u}} + \mathsf{F}_{s,\boldsymbol{u}} + \mathsf{F}_{b,\boldsymbol{u}} \\
\mathsf{F}_2 &= \mathsf{F}_p + \mathsf{F}_{s,p} + \mathsf{F}_{b,p}
\end{aligned}$$

The reader should note that, with respect to the incompressible Navier-Stokes case, the structure of the isentropic compressible problem is no longer of the saddle-point type, in other words, matrix A_{22} is not zero.

5.3 Predictor-corrector algorithm

This section is intended to include a description on the design of fractional step schemes for the isentropic compressible Navier-Stokes equations when the primitive variables are used. A usual approach to design fractional step methods is to extrapolate the variable that needs to be segregated from a certain equation, and correct the results once this variable has been computed somehow. For our specific case, in principle we will extrapolate the pressure in the momentun equation so that to compute an *intermediate or fractional velocity*. Later, the pres-

sure is computed and finally, the pertinent corrections are performed. This way of proceeding will give us an algorithm of *pressure-correction* type.

Let us start by writing the system (5.3a)–(5.3b) in the following *equivalent* manner,

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1} + \mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{S}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{K}_{b,\boldsymbol{u}}\mathsf{U}^{n+1} + \mathsf{G}\widehat{\mathsf{P}}^{n+1}_{k-1} + \mathsf{G}_b\widehat{\mathsf{P}}^{n+1}_{k-1} = \mathsf{F}_1 \tag{5.15a}$$

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}(\mathsf{U}^{n+1} - \widetilde{\mathsf{U}}^{n+1}) + (\mathsf{G}+\mathsf{G}_b)(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}^{n+1}_{k-1}) = \mathbf{0} \tag{5.15b}$$

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{D}\mathsf{U}^{n+1} + \mathsf{D}_b\mathsf{U}^{n+1} = \mathsf{F}_2 \tag{5.15c}$$

where $\widetilde{\mathsf{U}}^{n+1}$ is an auxiliary variable to which we shall refer as intermediate velocity. Likewise, $\widehat{\mathsf{P}}^{n+1}_{k-1}$ is an extrapolation of the pressure of order $k-1$ at time step t^{n+1}. See Equations (1.19a)–(1.19c) for details. The reader should note that adding up (5.15a) and (5.15b) we recover (5.3a).

Let us now do a little bit of manipulation with the previous equations, what will allow the final scheme to be easier to implement and practical. From Equation (5.15b) one can obtain a relation between the intermediate velocity $\widetilde{\mathsf{U}}^{n+1}$and the end-of-step velocity, namely U^{n+1}. It gives,

$$\mathsf{U}^{n+1} = \widetilde{\mathsf{U}}^{n+1} - \gamma_k \delta t \mathsf{M}_{\boldsymbol{u}}^{-1}(\mathsf{G}+\mathsf{G}_b)(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}^{n+1}_{k-1}) \tag{5.16}$$

and if we introduce this result into (5.15c) it would yield,

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} - \gamma_k \delta t(\mathsf{D}+\mathsf{D}_b)\mathsf{M}_{\boldsymbol{u}}^{-1}(\mathsf{G}+\mathsf{G}_b)(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}^{n+1}_{k-1}) = \mathsf{F}_2 - (\mathsf{D}+\mathsf{D}_b)\widetilde{\mathsf{U}}^{n+1} \tag{5.17}$$

but at this point, it is very convenient to make the following observations:

- One should notice that the term $\mathsf{D}\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}$ can be viewed as an approximation to the discrete version of the laplacian operator $\boldsymbol{\Delta}$, [23]. In order to avoid dealing with this matrix (which is computationally feasible only if $\mathsf{M}_{\boldsymbol{u}}$ is approximated by a diagonal matrix), we can work with,

$$\mathsf{D}\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G} \approx \mathsf{L} \qquad \text{with components} \qquad \mathsf{L}^{ab} := -\left\langle \frac{1}{\rho^{n+1}}\boldsymbol{\nabla}\pi_a, \boldsymbol{\nabla}\pi_b \right\rangle_K \tag{5.18}$$

- If we wanted to compute the pressure from Equation (5.17), we would still have to face

the problem of computing the terms $\mathsf{D}_b\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}$, $\mathsf{D}\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}_b$ and $\mathsf{D}_b\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}_b$. Such computations can be really time consuming and bourdersome. Note that an approximation similar to the one just commented is not possible due to the character of the boundary matrices D_b and G_b.

Having this in mind, what we propose in this work is to modify Equations (5.15a) and (5.15c) in such a way that we take the terms $\mathsf{G}_b\widehat{\mathsf{P}}_{k-1}^{n+1}$ and $\mathsf{D}_b\mathsf{U}^{n+1}$ to the right hand side, *by means of an extrapolation of order k* of both terms, what would maintain the temporal accuracy of the scheme. In addition, we will not include the boundary extrapolation for the pressure in the correction step. Thus, the relation between both intermediate and end-of step velocity established in (5.16) would be now,

$$\mathsf{U}^{n+1} = \widetilde{\mathsf{U}}^{n+1} - \gamma_k\delta t\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) \tag{5.19}$$

and when this expression is taken into account in the system for the pressure, (5.17) would simply become,

$$\begin{aligned}\frac{1}{\gamma_k\delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} \\ -\gamma_k\delta t\mathsf{D}\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathsf{F}_2 - \mathsf{D}\widetilde{\mathsf{U}}^{n+1} - \mathsf{D}_b\widehat{\mathsf{U}}_k^{n+1}\end{aligned} \tag{5.20}$$

This manipulation induces that the set of equations to be solved now yields,

$$\begin{aligned}\frac{1}{\gamma_k\delta t}\mathsf{M}_{\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1} + \mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{S}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} \\ +\mathsf{K}_{b,\boldsymbol{u}}\mathsf{U}^{n+1} = \mathsf{F}_1 - \mathsf{G}\widehat{\mathsf{P}}_{k-1}^{n+1} - \mathsf{G}_b\widehat{\mathsf{P}}_k^{n+1}\end{aligned} \tag{5.21}$$

$$\begin{aligned}\frac{1}{\gamma_k\delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} \\ +\gamma_k\delta t\mathsf{L}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathsf{F}_2 - \mathsf{D}\widetilde{\mathsf{U}}^{n+1} - \mathsf{D}_b\widehat{\mathsf{U}}_k^{n+1}\end{aligned} \tag{5.22}$$

$$\frac{1}{\gamma_k\delta t}\mathsf{M}_{\boldsymbol{u}}(\mathsf{U}^{n+1} - \widetilde{\mathsf{U}}^{n+1}) + \mathsf{G}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathbf{0} \tag{5.23}$$

where the equations are written in the order they are solved. The reader should note that now the products $\mathsf{D}_b\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}$, $\mathsf{D}\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}_b$ and $\mathsf{D}_b\mathsf{M}_{\boldsymbol{u}}^{-1}\mathsf{G}_b$ do not appear in the formulation and that we treat some terms on the boundary explicitly via extrapolations in $\mathsf{G}_b\widehat{\mathsf{P}}_k^{n+1}$ and $\mathsf{D}_b\widehat{\mathsf{U}}_k^{n+1}$.

Remark: In principle, we are interested in designing a second order segregation algorithm. It is well known that the extrapolation of second order of the term $\mathsf{G}\widehat{\mathsf{P}}_{k-1}^{n+1}$, i.e. taking $k = 3$ is unstable. In fact, this issue motivated the study of the so called velocity correction algorithms, which allow to design fractional step schemes of third order in time. Still, we do not expect

any unstable behaviour of the term $\mathsf{G}_b\widehat{\mathsf{P}}_k^{n+1}$ for the extrapolation of second order.

At this point, we can make the essential approximation,

$$\mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} \approx \mathsf{K}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1})\widetilde{\mathsf{U}}^{n+1} \tag{5.24}$$

and one can see from (5.23) that this perturbation is of order $\mathscr{O}(\delta t^k)$. In addition, $S_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} \approx \mathsf{S}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1})\widetilde{\mathsf{U}}^{n+1}$ and we will use $\widetilde{\mathsf{U}}^{n+1}$ to compute the adveciton velocity when solving for the pressure. This basically means that in practice, we assume $\mathsf{K}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} \simeq \mathsf{K}_p(\widetilde{\mathsf{U}}^{n+1})\mathsf{P}^{n+1}$ and $\mathsf{S}_p(\mathsf{U}^{n+1})\mathsf{P}^{n+1} \simeq \mathsf{S}_p(\widetilde{\mathsf{U}}^{n+1})\mathsf{P}^{n+1}$.

In principle, fractional step methods can be designed by choosing $k = 1,2$ and solving as follows,

1. Compute an intermediate velocity $\widetilde{\mathsf{U}}^{n+1}$ using extrapolations for the pressure terms and solving,

$$\frac{1}{\gamma_k\delta t}\mathsf{M}_{\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1} + \mathsf{K}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1})\widetilde{\mathsf{U}}^{n+1} + \mathsf{S}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1})\widetilde{\mathsf{U}}^{n+1} + \mathsf{K}_{b,\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1} = \mathsf{F}_1 - \mathsf{G}\widehat{\mathsf{P}}_{k-1}^{n+1} - \mathsf{G}_b\widehat{\mathsf{P}}_k^{n+1} \tag{5.25}$$

2. Solve for the pressure P^{n+1} using the intermediate velocity just computed above,

$$\frac{1}{\gamma_k\delta t}\mathsf{M}_p\mathsf{P}^{n+1} + \mathsf{K}_p(\widetilde{\mathsf{U}}^{n+1})\mathsf{P}^{n+1} + \mathsf{S}_p(\widetilde{\mathsf{U}}^{n+1})\mathsf{P}^{n+1} + \gamma_k\delta t\mathsf{L}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathsf{F}_2 - \mathsf{D}\widetilde{\mathsf{U}}^{n+1} - \mathsf{D}_b\widehat{\mathsf{U}}_k^{n+1} \tag{5.26}$$

3. Perform the correction in order to compute the end-of-step velocity U^{n+1} after solving,

$$\frac{1}{\gamma_k\delta t}\mathsf{M}_{\boldsymbol{u}}(\mathsf{U}^{n+1} - \widetilde{\mathsf{U}}^{n+1}) + \mathsf{G}(\mathsf{P}^{n+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathbf{0} \tag{5.27}$$

The reader should note that the only non-linear problem is in the first step, i.e. when solving for the intermediate velocity. Thus the next step is to linearize such problem. As already mentioned, we consider the Picard approach to solve the non-linearities, i.e. taking the known values from previous iterations or time step. In Algorithm 5.1 we include the final scheme in its matrix form, where the superscript i denotes the iteration counter. For a more detailed exposition, in the Appendix, we include also Algorithm 8.1 which contains a second order fractional step scheme in it semidiscrete form. There we offer a more computational viewpoint for actually implementing the method.

The previous scheme provides a practical way to decouple the solution of the velocity and

pressure unknowns for the isentropic compressible problem. However, some errors are introduced which can affect the accuracy of the final computation. The first error is due to the fact that the momentum equation (5.25) is solved for the intermediate velocity $\widetilde{\mathsf{U}}^{n+1,i+1}$ instead of the end of step velocity $\mathsf{U}^{n+1,i+1}$. This is usually referred as *fractional error* or error of fractioning. Another noticeable source of error may come from the fact of not including the time-dependency of subscales in the stabilization. As pointed out earlier in this work, this approach might show an unstable behavior for anisotropic space-time discretizations. Finally, and maybe the most remarkable error, an extra Dirichlet boundary condition needs to be provided for Equation (5.26) . The Dirichlet boundary condition we enforce (at the continuous level) is:

$$\bar{p} = 0 \qquad \text{in } \Gamma_O \tag{5.28}$$

$$p' = \rho c(\boldsymbol{u}' \cdot \boldsymbol{n}) \qquad \text{in } \Gamma_O \tag{5.29}$$

where we take into account the decomposition of the pressure in average (flow) and oscillatory (acoustic) components as suggested in the previous chapter. Equation (5.29) aids to impose the NRBC when solving the continuity equation. For low viscous flows this approximation is reasonable, since the effect of the viscosity is small and, in addition, we assume that Γ_O is located sufficiently far away from the solid boundary, so that the velocity gradients could be neglected. Finally, when computing $\boldsymbol{u}' = \boldsymbol{u} - \bar{\boldsymbol{u}}$, we make use of $\boldsymbol{u}^{n+1,i+1}$, solution of the algebraic problem (5.25).

Algorithm 5.1 Fist and second order fractional step schemes (k=1,2)

- Non-linear problem to compute the intermediate velocity $\widetilde{\mathsf{U}}^{n+1,i+1}$ using the pressure extrapolations:

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1,i+1} + \mathsf{K}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1,i})\widetilde{\mathsf{U}}^{n+1,i+1} + \mathsf{S}_{\boldsymbol{u}}(\widetilde{\mathsf{U}}^{n+1,i})\widetilde{\mathsf{U}}^{n+1,i+1} + \mathsf{K}_{b,\boldsymbol{u}}\widetilde{\mathsf{U}}^{n+1,i+1} = \mathsf{F}_1 - \mathsf{G}\widehat{\mathsf{P}}_{k-1}^{n+1} - \mathsf{G}_b\widehat{\mathsf{P}}_k^{n+1} \tag{5.30}$$

- Compute the pressure $\mathsf{P}^{n+1,i+1}$:

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_p\mathsf{P}^{n+1,i+1} + \mathsf{K}_p(\widetilde{\mathsf{U}}^{n+1,i+1})\mathsf{P}^{n+1,i+1} + \mathsf{S}_p(\widetilde{\mathsf{U}}^{n+1,i+1})\mathsf{P}^{n+1,i+1} + \gamma_k \delta t\mathsf{L}(\mathsf{P}^{n+1,i+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathsf{F}_2 - \mathsf{D}\widetilde{\mathsf{U}}^{n+1,i+1} - \mathsf{D}_b\widehat{\mathsf{U}}_k^{n+1} \tag{5.31}$$

- Velocity correction step, $\mathsf{U}^{n+1,i+1}$:

$$\frac{1}{\gamma_k \delta t}\mathsf{M}_{\boldsymbol{u}}(\mathsf{U}^{n+1,i+1} - \widetilde{\mathsf{U}}^{n+1,i+1}) + \mathsf{G}(\mathsf{P}^{n+1,i+1} - \widehat{\mathsf{P}}_{k-1}^{n+1}) = \mathbf{0} \tag{5.32}$$

Chapter 6

Numerical results

In this chapter we aim at presenting some results and comments for the developed formulation for the isentropic compressible problem. Here we will present a benchmark for the validation of the implemention of our algorithm (see Algorithm 8.1 in the Appendix for programming details).

6.1 Aeolian tones of a low Mach viscous flow

The benchmark we have chosen for a proper assessment of the proposed formulation consists in a 2D flow around a cylinder which allows to evaluate what in the literature is referred as aeolian tones (see [32]). From the point of view of the final user we recall that some additional parameters are needed when departing from an already implemented incompressible Navier-Stokes solver. Those are, the universal gas constant $R = 8.31\ J/kg \cdot mol$, the molar mass of the considered gas, the sound propagation speed and the bulk temperature. For this example, the values for air are considered, i.e., $\mathscr{M} = 28.97\ g/mol$, $c_0 = 343.26\ m/s$ and $T_0 = 293.15\ K$.

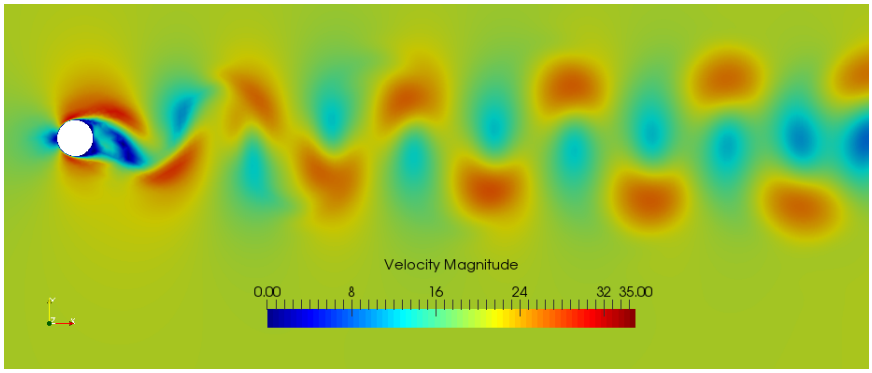
The rectangle where the cylinder is placed is $10 \times 30\ m$ in size and it is inserted inside a box of $450 \times 450\ m$ in order to describe the far field conditions of the problem (far away from the cylinder). The diameter is $D = 0.3\ m$. We have used the *GiD* prepocessor, developed at CIMNE, so that to set up the geometry and boundary conditions. The incident velocity has been chosen as $\boldsymbol{u}_{in} = [20, 0]\ m/s$ and we have taken a dynamic viscosity coefficient of $\mu = 0.006$. All this information leads to $Re = 1000$ and $M = 0.0583$ in the far field.

For the simulation we consider an unstructured mesh of nearly $300,000$ triangular linear elements and we use equal interpolation for velocity and pressure, thanks to the stabilized formulation proposed. The size of the mesh is of $3 \times 10^{-3} D$ near the cylinder surface, so that to be able to capture the expected high gradients in that zone. The case has been run up to

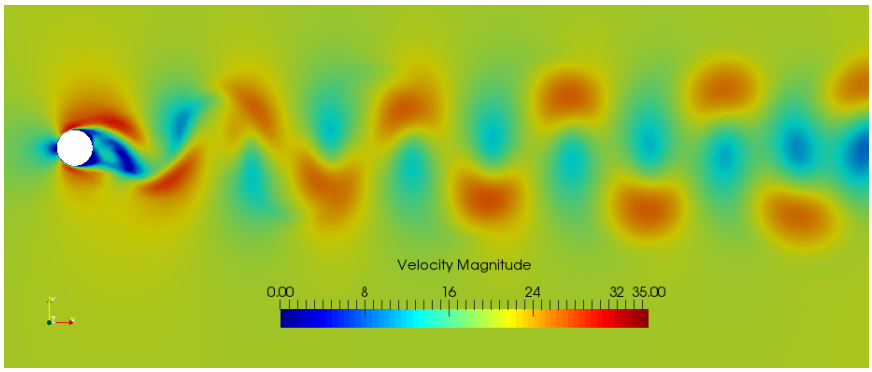
$t = 1.0\ s$ with a time step size $\delta t = 0.001\ s$. It is important to note that the time step has to be small enough to be able to reproduce the sound waves in an adequate manner.

We recall the necessity of letting the code run for several time steps prior to the application of the developed formulation for the special treatment of boundary conditions. This is to accumulate representative data in order to properly compute the mean flow values. In this sense, we considered 20 initial time steps. In addition, we depart from an incompressible solution as initial condition and the penalty parameter for the weak imposition of boundary conditions has been taken as $\beta = 10$.

Let us first assess the velocity results. In Figure 6.1 we compare both isentropic compressible and incompressible developed velocity profiles. For the solution of the incompressible case, we also considered the split OSS fractional step approach, previously programmed at the beginning of this master thesis.



(a) Incompressible velocity profile



(b) Isentropic velocity profile

Figure 6.1: Velocity profile comparison in the near field region of both incompressibe (also fractional step) and isentropic compressible formulations.

It is observed that both profiles basically coincide for the near-field region. Hence, this fact induces that, in practice, the isentropic compressible finite element solver could even replace

the incompressible one when its convergence is not adequate or when the compressibility feature of the formulation wants to be taken into account. In addition to this, those results confirm the good performance of the weakly imposed conditions.

The behavior of waves when reaching the domain exterior walls is one of the main concerns in compressible flow solvers. If the boundary formulation introduced in Chapter 4 was skipped, the raw isentropic formulation would lead to the reflection of waves into the computational domain by the exterior walls. This is what we present in Figure 6.2 down below.

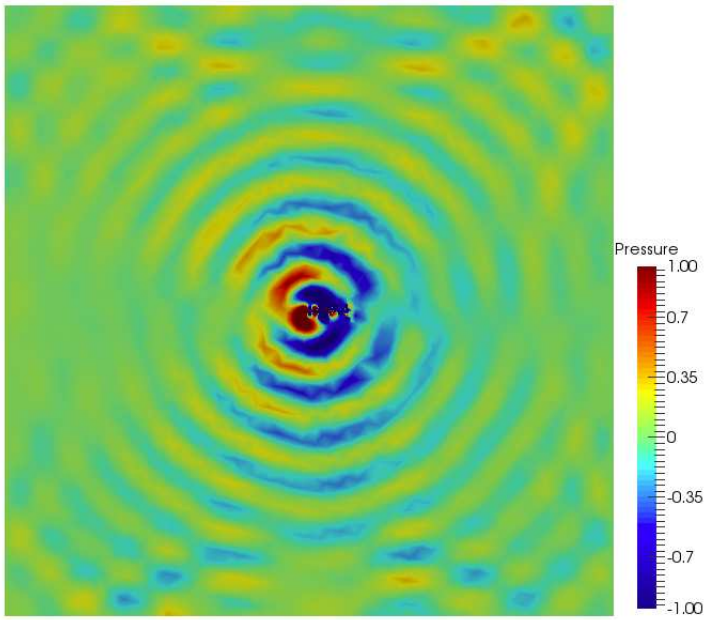
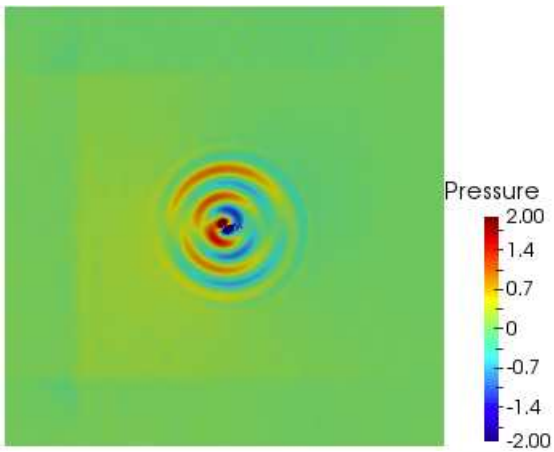
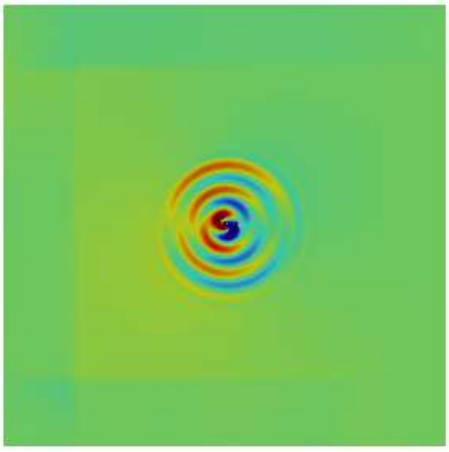


Figure 6.2: Waves being reflected by the external walls of the domain as a result of not including the especial treatment of boundary conditions. For this image we considered a way coarser mesh since the purpose was only to show the backscattering phenomena.

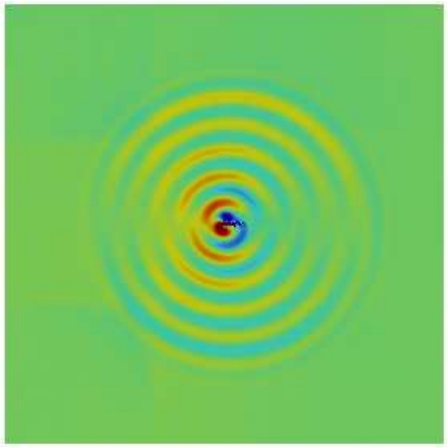
Such a problem is circumvented by the boundary formulation we reviewed and adapted previously. In Figure 6.3 we present snapshots for different time steps which validate the acoustic propagation at the far field with respect to the monolithic scheme proposed and validated in the literature [41]. Note that no waves are reflected by none of the external boundaries. It is observed how the waves evolve from the cylinder area, which causes the perturbation, and eventually leave the domain smoothly. Although the reader may notice some minor discrepancies, the results are equivalent in a reasonable manner. The differences should come from the errors introduced by the splitting approach and the approximate boundary condition. Additionally, also from the distinct diffusive character of the stabilization techniques as we proposed a term-by-term stabilization whereas the monolithc scheme considered an ASGS stabilization.



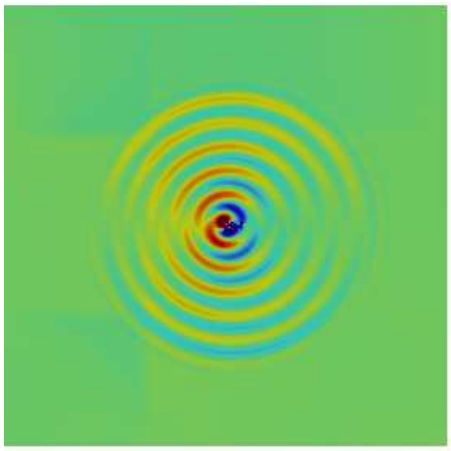
(a) Fractional step, $t = 0.2\ s$



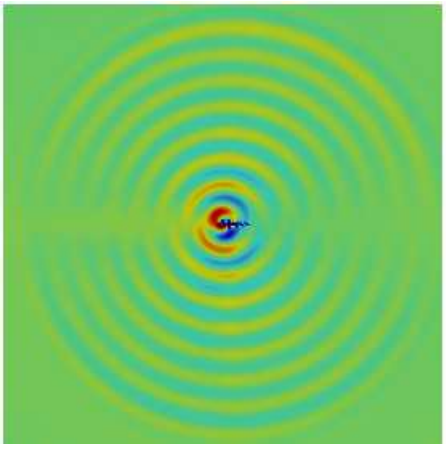
(b) Monolithic, $t = 0.2\ s$



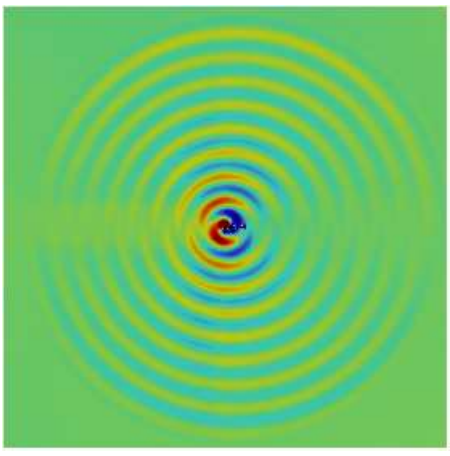
(c) Fractional step, $t = 0.4\ s$



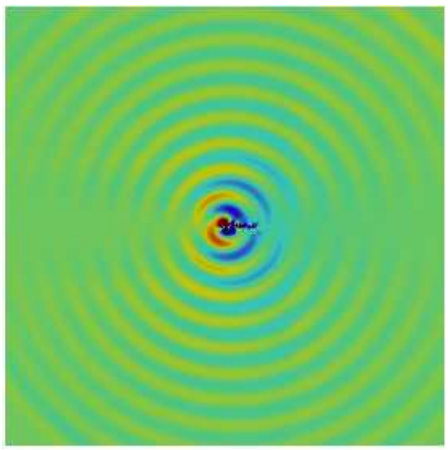
(d) Monolithic, $t = 0.4\ s$



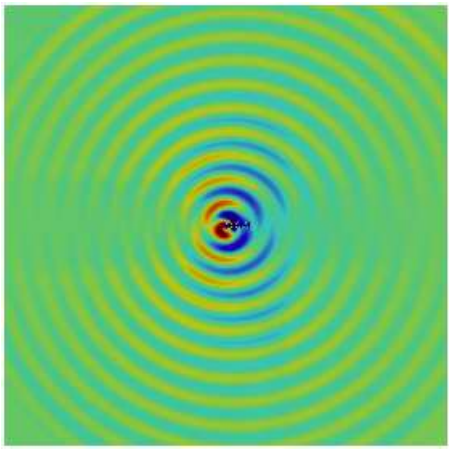
(e) Fractional step, $t = 0.6\ s$



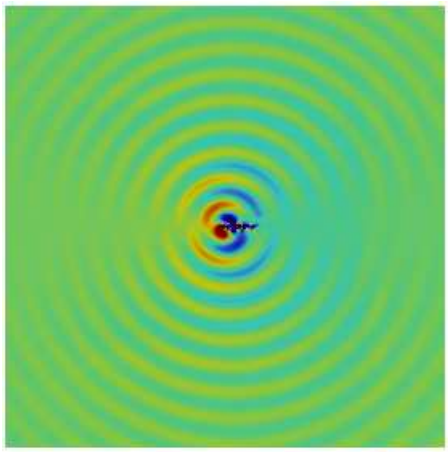
(f) Monolithic, $t = 0.6\ s$



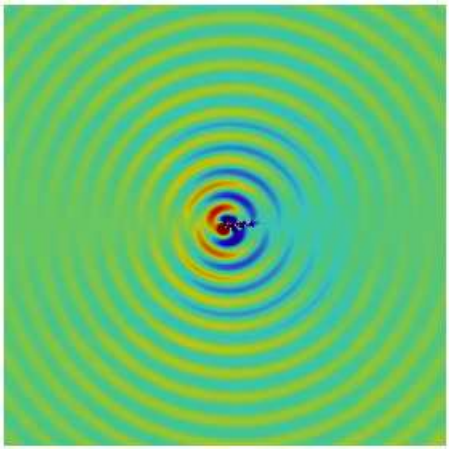
(g) Fractional step, $t = 0.8\ s$



(h) Monolithic, $t = 0.8\ s$



(i) Fractional step, $t = 1\ s$



(j) Monolithic, $t = 1\ s$

Figure 6.3: Comparison of screenshots of different time steps of the simulation, for the fractional step scheme proposed (left column) and the original monolithic scheme [41] (right column). The results are mainly equivalent although minor changes might be observed.

Chapter 7

Conclusions

In this final chapter we present the achievements and conclusions obtained during the preparation of the present work, and we state some of the possible future lines of research.

7.1 Achievements

In this masther thesis we have presented a stabilized second order fractional step scheme for the solution of the isentropic compressible Navier-Stokes problem, solved in a monolithic fashion in [40]. The algorithm has been coded following the structure of FEMUSS, an in-house parallel finite element code which allows large-scale computations using the object-oriented Fortran standards. The implementation of the method required the understanding of the object relations in a large-scale code.

Under the isentropic assumption, we have come up with a compressible formulation which is actually an extension of the incompressible Navier-Stokes problem and allows one to forget about solving the complete set of compressible equations, for some cases.

The proposed stabilization strategy was achieved by means of the split or term-by-term Orthogonal SubgridScale concept. This approach is not residual-based, and hence, it is not consistent, being consistency understood in the finite element context. The key idea behind this technique is the space where the subscales are defined, which we recall that is the orthogonal complement to the finite element space. This fact is what in practice ensures that the term-by-term formulation proposed has an optimal consistency error, as it has been shown in the literature.

One of the main challenges of the isentropic problem is to be able to avoid the backscattering of waves that the compressibility feature brings in to the problem. In this work, we have reviewed a technique already presented in [41] and have adapted it to a pressure-correction

scheme, in an algebraic manner. The reduced implementation cost when departing from a fractional step scheme for the incompressible Navier-Stokes finite element solver makes our formulation very attractive for solving aeroacoustic problems where heat transfer and shocks can be neglected.

In order to assess the correctness of our implementation we have presented a benchmark. We have validated the aeolian tones of a low Mach viscous flow over a cylinder, for $Re = 1000$ and $M = 0.0583$, where we have obtained comparable results to those for the monolitch case in [40].

7.2 Future work

Let us now briefly describe here some open research lines that could be considered.

- In the future, it might be of interest to design a third order scheme. This idea can be motivated due to the fact that the isentropic formulation showed less computational cost per iteration when compared to the full compressible and incompressible cases (see Conclusions in [41]). But one major thing needs to be taken into account for such a higher order scheme. An extrapolation of second order for the term $\mathsf{G}\hat{\mathsf{P}}_{k-1}$ ($k = 3$) is known to have an unstable behavior. Hence, a third order scheme needs to be developed by other means. One possibility might be to develop a formulation in a similar fashion as it was done in [12] for a three-field viscoelastic problem. At the base of this formulation is the fact that we *should not* do the approximation $\mathsf{K}_{\boldsymbol{u}}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} \approx \mathsf{K}_{\boldsymbol{u}}(\tilde{\mathsf{U}}^{n+1})\tilde{\mathsf{U}}^{n+1}$. This approach could be interpreted as a Yosida scheme.

- Another possibility for higher order schemes would be to change the algebraic procedure for designing the fractional step method. In this work we proposed a pressure-correction-type scheme, in which we extrapolated the pressure from the momentun equation so that to compute an intermediate velocity that would be corrected afterwards. A different technique is the socalled velocity-correction algorithm. In this approach the velocity is extrapolated first, so that to compute an intermediate pressure, with which the velocity is computed later. Finally, the pressure is corrected. This type of schemes are suitable since extrapolations for the velocity term are not unstable.

- The stabilized formulation proposed in Equation (3.17a)–(3.17b) assumed the subscales to be quasistatic. Of course, a more complete formulation would be the one including the dynamic behavior of subscales, what in practice would allow to consider anisotropic space-time discretizations.

- As we have pointed out in the introduction, one of the major facts that led to the development of the isentropic formulation is the high computational cost associated to the solution of the complete compressible Navier-Stokes problem. A technique that could lead to a remarkable reduction of this cost would be to design a fractional step algorithm for this three-field problem, (similar as in [25]).

- In the previous chapter we present a benchmark in order to asses the correctness of our implementation by direct comparison with the monolithc isentropic formulation already validated in the literature. Still, that numerical example considered a really low Mach number and, hence, it remains to test the code up to, for instance, $M = 0.75$, so that the full subsonic range is covered.

Bibliography

[1] M. Avila, J. Príncipe, and R. Codina. A finite element dynamical nonlinear subscale approximation for the low Mach number flow equations. *Journal of Computational Physics*, 230:7988–8009, 2011.

[2] S. Badia and R. Codina. Algebraic pressure segregation methods for the incompressible Navier–Stokes equations. *Archives of Computational Methods in Engineering*, 15(3): 1–52, 2007.

[3] S. Badia and R. Codina. On a multiscale approach to the transient Stokes problem: Dynamic subscales and anisotropic space–time discretization. *Applied Mathematics and Computation.*, 207(2):415 – 433, 2009.

[4] C. Bayona, J. Baiges, and R. Codina. Variational multi-scale finite element solution of the compressible Navier–Stokes equations. *International Journal of Numerical Methods for Heat and Fluid Flow*, 26:1240–1271, 2015.

[5] C. Bayona, J. Baiges, and R. Codina. Solution of low Mach number aeroacoustic flows using a Variational Multi-Scale Finite Element formulation of the compressible Navier–Stokes equations written in primitive variables. *Submitted*, 2018.

[6] J.P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114:185 – 200, 1994.

[7] J. Blasco and R. Codina. Space and time error estimates for a first order, pressure stabilized finite element method for incompressible Navier–Stokes equations. *Applied Numerical Mathematics.*, 38:475 – 497, 2004.

[8] P.B. Bochev, M.D. Gunzburger, and R.B. Lehoucq. On stabilized finite element methods

for the Stokes problem in the small time step limit. *International Journal for Numerical Methods in Fluids.*, 53(4):573 – 597, 2007.

[9] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods.* Springer, 1993.

[10] F. Brezzi and M. Fortin. Mixed and hybrid finite element methods. *Springer Verlag*, 1991.

[11] A.N. Brooks and T.R.J. Hughes. Streamline upwind / Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[12] E. Castillo and R. Codina. First, second and third order fractional step methods for the three-field viscoelastic flow problem. *Journal of Computational Physics*, 296:113–137, 2015.

[13] E. Castillo and R. Codina. Dynamic term-by-term stabilized finite element formulation using orthogonal subgrid scales for the navier-stokes problem. *Submitted*, 2018.

[14] A. J. Chorin. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics*, 2:12–26, 1967.

[15] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Computational Methods in Applied Mechanics and Engineering*, 190(2001): 2681–2706, 2000.

[16] R. Codina. On stabilized finite element methods for linear systems of convection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 188(1):61–82, 2000.

[17] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computational Methods in Applied Mechanics and Engineering*, 191:4295–4321, 2002.

[18] R. Codina. Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales. *Applied Numerical Mathematics.*, 58:264 – 283, 2008.

[19] R. Codina and J. Baiges. Approximate imposition of boundary conditions in immersed boudary methods. *International Journal for Numerical Methods in Engineering.*, 80(11): 1379 – 1405, 2009.

[20] R. Codina and J. Blasco. Analysis of a pressure–stabilized finite element approximation of the stationary Navier–Stokes equations. *Numerische Mathematik.*, 87:59 – 81, 2000.

[21] R. Codina and J. Blasco. Analysis of a stabilized finite element approximation of the transient convection–diffusion–reaction equation using orthogonal subscales. *Computing and Visualization in Science.*, 4:167 – 174, 2002.

[22] R. Codina and J. Blasco. Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales. *Computing and Visualization in Science.*, 4:167 – 174, 2002.

[23] R. Codina and A. Folch. A stabilized finite element predictor–corrector scheme for the incompressible Navier–Stokes equations using a nodal–based implementation. *International Journal for Numerical Methods in Fluids*, 44:483–503, 2004.

[24] R. Codina and O. Soto. Approximation of the incompressible Navier–Stokes equations using orthogonal subscale stabilization and pressure segregation on anisotropic finite element meshes. *Computer Methods in Applied Mechanics and Engineering.*, 193:1403 – 1419, 2004.

[25] R. Codina, M. Vázquez, and O.C. Zienkiewicz. A general algorithm for the compressible and incompressible flows. Part III: the semi–implicit form. *International Journal for Numerical Methods in Fluids*, 27:13–32, 1996.

[26] R. Codina, S. Badia, J. Baiges, and J. Príncipe. Variational Multiscale Methods in Computational Fluid Dynamics. *Encyclopedia of Computational Mechanics*, John Wiley and Sons Ltd., to appear.

[27] J. Donea. A Taylor–Galerkin method for convection transport problems. *International Journal for Numerical Methods in Engineering.*, 20:101–119, 1984.

[28] J. Douglas and T. Russel. Numerical methods for convection dominated problems based on combining the method of characteristics with finite elements or finite difference procedures. *SIAM Journal on Numerical Analysis.*, 19:871–885, 1982.

[29] H. Espinoza, R. Codina, and S. Badia. A Sommerfeld non-reflecting boundary condition for the wave equation in mixed form. *Computer Methods in Applied Mechanics and Engineering.*, 276:122 – 148, 2014.

[30] P. Fosso, H. Deniau, N. Lamarque, and T. Poinsot. Comparison of outflow boundary conditions for subsonic aeroacoustic simulations. *International Journal of Numerical Methods in Fluids*, 68(10):1207 – 1233, 2012.

[31] V. Granet, O. Vermorel, T. Léonard, I. Gicquel, and T. Poinsot. Comparison of nonreflecting outlet boundary conditions for compressible solver on unstructured grids. *AIAA*, 48(10):2348 – 2364, 2010.

[32] O. Guasch and R. Codina. An algebraic subgrid scale finite element method for the convected helmholtz equation in two dimensions with applications in aeroacoustics. *Computer Methods in Applied Mechanics and Engineering.*, 196(45):4672 – 4689, 2007.

[33] F.Q. Hu. A perfectly matched layer absorbing boundary condition for linearized Euler equations with a non–uniform mean flow. *Journal of Computational Physics*, 208:469 – 492, 2005.

[34] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.B. Quincy. The variational multiscale method–a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering.*, 166:3–24, 1998.

[35] T.J.R. Hughes, G. Scovazzi, and L.P. Franca. Multiscale and stabilized methods. *Encyclopedia of Computational Mechanics*, Wiley, 2004.

[36] T.R.J. Hughes. Multiscale phenomena: Green's function, the Dirichlet–to–Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering.*, 127:387–401, 1995.

[37] T.R.J. Hughes and A.N. Brooks. A multidimensional upwind scheme with no cross–wind diffusion. *FEM for Convection Dominated Flows.*, ASME, New York, 1979.

[38] T.R.J. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin / Least–Squares method for advectiv–diffusive equations. *Computer Methods in Applied Mechanics and Engineering.*, 73: 173–189, 1989.

[39] P.H. Oosthuizen and W.E. Carscallen. *Introduction to Compressible Fluid Flow*. CRC Press, 2013.

[40] A. Pont. Numerical Simulation of Aeroacoustics using the Variational Multiscale Method. Application to the problem of human phonation. *Polytechnic University of Catalonia, Barcelona - Tech. PhD. Thesis*, 2018.

[41] A. Pont, R. Codina, J. Baiges, and O. Guasch. Unified solver for fluid dynamics and aeroacoustics in isentropic gas flows. *Journal of Computational Physics*, 363:11–29, 2018.

[42] R. Teman. Sur l'approximation de la solution des equations de Navier–Stokes par la méthode des pas fractionaires (I). *Archives for Rational Mechanics and Analysis*, 32: 135–153, 1969.

Chapter 8

Appendix

In this appendix we include the following algorithm, with the purpose of offering a more computational point of view when it comes to the implementation of the method. This is a second order fractional step scheme, where we have already introduced the expression for the extrapolation operators to be considered (first and second order). In order to ease the identification of different terms in the algorithm, we highlight the stabilization terms in red, and those imposing the boundary conditions in blue.

Algorithm 8.1 Second order fractional step scheme for the isentropic compressible Navier-Stokes problem

Read (or compute) the initial values, $\boldsymbol{u}_h^1$ and p_h^1 .

FOR $n = 1, \ldots, N$ DO:

1. Solve the problem for the intermediate velocity, $\tilde{\boldsymbol{u}}_h^{n+1,i+1}$.

- Set $i = 1$
- Set $\tilde{\boldsymbol{u}}_h^{n+1,1} = \boldsymbol{u}_h^n$
- WHILE (not converged) DO:
  - $\diamond$ $i+1 \leftarrow i$
  - $\diamond$ Compute (at each integration point) $\tau_{1,K}$ and $\tau_{2,K}$ from Equations (3.12a) and (3.12b) and ρ from (5.12).
  - $\diamond$ Compute contributions to elemental matrix

$$\left(\boldsymbol{v}_h, \frac{\rho}{\gamma_2 \delta t}\tilde{\boldsymbol{u}}_h^{n+1,i+1}\right) + (\boldsymbol{v}_h, \rho\tilde{\boldsymbol{u}}_h^{n+1,i} \cdot \boldsymbol{\nabla}\tilde{\boldsymbol{u}}_h^{n+1,i+1}) + \mu(\boldsymbol{\nabla}\boldsymbol{v}_h, \boldsymbol{\nabla}\tilde{\boldsymbol{u}}_h^{n+1,i+1})$$
$$+ \frac{\mu}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_h, \boldsymbol{\nabla}\cdot\tilde{\boldsymbol{u}}_h^{n+1,i+1}) + \sum_{K\in\mathcal{T}_h} \left\langle \rho\tilde{\boldsymbol{u}}_h^{n+1,i} \cdot \boldsymbol{\nabla}\boldsymbol{v}_h, \tau_{1,K}\rho\tilde{\boldsymbol{u}}_h^{n+1,i} \cdot \boldsymbol{\nabla}\tilde{\boldsymbol{u}}_h^{n+1,i+1} \right\rangle_K$$

$$
\begin{aligned}
&+\sum_{K\in\mathcal{T}_h}\left\langle \nabla\cdot\boldsymbol{v}_h, \tau_{2,K}\nabla\cdot\tilde{\boldsymbol{u}}_h^{n+1,i+1}\right\rangle_K\\
&-\frac{\mu}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}(\nabla\tilde{\boldsymbol{u}}_h^{n+1})\right\rangle_{\Gamma_L}-\frac{\mu}{6N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}(\nabla\cdot\tilde{\boldsymbol{u}}_h^{n+1})\right\rangle_{\Gamma_L}\\
&-\frac{\mu}{2N_w}\left\langle \tilde{\boldsymbol{u}}_h^{n+1}, \boldsymbol{n}(\nabla\boldsymbol{v}_h)\right\rangle_{\Gamma_L}-\frac{\mu}{6N_w}\left\langle \tilde{\boldsymbol{u}}_h^{n+1}, \boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L}\\
&+\left(1-\frac{1}{2N_w}\right)\left\langle \rho c\boldsymbol{v}_h\cdot\boldsymbol{n}, \tilde{\boldsymbol{u}}_h^{n+1}\cdot\boldsymbol{n}\right\rangle_{\Gamma_L\cup\Gamma_O}+\frac{\beta}{2N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{v}_h, \tilde{\boldsymbol{u}}_h^{n+1}\right\rangle_{\Gamma_L}
\end{aligned}
$$

◇ Compute contribution to right hand side

$$
\begin{aligned}
&\left(\boldsymbol{v}_h, \frac{\rho}{\gamma_2\delta t}\sum_{i=0}^{1}\alpha^i\boldsymbol{u}^{n-i}\right)+(\nabla\cdot\boldsymbol{v}_h, p^n)\\
&+\sum_{K\in\mathcal{T}_h}\left\langle \rho\tilde{\boldsymbol{u}}_h^{n+1,i}\cdot\nabla\boldsymbol{v}_h, \tau_{1,K}\Pi_h\left[\rho\boldsymbol{u}_h^n\cdot\nabla\boldsymbol{u}_h^n\right]\right\rangle_K+\sum_{K\in\mathcal{T}_h}\left\langle \nabla\cdot\boldsymbol{v}_h, \tau_{2,K}\Pi_h\left[\nabla\cdot\boldsymbol{u}_h^n\right]\right\rangle_K\\
&-\frac{1}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}(2p_h^n-p_h^{n-1})\right\rangle_{\Gamma_L}+\left\langle \boldsymbol{v}_h, \boldsymbol{u}_L^{n+1}\right\rangle_{\Gamma_L}\\
&-\mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}(\nabla\boldsymbol{v}_h)\right\rangle_{\Gamma_L}-\frac{\mu}{3}\mu\left\langle \boldsymbol{u}_L^{n+1}, \boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L}\\
&-\frac{1}{N_w}\left\langle \boldsymbol{v}_h, \sum_{k=n-N_w+2}^{n}\left[\boldsymbol{n}p_h^k-\mu\boldsymbol{n}(\nabla\boldsymbol{u}_h^k)-\frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{u}_h^k)\right]\right\rangle_{\Gamma_L}\\
&-\frac{1}{2N_w}\left\langle \boldsymbol{v}_h, \boldsymbol{n}p_h^{n-N_w+1}-\mu\boldsymbol{n}(\nabla\boldsymbol{u}_h^{n-N_w+1})-\frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{u}_h^{n-N_w+1})\right\rangle_{\Gamma_L}\\
&+\frac{1}{N_w}\left\langle \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k, \mu\boldsymbol{n}(\nabla\boldsymbol{v}_h)+\frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L}\\
&+\frac{1}{2N_w}\left\langle \boldsymbol{u}_h^{n-N_w+1}, \mu\boldsymbol{n}(\nabla\boldsymbol{v}_h)+\frac{\mu}{3}\boldsymbol{n}(\nabla\cdot\boldsymbol{v}_h)\right\rangle_{\Gamma_L}\\
&+\frac{1}{N_w}\left\langle \rho c\boldsymbol{v}_h\cdot\boldsymbol{n}, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k\cdot\boldsymbol{n}+\frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\cdot\boldsymbol{n}\right\rangle_{\Gamma_L\cup\Gamma_O}\\
&-\frac{\beta}{N_w}\frac{\mu_p}{l_p}\left\langle \boldsymbol{v}_h, \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k+\frac{1}{2}\boldsymbol{u}_h^{n-N_w+1}\right\rangle_{\Gamma_L}
\end{aligned}
$$

◇ Assembly and call solver

◇ Check convergence

- END WHILE
- Store converged value, $\boldsymbol{u}_h^{n+1,i+1}$ to solve next steps.

2. *Solve problem for pressure,* $p^{n+1,i+1}$

- As previously, compute (at each integration point) $\tau_{1,K}$, $\tau_{2,K}$, ρ and now also c^2 from Equation (5.13).
- Compute contributions to elemental matrix

$$\left(q_h, \frac{1}{c^2\rho\gamma_2\delta t} p_h^{n+1,i+1}\right) + \left(q_h, \frac{1}{c^2\rho}\tilde{\boldsymbol{u}}_h^{n+1,i+1}\cdot\boldsymbol{\nabla} p_h^{n+1,i+1}\right) + \gamma_2\delta t\left(\boldsymbol{\nabla} q_h, \frac{1}{\rho}\nabla p_h^{n+1,i+1}\right)$$
$$+ \sum_{K\in\mathscr{T}_h}\left\langle \frac{1}{c^2\rho}\tilde{\boldsymbol{u}}_h^{n+1,i+1}\cdot\boldsymbol{\nabla} q_h, \tau_{2,K}\frac{1}{c^2\rho}\tilde{\boldsymbol{u}}_h^{n+1,i+1}\cdot\boldsymbol{\nabla} p_h^{n+1,i+1}\right\rangle_K + \sum_{K\in\mathscr{T}_h}\left\langle \boldsymbol{\nabla} q_h, \tau_{1,K}\nabla p_h^{n+1,i+1}\right\rangle_K$$

- Compute contribution to right hand side

$$\left(q_h, \frac{1}{c^2\rho\gamma_2\delta t}\sum_{i=0}^{1}\alpha^i p^{n-i}\right) + \gamma_2\delta t\left(\boldsymbol{\nabla} q_h, \frac{1}{\rho}\nabla p_h^n\right) - \left(q_h, \nabla\cdot\tilde{\boldsymbol{u}}_h^{n+1,i+1}\right)$$
$$+ \sum_{K\in\mathscr{T}_h}\left\langle \frac{1}{c^2\rho}\tilde{\boldsymbol{u}}_h^{n+1,i}\cdot\boldsymbol{\nabla} q_h, \tau_{2,K}\Pi_h\left[\frac{1}{c^2\rho}\boldsymbol{u}_h^n\cdot\boldsymbol{\nabla} p_h^n\right]\right\rangle_K + \sum_{K\in\mathscr{T}_h}\left\langle \boldsymbol{\nabla} q_h, \tau_{1,K}\Pi_h\left[\boldsymbol{\nabla} p_h^n\right]\right\rangle_K$$
$$- \frac{1}{2N_w}\left\langle (2\boldsymbol{u}_h^n - \boldsymbol{u}_h^{n-1}), \boldsymbol{n} q_h\right\rangle_{\Gamma_L} + \left\langle \boldsymbol{u}_L, \boldsymbol{n} q_h\right\rangle_{\Gamma_L}$$
$$- \frac{1}{2N_w}\left\langle \boldsymbol{u}_h^{n-N_w+1}, \boldsymbol{n} q_h\right\rangle_{\Gamma_L} - \frac{1}{N_w}\left\langle \sum_{k=n-N_w+2}^{n}\boldsymbol{u}_h^k, \boldsymbol{n} q_h\right\rangle_{\Gamma_L}$$

- Assembly and call solver
- Store $p_h^{n+1,i+1}$

3. *Correction step. Solve problem for end-of-step velocity,* $\boldsymbol{u}_h^{n+1,i+1}$
   - Compute ρ at each integration point
   - Contribution to elemental matrix

$$\left(\boldsymbol{v}_h, \frac{\rho}{\gamma_2\delta t}\boldsymbol{u}_h^{n+1,i+1}\right)$$

   - Contribution to right hand side

$$\left(\boldsymbol{v}_h, \frac{\rho}{\gamma_2\delta t}\tilde{\boldsymbol{u}}_h^{n+1,i+1}\right) + \left(\boldsymbol{\nabla}\cdot\boldsymbol{v}_h, p_h^{n+1,i+1} - p_h^n\right)$$

   - Assembly and call solver
   - Store end of step velocity $\boldsymbol{u}_h^{n+1,i+1}$

→ Compute and store the projections for next time step:

$$\Pi_h\left[\nabla p_h^{n+1,i+1}\right] = (\boldsymbol{v}_h, \nabla p_h^{n+1,i+1})$$

$$\Pi_h\left[\rho \boldsymbol{u}_h^{n+1,i+1} \cdot \nabla \boldsymbol{u}_h^{n+1,i+1}\right] = (\boldsymbol{v}_h, \rho \boldsymbol{u}_h^{n+1,i+1} \cdot \nabla \boldsymbol{u}_h^{n+1,i+1})$$

$$\Pi_h\left[\frac{1}{c^2\rho}\boldsymbol{u}_h^{n+1,i+1} \cdot \nabla p_h^{n+1,i+1}\right] = \left(q_h, \frac{1}{c^2\rho}\boldsymbol{u}_h^{n+1,i+1} \cdot \nabla p_h^{n+1,i+1}\right)$$

$$\Pi_h\left[\nabla \cdot \boldsymbol{u}_h^{n+1,i+1}\right] = (q_h, \nabla \cdot \boldsymbol{u}_h^{n+1,i+1})$$

→ Set up values for next time step, $\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^{n+1,i+1}$ and $p_h^{n+1} = p_h^{n+1,i+1}$
END DO