

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
Ingegneria Energetica, Nucleare e del Controllo Ambientale
CICLO XXII

SETTORE SCIENTIFICO-DISCIPLINARE DI AFFERENZA: ING/IND-19 IMPIANTI NUCLEARI

**A THREE-DIMENSIONAL FEM SOLVER OF
THE NAVIER-STOKES EQUATIONS WITH
APPLICATIONS TO TWO-PHASE FLOWS AND
INNOVATIVE NUCLEAR REACTOR CONCEPTS**

Presentata da ANTONIO CERVONE

Coordinatore Dottorato

Chiar.mo Prof. Ing. ANTONIO BARLETTA

Relatore

Chiar.mo Prof. Ing. RUBEN SCARDOVELLI

Esame finale anno 2010

Contents

. Introduction	1
I. Mathematical Model	5
1. Variational formulation of the Navier-Stokes problem	7
1.1. Function space notation	7
1.1.1. Integrable spaces $L^p(\Omega)$	7
1.1.2. Sobolev spaces $H^k(\Omega)$	8
1.2. The Stokes problem	9
1.2.1. Stokes variational equation	9
1.2.2. Boundary conditions	10
1.3. The Navier-Stokes problem	12
1.3.1. The steady-state problem	12
1.3.2. The time-dependent Navier-Stokes problem	12
2. Finite element approximation	15
2.1. The Galerkin method approximation	15
2.2. The Finite Element Method	17
2.2.1. One-dimensional case	17
2.2.2. Two-dimensional case	20
2.2.3. Three-dimensional case	25
2.3. Error estimation	27
2.3.1. Notation	27
2.3.2. Energy (H^1) norm error estimation	27
2.3.3. L^2 norm error estimation	29
3. Numerical solution	31
3.1. Discretization of the Navier-Stokes equations	31
3.1.1. Temporal derivative discretization	32
3.1.2. Algebraic formulation of the Navier-Stokes equations	33
3.2. Coupled solvers	34
3.2.1. Domain Decomposition Methods	35
3.2.2. Krylov subspace methods	36
3.2.3. Preconditioning	38
3.3. Segregated (split) solvers	39
3.3.1. The incremental pressure-correction scheme	40

3.3.2.	The rotational incremental pressure-correction scheme	41
3.3.3.	The penalty method	42
3.4.	Multigrid algorithm	44
3.4.1.	Multigrid cycles	45
3.4.2.	Prolongation and restriction	46
II.	Two-Phase Flow	51
4.	Two-phase flow and interface capturing	53
4.1.	Single fluid formulation of the Navier-Stokes equations for two-phase flow	53
4.2.	Volume-of-Fluid (VOF) method for interface capturing	55
4.2.1.	Interface reconstruction	56
4.2.2.	Interface advection	59
4.3.	Multilevel VOF method	63
4.3.1.	Numerical implementation of the multilevel VOF method	66
4.3.2.	Velocity refinement with an optimal control approach	68
4.4.	Surface tension modeling	70
4.4.1.	Numerical implementation of the capillary force	71
5.	Numerical simulations	73
5.1.	Linear dispersion equation for jets	73
5.1.1.	Analytical dispersion equation	73
5.1.2.	Numerical results	76
5.2.	Axisymmetric pulsating jets	78
III.	Nuclear Reactors Thermal-Hydraulics	91
6.	Porous media approach for nuclear reactor cores	93
6.1.	Energy equation	93
6.2.	Porous media approach	94
6.2.1.	Two-level finite element Navier-Stokes system	94
6.2.2.	Transfer operator modeling	95
6.3.	Numerical simulations	97
6.3.1.	Boundary conditions	97
6.3.2.	Thermophysical properties of liquid metals	98
6.3.3.	Working conditions	98
6.3.4.	Simulations of an open core reactor	99
.	Conclusions	105
	Bibliography	107

List of Figures

2.1.	Hat function for the interval K_i	18
2.2.	Linear shape functions on the EDGE2 canonical element.	19
2.3.	Quadratic shape functions on the EDGE3 canonical element.	20
2.4.	TRI3 (left) and TRI6 (right) canonical elements.	22
2.5.	QUAD4 (left) and QUAD9 (right) canonical elements.	23
2.6.	Shape functions on the QUAD9 canonical element: N_{01} (top left), N_{21} (top right), N_{22} (bottom).	24
2.7.	TET4 (left) and TET10 (right) canonical elements.	25
2.8.	HEX8 (left) and HEX27 (right) canonical elements.	26
3.1.	Left: Vanka subdomain made of 3×3 elements. Right: Vanka subdomain made of 2×2 elements. Unknowns are localized on the grid (\circ : velocity, \square : pressure).	36
3.2.	V-cycle multigrid scheme.	46
3.3.	W-cycle multigrid scheme.	47
3.4.	Nested cycle multigrid scheme.	48
4.1.	Computational domain. The liquid in Ω_l is the reference phase and the boundary between the two phases is denoted by Γ_s . Dirichlet boundary conditions are applied on Γ_d and Neumann boundary conditions on Γ_n	53
4.2.	Cell stencil used for the reconstruction.	56
4.3.	The reference phase occupies the area of the pentagon $ABFGD$	58
4.4.	Eulerian implicit method: (a) SLIC reconstruction of the interface, (b) implicit step fluxes.	60
4.5.	Lagrangian explicit method: (a) SLIC reconstruction of the interface, (b) final configuration.	60
4.6.	geometrical unsplit method: (a) initial configuration; (b) implicit step; (c) explicit step.	61
4.7.	Pre-image of the grid lines (dashed lines) and streamline through the cell vertices (dotted lines). The area of the polygon 1-2- B - D - A is equal to Φ_x^+ , that of 1-3- C - E - A to Φ_y^+ and the continuous piecewise-linear line B - D - A - E - C approximates the pre-image of the cell sides 1-2 and 1-3.	62
4.8.	Three different shapes of the flux polygon through the right side of a cell: (a) standard case with the flux polygon inside two consecutive vertical cells; (b) flux polygon defined across two horizontal cells; (c) polygon intersecting three different cells.	63

List of Figures

4.9.	The color function distribution on different meshes (top left) and on the coarse mesh (top right). The compact data memorization with two (bottom left) and four (bottom right) levels of grid refinement.	66
4.10.	The C data distribution on a 5×5 Cartesian mesh (top) and the compressed stored data (bottom): row number, number of cells n_c , color function in the mixed and consecutive full cells and column position.	67
4.11.	The coarse element with four nodes (left) and the refined one with nine nodes (right).	69
5.1.	Initial axisymmetric geometry.	75
5.2.	Time evolution of the growth rate α_r for the $m = 0$ mode during the initial transient (left) and after the restart of the computation (right). . .	75
5.3.	Numerical growth rate α for $\mu_l/\mu_g = 100$ for different mesh resolutions on the left (32 (A) , 64 (B), 128 (C)) and for a different number of refinement levels on the right ($f = c + 2$ (A), $f = c + 3$ (B), $f = c + 4$ (C)). In both graphs it is also plotted the inviscid theoretical curve of (5.7) (D).	76
5.4.	Numerical growth rate α for different viscosity ratios μ_l/μ_g (10 (A), 100 (B) and 1000 (C)). In the graph it is also plotted the inviscid theoretical curve of (5.7) (D).	77
5.5.	Numerical (+) and theoretical dispersion for the growth rate for $Oh = 0.205$ (left) and $Oh = 0.065$ (right).	78
5.6.	Evolution of the interface line for the three different grid resolutions: 32×192 (A), 64×384 (B) and 128×768 (C), at the non-dimensional times $t = 0.5, 1, 2, 3$	80
5.7.	Interface evolution (left) and pressure, color function and velocity profiles along the jet axis (right, lines A, B and C, respectively) at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (top to bottom).	82
5.8.	Jet evolution for the intermediate wave length λ_2 at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (left to right and top to bottom).	83
5.9.	Jet evolution for the shortest wave length λ_1 at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (left to right and top to bottom).	84
5.10.	Jet evolution for the longest wave length λ_3 at times 0.5, 1, 2, 2.5, 3, 3.5, 4, 5, 6, 7.5 (left to right and top to bottom).	85
5.11.	Jet evolution with inlet velocity $U_0 = 10 \text{ m/s}$ at times 1, 2, 3.5, 4, 4.5 (two different views in each row).	87
5.12.	Jet evolution with inlet velocity $U_0 = 100 \text{ m/s}$ at times 1.5, 2, 2.5, 3, 3.5 (two different views in each row).	88
6.1.	Full reactor (left) and computational domain (right).	97
6.2.	Power distribution. Each generating element is identified by its peak factor, all others are non generating elements.	99
6.3.	Velocity (left) and pressure (right) distribution on different planes: $z = 2$ (top), $y = 0$ (middle), $x = 0$ (bottom).	100

6.4. Density (left) and temperature (right) distribution on different planes:
 $z = 2$ (top), $y = 0$ (middle), $x = 0$ (bottom)... 101

6.5. Comparison between variable (A) and constant (B) physical properties
with respect to temperature: velocity and pressure (top), density and
temperature (bottom) on the line $x = y = 1.1$ 102

6.6. Comparison between variable (A) and constant (B) physical properties
with respect to temperature: velocity and temperature profiles on the line
 $y = x, z = 2$ 102

List of Tables

5.1. The maximum value of $\ S_c^{fc}\ _\infty/\ \mathbf{f}_s\ _\infty$ for the coarse grid with 32×192 cells, four different refinement levels, in the non-dimensional time interval $[0, 2]$	79
6.1. Geometrical and physical properties at $T=673.15 K$ in SI units.	98

Introduction

Nowadays the development of computational codes and the use of numerical simulations have become a common tool in research and innovation fields, given the possibility to explore different solutions at a reasonable cost when compared to experimental facilities. Project work-flows today involve numerical simulations to the point that computer assisted design is a required step and not an auxiliary tool as in the past.

Thermal-fluid-dynamics is one of the fields in which the integration of different computational tools has the greatest diffusion, since many of the applications that fall in this area of research and industrial design would require extensive and expensive experimental counterparts that can be afforded in a very limited number of cases. In this work isothermal two-phase flows and the thermal-hydraulics of the nuclear reactor core are explored.

The two-phase flow simulations that are considered in this thesis focus mainly on jets. These phenomena appear widely in natural processes, from atomic scales to galaxy proportions, and in particular in many engineering applications, such as heat removal or fuel injection. The dynamics of a jet is a very complex process that can evolve in many different ways, such as the formation of a cloud of small droplets, as in a combustion chamber of a diesel/gasoline engine, a sheet-like structure, that can occur in agricultural sewage and irrigation, or a column of fluid that evolves in a line of drops. The physical modeling of this wide range of phenomena is complex and still not completely resolved, especially for the detailed mechanism of breakup.

Nuclear reactor design has historically been one of the first field to massively employ computational tools and computers, since the difficulty to create test facilities that reproduce real operation conditions. The evolution of computers has not seen an equivalent rate of advance in nuclear reactor codes, but the trend is changing. More powerful, reliable and complex codes are required to design the new generation of nuclear power plants. The thermal-hydraulics of the core involves many processes with complex modeling, but the greatest challenge remains the impossibility to perform the simulation of the whole core in a three-dimensional geometry with adequate resolution. The number of components and physical processes that must be described overwhelms the present computational power.

This thesis is devoted to the analysis of an innovative code that has been developed by the research group under the supervision of professors S. Manservigi and R. Scardovelli to deal with this kind of applications. The fundamental idea has been to create a general and modular tool, where the solver of a coupled system of partial differential equations for diffusion/advection problems was the backbone upon which all other components rely. Generality means the possibility to cover a wide range of physical and engineering applications, including the two already mentioned. Modularity means the ability to load

dynamically different features suited for a particular application and not the whole system. This has led to a computational tool that looks like a single-purpose application when needed, but which is also able to easily manage the coupling among different phenomena, implemented in different software modules, when the physics requires it.

The development of this general and modular software has required the complete redesign of the very first finite element code written by E. Aulisa, now on tenure track at Texas Tech, and S. Manservigi. The code development is always in progress and has recently seen the introduction of a series of new modules, that will not be discussed in this thesis, that are relevant to magneto-hydrodynamics, turbulence, neutronics and fluid-structure interaction. The very first modules have been developed for the solution of the incompressible Navier-Stokes equations and the energy equation. Direct numerical simulations of two-phase flows are performed with the single fluid formulation of the equations together with a few routines to capture the evolution of the interface and to compute its geometrical properties.

The Finite Element Method (FEM) has been chosen for the solution of the system of partial differential equations, given its mathematical properties and points of strength, and it is described in the first part of the thesis. The first chapter describes with some detail the weak formulation of the Navier-Stokes equations and the basic analytical results that provide the mathematical framework of the successive work.

The second chapter describes the finite element method, with the introduction of the discretization spaces and the discretized elements. The Lagrangian elements for computational domains in one, two and three dimensions are depicted. Some theoretical results on the error of the finite element approximations close the chapter.

The third chapter is devoted to the numerical solution of the discretized problem that has been introduced in the first two chapters. A short introduction to the algorithms that are used for the solution of sparse linear systems is given here, with particular emphasis to the specific algorithms that are used in the numerical solution of the Navier-Stokes equations.

The second part of the thesis deals with the two-phase flow simulations. The fourth chapter describes the Volume-of-Fluid (VOF) method, that is one of the most popular techniques for interface capturing. This method describes the interface through the color function that represents the fraction of each cell that is occupied by the reference phase. The geometrical properties of the interface, such as the normal and the curvature, are derived from this color function usually with finite difference schemes. When the velocity field is divergence-free, the mass can be accurately conserved by the VOF method, differently from other approaches such as the Level Set method, for example. Another feature of the VOF approach resides in the automatic handling of topology changes. This aspect can be a serious limitation if not carefully handled, therefore we introduce the multilevel VOF method, that artificially separates the grid on which the velocity and pressure fields are computed from the grid where the interface is modeled. This technique enables us to use higher resolutions in VOF reconstruction, giving us more control on the scale at which the breakup events occur.

In the fifth chapter the results of the jet simulations are presented. In the first part of the chapter the code is tested against the analytical results of the stability of a column

of fluid, while in the second part the results of the simulations are presented that are relevant to liquid jets with different physical parameters, such as the wavelength of the applied disturbance or the magnitude of the velocity at the injector nozzle.

The third part of the thesis includes only the sixth chapter and shows the approach that has been adopted in innovative nuclear reactor design problems. The numerical algorithm that has been developed follows the so-called porous approach, which widely adopted in this field to overcome the impossibility to describe with enough resolution the components inside a reactor core. This technique is similar to that one introduced in the multilevel VOF, where the simulation is made on two different scales, a coarse scale that covers the entire core, and a fine one where the components are reproduced correctly, but with a spatial extension limited to a subchannel of the reactor core. Some results obtained for a new generation of liquid metal cooled reactors are shown to illustrate the main features of the model.

The last chapter is a short summary of the thesis, where the conclusions and future perspectives are outlined.

Part I.

Mathematical Model

1. Variational formulation of the Navier-Stokes problem

In this chapter we introduce the variational formulation of the Navier-Stokes equations and recall some analytical results on existence and uniqueness of solutions [1, 2, 3, 4].

1.1. Function space notation

1.1.1. Integrable spaces $L^p(\Omega)$

Let Ω be an open set in \mathbb{R}^n with locally Lipschitz boundary $\Gamma = \partial\Omega$ and $f : \Omega \mapsto \mathbb{R}$ be a real-valued or a component of a vector-valued function defined on Ω . We define the space $L^p(\Omega)$, $1 \leq p < \infty$ as the set of all functions that satisfy

$$\|f\|_p := \left(\int_{\Omega} |f|^p \, dV \right)^{1/p} < \infty, \quad (1.1)$$

where the integration is intended in Lebesgue measure. By extension, we also introduce $L^\infty(\Omega)$ as the set of all functions such as

$$\|f\|_\infty := \sup_{\Omega} |f| < \infty. \quad (1.2)$$

Both $L^p(\Omega)$ and $L^\infty(\Omega)$ are Banach spaces. The space $L^2(\Omega)$ is a Hilbert space with scalar product

$$(f, g) = \int_{\Omega} fg \, dV, \quad (1.3)$$

and corresponding norm

$$\|f\|_2 := \left(\int_{\Omega} |f|^2 \, dV \right)^{1/2}. \quad (1.4)$$

We can now introduce the Sobolev space $\mathcal{W}^{p,k}$ as the space of functions f in $L^p(\Omega)$ with $k-1$ derivatives $Df, D^2f, \dots, D^k f$ in $L^p(\Omega)$, as

$$\mathcal{W}^{p,k}(\Omega) := \left\{ f \in L^p(\Omega) : \sum_{j=0}^k \left(\int_{\Omega} |D^j f|^p \, dV \right)^{1/p} < \infty \right\}. \quad (1.5)$$

This definition introduces on these spaces a natural norm in the form

$$\|f\|_{p,k} := \sum_{j=0}^k \left(\int_{\Omega} |D^j f|^p \, dV \right)^{1/p}, \quad (1.6)$$

1. Variational formulation of the Navier-Stokes problem

and semi-norms

$$|f|_{p,k,j} := \left(\int_{\Omega} |D^j f|^p \, dV \right)^{1/p}. \quad (1.7)$$

1.1.2. Sobolev spaces $H^k(\Omega)$

The Sobolev spaces with $p = 2$ are Hilbert spaces when equipped with the scalar product

$$(f, g)_{H^k} := \sum_{j=0}^k \left(\int_{\Omega} D^j f D^j g \, dV \right)^{1/2}. \quad (1.8)$$

They will be denoted by $H^k(\Omega) := \mathcal{W}^{2,k}(\Omega)$.

Let $\mathcal{D}(\Omega)$ be the set of all infinite-derivable functions vanishing outside the compact set Ω . The closure of $\mathcal{D}(\Omega)$ in $H^k(\Omega)$ is an Hilbert space denoted usually by $H_0^k(\Omega)$. If Ω is bounded in some direction the Poincaré inequality holds

$$\|f\|_{L^2} \leq C(\Omega) \|Df\|_{L^2}, \quad \forall f \in H_0^1(\Omega), \quad (1.9)$$

where the derivative must be taken in the bounded direction.

Let γ_0 be a linear continuous operator such that

$$\gamma_0 : H^1(\Omega) \mapsto L^2(\Gamma), \quad \gamma_0 f = f|_{\Gamma}. \quad (1.10)$$

The operator γ_0 is called *trace operator* of $H^1(\Omega)$. The space $H_0^1(\Omega)$ is the kernel of γ_0 , while the image is a dense subspace of L^2 denoted by $H^{1/2}(\Gamma)$. This operator is introduced to seek solutions in H^1 when Dirichlet boundary conditions are applied. In fact the boundary conditions can be thought as trace functions. Let $H_{div}(\Omega)$ be the subset of $L^2(\Omega)$ for which $\nabla \cdot \mathbf{f} \in L^2(\Omega)$, then we can introduce the trace operator γ_n that satisfies

$$\gamma_n : H_{div}(\Omega) \mapsto H^{-1/2}(\Gamma), \quad \gamma_n \mathbf{f} = \mathbf{f} \cdot \mathbf{n}|_{\Gamma}. \quad (1.11)$$

It is possible to prove that $H^{-1/2}(\Gamma)$ is the dual space of $H^{1/2}(\Gamma)$.

Let \mathcal{V} be the set of all divergence-free functions in \mathcal{D} ,

$$\mathcal{V}(\Omega) := \{ \mathbf{f} \in \mathcal{D} : \nabla \cdot \mathbf{f} = 0 \}, \quad (1.12)$$

and $H(\Omega)$, $V(\Omega)$ be the closures of \mathcal{V} in $L^2(\Omega)$, $H_0^1(\Omega)$ respectively. It is possible to show that

$$\begin{aligned} H(\Omega) &= \{ \mathbf{f} \in L^2(\Omega) : \nabla \cdot \mathbf{f} = 0, \gamma_n \mathbf{f} = 0 \}, \\ H^\perp(\Omega) &= \{ \mathbf{f} \in L^2(\Omega) : \mathbf{f} = \nabla p, p \in H^1(\Omega) \}, \\ V(\Omega) &= \{ \mathbf{f} \in H_0^1(\Omega) : \nabla \cdot \mathbf{f} = 0 \}, \end{aligned}$$

where $H^\perp(\Omega)$ is the orthogonal complement of $H(\Omega)$ in $L^2(\Omega)$. Furthermore, it holds that

$$\begin{aligned} L^2(\Omega) &= H(\Omega) + H_1(\Omega) + H_2(\Omega), \\ H_1(\Omega) &= \left\{ \mathbf{f} \in L^2(\Omega) : \mathbf{f} = \nabla p, p \in H^1(\Omega), \nabla^2 p = 0 \right\}, \\ H_2(\Omega) &= \left\{ \mathbf{f} \in L^2(\Omega) : \mathbf{f} = \nabla p, p \in H_0^1(\Omega) \right\}. \end{aligned}$$

1.2. The Stokes problem

1.2.1. Stokes variational equation

The Navier-Stokes problem is typically described as a set of partial differential equations of convective-diffusive type. In order to simplify the notation we start to analyze the variational form of the associated Stokes problem. This implies a brief analysis of the its strong and weak form. For the Stokes problem the set of differential equation, in the steady case, reduces to

$$\begin{aligned} -\nu \nabla^2 \mathbf{u} + \nabla p &= \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega, \\ \mathbf{u} &= 0 & \text{in } \Gamma, \end{aligned} \tag{1.13}$$

where \mathbf{u} is the velocity field, ν the cinematic viscosity, p the pressure field and \mathbf{f} the external force on a given domain Ω . The boundary condition applied here is a Dirichlet homogeneous one but other type of conditions are addressed in section 1.2.2.

In this strong formulation of the problem, the velocity and the pressure fields \mathbf{u} and p require continuous second and first derivatives, respectively.

By using standard variational methods, i.e. *weighted residual method*, it is possible to convert (1.13) into a variational form in order to get weaker requirements on the smoothness of the solution. The method consists of multiplying by *test functions* and integrating the equations over all the domain Ω . In other words, we take the scalar product of (1.13) with test functions in $\mathbf{H}_0^1(\Omega)$ and $L^2(\Omega)$ and seek solutions for the projected equations. We get

$$\begin{aligned} - \int_{\Omega} \nu \nabla^2 \mathbf{u} \cdot \mathbf{v} \, dV + \int_{\Omega} \nabla p \cdot \mathbf{v} \, dV &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dV, \quad \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ \int_{\Omega} q \nabla \cdot \mathbf{u} \, dV &= 0, \quad \forall q \in L^2(\Omega). \end{aligned} \tag{1.14}$$

We can now integrate by parts and, using Green's identity, we obtain

$$\begin{aligned} \int_{\Omega} (\nu \nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) \, dV - \int_{\Gamma} \nu (\nabla \mathbf{u} \cdot \mathbf{n}) \cdot \mathbf{v} \, dS + \\ - \int_{\Omega} p \nabla \cdot \mathbf{v} \, dV + \int_{\Gamma} p \mathbf{v} \cdot \mathbf{n} \, dS &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dV, \quad \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ \int_{\Omega} q \nabla \cdot \mathbf{u} \, dV &= 0, \quad \forall q \in L^2(\Omega). \end{aligned} \tag{1.15}$$

1. Variational formulation of the Navier-Stokes problem

The integration by parts leads to two new boundary terms, that can be combined into one as

$$\int_{\Gamma} \left(p\mathbf{n} - \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \cdot \mathbf{v} \, dS. \quad (1.16)$$

Because the velocity field is known on Γ from the boundary condition, the test function \mathbf{v} must be zero on the boundary and divergence-free on the domain then (1.15) becomes

$$(\nu \nabla \mathbf{u}, \nabla \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}. \quad (1.17)$$

On the other hand, if \mathbf{u} is in \mathbf{H}_0^1 and satisfies (1.17), then we can prove that \mathbf{u} satisfies the original problem (1.13). We can therefore write that

$$\begin{aligned} &\text{there exists } p \in L^2(\Omega) \text{ such that} \\ &-\nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}, \\ &\nabla \cdot \mathbf{u} = 0, \\ &\text{with } \gamma_0 \mathbf{u} = 0. \end{aligned} \quad (1.18)$$

We can introduce the continuous bilinear form $a(\cdot, \cdot)$ as

$$a : \mathbf{V} \times \mathbf{V} \mapsto \mathbb{R}, \quad a(\mathbf{u}, \mathbf{v}) = (\nu \nabla \mathbf{u}, \nabla \mathbf{v}) \quad (1.19)$$

and the linear form \mathbf{F} as

$$\mathbf{F}(\mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad (1.20)$$

and write the problem in a more compact form

$$a(\mathbf{u}, \mathbf{v}) = \mathbf{F}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V}. \quad (1.21)$$

The form $a(\cdot, \cdot)$ is bounded (or continuous), i.e. there exists $\beta > 0$ such that

$$a(\mathbf{v}, \mathbf{v}) < \beta \|\mathbf{v}\|_{\mathbf{V}}^2, \quad \forall \mathbf{v} \in \mathbf{V}. \quad (1.22)$$

Also the form $a(\cdot, \cdot)$ is coercive, namely there exists $\alpha > 0$ such that

$$a(\mathbf{v}, \mathbf{v}) > \alpha \|\mathbf{v}\|_{\mathbf{V}}^2, \quad \forall \mathbf{v} \in \mathbf{V}. \quad (1.23)$$

These are the two conditions needed for the applicability of the *Lax-Milgram* theorem, which demonstrates that the solution of (1.21) exists and is unique for any bounded open set Ω .

1.2.2. Boundary conditions

In the previous section we have introduced only homogeneous boundary conditions but in general the physical situations that will be analyzed require more general conditions at the boundary. In order to extend the analysis to non-homogeneous boundary conditions we can split the whole boundary Γ in two regions Γ_D and Γ_N where

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_0, \quad \mathbf{u} \in \Gamma_D \subseteq \Gamma, \\ \frac{\partial \mathbf{u}}{\partial \mathbf{n}} &= 0, \quad \mathbf{u} \in \Gamma_N \subseteq \Gamma, \end{aligned} \quad (1.24)$$

with $\Gamma_D + \Gamma_N = \Gamma$. On Γ_D Dirichlet conditions are applied.

If \mathbf{u}_0 is not zero, the solution \mathbf{u} is not in $H_0^1(\Omega)$. We can, however, take the problem back to the homogeneous one with the introduction of a *lifting operator* ℓ_Ω such that

$$\ell_\Omega : H^{1/2}(\Gamma) \mapsto H^1(\Omega), \quad \gamma_0 \ell_\Omega = I. \quad (1.25)$$

We can now consider the new unknown $\tilde{\mathbf{u}} = \mathbf{u} - \ell_\Omega \mathbf{u}_0$ and rewrite (1.17) as

$$(\nu \nabla \tilde{\mathbf{u}}, \nabla \mathbf{v}) = F(\mathbf{v}) = (\mathbf{f}, \mathbf{v}) - (\nu \nabla \ell_\Omega \mathbf{u}_0, \nabla \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \quad (1.26)$$

where $F(\cdot)$ is a continuous and linear functional. It is easy to see that $\tilde{\mathbf{u}}$ now satisfies homogeneous boundary conditions on Γ_D . Existence and uniqueness of solutions can be recovered with the same hypotheses used for the homogeneous equation (1.17).

With Neumann boundary conditions applied on Γ_N , we must consider the two boundary terms from (1.15) and rewrite

$$\int_\Gamma \boldsymbol{\tau} \cdot \mathbf{v} \, dS = \int_\Gamma (\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{v} \, dS = \int_\Gamma ((pI - \nu \nabla \mathbf{u}) \cdot \mathbf{n}) \cdot \mathbf{v} \, dS, \quad (1.27)$$

where $\boldsymbol{\tau}$ is the boundary stress term and \mathbf{T} is the stress tensor. When the condition applied is of no-stress, such as symmetry conditions or out-flow regime, this term is again equal to zero. In other conditions, such as turbulent flow near a wall, the term cannot be neglected and appropriate values of the stress tensor must be applied to the equations.

If $\Gamma_D = \Gamma$, so Dirichlet boundary conditions are specified on all the boundary Γ , we can see that the pressure p is defined in (1.13) up to a constant. Therefore, if $\{\mathbf{u}, p\}$ is a solution of (1.13) or (1.17), also $\{\mathbf{u}, p + c\}$ is a solution. In order to avoid the non-uniqueness we must fix the pressure value on boundary points as

$$p(\mathbf{x}_b) = p_0, \quad \mathbf{x}_b \in \Gamma,$$

or we can set the mean pressure value to zero,

$$\int_\Omega p \, dV = (p, 1) = 0.$$

The first approach is theoretically more complex since the solution space is not a well characterized subspace. The second approach is instead viable in the weak formulation by defining a new subspace. We define $L_0^2(\Omega)$ the space such that

$$L_0^2(\Omega) = \left\{ p \in L^2(\Omega) : (p, 1) = 0 \right\}. \quad (1.28)$$

When Γ_N is not empty, the pressure is defined by the boundary condition since no derivatives appear in (1.27).

1.3. The Navier-Stokes problem

1.3.1. The steady-state problem

In this section we simply introduce the Navier-Stokes system by adding the advection term to (1.13) and get

$$\begin{aligned} -\nu\nabla^2\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= \mathbf{f}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0, & \text{in } \Omega, \\ \mathbf{u} &= 0, & \text{in } \Gamma. \end{aligned} \tag{1.29}$$

We can now derive the weak form of this problem following the same procedure of the previous section and obtain

$$(\nu\nabla\mathbf{u}, \nabla\mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \tag{1.30}$$

where

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \int_{\Omega} ((\mathbf{u} \cdot \nabla)\mathbf{v}) \cdot \mathbf{w} \, dV, \tag{1.31}$$

is a trilinear continuous form on $\mathbf{H}_0^1(\Omega)$. This equation is equivalent to the first one, i. e., if \mathbf{u} satisfies (1.30) we can find $p \in L^2(\Omega)$ that satisfies (1.29), and $\nabla \cdot \mathbf{u} = 0$.

Existence of solutions for this problem is proved in [2]. Uniqueness is now linked to the value of the constant ν , that represents the inverse of the Reynolds number. When $\nu^2 > c\|\mathbf{f}\|$ the solution is unique, otherwise there may be multiple solution to (1.30) corresponding to different configurations of the flow.

Also in this case non-homogeneous boundary conditions can be treated with the introduction of a lifting operator and the definition of an appropriate functional of the external force and boundary data. Uniqueness is also subject to the condition

$$|c(\mathbf{v}, \ell_{\Omega}\mathbf{u}_0, \mathbf{v})| \leq \frac{\nu}{2}\|\mathbf{v}\|^2, \quad \forall \mathbf{v} \in \mathbf{V} \tag{1.32}$$

and the external force \mathbf{f} is substituted by

$$\hat{\mathbf{f}} = \mathbf{f} + \nu\nabla^2\ell_{\Omega}\mathbf{u}_0 - (\ell_{\Omega}\mathbf{u}_0 \cdot \nabla)\ell_{\Omega}\mathbf{u}_0. \tag{1.33}$$

1.3.2. The time-dependent Navier-Stokes problem

The complete time-dependent Navier-Stokes equation can be written as

$$\begin{aligned} \frac{\partial\mathbf{u}}{\partial t} - \nu\nabla^2\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega, \\ \mathbf{u} &= 0 & \text{in } \Gamma, \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0 & \text{in } \Omega, \end{aligned} \tag{1.34}$$

1.3. The Navier-Stokes problem

where \mathbf{u}_0 is the initial condition defined over all the domain Ω . The system must be solved over the interval $[0, T]$. The conversion to the variational formulation follows the line of the previous ones to give

$$\begin{aligned} \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, dV + \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \, dV + \int_{\Omega} (\nu \nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) \, dV - \int_{\Gamma} \mathbf{v} \nu \nabla \mathbf{u} \cdot \mathbf{n} \, dS + \\ - \int_{\Omega} p \nabla \cdot \mathbf{v} \, dV + \int_{\Gamma} p \mathbf{v} \cdot \mathbf{n} \, dS = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dV \quad \forall \mathbf{v} \in \mathbf{V}, \\ \int_{\Omega} q \nabla \cdot \mathbf{u} \, dV = 0 \quad \forall q \in L_0^2. \end{aligned} \quad (1.35)$$

The boundary terms from the integration by parts can be treated in the same way of Section 1.2.2. Introducing the scalar product notation and the properties of the spaces that we already analyzed we can write

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \right) + (\nu \nabla \mathbf{u}, \nabla \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}. \quad (1.36)$$

We observe that

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \right) = \frac{d}{dt} (\mathbf{u}, \mathbf{v}),$$

so the weak problem can be rewritten as

$$\begin{aligned} \text{find } \mathbf{u} \in L^2(0, T; \mathbf{V}) \text{ satisfying} \\ \frac{d}{dt} (\mathbf{u}, \mathbf{v}) + (\nu \nabla \mathbf{u}, \nabla \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \\ \mathbf{u}(0) = \mathbf{u}_0, \end{aligned} \quad (1.37)$$

where $L^2(0, T; \mathbf{V})$ is the space of functions in \mathbf{V} that are integrable in $[0, T]$, i.e.,

$$L^2(a, b; \mathbf{V}) = \left\{ \mathbf{f} \in \mathbf{V} : \int_a^b \|\mathbf{f}\|_{\mathbf{V}}^2 \, dt < \infty \right\}. \quad (1.38)$$

Existence of solutions of (1.37) is proved in [2]. Again, uniqueness can be proved only for *small* values of \mathbf{f} and of the advection term, or when the *energy inequality*

$$|\mathbf{u}(t)|^2 + 2\nu \int_0^t \|\mathbf{u}(s)\|^2 \, ds \leq |\mathbf{u}(0)|^2 + 2 \int_0^t (\mathbf{f}, \mathbf{u}(s)) \, ds, \quad \forall t \in [0, T], \quad (1.39)$$

is satisfied. Strong solutions, if there exist any, satisfy (1.39) with an equal sign.

2. Finite element approximation

2.1. The Galerkin method approximation

In order to solve the weak problems analyzed in the previous chapter, we introduce the finite element method. We consider a generic differential equation expressed in its weak formulation

$$a(u, v) = f(v), \quad \forall v \in V, \quad (2.1)$$

where V is a suitable space, $a(\cdot, \cdot)$ is a bilinear form and $f(\cdot)$ a bounded linear operator.

The Galerkin method consists of finding an approximated solution u_h in a finite-dimensional subspace $V_h \subset V$. For this purpose we define a family of subspaces with a sequence of grid discretizations characterized by the parameter h . We can now solve, over the subspace of dimension n_h , for the function $u_h \in V_h$ the finite dimensional problem

$$a(u_h, v_h) = f(v_h), \quad \forall v_h \in V_h. \quad (2.2)$$

The problem in (2.2) is called *Galerkin approximation* of the continuous problem in (2.1). Let us consider the difference $e_h = u - u_h$ between the solution u of the continuous problem and the solution u_h of the discrete one. We can set v in (2.1) as v_h and get

$$a(e_h, v_h) = a(u - u_h, v_h) = a(u, v_h) - a(u_h, v_h) = f(v_h) - f(v_h) = 0. \quad (2.3)$$

The error e_h is therefore a -orthogonal to v_h , with the energy inner product $a(u, v)$ on V . This property is used to prove the *Céa's lemma*. Assume that $a(\cdot, \cdot)$ in (2.1) is a bounded, coercive bilinear form, i.e., there exist positive constants C and α such that

$$\begin{aligned} |a(u, v)| &\leq C \|u\| \|v\|, \quad \forall u, v \in V \\ a(u, u) &\geq \alpha \|u\|^2, \quad \forall u \in V, \end{aligned}$$

and that $f(\cdot)$ in (2.1) is bounded. Then, if u and u_h are the solutions of (2.1) and (2.2) respectively, we have that

$$\|u - u_h\| \leq \frac{C}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|. \quad (2.4)$$

The proof is obtained observing that

$$\begin{aligned} \alpha \|u - u_h\|^2 &\leq a(u - u_h, u - u_h) \leq a(u - u_h, u - u_h) + \\ &\quad + a(u - u_h, v_h - u_h) \leq C \|u - u_h\| \|u - v_h\|. \end{aligned}$$

This implies that the solution u_h is the best approximation of u in V_h .

2. Finite element approximation

If we consider a basis $\{\varphi_i, i = 1, 2, \dots, n_h\}$ of V_h we can rewrite (2.2) as

$$a(u_h, \varphi_i) = f(\varphi_i), \quad i = 1, 2, \dots, n_h. \quad (2.5)$$

The solution u_h can be written as a linear combination of the basis $\{\varphi_j\}$, $u_h = \sum_{j=1}^{n_h} u_j \varphi_j$. If we use this expression in (2.5) we get

$$\sum_{j=1}^{n_h} u_j a(\varphi_i, \varphi_j) = f(\varphi_i), \quad i = 1, 2, \dots, n_h. \quad (2.6)$$

We can define the *stiffness matrix* \mathbf{A} as the matrix with components $a_{ij} = a(\varphi_i, \varphi_j)$ and \mathbf{f} the vector with components $f_i = f(\varphi_i)$. If the unknowns u_j are collected in a vector \mathbf{u} then we can rewrite (2.6) as the following linear system

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \quad (2.7)$$

The properties of the stiffness matrix depend on the selected basis in V_h and the type of the problem studied. Relevant characteristics for the numerical solution of the system, such as the condition number and sparsity pattern of \mathbf{A} , depend strongly on the form of the basis functions. For this reason, basis with smaller support are preferred. In fact when the support of two element φ_i, φ_j does not overlap, their energy scalar product is zero. Another important basis feature for the numerical solution is the computational effort needed to calculate the elements of \mathbf{A} and \mathbf{f} .

In the same way we can associate to each vector \mathbf{v} in \mathbb{R}^{n_h} the function $v_h = \sum_{i=1}^{n_h} v_i \varphi_i$ in V_h . We note that, under the same hypotheses of coercivity and boundlessness for $a(\cdot, \cdot)$ we have

$$\begin{aligned} \mathbf{v}^\top \mathbf{A}\mathbf{v} &= \sum_{i=1}^{n_h} \sum_{j=1}^{n_h} v_i a_{ij} v_j = \sum_{i=1}^{n_h} \sum_{j=1}^{n_h} v_i a(\varphi_i, \varphi_j) v_j = \sum_{i=1}^{n_h} \sum_{j=1}^{n_h} a(v_i \varphi_i, v_j \varphi_j) = \\ &= a\left(\sum_{i=1}^{n_h} v_i \varphi_i, \sum_{j=1}^{n_h} v_j \varphi_j\right) = a(v_h, v_h) \geq \alpha \|v_h\|^2 \geq 0. \end{aligned} \quad (2.8)$$

We can see also that $\mathbf{v}^\top \mathbf{A}\mathbf{v} = 0$ if and only if $\mathbf{v} = 0$, so the stiffness matrix is positive definite.

Another critical property of the Galerkin approximation is that the solution u_h tends to the exact solution u when the dimension of V_h tends to infinity ($h \rightarrow 0$). From (2.4), we must only prove that, as h tends to zero, V_h becomes the space V . This ensures that

$$\lim_{h \rightarrow 0} \|u - u_h\| = 0. \quad (2.9)$$

The different choices of V_h affect the convergence velocity and the error order.

2.2. The Finite Element Method

The *Finite Element Method (FEM)* is based on the Galerkin method that decomposes the domain in a set of elements on which the solution is discretized on the nodes of each element [5]. A suitable choice for V_h is adopted on each element in order to satisfy (2.9) and improve sparsity patterns of the resulting matrix. We start the description of a one-dimensional problem and then analyze the two- and three-dimensional cases [6].

2.2.1. One-dimensional case

Let us consider the interval $(a, b) \subset \mathbb{R}$ and a partition \mathbb{T}_h of (a, b) in $N + 1$ intervals $K_j = (x_j, x_{j+1})$ of length $h_j = x_{j+1} - x_j$, with

$$a = x_0 < x_1 < \dots < x_N < x_{N+1} = b, \quad (2.10)$$

and $h = \max_j(h_j)$. A natural choice for the basis function $\{\varphi_i\}$ of $H^1(a, b)$ is a polynomial basis. For this purpose we can introduce the family of spaces

$$X_h^r = \left\{ v_h \in C^0[a, b] : v_h|_{K_j} \in \mathbb{P}_r, \forall K_j \in \mathbb{T}_h \right\}, \quad r = 1, 2, \dots,$$

where \mathbb{P}_r is the space of polynomials of order r . We recall that all X_h^r are subspaces of $H^1(a, b)$. In order to guarantee the maximum extension of the solution space, we can use the lower order polynomials that satisfy the existence of the integrals involved. We also select the polynomials with a small support to obtain a highly sparse and well-conditioned system. We will analyze the widely used Lagrangian basis for X_h^1 and X_h^2 .

First degree polynomials - X_h^1

The functions in the space X_h^1 are continuous and linear. It is easy to see that the degrees of freedom on \mathbb{T}_h are $N + 2$. In the Lagrangian approach the points in analysis are the points with coordinates $\{x_i\}$ equal to those in (2.10) and the basis $\{\varphi_i\}$ satisfies the property

$$\varphi_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (2.11)$$

where δ_{ij} is the *Kronecker delta*. In this way, the weights of the solution u_h with respect to the basis $\{\varphi_i\}$ are the value of the function on each point, i.e.,

$$u_h(x) = \sum_{i=0}^{N+1} u_i \varphi_i(x) = \sum_{i=0}^{N+1} u_h(x_i) \varphi_i(x). \quad (2.12)$$

The complete form of an element of this basis, as shown in Fig. 2.1, is therefore

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{h_{i-1}}, & x_{i-1} \leq x \leq x_i, \\ \frac{x_{i+1}-x}{h_i}, & x_i \leq x \leq x_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (2.13)$$

2. Finite element approximation

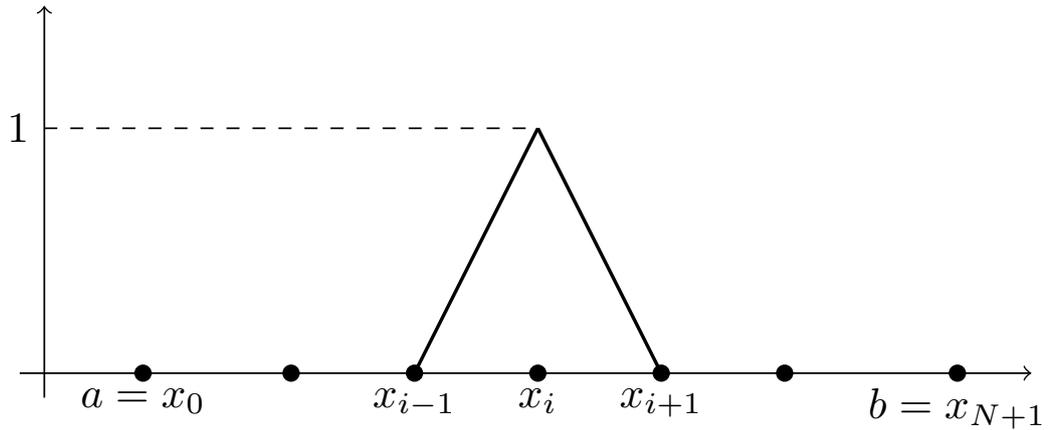


Figure 2.1.: Hat function for the interval K_i .

that is often referred to as *hat function*. The support for each φ_i is limited to the interval $[x_{i-1}, x_{i+1}]$, with the exception of the first element φ_0 and last one φ_{N+1} , for which the support is limited to one interval, $[a, x_1]$ and $[x_N, b]$ respectively. This means that each function support overlaps only with the support of the φ_{i-1} and φ_{i+1} functions. Therefore the stiffness matrix formed with this basis will be tri-diagonal.

In order to build the matrix for the discretization adopted, we add the contributions element by element. On each sub-interval $K_i = [x_i, x_{i+1}]$, the only basis functions different from zero are the ones that are centered at the two end-points, φ_i and φ_{i+1} . The restrictions of these two bases to the interval K_i are called *shape functions*. They are expressed by

$$N_{i,l} = \frac{x_{i+1} - x}{h_i},$$

for the left one and

$$N_{i,r} = \frac{x - x_{i-1}}{h_{i-1}},$$

for the right one. The definition of shape function can be extended to the whole domain $[a, b]$, but any function will be zero out of K_i .

The shape of these functions are the same on every interval K_i but the length of the interval h_i may be different. For this reason it is clever to transform each element into a *canonical element*, in order to calculate the integrals only once. The reference interval used is typically $[-1, 1]$, called *EDGE2*, and the linear transformation between $[-1, 1]$ and K_i is

$$x(\xi) = \frac{1 - \xi}{2}x_i + \frac{1 + \xi}{2}x_{i+1}, \quad \xi \in [-1, 1]. \quad (2.14)$$

We can now introduce the shape functions N_0 and N_1 , plotted in Fig.2.2, associated to

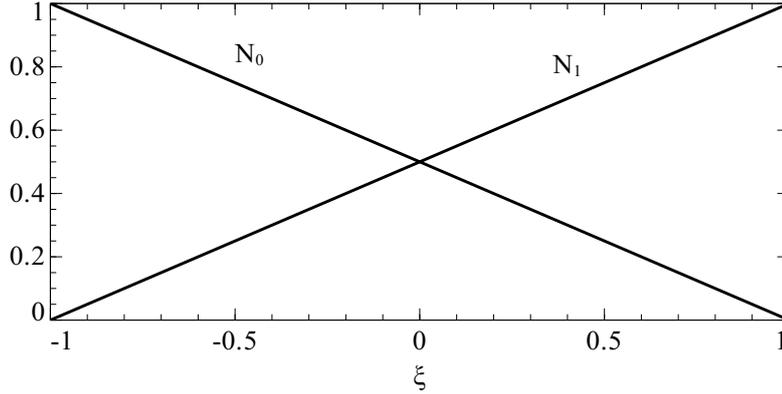


Figure 2.2.: Linear shape functions on the EDGE2 canonical element.

the canonical element

$$N_0(\xi) = \frac{1 - \xi}{2}, \quad N_1(\xi) = \frac{1 + \xi}{2}, \quad (2.15)$$

that can be used to calculate all the integrals involved in the building of the stiffness matrix, once we know the Jacobian of the transformation between the original element and the canonical one.

Second degree polynomials - X_h^2

The degrees of freedom granted by a polynomial of second order are three for each element K_i of the partition \mathbb{T}_h . In order to impose the continuity of the approximation among elements we fix the end-points of each interval in the same way as we did for the first order case. In addition we need to fix a further point, for example the midpoint. The Lagrangian method sets the value of each function φ_i in the basis at each point following (2.11) which recovers again (2.12). The generic element of this basis becomes

$$\varphi_i(x) = \begin{cases} 1 + 3\left(\frac{x-x_i}{h_{i-1}}\right) + 2\left(\frac{x-x_i}{h_{i-1}}\right)^2 & x_{i-1} \leq x \leq x_i \\ 1 - 3\left(\frac{x-x_i}{h_i}\right) + 2\left(\frac{x-x_i}{h_i}\right)^2 & x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise,} \end{cases} \quad (2.16)$$

with x_i an end-point of an element, or

$$\varphi_i(x) = \begin{cases} 1 - 4\left(\frac{x-x_{i+1/2}}{h_i}\right) & x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise,} \end{cases} \quad (2.17)$$

with $x_{i+1/2}$ a mid-point of an element. This choice for the basis $\{\varphi_i\}$ leads to a pentadiagonal stiffness matrix.

2. Finite element approximation

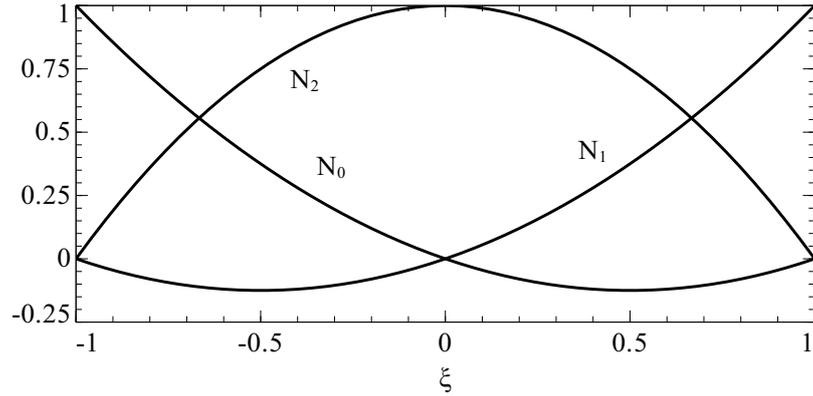


Figure 2.3.: Quadratic shape functions on the *EDGE3* canonical element.

In a similar way also in this case the transformation into the canonical element simplifies the amount of calculations needed to build the stiffness matrix. The canonical shape functions are

$$N_0(\xi) = \frac{\xi(\xi - 1)}{2}, \quad N_1(\xi) = \frac{\xi(\xi + 1)}{2}, \quad N_2(\xi) = 1 - \xi^2. \quad (2.18)$$

These are plotted in Fig. 2.3. The three nodes of the canonical element, called *EDGE3*, are -1 , 0 and 1 .

The way to obtain the Lagrangian bases for higher degrees is straightforward. We can generalize (2.18) to a generic order r with the *Lagrange formula*

$$N_k(\xi) = \prod_{\substack{l=0 \\ l \neq k}}^r \frac{\xi - \xi_l}{\xi_k - \xi_l}. \quad (2.19)$$

It is important to underline that using high order polynomials affects the sparsity ratio of the stiffness matrix and therefore the computational costs. Moreover, linear interpolation theory shows that using equidistant nodes with high degree polynomials can lead to unstable approximations [7].

2.2.2. Two-dimensional case

We start considering a polygonal open domain $\Omega \in \mathbb{R}^2$ partitioned with a \mathbb{T}_h that covers completely Ω without overlapping. The discretized domain Ω_h is

$$\Omega_h = \Omega = \text{int} \left(\bigcup_{K \in \mathbb{T}_h} K \right), \quad (2.20)$$

where int does not include the boundary of the union. We can introduce the diameter of an element K as

$$\text{diam}(K) = h_K = \max_{x, y \in K} \|x - y\|, \quad (2.21)$$

and define the spatial step h for the partition \mathbb{T}_h as

$$h = \max_{K \in \mathbb{T}_h} h_K. \quad (2.22)$$

The partitioning $\{\mathbb{T}_h, h\}$ must satisfy also a regularity constraint. If ρ_K is the diameter of the inscribed circle of K and δ a positive constant, we say that \mathbb{T}_h is regular if

$$\frac{h_K}{\rho_K} \leq \delta \quad \forall K \in \mathbb{T}_h. \quad (2.23)$$

We define the finite element space X_h^r as

$$X_h^r = \left\{ v_h \in C^0(\overline{\Omega}) : v_h|_K \in \mathbb{P}_r \quad \forall K \in \mathbb{T}_h \right\}, \quad (2.24)$$

and

$$X_{h,0}^r = \{ v_h \in X_h^r : v_h|_{\partial\Omega} = 0 \}, \quad (2.25)$$

which are our approximation spaces for $H^1(\Omega)$ and $H_0^1(\Omega)$ respectively. We can also introduce a rigorous definition of the finite element as the structure $(K, \mathbb{P}_r, \{\varphi_i\})$ such that

- $K \subseteq \mathbb{R}^n$ is a non-empty, closed and bounded set with a polygonal boundary,
- \mathbb{P}_r is a finite-dimensional function space on K called *shape functions space*,
- $\{\varphi_i\}$ is a basis of \mathbb{P}_r ,
- $\{\sigma_i : \mathbb{P}_r \rightarrow \mathbb{R}\}$ is a set of functional on \mathbb{P}_r such that $\sigma_i(\varphi_j) = \delta_{ij}$.

Each functional σ_i identifies uniquely the coefficient $\alpha_i = \sigma_i(p)$ for any polynomial p in \mathbb{P}_r , so the linear combination $p(x) = \sum_{i=1}^r \alpha_i \varphi_i(x)$ is correctly defined. $\{\alpha_i\}$ are the degree of freedom of the finite element. In the Lagrangian case, each degree fixes the value of the function at a point of the element, called *node*, $\alpha_i = p(\mathbf{a}_i)$. Therefore for the Lagrangian basis the set of nodes $\{\mathbf{a}_i\}$ is equivalent to the set of functionals $\{\sigma_i\}$. These nodes must be chosen wisely in order to assure that the interpolation problem is well-posed. This property is true if the set $\{\mathbf{a}_i\}$ is *unisolvant* on \mathbb{P}_r ,

$$p(\mathbf{a}_i) = \alpha_i, \quad i = 0, 1, \dots, r, \quad (2.26)$$

where $p \in \mathbb{P}_r$ is unique and $\{\alpha_i\}$ is an arbitrary set of $r + 1$ scalars.

We now analyze the two most used Lagrangian elements in two dimensions: the triangular and the quadrangular finite elements.

2. Finite element approximation

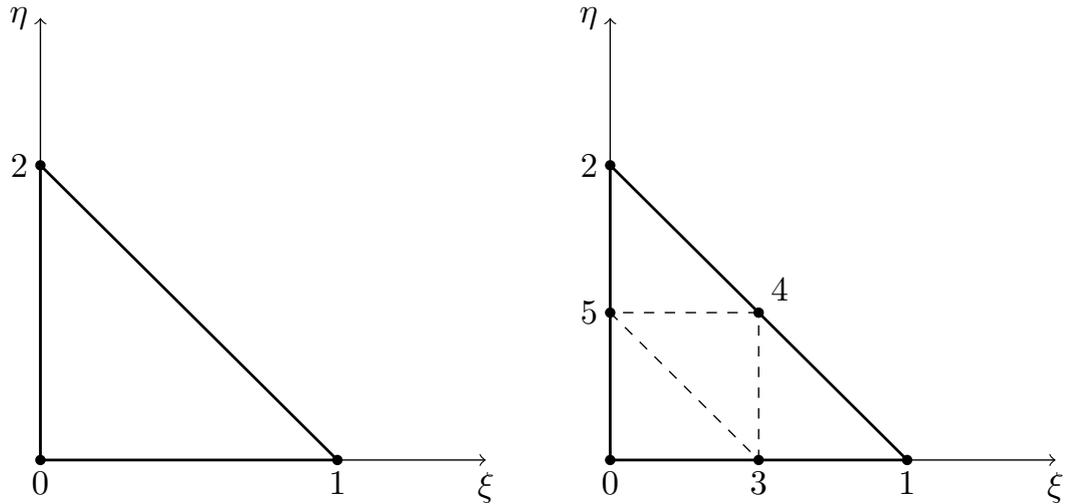


Figure 2.4.: TRI3 (left) and TRI6 (right) canonical elements.

Triangular finite element

The linear case is an extension of the mono-dimensional one, with linear basis that are similar to the hat functions. The shape functions $\{N_i\}$ that come from this basis satisfy

$$N_i(\mathbf{x}_j) = \delta_{ij}, \quad i, j = 0, 1, 2, \quad (2.27)$$

where \mathbf{x}_j are the vertices of the triangular element. The basis functions associated with the shape functions satisfy (2.27) and have a support over four elements that are connected to the node on which the function is 1. This choice implies continuity of the solution on the whole boundary of the element, where the solution is the linear interpolation between two nodal values. Also in this case it is appropriate to introduce a canonical element, called *TRI3*, which is the triangle of vertices $(0, 0)$, $(1, 0)$, $(0, 1)$ and shown in Fig. 2.4. The shape functions on this element are

$$N_0(\xi, \eta) = 1 - \xi - \eta, \quad N_1(\xi, \eta) = \xi, \quad N_2(\xi, \eta) = \eta. \quad (2.28)$$

The extension to bases of higher order is straightforward. We note that the number of nodes n_r that must be fixed for a polynomial basis of order r is

$$n_r = \frac{(r+1)(r+2)}{2}, \quad (2.29)$$

for example the quadratic Lagrangian basis is unique if it involves six nodes: the three vertices and the three midpoints of the edges. The shape functions associated to this

basis on the canonical element are therefore

$$\begin{aligned}
 N_0(\xi, \eta) &= 1 - 3(\xi + \eta) + 2(\xi + \eta)^2, & N_1(\xi, \eta) &= \xi(2\xi - 1), \\
 N_2(\xi, \eta) &= \eta(2\eta - 1), & N_3(\xi, \eta) &= 4\xi(1 - \xi - \eta), \\
 N_4(\xi, \eta) &= 4\xi(1 - \xi - \eta), & N_5(\xi, \eta) &= 4\xi\eta.
 \end{aligned}
 \tag{2.30}$$

The canonical triangular element of second order is called *TRI6* and shown in Fig. 2.4.

Quadrangular finite element

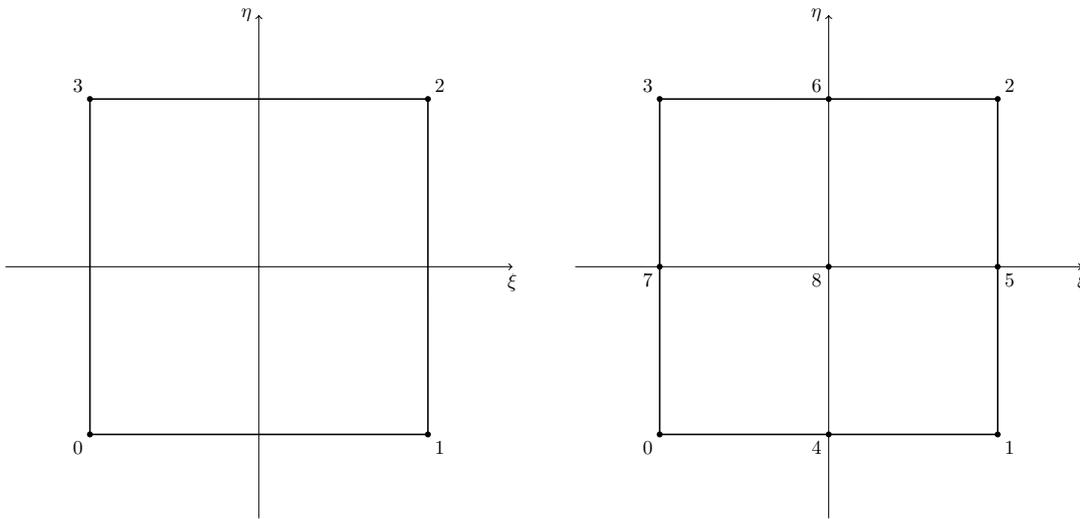


Figure 2.5.: QUAD4 (left) and QUAD9 (right) canonical elements.

The building of a quadrangular finite element is easier than that of a triangle because for Lagrangian elements it is a direct extension of the mono-dimensional case. The shape functions are a product of the mono-dimensional shapes associated with each direction. We can identify each node and each shape with a pair of indices, one for each dimension, and write the shape functions as

$$N_{ik}(\mathbf{x}_{jl}) = \delta_{ij}\delta_{kl}.
 \tag{2.31}$$

If the polynomial approximation is linear in each direction, the nodes needed to identify in a unique way the shape functions are four and lie on the vertices of the quadrangle. The basis function has a support made by four quadrangular elements that share one vertex. The associated canonical element is the square $[-1, 1] \times [-1, 1]$ shown in Fig. 2.5

2. Finite element approximation

and the shape functions are

$$\begin{aligned} N_{00}(\xi, \eta) &= N_0(\xi)N_0(\eta) = \frac{1-\xi}{2} \frac{1-\eta}{2}, \\ N_{10}(\xi, \eta) &= N_1(\xi)N_0(\eta) = \frac{1+\xi}{2} \frac{1-\eta}{2}, \\ N_{11}(\xi, \eta) &= N_1(\xi)N_1(\eta) = \frac{1+\xi}{2} \frac{1+\eta}{2}, \\ N_{01}(\xi, \eta) &= N_0(\xi)N_1(\eta) = \frac{1-\xi}{2} \frac{1+\eta}{2}. \end{aligned}$$

All $\{N_{ij}\}$ are bilinear on the canonical quadrangle, and, even if they contain terms of second order in $\xi\eta$, they cannot reproduce all polynomials of second order, but only polynomials of first order are reproduced completely.

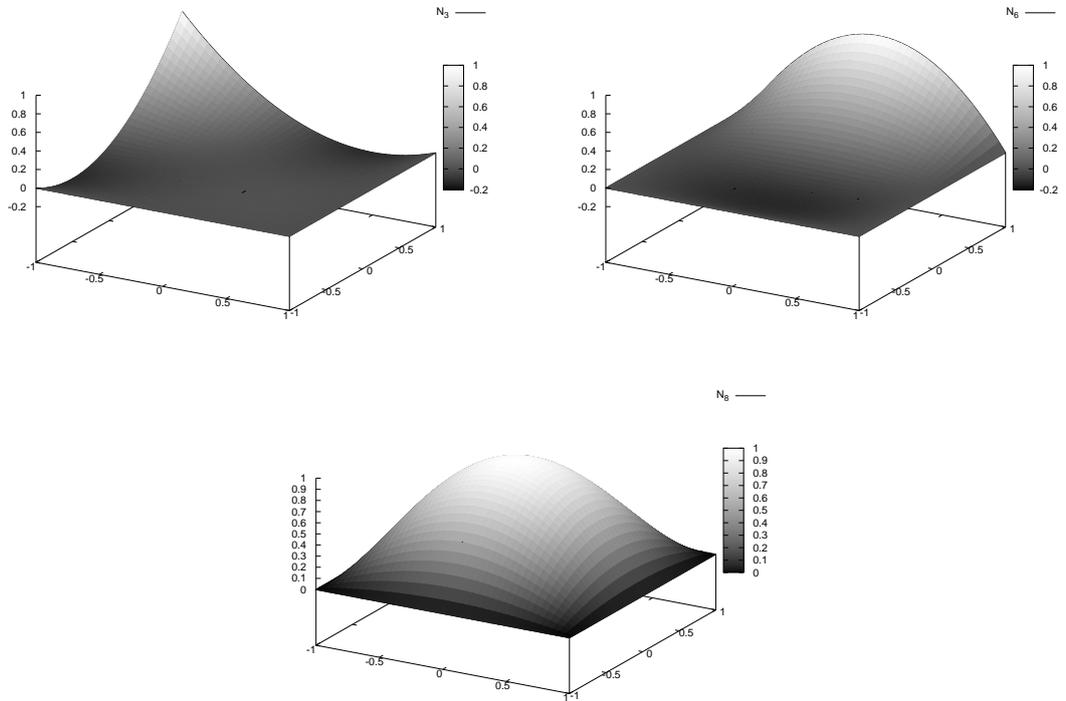


Figure 2.6.: Shape functions on the QUAD9 canonical element: N_{01} (top left), N_{21} (top right), N_{22} (bottom).

When the order of the approximating polynomial is greater than one, the number of nodes needed to define the Lagrangian basis grows quadratically with respect to the mono-dimensional case. For example, as shown in Fig. 2.5, the quadratic approximation requires nine nodes, the four vertices, the four midpoints on each side and the central point. The shape functions can again be obtained by multiplying the shape functions

of the mono-dimensional case in (2.18), and three different cases of them are shown in Fig. 2.6. These function contain also terms of higher order than the second, but they can reproduce completely only polynomials of second order.

2.2.3. Three-dimensional case

A three-dimensional Lagrangian element respects the general definition given in Par. 2.2.2 and is therefore build in the same way of the two-dimensional case. The most widely used elements are tetrahedra and hexahedra.

Tetrahedral finite element

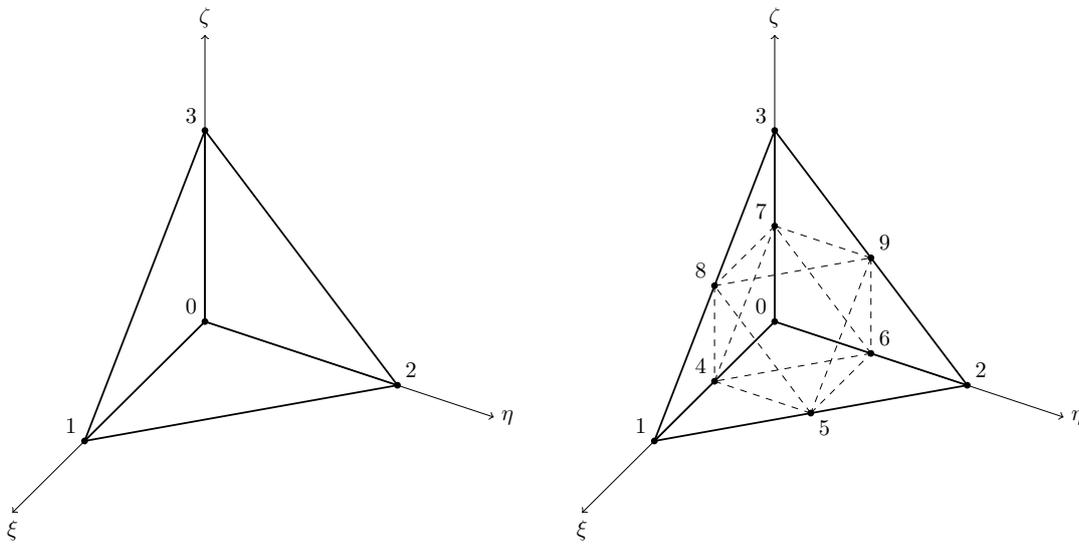


Figure 2.7.: TET4 (left) and TET10 (right) canonical elements.

The linear Lagrangian tetrahedron has the nodes on the four vertices. The shape functions satisfy again the relation

$$N_i(\mathbf{x}_j) = \delta_{ij}. \quad (2.32)$$

We introduce also the canonical tetrahedron TET_4 with vertices in $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, shown in Fig. 2.7. The shape functions on this element are

$$\begin{aligned} N_0(\xi, \eta, \zeta) &= 1 - \xi - \eta - \zeta, & N_1(\xi, \eta, \zeta) &= \xi, \\ N_2(\xi, \eta, \zeta) &= \eta, & N_3(\xi, \eta, \zeta) &= \zeta. \end{aligned}$$

These are linear on each face and are completely defined by its three vertices. In fact, each face can be seen as a TRI3 finite element with respect to its own coordinates.

2. Finite element approximation

If the order of the interpolating polynomial is r , the number of nodes n_r of the tetrahedron is

$$n_r = \frac{(r+1)(r+2)(r+3)}{6}. \quad (2.33)$$

In the quadratic case, n_r is equal to 10, with the addition of the midpoints of each edge, numbered as in Fig. 2.7 where the *TET10* canonical element is shown. The shape functions are

$$\begin{aligned} N_0(\xi, \eta, \zeta) &= 1 - 3(\xi + \eta + \zeta) + 3(\xi + \eta + \zeta)^2, & N_1(\xi, \eta, \zeta) &= \xi(2\xi - 1), \\ N_2(\xi, \eta, \zeta) &= \eta(2\eta - 1), & N_3(\xi, \eta, \zeta) &= \zeta(2\zeta - 1), \\ N_4(\xi, \eta, \zeta) &= 4\xi(1 - \xi - \eta - \zeta), & N_5(\xi, \eta, \zeta) &= 4\xi\eta, \\ N_6(\xi, \eta, \zeta) &= 4\eta(1 - \xi - \eta - \zeta), & N_7(\xi, \eta, \zeta) &= 4\zeta(1 - \xi - \eta - \zeta), \\ N_8(\xi, \eta, \zeta) &= 4\xi\zeta, & N_9(\xi, \eta, \zeta) &= 4\eta\zeta, \end{aligned}$$

and each face is a TRI6 with respect to its own coordinates.

Hexahedral finite element

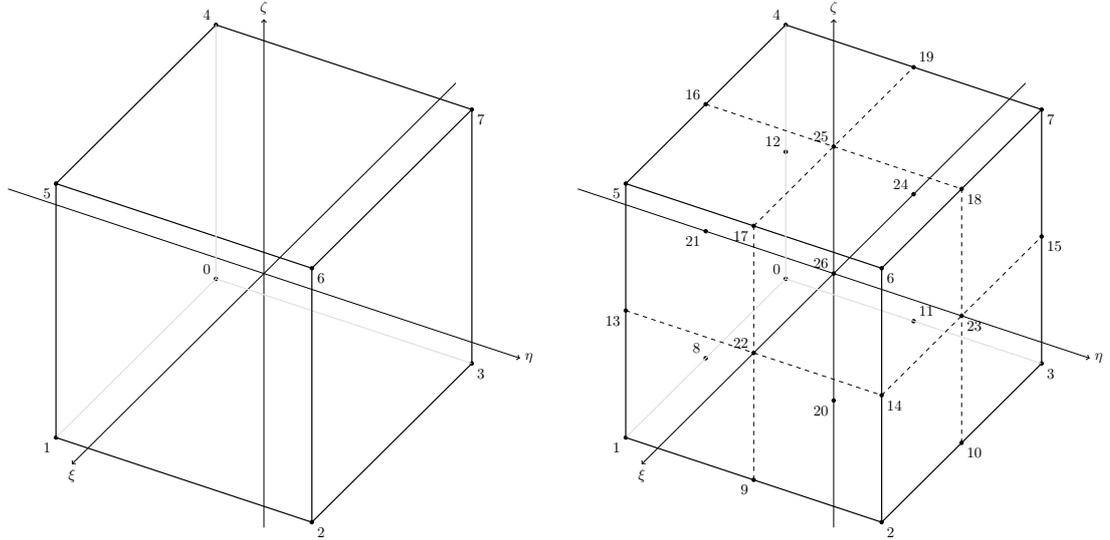


Figure 2.8.: HEX8 (left) and HEX27 (right) canonical elements.

Just like the two-dimensional case, the hexahedral Lagrangian finite element can be seen as the combination of three mono-dimensional edge elements. In the linear case, the nodes are eight and correspond to the vertices of the hexahedron. The canonical element associated is the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$, called *HEX8*, shown in Fig. 2.8, with the shape functions $\{N\}$ denoted by three indices that identify the mono-dimensional shape associated, that is

$$N_{ijk}(\xi, \eta, \zeta) = N_i(\xi)N_j(\eta)N_k(\zeta), \quad i, j, k = 0, 1. \quad (2.34)$$

For the quadratic case, the number of nodes is 27, with the addition of edge middle points, face centers and volume central point. The element is called *HEX27* and shown in Fig. 2.8. The shape function are a combination of the mono-dimensional *EDGE3* element, and can be written as

$$N_{ijk}(\xi, \eta, \zeta) = N_i(\xi)N_j(\eta)N_k(\zeta), \quad i, j, k = 0, 1, 2. \quad (2.35)$$

2.3. Error estimation

2.3.1. Notation

In order to give an estimate of the error in the finite element method, let us consider a generic domain Ω . For every $v \in C^0(\bar{\Omega})$ we introduce the interpolator $\Pi_h^r v$ in X_h^r as the function for which

$$\Pi_h^r v(\mathbf{x}_i) = v(\mathbf{x}_i), \quad \forall \mathbf{x}_i \in \mathbb{T}_h, i = 0, 1, \dots, N_h, \quad (2.36)$$

where $\{\mathbf{x}_i\}$ are the nodes of the triangulation \mathbb{T}_h . If $\{\varphi_i\}$ is a Lagrangian base of X_h^r , we can write

$$\Pi_h^r v(\mathbf{x}) = \sum_{i=1}^{N_h} v(\mathbf{x}_i) \varphi_i(\mathbf{x}), \quad (2.37)$$

where this can be also seen as an operative way of calculating the interpolator. The operator Π_h^r maps a continuous function v with its interpolator $\Pi_h^r v$ and is called *interpolating operator*. We can also introduce a local interpolating operator Π_K^r associated to a single element K of the partition \mathbb{T}_h as

$$\Pi_K^r v|_K = (\Pi_h^r v)|_K. \quad (2.38)$$

We also introduce the sphericity ρ_K and the diameter h_K of the element K , and the invertible affine transformation $F_K : \hat{K} \rightarrow K$ between the canonical element \hat{K} and the element K . This map can be expressed with

$$F_K(\hat{\mathbf{x}}) = A_K(\hat{\mathbf{x}}) + \mathbf{b}_K, \quad (2.39)$$

with $A_K \in \mathbb{R}^{n \times n}$ and $\mathbf{b}_K \in \mathbb{R}^n$.

2.3.2. Energy (H^1) norm error estimation

In order to study the energy norm error we introduce the error on seminorms, which are defined in (1.7) [8]. For every $m \geq 0$ and $v \in H^m(K)$, let $\hat{v} : \hat{K} \rightarrow \mathbb{R}$ be the function $\hat{v} = v \circ F_K$ for each $\hat{v} \in H^m(\hat{K})$ such that there exists $C = C(m) > 0$ and

$$\|\hat{v}\| \leq C \|A_K\|^m |\det A_K|^{-1/2} \|v\|, \quad (2.40)$$

$$\|v\| \leq C \|A_K^{-1}\|^m |\det A_K|^{1/2} \|\hat{v}\|. \quad (2.41)$$

2. Finite element approximation

The matrix norm $\|A_K\|$ is the norm associated with the Euclidean norm,

$$\|A_K\| = \sup_{\boldsymbol{\xi} \in \mathbb{R}^n, \boldsymbol{\xi} \neq 0} \frac{\|A_K \boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|}, \quad (2.42)$$

or, equivalently,

$$\|A_K\| = \frac{1}{\hat{\rho}} \sup_{\boldsymbol{\xi} \in \mathbb{R}^n, \|\boldsymbol{\xi}\| = \hat{\rho}} \|A_K \boldsymbol{\xi}\|, \quad (2.43)$$

where $\hat{\rho}$ is the sphericity of the canonical element \hat{K} . For every $\boldsymbol{\xi}$ with $\|\boldsymbol{\xi}\| = \hat{\rho}$, there exist two points $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ in \hat{K} such that $\hat{\boldsymbol{x}} - \hat{\boldsymbol{y}} = \boldsymbol{\xi}$. We notice that $A_K \boldsymbol{\xi} = F_K(\hat{\boldsymbol{x}}) - F_K(\hat{\boldsymbol{y}})$, so

$$\|A_K\| \leq \frac{h_K}{\hat{\rho}}. \quad (2.44)$$

Similarly we get

$$\|A_K^{-1}\| \leq \frac{\hat{h}}{\rho_K}, \quad (2.45)$$

where \hat{h} is the diameter of \hat{K} .

We try now to estimate the value of the norm of $(v - \Pi_K^r v) \circ F_K$ for every function $v \in H^m(K)$ with reference to the canonical element \hat{K} . If $\boldsymbol{x}_i = F_K(\hat{\boldsymbol{x}}_i)$ are the nodes on K and $\{\varphi_i\} = \{\hat{\varphi}_i \circ F_K^{-1}\}$ the basis of K we can write

$$\Pi_K^r v \circ F_K = \sum_{i=0}^{M_K-1} v(\boldsymbol{x}_i) \varphi_i \circ F_K = \sum_{i=0}^{M_K-1} v(F_K(\hat{\boldsymbol{x}}_i)) \hat{\varphi}_i = \Pi_{\hat{K}}^r \hat{v}, \quad (2.46)$$

where M_K is the number of nodes on K . We can infer therefore that

$$\|(v - \Pi_K^r v) \circ F_K\| = \|\hat{v} - \Pi_{\hat{K}}^r \hat{v}\|. \quad (2.47)$$

We are now ready to introduce some results on the local and the global estimate.

- *Local estimate for the interpolating error.* Let $r \geq 1$ and $0 \leq m \leq r + 1$. There exists a constant $C = C(r, m, \hat{K})$ such that

$$\|v - \Pi_K^r v\|_{H^m(K)} \leq C \frac{h_K^{r+1}}{\rho_K^m} \|v\|_{H^{r+1}(K)}, \quad \forall v \in H^{r+1}(K). \quad (2.48)$$

- *Global estimate for the interpolating error.* Let $\{\mathbb{T}_h\}$ be a family of regular triangulation on ω , $m = 0, 1$ and $r \geq 1$. There exists a constant $C = C(r, m, \hat{K})$ such that

$$\|v - \Pi_h^r v\|_{H^m(\Omega)} \leq C \left(\sum_{K \in \mathbb{T}_h} h_K^{2(r+1-m)} \|v\|_{H^{r+1}(K)}^2 \right)^{1/2}, \quad \forall v \in H^{r+1}(\Omega). \quad (2.49)$$

In particular

$$\|v - \Pi_h^r v\|_{H^m(\Omega)} \leq C h^{r+1-m} \|v\|_{H^{r+1}(K)}, \quad \forall v \in H^{r+1}(\Omega). \quad (2.50)$$

With (2.50) we can get an estimate for the Galerkin approximation method using (2.4). Let $u \in V$ be the exact solution of a variational problem (2.1) and u_h its approximation obtained with the finite element method of order r . If $u \in H^{r+1}(\Omega)$ it holds

$$\|u - u_h\|_{H^1(\Omega)} \leq \frac{M}{\alpha} C \left(\sum_{K \in \mathbb{T}_h} h_K^{2r} \|u\|_{H^{r+1}(K)}^2 \right)^{1/2}, \quad (2.51)$$

and

$$\|u - u_h\|_{H^1(\Omega)} \leq \frac{M}{\alpha} C h^r \|u\|_{H^{r+1}(K)}, \quad (2.52)$$

where C does not depend on h or u .

This result demonstrates that we can follow two ways to improve the accuracy of the solution. We can refine the computational grid, reducing the value of h , or increase the polynomial order r of the approximation. The second approach is viable only for enough regular solution, namely if $u \in V \cap H^{p+1}(\Omega)$ the maximum order of the approximation to use is $r = p$. Even if the solution has only the lower possible regularity ($p = 0$), the Céa's lemma guarantees the convergence, but the estimate (2.52) is no longer valid. We can summarize the results saying that, if $u \in H^{p+1}(\Omega)$, then we have

$$\|u - u_h\|_{H^1(\Omega)} \leq C h^s \|u\|_{H^{s+1}(K)}, \quad s = \min(r, p). \quad (2.53)$$

2.3.3. L^2 norm error estimation

We can get an estimate for the error also in L^2 norm. This norm is lesser strict than the energy one and we expect a faster convergence with h . With the same hypotheses of (2.52), let us consider the adjoint problem of (2.1), i.e. the problem of finding $\phi = \phi(g) \in V$ such that

$$a(v, \phi) = (g, v) \quad \forall v \in V. \quad (2.54)$$

If we assume a result of elliptic regularity for (2.1), the adjoint problem (2.54) has the same property, or there exists a $C > 0$ such that

$$\|\phi(g)\|_{H^2(\Omega)} \leq C \|g\|_{L^2(\Omega)}, \quad \forall g \in L^2(\Omega). \quad (2.55)$$

This property holds for an elliptic problem on a polygonal and convex domain with Dirichlet or Neumann (but not mixed) boundary conditions [9]. Let $e_h = u - u_h$ be the approximation error, if we choose $g = e_h$ and $v = e_h$ we get

$$\|e_h\|_{L^2(\Omega)}^2 = a(e_h, \phi(e_h)). \quad (2.56)$$

For the adjoint problem we have $\phi \in H^2$ and, using the Galerkin orthogonality (2.3), we can write

$$\begin{aligned} \|e_h\|_{L^2(\Omega)}^2 &= a(e_h, \phi) = a(e_h, \phi - \Pi_h^1 \phi) \leq C_1 \|e_h\|_{H^1(\Omega)} \|\phi - \Pi_h^1 \phi\|_{H^1(\Omega)} \leq \\ &\leq C_2 \|e_h\|_{H^1(\Omega)} h \|\phi\|_{H^2(\Omega)} \leq C_3 \|e_h\|_{H^1(\Omega)} h \|e_h\|_{L^2(\Omega)}. \end{aligned} \quad (2.57)$$

2. Finite element approximation

By using the continuity of the bilinear form $a(\cdot, \cdot)$ and (2.53) we obtain

$$\|e_h\|_{L^2(\Omega)} = C_3 h \|e_h\|_{H^1(\Omega)}, \quad (2.58)$$

so the order of convergence is one order greater than that in H^1 ,

$$\|u - u_h\|_{L^2(\Omega)} = C h^{s+1} \|u\|_{H^{s+1}(\Omega)}, \quad s = \min(r, p). \quad (2.59)$$

with C independent from h and u .

3. Numerical solution

3.1. Discretization of the Navier-Stokes equations

We recall the strong form of the Navier-Stokes incompressible problem of (1.34) rewritten in a slightly more general way

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u}^\top))) + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \quad \mathbf{x} \in \Omega, t \in [0, T], \end{aligned} \quad (3.1)$$

with appropriate boundary and initial conditions. The domain considered is again Ω , with $\Gamma = \partial\Omega$ and \mathbf{n} the external normal to Γ . Here the density and the viscosity may not be constant. We will suppose that $\mathbf{f} \in \mathbf{L}^2([0, T]; [\mathbf{L}^2(\Omega)])$. The variational formulation is similar to (1.36). We can introduce a new trilinear form $d(\cdot, \cdot, \cdot)$ defined by

$$d(\mathbf{u}, \mathbf{v}, \mathbf{w}) = ((\mathbf{u} \cdot \nabla)\mathbf{v}, \mathbf{w}) + \frac{1}{2}(\nabla \cdot \mathbf{u}, \mathbf{v} \cdot \mathbf{w}), \quad (3.2)$$

that is equivalent to $c(\cdot, \cdot, \cdot)$ of (1.31) when \mathbf{u} is incompressible. The introduction of this additional term leads to a more stable solver (see [10]). We can then rewrite the system as

$$\left(\rho \frac{\partial \mathbf{u}}{\partial t}, \mathbf{v}\right) + a(\mathbf{u}, \mathbf{v}) + d(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \quad (3.3)$$

where the density ρ and the viscosity μ are now embedded in $d(\cdot, \cdot, \cdot)$ and $a(\cdot, \cdot)$, respectively. The bilinear operator $b(\cdot, \cdot)$ is here introduced because we do not require that all the functions in \mathbf{V} are divergence free, differently from (1.37). It is defined by

$$b(\mathbf{v}, q) = - \int_{\Omega} q \nabla \cdot \mathbf{v} dV, \quad \forall \mathbf{v} \in \mathbf{V}, \forall q \in Q. \quad (3.4)$$

We can introduce an operational formulation of these equations that is suitable for understanding the successive discretization that will convert each differential operator in a matrix. We therefore introduce the operators

$$\begin{aligned} A : \mathbf{V} &\rightarrow \mathbf{V}', & \langle A\mathbf{u}, \mathbf{v} \rangle &= a(\mathbf{u}, \mathbf{v}), & \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}, \\ B : \mathbf{V} &\rightarrow Q', & \langle B\mathbf{v}, p \rangle &= b(\mathbf{v}, p), & \forall \mathbf{v} \in \mathbf{V}, \forall p \in Q, \\ D : \mathbf{V}^2 &\rightarrow \mathbf{V}', & \langle D(\mathbf{u}, \mathbf{u}), \mathbf{v} \rangle &= d(\mathbf{u}, \mathbf{u}, \mathbf{v}), & \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}, \end{aligned} \quad (3.5)$$

where \mathbf{V}' and Q' are the dual spaces of \mathbf{V} and Q respectively, and $\langle \cdot, \cdot \rangle$ designates the dual product. The operator B represents the divergence operator, and we can easily see that its adjoint operator is the gradient, indeed

$$\langle B^*p, \mathbf{v} \rangle = \langle B\mathbf{v}, p \rangle = b(\mathbf{v}, p) \quad \forall \mathbf{v} \in \mathbf{V}, \forall p \in Q. \quad (3.6)$$

3. Numerical solution

Furthermore, the adjoint operator is the transpose of the original operator, $B^* = B^\top$. Using (3.5-3.6) we rewrite (1.34) as

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + D(\mathbf{u}, \mathbf{u}) + A\mathbf{u} + B^\top p &= \mathbf{f} \quad \text{in } \mathbf{V}', \\ B\mathbf{u} &= 0 \quad \text{in } Q'. \end{aligned} \quad (3.7)$$

Let us now introduce a quasi-uniform regular triangulation \mathbb{T}_h of the domain Ω and the two families of finite-dimensional subspaces $\mathbf{V}_h \subset \mathbf{V}$ and $Q_h \subset Q$ that depend from the grid discretization parameter h . We need to assure some properties of these spaces in order to guarantee convergence of the solution as h tends to zero. In particular, if $\mathbf{v}_h \in \mathbf{V}_h$ and $q_h \in Q_h$ are approximations of $\mathbf{v} \in \mathbf{V}$ and $q \in Q$ respectively, then for $l \geq 1$ and for every $0 \leq r \leq l$ there must exist $C > 0$ such that

$$\begin{aligned} \inf_{\mathbf{v}_h \in \mathbf{V}_h} (\|\mathbf{v} - \mathbf{v}_h\|_0 + h\|\mathbf{v} - \mathbf{v}_h\|_1) &\leq Ch^{r+1}\|\mathbf{v}\|_{r+1}, \quad \forall \mathbf{v} \in \mathbf{H}^{r+1}(\Omega) \cap \mathbf{V}, \\ \inf_{\mathbf{v}_h \in \mathbf{V}_h} \|\mathbf{v} - \mathbf{v}_h\|_{1,p} &\leq Ch^r\|\mathbf{v}\|_{r+1,p}, \quad \forall \mathbf{v} \in \mathbf{W}^{r+1,p}(\Omega) \cap \mathbf{V}, \\ \inf_{q_h \in Q_h} \|q - q_h\|_0 &\leq Ch^r\|q\|_r, \quad \forall q \in H^r(\Omega) \cap Q. \end{aligned} \quad (3.8)$$

We also introduce the discretized version of (3.5) with

$$\begin{aligned} A_h : \mathbf{V}_h &\rightarrow \mathbf{V}'_h, & \langle A_h \mathbf{u}_h, \mathbf{v}_h \rangle &= a(\mathbf{u}_h, \mathbf{v}_h), & \forall \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h, \\ B_h : \mathbf{V}_h &\rightarrow Q'_h, & \langle B_h \mathbf{v}_h, p_h \rangle &= b(\mathbf{v}_h, p_h), & \forall \mathbf{v}_h \in \mathbf{V}_h, \forall p_h \in Q_h, \\ D_h : \mathbf{V}_h^2 &\rightarrow \mathbf{V}'_h, & \langle D_h(\mathbf{u}_h, \mathbf{u}_h), \mathbf{v}_h \rangle &= d(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h), & \forall \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h. \end{aligned} \quad (3.9)$$

The last hypothesis that the discretization must fulfill, in order to guarantee the convergence of the numerical solution to the variational one, is the so-called *InfSup* or *Ladyshenskaya-Babuska-Brezzi (LBB)* condition [11], that states that for \mathbf{V}_h and Q_h there must exist a constant $C > 0$ such that

$$\inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{(B_h \mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_1 \|q_h\|_0} \geq C. \quad (3.10)$$

We can finally write down the space-discretized version of (3.7) as

$$\begin{aligned} \frac{\partial \mathbf{u}_h}{\partial t} + D_h(\mathbf{u}_h, \mathbf{u}_h) - A_h \mathbf{u}_h + B_h^\top p_h &= i_h^\top \mathbf{f}, \\ B_h \mathbf{u}_h &= 0, \end{aligned} \quad (3.11)$$

where $i_h : \mathbf{V}_h \rightarrow \mathbf{V}$ is the continuous injection between \mathbf{V}_h and \mathbf{V} used to get $\mathbf{f}_h = i_h^\top \mathbf{f}$.

3.1.1. Temporal derivative discretization

We discretize the temporal derivative with a straightforward first-order finite difference scheme, where k denotes the time step and Δt is the time difference between two

3.1. Discretization of the Navier-Stokes equations

consecutive time steps, and write down the discretized version of (3.7) as

$$\begin{aligned} \frac{\mathbf{u}_h^{k+1} - \mathbf{u}_h^k}{\Delta t} + D_h(\mathbf{u}_h^{k+1}, \mathbf{u}_h^{k+1}) - A_h \mathbf{u}_h^{k+1} + B_h^\top p_h^{k+1} &= \mathbf{f}_h^{k+1}, \\ B_h \mathbf{u}_h^{k+1} &= 0. \end{aligned} \quad (3.12)$$

Recalling the definition of $d(\cdot, \cdot, \cdot)$ in (3.2), we see that both its components are non-linear. We must linearize these terms in order to include them in a matricial formulation of the discretized equation. The term that corresponds to the $c(\cdot, \cdot, \cdot)$ trilinear form must be linearize it in order to include it in a matricial formulation of the discretized equation. The easiest way to linearize the term that corresponds to the $c(\cdot, \cdot, \cdot)$ trilinear form is by taking the discretized advection term as $((\mathbf{u}_h^k \cdot \nabla) \mathbf{u}_h^{k+1}, \mathbf{v}_h)$ with a first order approximation. We instead implement a second order approximation in the form

$$((\mathbf{u}_h^k \cdot \nabla) \mathbf{u}_h^{k+1}, \mathbf{v}_h) + ((\mathbf{u}_h^{k+1} \cdot \nabla) \mathbf{u}_h^k, \mathbf{v}_h) - ((\mathbf{u}_h^k \cdot \nabla) \mathbf{u}_h^k, \mathbf{v}_h), \quad (3.13)$$

that is symmetric and guarantees faster convergence when the advection term plays a strong role on the evolution of the system. We will denote $D(\mathbf{u}_h^{k+1/2}, \mathbf{u}_h^{k+1/2})$ this second order approximation. The other term is linearized with

$$(\nabla \cdot \mathbf{u}_h, \mathbf{u}_h \cdot \mathbf{v}_h) = (\nabla \cdot \mathbf{u}_h^k, \mathbf{u}_h^{k+1} \cdot \mathbf{v}_h). \quad (3.14)$$

3.1.2. Algebraic formulation of the Navier-Stokes equations

At this point we have all the instruments needed to introduce the linear algebraic system associated with the original Navier-Stokes problem (3.1) that can be solved numerically. Let $\{\varphi_j\}$ and $\{\phi_k\}$ be some bases for V_h and Q_h , respectively. Following the Galerkin formulation we write the unknowns u_h and p_h as

$$u_h(\mathbf{x}) = \sum_{j=1}^{n_V} u_j \varphi_j(\mathbf{x}), \quad p_h(\mathbf{x}) = \sum_{k=1}^{n_Q} p_k \phi_k(\mathbf{x}), \quad (3.15)$$

where n_V and n_Q designates the dimension of the spaces V_h and Q_h . We select as test functions the basis functions and write the associated linear system as

$$\begin{aligned} M \mathbf{u} + B^\top p &= \mathbf{f}, \\ B \mathbf{u} &= 0, \end{aligned} \quad (3.16)$$

where the matrices are defined by

$$M = (m_{ij}) = \left(\frac{\varphi_i \varphi_j}{\Delta t} + a(\varphi_i, \varphi_j) + d(\mathbf{u}^k, \varphi_i, \varphi_j) \right), \quad (3.17)$$

$$B = (b_{ij}) = (b(\varphi_i, \phi_j)), \quad (3.18)$$

$$\mathbf{u} = (u_j), \quad (3.19)$$

$$p = (p_j), \quad (3.20)$$

$$\mathbf{f} = (f_j) = \left(f_j + \frac{u^k \varphi_j}{\Delta t} \right). \quad (3.21)$$

3. Numerical solution

We can also define

$$A = \begin{pmatrix} M & B^\top \\ B & 0 \end{pmatrix}, \quad (3.22)$$

the global matrix of the system. A necessary and sufficient condition for the solution of (3.16) to exist is that $\det(A) \neq 0$. It can be demonstrated that this property is equivalent to the LBB condition (3.10). It can also be proved that M is non-singular and positive-definite, so that the determinant of A is different from zero when the kernel of B^\top is zero-dimensional. When this is not true there exists a vector $\tilde{p} \in \mathbb{R}^{n_Q}$ different from zero such that $B^\top \tilde{p} = 0$, so to say $b(\varphi_m, \tilde{p}) = 0$. This is equivalent to $b(v_h, \tilde{p}) = 0, \forall v_h \in V_h$, that violates the LBB condition. On the other way, if the LBB condition is not valid, we can find a \tilde{q} in Q_h such that $b(v_h, \tilde{q}) = 0, \forall v_h \in V_h$. If (u_h, p_h) is a solution of (3.16), also $(u_h, p_h + \tilde{q})$ is a solution. This means that the solution for the pressure is not unique, and we can get the so-called *spurious modes* that lead to an unstable solver. For this reason the finite element spaces must be carefully selected in order to guarantee that the pressure solution is unique.

The minimal properties of p imposed by the variational formulation (3.3) do not include continuity. This means that the lower space we can use for the pressure is \mathbb{P}_0 . Since the velocity appears with one order of derivation more than the pressure, we can use \mathbb{P}_1 polynomials for its representation. This couple does not satisfy the LBB condition, and can be used only with a proper method that assures the elimination of the spurious pressure solutions. If we want to keep a discontinuous pressure, we must couple \mathbb{P}_0 with \mathbb{P}_2 . If we instead want a continuous pressure solution, we can not use \mathbb{P}_1 as the approximation space for both p and u . A suitable couple is \mathbb{P}_1 for p and \mathbb{P}_2 for u , and we will use this one in the rest of this work.

Once obtained an algebraic version (3.16) with the solution granted, we can solve the system as any linear system by inverting the matrix A . The sparsity of the matrix suggests to use an iterative solver. We define the sparsity ratio as the ratio between the number of elements different from zero in the matrix and the total number of elements. We will now analyze some basic techniques to deal with this system. There are two main families of solver used for the Navier-Stokes system (3.12). A feature that distinguishes between the two is if the velocity and pressure fields are solved together, or if the two fields are solved using the other variable as a known value. The former solvers are called *coupled*, while the latter ones are often referred to as *segregated* or *split* solvers.

3.2. Coupled solvers

These algorithms solve the system (3.12) as a whole and are suitable for any system of differential equations. We will analyze shortly the principal characteristics of these techniques with respect to the application to the Navier-Stokes equations. From now on we will drop the h subscript for clarity.

3.2.1. Domain Decomposition Methods

Domain decomposition methods try to reduce the complete boundary problem to a set of sub-problems on smaller domains or group of variables. Then, the boundaries between the sub-domains are updated with the value coming from the adjacent regions and the global solution is obtained iteratively. This is different from what is done in a segregated algorithm, where the splitting is made directly in the equations (see Sec. 3.3). The algorithms which are described in this section are also frequently adopted as preconditioner and smoother for other methods, in this case they are used with a fixed and small number of iterations.

Schur complement

Let us rewrite the discretized system (3.16) in the matricial form as

$$\begin{pmatrix} M & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}. \quad (3.23)$$

Our goal is to make a block diagonalization, so we left multiply the system with a suitable matrix to get

$$\begin{pmatrix} I & 0 \\ -BM^{-1} & I \end{pmatrix} \begin{pmatrix} M & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} I & 0 \\ -BM^{-1} & I \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}. \quad (3.24)$$

This choice leads to

$$\begin{pmatrix} M & B^\top \\ 0 & -BM^{-1}B^\top \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ -BM^{-1}\mathbf{f} \end{pmatrix}. \quad (3.25)$$

The *Schur complement* S is defined as the the matrix $(-BM^{-1}B^\top)$. We can see that in this form the solution of the pressure field does not depend on the velocity field. This means that, if we are able to build the S matrix, we can solve directly the pressure field and then substitute it in the momentum equation, that remains unchanged in this process. The difficulty in the calculation of the Schur complement resides in the fact that the calculation of M^{-1} can be as complex as the solution of the complete system. However, the Schur complement inherits the properties of M , that is, if M is symmetric positive defined, so is S . The building of the Schur complement is often approximated using an approximated inverse of M and used as a preconditioner.

Vanka-type smoother

A class of solvers that can show an excellent rate of convergence is named Vanka-type smoother, because it is often implemented as a pre-smoothing algorithm for a generic solver. The idea is to split the domain in several problems over small overlapping domains. The algorithm can be seen as a block Gauss-Seidel algorithm where each iteration is the solution of a block corresponding to one of the subdomain introduced.

3. Numerical solution

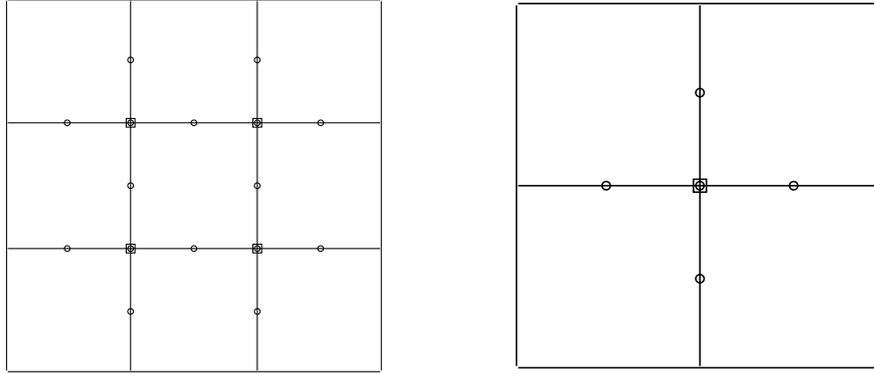


Figure 3.1.: Left: Vanka subdomain made of 3×3 elements. Right: Vanka subdomain made of 2×2 elements. Unknowns are localized on the grid (\circ : velocity, \square : pressure).

Since the subdomain are solved independently, this algorithm is very suited for parallel implementations. In the framework of finite elements the subdomain can be identified directly with a small group of elements. We call Ω_i the generic subdomain, and with \mathbb{T}_v the partition made of all the subdomains, $\mathbb{T}_v = \{\Omega_i\}$, that can coincide with the finite element partitioning \mathbb{T}_h . It must hold

$$\bigcup_{i=0}^m \Omega_i = \Omega, \quad (3.26)$$

where each Ω^i has a smooth boundary Γ^i and the subdomains are overlapping in the sense that $\mathbf{H}_0^1(\Omega) = \mathbf{H}_0^1(\Omega_1) + \mathbf{H}_0^1(\Omega_2) + \dots + \mathbf{H}_0^1(\Omega_m)$. On each subdomain consistent boundary conditions must be applied. If the selected subdomain is made up of a 3×3 stencil of elements in a two-dimensional domain, the unknowns updated are the ones shown in the left of Fig. 3.1 for for QUAD9/QUAD4 elements, with the boundary of the stencil fixed with Dirichlet boundary conditions. In order to update all the variables we need to repeat the procedure on every finite element of the grid, thus the domains overlap. If we select a subdomain made up of four cells, there is no necessity to extend the computational domain, and the Dirichlet conditions are applied directly on the edges of the subdomain, as shown in the right of Fig. 3.1. The three-dimensional extension is straightforward. After the solution on each subdomain, the boundary values are updated from the neighboring elements and the iteration is repeated until convergence. Since the subdomain Ω_i is small, the local solution of the Navier-Stokes equations can be executed with a direct solver, for example a Schur complement approach. Convergence properties and accuracy of the solution are analyzed in the literature [12].

3.2.2. Krylov subspace methods

The Krylov subspace method is a particular application of the *projection method* in which we seek a solution for a n -dimensional linear system $Ax = b$ in a subspace of \mathbb{R}^n .

If the dimension of this subspace is m , we must impose m independent constraints to get the solution. The general approach is to impose the orthogonality with m linearly independent vectors. In Krylov subspace methods the subspace \mathcal{K}_m chosen is

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}, \quad (3.27)$$

where $r_0 = b - Ax_0$ is the initial residual obtained with an arbitrary initial guess x_0 . This choice leads to an approximate solution x_m in the form

$$A^{-1}b \approx x_m = x_0 + q_{m-1}(A)r_0, \quad (3.28)$$

where q_{m-1} is a polynomial of degree $m - 1$ (dependent on the specific implementation of the Krylov subspace). If we choose $x_0 = 0$ we can see that

$$A^{-1}b \approx q_{m-1}(A)b, \quad (3.29)$$

meaning that the approximation we take for A^{-1} is $q_{m-1}(A)$. Even if all Krylov subspace methods share this approximation of the inverse of A , the selected constraints impact greatly on the iterative properties of the method. If we call \mathcal{L}_m the space defined by the constraints, we can identify two great families of methods, the first with $\mathcal{L}_m = \mathcal{K}_m$ and the minimum-residual variation $\mathcal{L}_m = A\mathcal{K}_m$, the second with $\mathcal{L}_m = \mathcal{K}_m(A^\top, r_0)$.

Arnoldi's procedure is a commonly adopted algorithm for building an orthogonal basis of the Krylov subspace. It can be described with the following pseudo-algorithm

- choose v_1 such that $\|v_1\| = 1$,
- for $j = 1, 2, \dots, m$:
 - compute $h_{ij} = (Av_j, v_i)$ for $i = 1, 2, \dots, j$,
 - compute $w_j = Av_j - \sum_{i=1}^j h_{ij}v_i$,
 - $h_{j+1,j} = \|w_j\|_2$,
 - if $h_{j+1,j} = 0$ stop,
 - $v_{j+1} = w_j/h_{j+1,j}$,
- endfor.

It is an iterative method that uses the classic Gram-Schmidt orthogonalization method to obtain a basis $\{v_i\}$ for

$$\mathcal{K}_m = \text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}. \quad (3.30)$$

We denote with V_m an $n \times m$ matrix with the i -th column set to v_i and by \bar{H}_m an $(m + 1) \times m$ Hessemberg matrix where h_{ij} comes from the Arnoldi's algorithm. Let also H_m be the matrix obtained from \bar{H}_m by deleting the last row. The following relation holds

$$AV_m = V_m H_m + w_m e_m^\top = V_{m+1} \bar{H}_m, \quad (3.31)$$

$$V_m^\top AV_m = H_m. \quad (3.32)$$

3. Numerical solution

Generalized Minimum RESidual (GMRES) method

The GMRES method takes $\mathcal{K} = \mathcal{K}_m$, $\mathcal{L} = A\mathcal{K}_m$ and $v_1 = r_0/\|r_0\|_2$. We note that any vector in $x_0 + \mathcal{K}_m$ can be written as

$$x = x_0 + V_m y, \quad (3.33)$$

with y a vector of m components. We can also define $J(y)$ as

$$J(y) = \|b - Ax\|_2 = \|b - a(x_0 + V_m y)\|_2. \quad (3.34)$$

Using the relation (3.31), valid for any Krylov subspace method, we can get

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) = r_0 - AV_m y = \\ &= \beta v_1 - V_{m+1} \bar{H}_m y = V_{m+1}(\beta e_1 - \bar{H}_m y), \end{aligned} \quad (3.35)$$

therefore

$$J(y) = \|\beta e_1 - \bar{H}_m y\|_2, \quad (3.36)$$

where β is a real number different from zero. The GMRES approximation is the vector in $x_0 + \mathcal{K}_m$ which minimizes J . The algorithm gives an approximation $x_m = x_0 + V_m y_m$ calculated with the y_m that minimizes (3.36). The minimizer y_m comes from the least-square solution of the inexpensive $(m + 1) \times m$ system.

3.2.3. Preconditioning

Preconditioning is a technique frequently adopted in numerical resolution of linear systems. The aim is to manipulate the system matrix in order to get an equivalent matrix that has better properties in terms of convergence and easiness of inversion. The simplest form of preconditioning we can think of is to multiply every row of the matrix A such that the resulting matrix has all ones on the diagonal. This can lead to faster convergence of the solver because all equations of the system are of the same order of magnitude because the algorithm is less prone to numerical cancellations. Theoretical results in preconditioning field are difficult to get and in real cases this simple technique can even lead to slower converging algorithms.

In general we can define a preconditioning method as a matrix T that premultiplies the system

$$TAx = Tb. \quad (3.37)$$

The construction of T can involve complex elaborations of the elements of A , or be based on physical considerations on the system we are studying. Either way, we should always look for a trade-off between the quality of the preconditioner and the computational costs of its building. Even if there exist special preconditioners oriented to the Navier-Stokes system, in general they are of difficult application to sparse matrices, since they can lead to less sparse matrix after their application. Other techniques, such as the ILU factorization, are instead oriented to generic matrices, but they can also be used for sparse matrices if they keep the sparsity ratio low.

Incomplete LU (ILU) factorization

The ILU method is based on a partial Lower-Upper factorization [13]. Instead of finding the two matrix L and U that are strictly lower and upper triangular, we get a decomposition such that

$$A = LU - R, \quad (3.38)$$

where R is the residual or error of the factorization. The level of factorization affects the acceleration gained in the successive solver, but it is obviously more expensive to calculate as the factorization improves. In general a very accurate factorization is favorable because it leads to more stable and robust solvers afterwards.

If we apply the Gaussian elimination algorithm directly, we fill empty spaces on the matrix structure, incrementing its sparsity ratio. Alternatively, we can perform the Gaussian elimination by dropping some off-diagonal elements, i.e. we put them to zero and the dropped values go into the residual matrix R . The pattern of the zeroed values must not contain diagonal elements and can be specified statically at the beginning of the algorithm as a partition P of $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$, that is

$$P \subset \{(i, j) : i \neq j, 1 \leq i, j \leq n\}. \quad (3.39)$$

It can be demonstrated that the incomplete factorization obtained with any pattern P is a regular splitting of A [13].

If P is chosen exactly as the zero pattern of the matrix A we get the so-called *zero fill-in ILU (ILU(0))* decomposition. This approach preserves exactly the sparsity pattern of A . We can define an ILU(p) technique ($p = 1, 2, \dots$) as the algorithm that allows p -th order fill-ins in the matrix. With $p = 1$ the first order fill-ins are the position different from zero of the product of the L and U matrices obtained with the ILU(0) algorithm, and by induction for other p . These approaches are suitable for matrices with a regular sparsity pattern, such as discretized differential equations, for which the sparsity pattern of the ILU(p) algorithm is predictable. There are also modified ILU approaches for which the element in the zero pattern are not simply discarded, but are used in some compensating algorithms.

3.3. Segregated (split) solvers

Segregated solvers are specifically designed for the Navier-Stokes equations and try to simplify the solution of the system by splitting the solution for the velocity and pressure fields. This division not only leads to smaller matrices to work with, but also converts a complex saddle point problem in two simpler elliptic problems, with higher stability and faster convergence. We have to remark, however, that, in order to get the split equations, we operate an arbitrary decomposition that can lead to unphysical results if not analyzed accurately. We will discuss these issues after showing the split equations.

3. Numerical solution

3.3.1. The incremental pressure-correction scheme

The projection of the equations is obtained by considering an intermediate velocity \mathbf{q} that satisfies an elliptic boundary value problem similar to the momentum balance equation, but without requiring that \mathbf{q} is divergence-free. Afterwards, the pressure field is calculated through an inviscid equation with suitable boundary conditions, and the final velocity \mathbf{u} is computed from \mathbf{q} and p . The split system of equations comes from (3.12) where we formally add and subtract the term $\mathbf{q}^{k+1}/\Delta t$ and split it in the following way

$$\begin{aligned}\frac{\mathbf{q}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{u}^{k+1}) - A\mathbf{u}^{k+1} &= \mathbf{f}^{k+1}, \\ \frac{\mathbf{u}^{k+1} - \mathbf{q}^{k+1}}{\Delta t} &= -B^\top p^{k+1}, \\ B\mathbf{u}^{k+1} &= 0,\end{aligned}\tag{3.40}$$

with appropriate boundary conditions. If we express \mathbf{u}^{k+1} from the second of (3.40), substitute it in the first and neglect the terms of higher order in Δt , we get

$$\begin{aligned}\frac{\mathbf{q}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A\mathbf{q}^{k+1} &= \mathbf{f}^{k+1}, \\ \frac{\mathbf{u}^{k+1} - \mathbf{q}^{k+1}}{\Delta t} &= -B^\top p^{k+1}, \\ B\mathbf{u}^{k+1} &= 0.\end{aligned}\tag{3.41}$$

We can also write the system in the incremental form for the pressure by introducing the increment $\delta p^{k+1} = p^{k+1} - p^k$ [14]. Following the same steps as before we get

$$\begin{aligned}\frac{\mathbf{q}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A\mathbf{q}^{k+1} &= \mathbf{f}^{k+1} - B^\top p^k, \\ \frac{\mathbf{u}^{k+1} - \mathbf{q}^{k+1}}{\Delta t} &= -B^\top \delta p^{k+1}, \\ B\mathbf{u}^{k+1} &= 0.\end{aligned}\tag{3.42}$$

Now we can use the second of (3.42), evaluated at the time step k , into the first to finally get

$$\begin{aligned}\frac{\mathbf{q}^{k+1} - \mathbf{q}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A\mathbf{q}^{k+1} &= \mathbf{f}^{k+1} - B^\top(p^k + \delta p^k) = \\ &= \mathbf{f}^{k+1} - B^\top(2p^k - p^{k-1}).\end{aligned}\tag{3.43}$$

We can also replace the last \mathbf{u} in the $D(\cdot, \cdot)$ operator neglecting higher order terms in Δt to get an equation for the intermediate velocity \mathbf{q} alone. Applying the divergence operator B to the second of (3.42) we get an equation for δp that depends only on \mathbf{q}

$$BB^\top \delta p^{k+1} = \frac{B\mathbf{q}^{k+1}}{\Delta t}.\tag{3.44}$$

3.3. Segregated (split) solvers

This equation requires the introduction of further boundary conditions on p , that are arbitrary and taken as $\partial p / \partial \mathbf{n} = 0$. We can therefore calculate in sequence \mathbf{q}^{k+1} , δp^{k+1} and \mathbf{u}^{k+1} from

$$\mathbf{u}^{k+1} = \mathbf{q}^{k+1} - \Delta t B^\top \delta p^{k+1}. \quad (3.45)$$

This approach is based on the orthogonal decomposition theorem [1], for which every vector field \mathbf{a} has a unique decomposition

$$\mathbf{a} = \mathbf{b} + \nabla \varphi, \quad (3.46)$$

with

$$\nabla \cdot \mathbf{b} = 0, \quad \left. \frac{\partial \mathbf{b}}{\partial \mathbf{n}} \right|_\Gamma = 0.$$

This means that \mathbf{q} can be decomposed in \mathbf{u} and p in a unique way.

The velocity field \mathbf{u}^{k+1} obtained with this technique is divergence-free, but does not satisfy the original boundary conditions as a consequence of the auxiliary Neumann conditions introduced to solve (3.44). If we calculate the normal component of (3.45) at the boundary we get

$$\left(\frac{\mathbf{u}^{k+1} - \mathbf{q}^{k+1}}{\Delta t} + B^\top \delta p^{k+1} \right) \cdot \mathbf{n} \Big|_\Gamma = \left. \frac{\partial \delta p^{k+1}}{\partial \mathbf{n}} \right|_\Gamma = 0, \quad (3.47)$$

so the boundary conditions applied to p are actually

$$B^\top \delta p^{k+1} \cdot \mathbf{n} \Big|_\Gamma = B^\top \delta p^k \cdot \mathbf{n} \Big|_\Gamma = \dots = B^\top \delta p^0 \cdot \mathbf{n} \Big|_\Gamma. \quad (3.48)$$

For the non-incremental scheme we would have found $B^\top p^{k+1} \cdot \mathbf{n} \Big|_\Gamma = 0$. Both conditions are non-physical and can lead to a numerical boundary layer resulting in an inexact value of the solution near the boundary.

3.3.2. The rotational incremental pressure-correction scheme

The rotational incremental pressure-correction scheme tries to decrease the error introduced by the auxiliary Neumann boundary conditions described in the previous section [15]. The equation for \mathbf{q} is again the (3.42)

$$\frac{\mathbf{q}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A \mathbf{q}^{k+1} = \mathbf{f}^{k+1} - B^\top p^k, \quad (3.49)$$

with $\mathbf{q}^{k+1} \Big|_\Gamma = 0$, but a modified projection step is considered in which

$$\frac{\mathbf{u}^{k+1} - \mathbf{q}^{k+1}}{\Delta t} = -B^\top \phi^{k+1}, \quad (3.50)$$

$$B \mathbf{u}^{k+1} = 0, \quad (3.51)$$

where $\phi^{k+1} = p^{k+1} - p^k + \nu B \mathbf{q}^{k+1}$ and $\mathbf{u}^{k+1} \cdot \mathbf{n} \Big|_\Gamma = 0$.

3. Numerical solution

By substituting the velocity from (3.50) in (3.49) as before we get

$$\frac{\mathbf{q}^{k+1} - \mathbf{q}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A\mathbf{q}^{k+1} = \mathbf{f}^{k+1} - B^\top(p^k + \phi^k), \quad (3.52)$$

with $\mathbf{q}^{k+1}|_\Gamma = 0$, while equation (3.44) is substituted by

$$BB^\top \phi^{k+1} = \frac{B\mathbf{q}^{k+1}}{\Delta t}, \quad (3.53)$$

with $\partial\phi^{k+1}/\partial\mathbf{n}|_\Gamma = 0$. This formulation mitigates some of the drawbacks highlighted in the previous section. To show this, we sum (3.49) and (3.50) to get

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) - A\mathbf{q}^{k+1} + B^\top \nu B\mathbf{q}^{k+1} = \mathbf{f}^{k+1} - B^\top p^{k+1}. \quad (3.54)$$

We consider for simplicity the case with ν constant. Using the vector identity $\nabla \times \nabla \times \mathbf{a} = \nabla(\nabla \cdot \mathbf{a}) - \nabla^2 \mathbf{a}$ and noting from (3.50) that $\nabla \times \nabla \times \mathbf{q}^{k+1} = \nabla \times \nabla \times \mathbf{u}^{k+1}$ we get

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{q}^{k+1}) + \nu \nabla \times \nabla \times \mathbf{u}^{k+1} + B^\top p^{k+1} = \mathbf{f}^{k+1}. \quad (3.55)$$

Looking at the Stokes problem associated with (3.55) or just introducing the further approximation $\mathbf{u}^k \approx \mathbf{q}^k$ which is of higher order in Δt , we can get for the normal component of the pressure at the boundary

$$\left. \frac{\partial p^{k+1}}{\partial \mathbf{n}} \right|_\Gamma = \left(\mathbf{f}^{k+1} - \nu \nabla \times \nabla \times \mathbf{u}^{k+1} \right) \cdot \mathbf{n} \Big|_\Gamma, \quad (3.56)$$

which is a boundary condition consistent with the original problem. Therefore this technique solves the problem on the pressure field boundary condition, at least for the normal component. On the contrary, the tangential boundary condition on the velocity field is still wrong, and it requires further analysis.

3.3.3. The penalty method

The basic idea of the penalty method is to replace a constrained optimization problem by a series of unconstrained problems whose solution ideally converges to the solution of the original constrained problem. Let us consider the following constrained optimization problem, that is to minimize the functional $J(\mathbf{u})$ subject to the constraints

$$G_i(\mathbf{u}) = 0, \quad i = 1, 2, \dots, n. \quad (3.57)$$

We call $\gamma_i, i = 1, 2, \dots, n$ the set of Lagrangian multipliers and define the augmented Lagrangian functional P as

$$P(\mathbf{u}, \gamma_i) = J(\mathbf{u}) + \sum_{i=1}^n \langle \gamma_i, G_i(\mathbf{u}) \rangle, \quad (3.58)$$

3.3. Segregated (split) solvers

where $\langle \cdot, \cdot \rangle$ is again the dual product. In this analysis we will consider only regular function for which the dual product can be substituted by the inner product, that is necessary for a numerical implementation of this method. The functional P is clearly minimized by the extremal points of the functional $J(\mathbf{u})$. The solution \mathbf{u} of the problem satisfies the Euler equation associated to the new functional

$$J'(\mathbf{u}) + \sum_{i=1}^n (\gamma'_i, G_i(\mathbf{u})) + \sum_{i=1}^n (\gamma_i, G'_i(\mathbf{u})) = 0, \quad (3.59)$$

where the prime denotes the partial derivative with respect to \mathbf{u} . The assumed regularity lets us substitute the Gateaux derivative with the partial derivative.

We can approximate the original minimization problem by introducing a penalty method, where we consider a single term $S(\mathbf{u})$ such that

$$S(\mathbf{u}) = 0 \quad \text{if } G_i(\mathbf{u}) = 0, \quad S(\mathbf{u}) > 0 \quad \text{if } G_i(\mathbf{u}) \neq 0, \quad (3.60)$$

multiplied by a penalty parameter $\gamma^* > 0$. The augmented Lagrangian becomes $P = J + \gamma^* S(\mathbf{u})$. For the penalty function we may assume the form

$$S(\mathbf{u}) = \frac{1}{2} \sum_{i=1}^n (G_i(\mathbf{u}), G_i(\mathbf{u})). \quad (3.61)$$

With these conditions the penalty augmented Lagrangian functional becomes

$$P(\mathbf{u}) = J(\mathbf{u}) + \frac{\gamma^*}{2} \sum_{i=1}^n (G_i(\mathbf{u}), G_i(\mathbf{u})). \quad (3.62)$$

The penalty solution $\tilde{\mathbf{u}}$ satisfies the Euler equation

$$J'(\tilde{\mathbf{u}}) + \gamma^* \sum_{i=1}^n (G_i(\tilde{\mathbf{u}}), G'_i(\tilde{\mathbf{u}})) = 0. \quad (3.63)$$

Under some conditions, when γ^* tends to infinity $\tilde{\mathbf{u}}$ tends to the solution \mathbf{u} of (3.59) [16].

Now we apply the penalty method to (3.52) with the divergence-free constraint (3.51). We try to minimize the functional $J(\mathbf{s})$, with $\mathbf{s} = \mathbf{q}^{k+1}$, defined as

$$J(\mathbf{s}) = \frac{1}{2} \frac{(\mathbf{s} - \mathbf{s}^k)^2}{\Delta t} + \frac{1}{2} D(\mathbf{u}^k, \mathbf{s}^2) + \frac{1}{2} A \mathbf{s}^2 - (\mathbf{f}^{k+1} - B^\top(p^k + \phi^k)) \cdot \mathbf{s}, \quad (3.64)$$

subject to the single constraint

$$G_1(\mathbf{s}) = -B\mathbf{s} = 0. \quad (3.65)$$

The penalty function becomes

$$S(\mathbf{s}) = \frac{1}{2} G_1^2(\mathbf{s}), \quad (3.66)$$

3. Numerical solution

and the augmented Lagrangian functional P has the form

$$P(\mathbf{s}) = J(\mathbf{s}) + \frac{1}{2}\gamma^*G^2(\mathbf{s}) = \frac{1}{2}\frac{(\mathbf{s} - \mathbf{s}^k)^2}{\Delta t} + \frac{1}{2}D(\mathbf{u}^k, \mathbf{s}^2) + \frac{1}{2}A\mathbf{s}^2 + (\mathbf{f}^{k+1} - B^\top(p^k + \phi^k)) \cdot \mathbf{s} + \frac{1}{2}\gamma^*(B\mathbf{s})^2. \quad (3.67)$$

We find an extremum of (3.67) by solving

$$P'(\mathbf{s}) = J'(\mathbf{s}) + \gamma^*G'(\mathbf{s}) = 0, \quad (3.68)$$

or equivalently by setting equal to zero the first variation of P

$$\delta P = \left(\frac{\mathbf{s} - \mathbf{s}^k}{\Delta t} + D(\mathbf{u}^k, \mathbf{s}) + A\mathbf{s} - (\mathbf{f}^{k+1} - B^\top(p^k + \phi^k)) - \gamma^*B^\top B\mathbf{s} \right) \cdot \delta\mathbf{s} = 0, \quad (3.69)$$

together with the boundary condition $\delta\mathbf{s}|_\Gamma = 0$, which is consistent with $\mathbf{u}|_\Gamma = \mathbf{q}|_\Gamma = 0$. The first variation of P should be zero for any choice of $\delta\mathbf{s}$, thus we obtain

$$\frac{\mathbf{s} - \mathbf{s}^k}{\Delta t} + A\mathbf{s} + D(\mathbf{u}^k, \mathbf{s}) - \mathbf{f}^{k+1} + B^\top(p^k + \phi^k - \gamma^*B\mathbf{s}) = 0, \quad (3.70)$$

where the pressure term is substituted by

$$p = p^k + \phi^k - \gamma^*B\mathbf{s}. \quad (3.71)$$

Looking at (3.70) we note that when $\gamma^* \rightarrow \infty$, \mathbf{s} becomes a penalty approximation of the solution of the original Navier-Stokes equations (3.11) and the projection step may not be strictly necessary. However, very high values would lead to an ill-conditioned matrix and the convergence of the method worsen and could not be assured. Therefore we implement the penalty method on top of the rotational incremental pressure-correction scheme of Sec. 3.3.2.

3.4. Multigrid algorithm

A multigrid algorithm works on top of a classical solver using a hierarchy of discretizations, with the aim to reduce the computational effort which is required to obtain the solution [17]. The use of different grids gives the possibility to accelerate the convergence reducing the error of the approximate solution on the different wavelengths connected with the grid spacing. The basic idea is that the improvement of the solution on each cell is seen only by the adjacent cells. This requires a large number of timesteps for the solution to propagate through the whole domain. If we use a coarser grid, with a smaller number of elements, the solution can reach all the cells in a smaller number of time steps. The strength of this approach resides also in the fact that it can be applied to any kind of system, without any restriction. A well-implemented multigrid method often scales linearly with the number of unknowns, putting it among the fastest algorithms available.

We consider the general problem of a linear algebraic system $A\mathbf{x} = \mathbf{b}$ and rewrite it as a function of the residual $\mathbf{r} = A\tilde{\mathbf{x}} - \mathbf{b}$, where $\tilde{\mathbf{x}}$ is an approximate solution of the system. If \mathbf{x} is the exact solution of the system, then

$$A\mathbf{e} = A(\tilde{\mathbf{x}} - \mathbf{x}) = \mathbf{r} + \mathbf{b} - \mathbf{b} = \mathbf{r}, \quad (3.72)$$

with \mathbf{e} the error between the approximation and the exact solution. We now introduce two different levels, a coarse one and a fine one, denoted by the superscript c and f respectively. The resolution on the coarse grid is performed for the residual \mathbf{r} and then projected back to the fine grid to correct the solution. We need two operators that transport the vectors \mathbf{e} and \mathbf{r} between the two grids. We call them *restriction* R , that goes from fine to coarse, and *prolongation* P , that goes from coarse to fine. We can write for a generic vector \mathbf{x}

$$\mathbf{x}^c = R\mathbf{x}^f, \quad \mathbf{x}^f = P\mathbf{x}^c. \quad (3.73)$$

We can introduce an algebraic system on the coarse grid as

$$A^c\mathbf{e}^c = \mathbf{r}^c, \quad (3.74)$$

where A^c is the system matrix defined on the coarse mesh. A^c can be build in the same way as we do on the fine grid by considering the coarse elements. Alternatively, the coarse matrix can be derived in a purely algebraic fashion from the fine one

$$\begin{aligned} A^f\mathbf{e}^f &= \mathbf{r}^f, \\ A^fP\mathbf{e}^c &= \mathbf{r}^f, \\ RA^fP\mathbf{e}^c &= R\mathbf{r}^f = \mathbf{r}^c, \end{aligned} \quad (3.75)$$

so we can set $A^c = RA^fP$. This approach is named *Algebraic MultiGrid (AMG)* and is suitable for any generic matrix. The other approach, referred to as *Full Approximation System (FAS)* can offer better performances, since it is based on the grid structure and can take advantage of the physical properties of the system. Furthermore, since the AMG deals only with the already linearized system, it may behave poorly with systems with non-linearities, such as the Navier-Stokes system, while in the FAS discretization process is executed entirely on the coarse grid.

3.4.1. Multigrid cycles

The number of coarse levels can be extended beyond one to reduce the error on all the wavelengths available in the grid. The arrangement of the operations to be performed on each level is called *cycle* and each of them can be described with a combinations of the following elementary steps which depend on the set of parameters β_1 , β_2 and β_3 .

- ◇ Execute the solver on the fine grid for β_1 times. This iterations are called *pre-relaxation iterations* because are executed before going to the coarse level.

3. Numerical solution

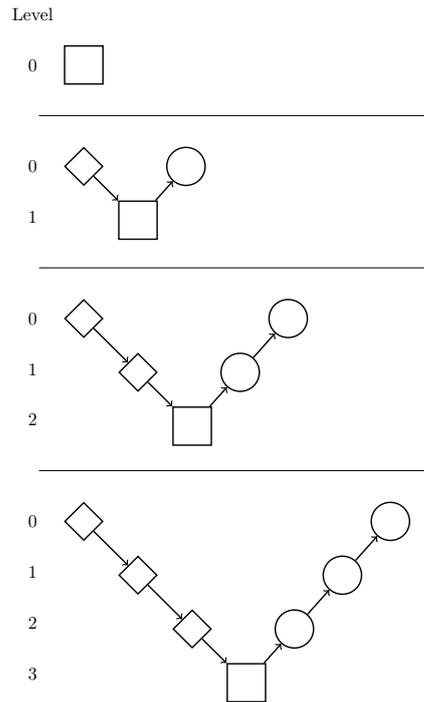


Figure 3.2.: V-cycle multigrid scheme.

- ↘ Restrict the problem to the coarse grid.
- Solve on the coarse grid for β_2 iterations.
- ↗ Prolong the solution on the fine grid.
- Solve on the fine grid for β_3 times. This iterations are the *post-relaxation iterations*.

The cycles that can be adopted are called V,W and Nested (also known as *Full Multigrid*). They differ in the sequence of operations when the number of coarse levels is greater than one, but they all reduce to the scheme described before when there is only one coarse level. The different cycles are described in Figs. 3.2, 3.3 and 3.4. The choice for β_1 , β_2 and β_3 can vary from case to case. In general, as a pre- and post-smoothing operator it can be adopted any preconditioner or solver which is available. On the coarser grid, especially when the number of coarse levels is high, it is often implemented a direct solver, such as the Gaussian elimination scheme or a Schur complement approach, that is not expensive since the number of nodes is very limited.

3.4.2. Prolongation and restriction

In the framework of finite element methods, the prolongation and restriction operators assume a particular form that is related to the shape functions defined on each element

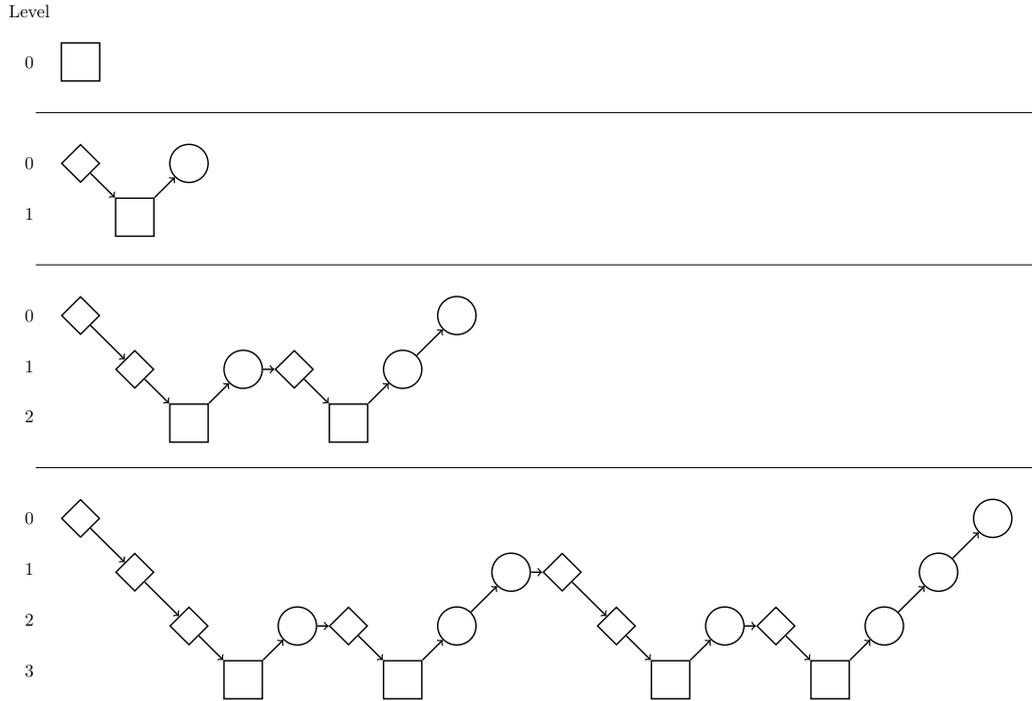


Figure 3.3.: W-cycle multigrid scheme.

[18]. Remembering the Galerkin discretized system of (3.16), these two operator are converted in rectangular matrices that work between the two levels. In particular, the restriction will have $n_c \times n_f$ elements, where n_c and n_f designate the number of nodes on the coarse and fine grids.

The simplest idea for a restriction is to locate a coarse node on top of a fine node. Therefore we can take the value of the fine node directly, so the i -th line of the restriction matrix R_i will look like

$$R_i = \{0, 0, \dots, 0, 1, 0, \dots, 0\}, \quad (3.76)$$

where the unique position different from zero is the corresponding i -node on the fine grid. If the common nodes are numbered coherently on both grids, we can see that

$$(R) = \begin{pmatrix} I_{n_c} \\ \emptyset \end{pmatrix}, \quad (3.77)$$

where I_{n_c} is the identity matrix of dimension n_c , while the lower part of the matrix is identically zero. Even if this approach is very simple, the results obtained are not very accurate, because all the nodes outside the coarse grid are ignored. When the number of levels grows, the number of nodes taken into account becomes smaller and smaller and the performance worsens. For this reason in our code we have implemented a *weighted* restriction operator, where the simple picking of the corresponding value on the fine grid

3. Numerical solution

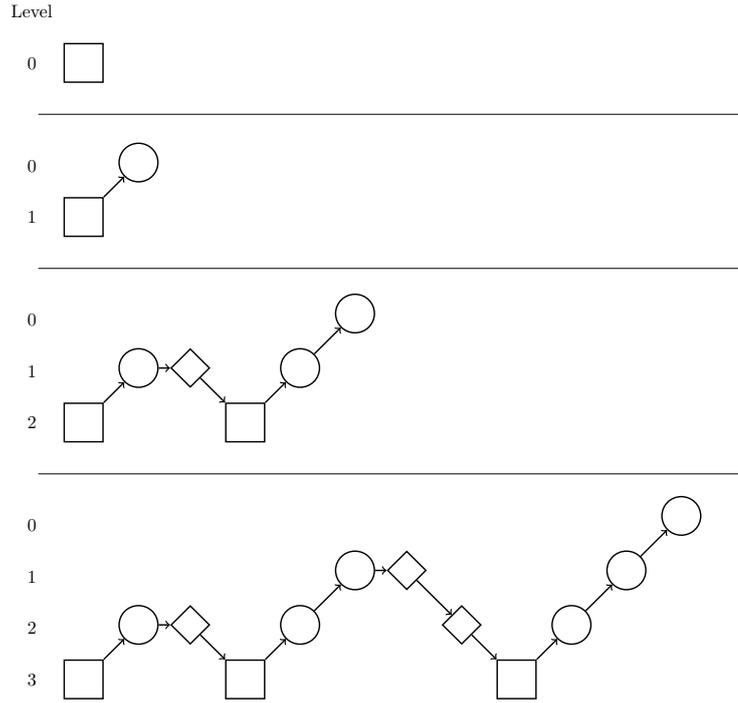


Figure 3.4.: Nested cycle multigrid scheme.

is substituted by a mean-value obtained by integration on a suitable support. In the finite element framework this support is chosen to be the coarse element where the node is placed. Remembering (3.15), we can easily get that each unknown $u^c(x_i) = u_i^c$ on the coarse grid is

$$u_i^c = \sum_{j=0}^{n_{sc}} r_{ij} u_j^f = \sum_{j=0}^{n_{sc}} \varphi_i^c(x_j^f) u_j^f, \quad (3.78)$$

where n_{sc} is the number of shape functions of the coarse element chosen. Each element r_{ij} of the restriction operator will be

$$r_{ij} = \varphi_i^c(x_j^f). \quad (3.79)$$

This value should be divided by the area of the element. In general, the shape functions used in the restriction process can be different from the one used in the solution approximation.

The projection operator must provide values of the solution on a number of points greater than the number of nodes on the coarse grid. Using again (3.15), we can write the value of the solution on any point of the computational domain, in particular each node on the fine grid. The support for any node is limited to the fine element. In this

way the generic unknown $u^f(x_i) = u_i^f$ on the fine grid is

$$u_i^f = \sum_{j=0}^{n_{sf}} p_{ij} u_j^c = \sum_{j=0}^{n_{sf}} \varphi_j^f(x_i^c) u_j^c, \quad (3.80)$$

where n_{sf} is the number of shapes on the fine element. We can infer that the element p_{ij} of the projection operator is

$$p_{ij} = \varphi_j^f(x_i^c). \quad (3.81)$$

We note that when an additional node is on the boundary of an element, then it can belong to more than one element. The value obtained with the projection just seen is the same, independently from the element in which it is calculated, because, as we have already seen in the shape function definition, a value calculated on a boundary depends only on the coarse node that are on that boundary when the approximation polynomial guarantees the continuity of the solution across the elements.

Part II.

Two-Phase Flow

4. Two-phase flow and interface capturing

4.1. Single fluid formulation of the Navier-Stokes equations for two-phase flow

The main application for which the finite element code has been developed is to perform direct numerical simulations (DNS) of basic two-phase phenomena that occur in many physical situations and industrial applications. Starting from the single phase equations that we have analyzed in the previous chapters and its numerical implementation, we now introduce a two-phase system and the mathematical model that governs its evolution. In the rest of the chapter we will stick to two-dimensional analysis.

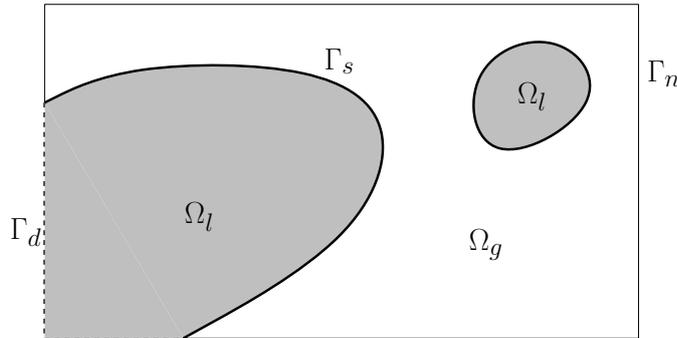


Figure 4.1.: Computational domain. The liquid in Ω_l is the reference phase and the boundary between the two phases is denoted by Γ_s . Dirichlet boundary conditions are applied on Γ_d and Neumann boundary conditions on Γ_n .

One of the formulations for the Navier-Stokes system is the *single fluid* formulation, in which the two-phase mixture is seen as a whole, with discontinuous physical properties at the interface of separation of the two phases, while they are supposed constant inside each of the two phases. We consider a domain like that on Fig. 4.1, where Ω , as before, designates the whole domain. Ω_l is the portion of the domain occupied by the reference phase, while $\Omega_g \subset \Omega$ is the subdomain with the secondary phase. The boundary between the two immiscible fluids is denoted by Γ_s , and its topology can vary during the evolution of the system since each subdomain evolves in time. The physical boundary is here subdivided in two zones, on which different boundary conditions are applied. The zone denoted by Γ_d has Dirichlet boundary conditions, while the zone named Γ_n has Neumann boundary conditions. The Navier-Stokes system is formally written as in single phase

4. Two-phase flow and interface capturing

formulation

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)) + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, t \in [0, T], \end{aligned} \quad (4.1)$$

but the difference is hidden in the definition of the density ρ , the viscosity μ and the force \mathbf{f} .

If we denote with l the properties of the reference phase and with g the values associated to the secondary phase, we define the physical properties ρ and μ as

$$\rho = \rho_l \chi + \rho_g(1 - \chi), \quad (4.2)$$

$$\mu = \mu_l \chi + \mu_g(1 - \chi), \quad (4.3)$$

where χ is the *characteristic function* or *indicator function*. This function describes the distribution of the two phases in the domain. It is equal to 1 in the reference phase and 0 in the secondary phase. We note that the function is discontinuous on the interface Γ_s . We can define χ as

$$\chi(\mathbf{x}, t) = \int_{\Omega_l(t)} \delta(\mathbf{x}' - \mathbf{x}) d\mathbf{x}' \quad \forall \mathbf{x} \in \Omega. \quad (4.4)$$

The indicator function is therefore a multidimensional Heaviside function that changes value on Γ_s . We can also write that

$$\nabla \chi = - \int_{\Gamma_s} \delta(\mathbf{x}' - \mathbf{x}) \mathbf{n}' dS' = -\mathbf{n} \int_{\Gamma_s} \delta(\mathbf{x}' - \mathbf{x}) dS' = -\mathbf{n} \delta_s(\mathbf{x}), \quad (4.5)$$

where $\delta_s(\mathbf{x})$ is the Dirac delta function that is discontinuous on Γ_s . Under the hypotheses of immiscible fluids with no phase change, the characteristic function behaves like a passive scalar and is purely transported by the velocity field, following the simple advection equation

$$\frac{\partial \chi}{\partial t} + (\mathbf{u} \cdot \nabla) \chi = 0, \quad \text{in } \Omega \times [0, T]. \quad (4.6)$$

The force term presents a sensible difference with respect to the single phase formulation, where it indicates only body forces such as gravity or electro-magnetic fields. Here we must take into account also the *surface tension*, that is modeled as a force applied only on the interface

$$\mathbf{f}_s(\mathbf{x}) = \int_{\Gamma_s} \sigma \kappa \mathbf{n} \delta_s(\mathbf{x}) dS, \quad (4.7)$$

where σ is a constant surface tension coefficient, as we do not consider temperature gradients or varying concentration of surfactants, κ the sum of the principal curvatures (in our convention $\kappa < 0$ for a liquid drop), \mathbf{n} the unit external normal to Γ_s and \mathbf{x}_s a point on Γ_s . The implementation of this term plays an important role in any two-phase model and it will be extensively analyzed in Sec. 4.4, because we first need to introduce a numerical modelization of the interface.

4.2. Volume-of-Fluid (VOF) method for interface capturing

The Volume-of-Fluid method is one of the most popular techniques adopted to model numerically an interface of separation between two phases. In this approach we define a *color function* C on each of the cells that are part of the computational domain. The value of C is taken as the integral of the characteristic function χ on the cell

$$C_i(t) = \frac{1}{\text{meas}(\Omega_i)} \int_{\Omega_i} \chi(\mathbf{x}, t) dV, \quad (4.8)$$

where Ω_i is one of the cells of the partition \mathbb{T}_h of Ω , and $\text{meas}(\Omega_i) = \int_{\Omega_i} dV$. It is easy to see that

$$\begin{cases} C_i(t) = 1 & \text{if } \Omega_i \subset \Omega_l, \\ C_i(t) = 0 & \text{if } \Omega_i \subset \Omega_g, \\ 0 < C_i(t) < 1 & \text{if } \Omega_i \cap \Gamma_s \neq \emptyset. \end{cases}$$

The interface of separation is located on the mixed cells. Once introduced the color function, we can define the physical properties at cell level

$$\rho_i = \rho_l C_i + \rho_g (1 - C_i), \quad (4.9)$$

$$\mu_i = \mu_l C_i + \mu_g (1 - C_i). \quad (4.10)$$

In each mixed cell, the interface is represented by a single segment that is oriented and positioned in order to approximate as well as possible the real interface. The overall reconstruction of the interface is therefore piece-wise linear. In order to advance in time the interface reconstruction, we integrate (4.6) on the cell Ω_i to get

$$\text{meas}(\Omega_i) \frac{\partial C_i}{\partial t} + \int_{\Gamma_i} \chi \mathbf{u} \cdot \mathbf{n} dS = 0, \quad (4.11)$$

where the integral is extended to the boundary Γ_i of Ω_i . Since this equation is still discontinuous, it can not be integrated with standard partial differential equation methods, that tend to diffuse the interface, and a geometrical approach is usually preferred. In summary, a VOF advection algorithm will require two steps

- **Reconstruction.** In every mixed cell a segment is placed to reproduce the interface. This step is further subdivided in two procedures

Orientation. The normal to the segment is determined from the color function distribution. A great number of schemes has been proposed, often based on finite differences.

Positioning. The segment is placed inside the mixed cell imposing the constraint that the underlying volume is equal to the color function value in the cell.

- **Advection.** Once we have a reconstruction segment in each cell, the interface is advanced in time using (4.11) and the color function is updated in each cell.

Since the volume in each cell is constrained to be the integral of the indicator function in that cell, the VOF method shows excellent mass conservation properties. We will now analyze in detail each step.

4. Two-phase flow and interface capturing

4.2.1. Interface reconstruction

The historically first reconstruction techniques available for VOF method were called *Single Line Interface Calculation (SLIC)* and only provided segments parallel to one of the edges of the cell boundary. In two-dimensions, this leads to an equation for the segment in the form

$$x = \alpha_1, \quad \text{or} \quad y = \alpha_2, \quad (4.12)$$

where α_1 and α_2 are set by the volume fraction conservation.

Modern techniques allow the segment to be oriented freely inside the cell and are therefore known as *Piecewise Linear Interface Calculation (PLIC)* methods. Sticking to the two-dimensional case, the segment line can be represented by

$$m_x x + m_y y = \alpha, \quad (4.13)$$

where $\mathbf{m} = (m_x, m_y)$ is a vector normal to the reconstruction segment. The volume conservation constraint sets the value of α . It is important to note that this imposition leads to an interface reconstruction that is not continuous across cell boundaries. Some of the most used techniques for the determination of \mathbf{m} rely on a discretized derivation of the color function distribution.

Parker-Youngs method

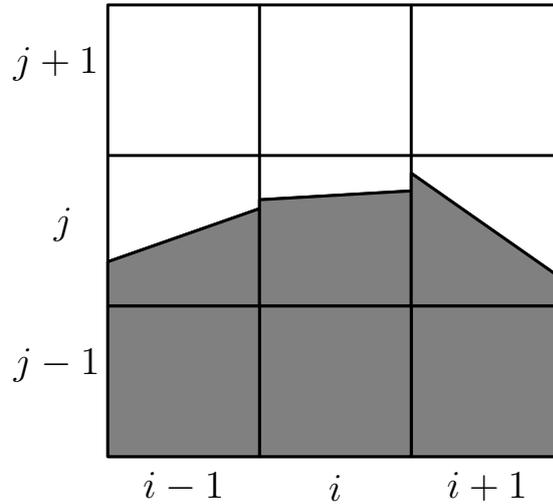


Figure 4.2.: Cell stencil used for the reconstruction.

We illustrate the method in two dimensions by considering a Cartesian grid where each cell is identified by the couple of indices (i, j) of integer values. Let us consider a 3×3 stencil of cells around the cell (i, j) , as shown in Fig. 4.2. The normal \mathbf{m} is first

4.2. Volume-of-Fluid (VOF) method for interface capturing

computed on the four cell vertices. For example, in the upper right corner, identified by the indices $(i + \frac{1}{2}, j + \frac{1}{2})$, we get

$$\begin{aligned} m_{x,i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{1}{2h_x}(C_{i+1,j} - C_{i,j} + C_{i+1,j+1} - C_{i,j+1}), \\ m_{y,i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{1}{2h_y}(C_{i,j+1} - C_{i,j} + C_{i+1,j+1} - C_{i+1,j}), \end{aligned} \quad (4.14)$$

where h_x and h_y are the grid steps in the two coordinate directions. When $h_x = h_y$ the normal at the cell center is obtained by taking the average of the four vertex values

$$\mathbf{m}_{ij} = \frac{1}{4}(\mathbf{m}_{i+\frac{1}{2},j-\frac{1}{2}} + \mathbf{m}_{i-\frac{1}{2},j-\frac{1}{2}} + \mathbf{m}_{i+\frac{1}{2},j+\frac{1}{2}} + \mathbf{m}_{i-\frac{1}{2},j+\frac{1}{2}}), \quad (4.15)$$

while the averaging changes slightly when $h_x \neq h_y$. This method is quite simple and has the great advantage of being easily extended to three-dimensional domains. Furthermore, when the resolution is low it shows better performances than many other methods which are more complex [19].

ELVIRA method

ELVIRA stands for *Efficient Least-squares Volume-of-fluid Interface Reconstruction Algorithm* [20] and uses the same stencil of cells of the previous method. We can calculate a discretized value of the height y on each of the columns of the stencil as the sum of the volume fractions on that column, $h_x y_i = \sum_{k=-1}^1 C_{i,j+k} h_x h_y$. If we consider the approximation $y = m_x x + \alpha$ on the central cell (i, j) , we can choose for m_x between the three values obtained with a backward (m_{xb}), centered (m_{xc}) or forward (m_{xf}) finite difference method given by

$$m_{xc} = \frac{1}{2h_x}(y_{i+1} - y_{i-1}) = \frac{1}{2h_x} \sum_{k=-1}^1 (C_{i+1,j+k} - C_{i-1,j+k}), \quad (4.16a)$$

$$m_{xf} = \frac{1}{h_x}(y_{i+1} - y_i) = \frac{1}{h_x} \sum_{k=-1}^1 (C_{i+1,j+k} - C_{i,j+k}), \quad (4.16b)$$

$$m_{xb} = \frac{1}{h_x}(y_i - y_{i-1}) = \frac{1}{h_x} \sum_{k=-1}^1 (C_{i,j+k} - C_{i-1,j+k}). \quad (4.16c)$$

We can also repeat the argument for the horizontal direction and approximate the interface with the line $x = m_y y + \alpha$. In this case we get for m_y

$$m_{yc} = \frac{1}{2h_y}(x_{j+1} - x_{j-1}) = \frac{1}{2h_y} \sum_{k=-1}^1 (C_{i+k,j+1} - C_{i+k,j-1}), \quad (4.16d)$$

$$m_{yf} = \frac{1}{h_y}(x_{j+1} - x_j) = \frac{1}{h_y} \sum_{k=-1}^1 (C_{i+k,j+1} - C_{i+k,j}), \quad (4.16e)$$

$$m_{yb} = \frac{1}{h_y}(x_j - x_{j-1}) = \frac{1}{h_y} \sum_{k=-1}^1 (C_{i+k,j} - C_{i+k,j-1}). \quad (4.16f)$$

4. Two-phase flow and interface capturing

We consider each of the six cases (4.16) and use them to reconstruct the line on the whole 3×3 stencil of cells defining in this way an approximated volume fraction distribution \tilde{C} . We consider the discretized error E in L_2 between the real data C and the approximated values \tilde{C}

$$E(\tilde{m}) = \left(\sum_{k=-1}^1 \sum_{l=-1}^1 (\tilde{C}_{i+k,j+l}(\tilde{m}) - C_{i+k,j+l})^2 \right)^{\frac{1}{2}}, \quad (4.17)$$

where \tilde{m} is one of the coefficients defined in (4.16). The value of \tilde{m} that minimizes E is chosen as the normal of the segment. This technique reproduces exactly any linear interface [20] and shows better convergence rates in basic VOF tests when the resolution is not too small. The three-dimensional equivalent of this algorithm requires however a stencil of cells that extends to 5 cells in each direction, in order to reproduce exactly any planar interface.

Segment positioning

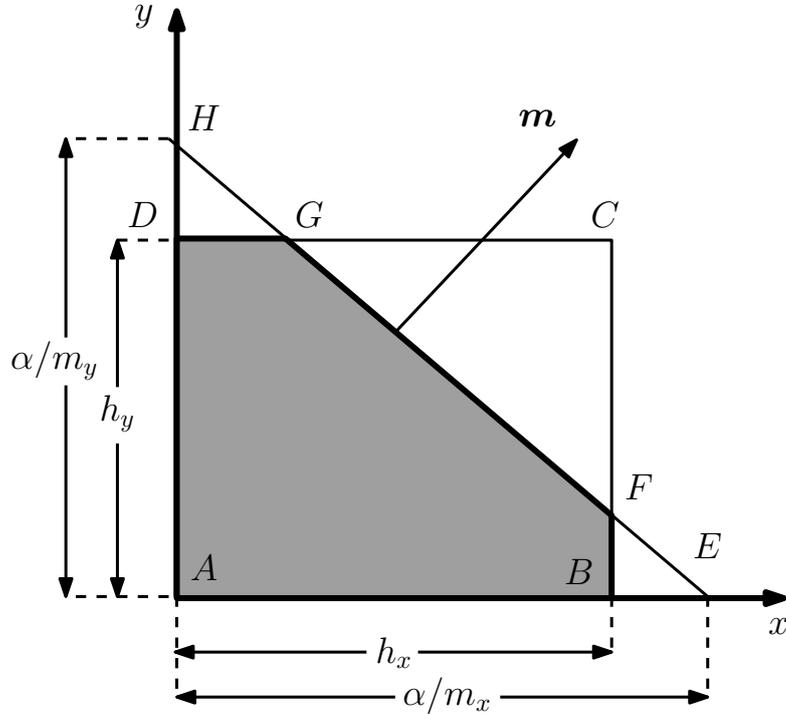


Figure 4.3.: The reference phase occupies the area of the pentagon $ABFGD$.

The correct positioning of the segment in the cell can be obtained by geometrical considerations, that lead to a unique relation between α and the color function value C_{ij} . With reference to Fig. 4.3, we want to calculate the area of the pentagon $ABFGD$. We can suppose that m_x and m_y are both positive, even if this is not the case we can apply

4.2. Volume-of-Fluid (VOF) method for interface capturing

some mirror reflections to go back to the reference situation. The area of the triangle AEH is $\alpha^2/(2m_x m_y)$. If the points E and H are inside the cell, the reference phase occupies exactly the area of this triangle. When the points move outside the cell, we must subtract the areas of the triangles BEF and DGH . Therefore we get

$$A_1(\alpha) = \frac{\alpha^2}{2m_x m_y} \left[1 - H(\alpha - m_x h_x) \left(\frac{\alpha - m_x h_x}{\alpha} \right)^2 - H(\alpha - m_y h_y) \left(\frac{\alpha - m_y h_y}{\alpha} \right)^2 \right], \quad (4.18)$$

where $H(x)$ is the *Heaviside function* and $A_1 = h_x h_y C_{ij}$. The second term is different from zero when E is outside the cell, namely $\alpha > m_x h_x$, while the third appears when H is beyond D , $\alpha > m_y h_y$. The area of this two smaller triangles can be easily computed noting that they are similar to AEH . We get

$$\frac{\text{meas}(BEF)}{\text{meas}(AEH)} = \left(\frac{\alpha - m_x h_x}{\alpha} \right)^2, \quad \frac{\text{meas}(DGH)}{\text{meas}(AEH)} = \left(\frac{\alpha - m_y h_y}{\alpha} \right)^2.$$

We remark that (4.18) is a strictly monotonic function, and is a polynomial of first or second order depending on the Heaviside functions into play. The properties of (4.18) guarantee that the inverse function $\alpha = \alpha(C)$ exists and can be determined easily [21].

4.2.2. Interface advection

The algorithms to propagate the interface can be divided in two broad categories. *Split* algorithms decompose the motion along the coordinate directions and advance the color data separately, by creating an intermediate field \tilde{C} after each of the steps, while *unsplit* algorithms define two-dimensional fluxes and advance the volume fraction distribution in a single step. Three dimensional algorithm of this type are still too complex geometrically. We will now describe some of mono-dimensional techniques implemented in the code, and an innovative two-dimensional unsplit algorithm.

Eulerian implicit method

We recall (4.6) and rewrite it in the conservative form

$$\frac{\partial \chi}{\partial t} + \nabla \cdot (\chi \mathbf{u}) = \chi \nabla \cdot \mathbf{u} = 0. \quad (4.19)$$

For a split algorithm we can consider the mono-dimensional case in the x direction and write for the cell (i, j)

$$h_x h_y \frac{\partial C_{ij}(t)}{\partial t} + \oint_{\Gamma_{ij}} \chi(\mathbf{x}, t) \mathbf{v} \cdot \mathbf{n} dl = h_x h_y C_{ij} \frac{\partial u}{\partial x}, \quad (4.20)$$

that is the integration of (4.19) on the cell. The term $\partial u / \partial x$ can be assumed as a mean value of the derivative and is different from zero even when the two-dimensional field is incompressible.

4. Two-phase flow and interface capturing

If we consider two temporal steps t^k e $t^{k+1} = t^k + \Delta t$, use nondimensional variables and approximate the spatial derivative with centered finite differences we get

$$C_{ij}^{k+1} = C_{ij}^k - \tilde{\Phi}_{i+1/2,j} + \tilde{\Phi}_{i-1/2,j} + \tilde{C}_{ij}(u_{i+1/2,j} - u_{i-1/2,j}), \quad (4.21)$$

where $\tilde{\Phi}$ is the normalized flux and it depends on the choice of \tilde{C} . We will assume $h_x = h_y$ to simplify the notation.

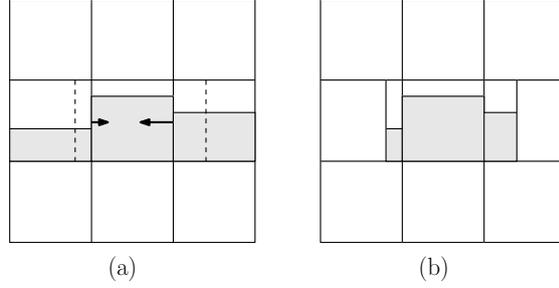


Figure 4.4.: Eulerian implicit method: (a) SLIC reconstruction of the interface, (b) implicit step fluxes.

The Eulerian implicit (EI) method sets $\tilde{C}_{ij} = C_{ij}^{k+1}$, therefore (4.21) becomes

$$C_{ij}^{k+1} = a \left(C_{ij}^k - \tilde{\Phi}_{i+1/2,j} + \tilde{\Phi}_{i-1/2,j} \right), \quad (4.22)$$

where $a = 1/(1 - u_{i+1/2,j} + u_{i-1/2,j})$ is the expansion/contraction coefficient of the Eulerian step. The geometrical procedure is shown in Fig. 4.4.

Lagrangian explicit method

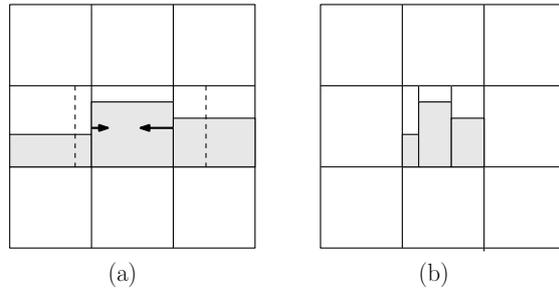


Figure 4.5.: Lagrangian explicit method: (a) SLIC reconstruction of the interface, (b) final configuration.

The Lagrangian explicit (LE) method derives also from (4.21) setting $\tilde{C}_{ij} = C_{ij}^k$ to get

$$C_{ij}^{k+1} = b C_{ij}^k - \tilde{\Phi}_{i+1/2,j} + \tilde{\Phi}_{i-1/2,j}, \quad (4.23)$$

4.2. Volume-of-Fluid (VOF) method for interface capturing

where $b = (1 + u_{i+1/2,j} - u_{i-1/2,j})$ is the Lagrangian expansion/contraction coefficient. The procedure is shown in Fig. 4.5. When the reconstruction is made with a SLIC method as in the figures, the Eulerian method and the Lagrangian one produce the same result, but this is clearly not true for PLIC reconstructions that lead to a different value for C_{ij}^{k+1} .

If the fluxes are not computed using these two schemes the final volume fraction distribution can be not consistent, with values of C that exceed one or are less than zero. This events are called *overshoots* and *undershoots* and must be avoided as much as possible.

Geometrical unsplit method

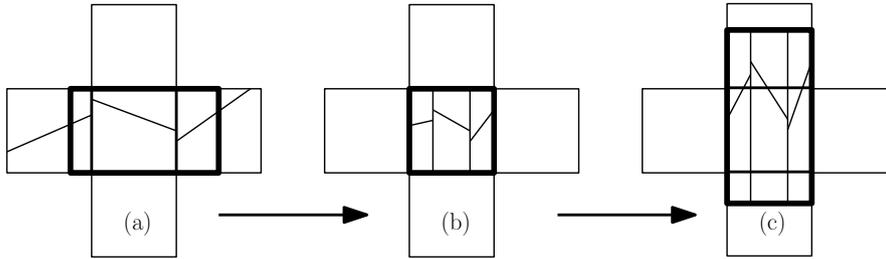


Figure 4.6.: geometrical unsplit method: (a) initial configuration; (b) implicit step; (c) explicit step.

The two previous one-dimensional methods seen can be combined in an unsplit method that does not require an intermediate volume fraction distribution \tilde{C} . From the geometrical point of view this method corresponds to a linear mapping between two tessellations of the computational domain and is shown in Fig. 4.6. The method is made up of an Eulerian step in the x direction followed by a Lagrangian step in the y direction through the linear transformation [22]

$$\Pi_{xy} = \begin{cases} x' = a(x + u_{ij}) \\ y' = by + v_{ij} \end{cases}, \quad (4.24)$$

where $a = 1/(1 - u_{i+1,j} + u_{ij})$ and $b = 1 + v_{i,j+1} - v_{ij}$, coherently with the mono-dimensional methods. If we recall the mass conservation equation and consider its discretized version on the cell, $u_{i+1,j} - u_{ij} + v_{i,j+1} - v_{ij} = 0$, we can easily get that $ab = 1$. This means that the transformation of the plane preserves the area, and guarantees no generation of overshoots or undershoots. By alternating the implicit and explicit direction or using the transformation $\Pi = (\Pi_{xy} + \Pi_{yx})/2$ no axis is favored.

Modified Eulerian implicit method

When dealing with full three-dimensional simulations, the combinations of EI and LE mono-dimensional steps does not lead to a volume-preserving algorithm. For this reason,

4. Two-phase flow and interface capturing

we introduce a modified Eulerian technique that satisfies the mass conservation constraint. Let us consider the discretized three-dimensional version of the continuity equation

$$u_{i+1,j,k} - u_{ijk} + v_{i,j+1,k} - v_{ijk} + w_{i,j,k+1} - w_{ijk} = 0. \quad (4.25)$$

If we couple an EI step with a modified Eulerian step and a LE final step, we can impose mass conservation using a coefficient c in the intermediate step such as $acb = 1$. From (4.25) we get

$$c = \frac{1}{ab} = \frac{1 - u_{i+1,j,k} + u_{ijk}}{1 + v_{i,j+1,k} - v_{ijk}} = 1 + \frac{w_{i,j,k+1} - w_{ijk}}{1 + v_{i,j+1,k} - v_{ijk}}. \quad (4.26)$$

The imposition of this coefficient, however, can lead to overshoots and undershoots in the second intermediate C data. The consequent reconstruction must cut away this inconsistencies, leading to an overall non-conservative algorithm. These problems become less and less important as the time step is decreased.

Geometrical predictor-corrector unsplit method

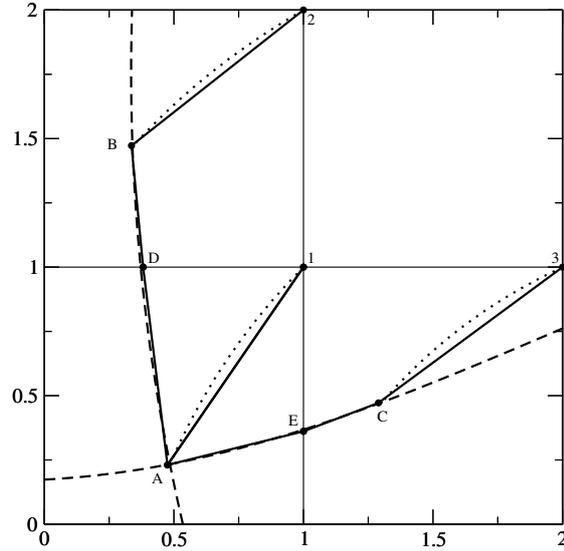


Figure 4.7.: Pre-image of the grid lines (dashed lines) and streamline through the cell vertices (dotted lines). The area of the polygon 1-2-B-D-A is equal to Φ_x^+ , that of 1-3-C-E-A to Φ_y^+ and the continuous piecewise-linear line B-D-A-E-C approximates the pre-image of the cell sides 1-2 and 1-3.

Differently from the geometrical unsplit method, that is a combination of mono dimensional split methods, this algorithm is a true unsplit method, meaning that the fluxes are calculated in a two-dimensional fashion. We can write the discretized advection equation as

$$C_{ij}^{n+1} = C_{ij}^n - \Phi_x^+ + \Phi_x^- - \Phi_y^+ + \Phi_y^-, \quad (4.27)$$

where Φ designates the incoming and outgoing fluxes.

We need to identify the area that corresponds to each of the fluxes. In order to do so, the velocities at the node of the cell are first traced backwards for the time Δt to compute the pre-image with a predictor-corrector method. With reference to Fig. 4.7, we identify the points A, B, C starting from 1, 2, 3 respectively. In general, the area of the quadrilateral 1- A - B -2 is not equal to the flux Φ_x^+ . In order to guarantee the equality, that leads to an area-preserving algorithm, the intersection between the line A - B and the Cartesian grid, identified by D in the figure, is moved horizontally to get that the polygon 1-2- B - D - A area is equal to the flux. The same procedure is followed to get the flux in the y direction. The procedure is straightforward and can be applied for any combination of vertex velocities, as shown in Fig. 4.8. The procedure has been here simplified, but full details on the method have been published in [23]. It can also be applied to unstructured grids without modifications. A three-dimensional version, however, is much more complex, given the greatly higher number of different topologies that the volume representing the flux can assume.

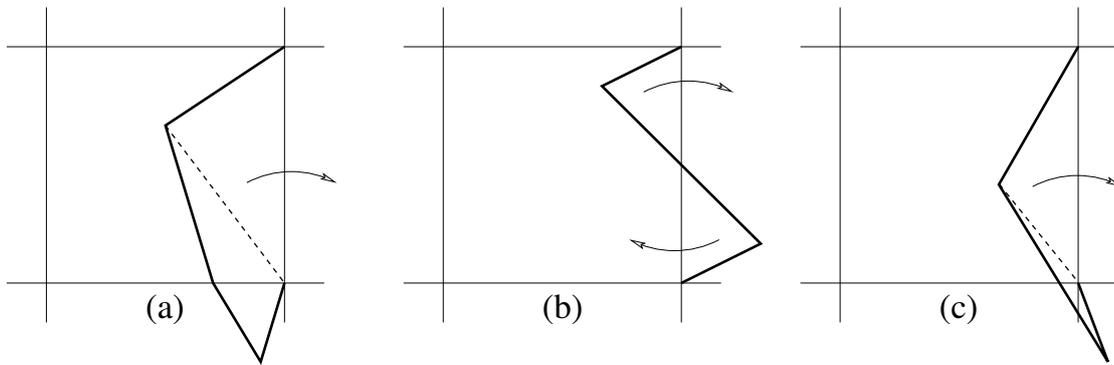


Figure 4.8.: Three different shapes of the flux polygon through the right side of a cell:
 (a) standard case with the flux polygon inside two consecutive vertical cells;
 (b) flux polygon defined across two horizontal cells; (c) polygon intersecting three different cells.

4.3. Multilevel VOF method

VOF methods show their weaknesses when dealing with structures with characteristic length comparable to the grid spacing. In particular, thin filaments or drops that have a characteristic length of three to four cells can still be reproduced accurately, but for smaller dimensions the interface reconstruction method is not able to describe accurately the shape of the interface. This feature may lead to artificial changes of topology that are unphysical. The formation of a pinch and the subsequent detachment of a drop should be driven by a physical model and not dictated by the computational grid. However, this model can be very complex and the physical process is still not fully understood.

4. Two-phase flow and interface capturing

To mitigate this effect, we have developed an innovative approach that tries to increase the resolution achievable with the VOF method, while the overhead induced on the Navier-Stokes solver is kept to a minimum. The basic idea is to separate the grid used for the Navier-Stokes system from the one used for interface evolution, in particular we use a finer grid obtained from the coarser one with repeated mid-point refinement. In this way, the number of segments in each cell is increased without the necessity to solve the velocity and pressure fields in a greater number of nodes.

We will now describe the equations on these grids with detail, suppressing the h subscript that should appear on all the discretized variables. Starting from the coarse level (c) we refine up to a fine level (f) with $f = c + l$. We indicate with \mathbf{V}^c , S^c and \mathbf{V}^f , S^f the families of subspaces defined at the coarse and fine levels and with Ω_i^c and Ω_i^f the corresponding generic cell. We can also introduce some transfer operators from the fine to the coarse levels, that take into account the different resolution at which the equations are solved. If some phase structure is present only at the fine level (f), the solution (p^c, \mathbf{u}^c) at the coarse level is different from (p^f, \mathbf{u}^f) , which satisfies the Navier-Stokes equations with different test functions.

We can start from the continuity equation, the second of (4.1), and assume that it is satisfied by the velocity field at both levels,

$$b(q^c, \mathbf{u}^c) = 0, \quad b(q^f, \mathbf{u}^f) = 0, \quad (4.28)$$

where q^c now designates the test function on S^c . We substitute in the relation at the fine level the coarse velocity field \mathbf{u}^c to get

$$b(q^f, \mathbf{u}^c) = \int_{\Omega} q^f R^{fc}(\mathbf{u}^c, \mathbf{u}^f) dV, \quad (4.29)$$

where we introduce the fine-to-coarse mass transfer operator R^{fc} , defined by

$$R^{fc}(\mathbf{u}^f, \mathbf{u}^c) = \nabla \cdot (\mathbf{u}^f - \mathbf{u}^c). \quad (4.30)$$

The meaning of this operator is to quantify the residual error of the mass conservation equation when we assume that the coarse level solution \mathbf{u}^c is valid at fine level. Even if the discrete solutions are divergence-free functions over the finite element mesh, the point-wise divergences $\nabla \cdot \mathbf{u}^f$ and $\nabla \cdot \mathbf{u}^c$ may be different from zero for all $\mathbf{x} \in \Omega$, because the divergence-free constraints are imposed in an integral fashion, i.e. $\int_{\Omega} q^f \nabla \cdot \mathbf{u}^f dV = 0$ and $\int_{\Omega} q^c \nabla \cdot \mathbf{u}^c dV = 0$, but $\int_{\Omega} q^f \nabla \cdot \mathbf{u}^c dV = 0$ is not imposed.

Since the fine mesh is obtained by mid-point refinement, $S^c(\Omega) \subset S^f(\Omega)$ and therefore any test function q^c can be written as a linear combination of the test functions q^f at the fine level,

$$q^c(\mathbf{x}) = \sum_i a_i q_i^f(\mathbf{x}). \quad (4.31)$$

From this we get $b(q^c, \mathbf{u}^f) = \sum_i a_i b(q_i^f, \mathbf{u}^f) = 0$ and therefore

$$0 = b(q^c, \mathbf{u}^c) = b(q^c, \mathbf{u}^c - \mathbf{u}^f) = \int_{\Omega} q^c R^{fc}(\mathbf{u}^c, \mathbf{u}^f) dV. \quad (4.32)$$

Under the above assumptions, there is no net mass transfer from the fine to the coarse level. Therefore, if we implement a projection of the velocity that preserves the divergence-free constraint, such as the one shown in Sec. 4.3.2, the transfer operator is identically equal to zero.

We repeat the same argument for the momentum conservation equation shown in the first of (4.1). Let (p^f, \mathbf{u}^f) be the solution of the Navier-Stokes equation at the fine level

$$\left(\rho \frac{\partial \mathbf{u}^f}{\partial t}, \mathbf{v}^f\right) + c(\rho, \mathbf{u}^f, \mathbf{u}^f, \mathbf{v}^f) + b(p^f, \mathbf{v}^f) + a(\mu, \mathbf{u}^f, \mathbf{v}^f) = (\mathbf{f}, \mathbf{v}^f) + (\mathbf{f}_s^f, \mathbf{v}^f), \quad (4.33)$$

with \mathbf{v}^f the test function in \mathbf{V}^f . Here the density ρ and the viscosity μ are explicitly written since they are now discontinuous across the interface. Now we substitute the solution (p^c, \mathbf{u}^c) of the coarse grid in (4.33)

$$\begin{aligned} \left(\rho \frac{\partial \mathbf{u}^c}{\partial t}, \mathbf{v}^f\right) + c(\rho, \mathbf{u}^c, \mathbf{u}^c, \mathbf{v}^f) + b(p^c, \mathbf{v}^f) + a(\mu, \mathbf{u}^c, \mathbf{v}^f) = \\ = (\mathbf{f}, \mathbf{v}^f) + (\mathbf{f}_s^f, \mathbf{v}^f) + (P^{fc}(p^c, p^f, \mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f) + (T^{fc}(\mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f), \end{aligned} \quad (4.34)$$

and introduce the fine-to-coarse momentum transfer operator P^{fc} defined by

$$\begin{aligned} (P^{fc}(p^c, p^f, \mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f) = \left(\rho \frac{\partial \mathbf{u}^c}{\partial t}, \mathbf{v}^f\right) + b(p^c, \mathbf{v}^f) + a(\mu, \mathbf{u}^c, \mathbf{v}^f) + \\ - c(\rho, \mathbf{u}^c - \mathbf{u}^f, \mathbf{u}^c - \mathbf{u}^f, \mathbf{v}^f) - \left(\rho \frac{\partial \mathbf{u}^f}{\partial t}, \mathbf{v}^f\right) - b(p^f, \mathbf{v}^f) - a(\mu, \mathbf{u}^f, \mathbf{v}^f), \end{aligned} \quad (4.35)$$

and the fine-to-coarse turbulent transfer operator T^{fc} defined by

$$(T^{fc}(\mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f) = c(\rho, \mathbf{u}^c - \mathbf{u}^f, \mathbf{u}^c - \mathbf{u}^f, \mathbf{v}^f) + c(\rho, \mathbf{u}^c, \mathbf{u}^c, \mathbf{v}^f) - c(\rho, \mathbf{u}^f, \mathbf{u}^f, \mathbf{v}^f). \quad (4.36)$$

We split the contribution from the fine grid to the coarse one in two terms to get a term that can be associated to the well-known turbulence contribution from the sub-grid velocity field. The other term summarizes the difference of virtual work between the two levels.

When the spaces are embedded, $\mathbf{V}^c(\Omega) \subset \mathbf{V}^f(\Omega)$, (4.34) holds for any test function on the coarse grid and

$$\begin{aligned} \left(\rho \frac{\partial \mathbf{u}^c}{\partial t}, \mathbf{v}^c\right) + c(\rho, \mathbf{u}^c, \mathbf{u}^c, \mathbf{v}^c) + b(p^c, \mathbf{v}^c) + a(\mu, \mathbf{u}^c, \mathbf{v}^c) = \\ = (\mathbf{f}, \mathbf{v}^c) + (\mathbf{f}_s^f, \mathbf{v}^c) + (S^{fc}(p^c, p^f, \mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^c). \end{aligned} \quad (4.37)$$

The operator S^{fc} models the whole residual between the fine grid Navier-Stokes solution and the coarse one and depends only on the variables of the coarse grid

$$\begin{aligned} (S^{fc}, \mathbf{v}^f) = (P^{fc}(p^c, p^f, \mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f) + (T^{fc}(\mathbf{u}^c, \mathbf{u}^f), \mathbf{v}^f) = \\ = \left(\rho \frac{\partial \mathbf{u}^c}{\partial t}, \mathbf{v}^f\right) + c(\rho, \mathbf{u}^c, \mathbf{u}^c, \mathbf{v}^f) + b(p^c, \mathbf{v}^f) + a(\mu, \mathbf{u}^c, \mathbf{v}^f) - (\mathbf{f}_s^f, \mathbf{v}^f) - (\mathbf{f}, \mathbf{v}^f). \end{aligned} \quad (4.38)$$

4. Two-phase flow and interface capturing

When S^{fc} is small it means that a further refinement of the VOF grid does not modify the velocity and pressure fields, and the capillary force calculated on the fine grid is well-resolved. When this is not true, S^{fc} can be directly calculated with (4.38) and projected on the coarse grid, or can be modeled in some way at the coarse grid level. The fine grid is used only to get a higher resolution on interface reconstruction, while the relevant physics must be resolved completely at the coarse level. In the simulations presented in the next chapter, the value of S^{fc} has been monitored and kept small at any point of the simulations.

4.3.1. Numerical implementation of the multilevel VOF method

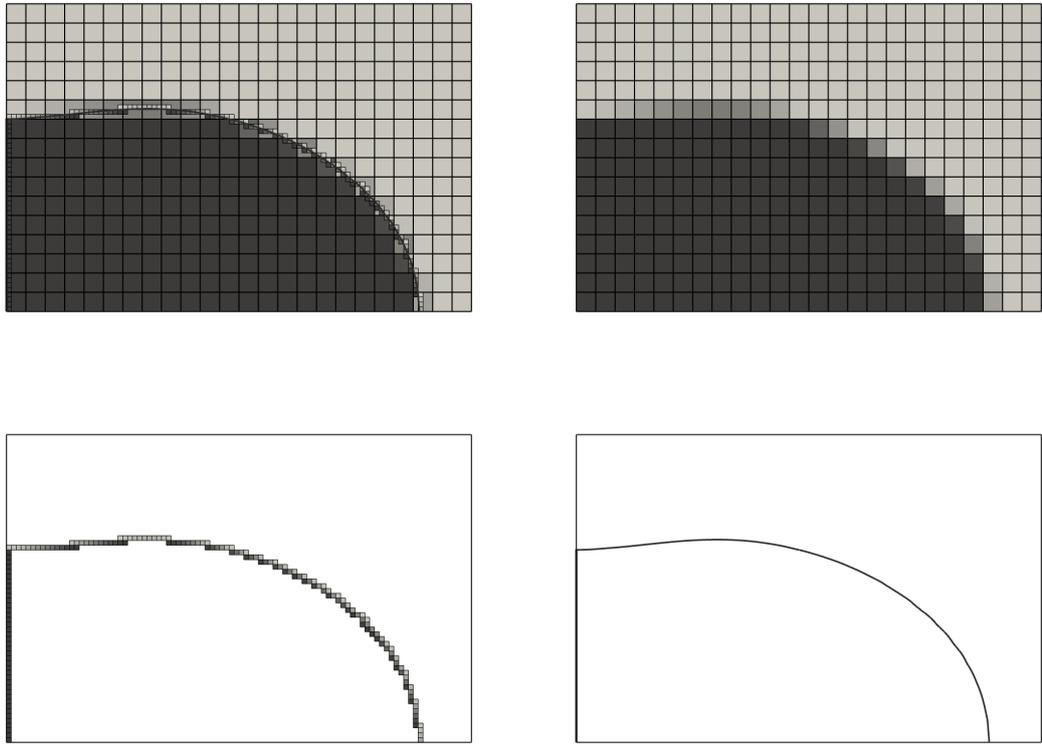


Figure 4.9.: The color function distribution on different meshes (top left) and on the coarse mesh (top right). The compact data memorization with two (bottom left) and four (bottom right) levels of grid refinement.

We consider a few features of the multilevel in order to describe the algorithms that keep the induced overhead to a minimum. In particular, the fine grid is used to compute the surface tension force \mathbf{f}_s^f that is then inserted in the coarse grid equations. Since each

0	0.22	0.31	0.14	0
0.33	0.99	1	0.87	0.13
0.39	1	1	1	0.25
0.04	0.63	0.77	0.71	0.14
0	0	0	0	0

row	n_c	C					column				
1	3	0.22	0.31	0.14							
2	5	0.33	0.99	1	0.87	0.13	1	2	3	4	5
3	3	0.39	3	0.25			1	2	5		
4	5	0.04	0.63	0.77	0.71	0.14	1	2	3	4	5
5	0										

Figure 4.10.: The C data distribution on a 5×5 Cartesian mesh (top) and the compressed stored data (bottom): row number, number of cells n_c , color function in the mixed and consecutive full cells and column position.

level of refinement multiplies by a factor of four in two dimensions (or eight in three dimensions) in the number of points and cells, the complete memorization of the VOF data would increase the memory footprint quickly. For this reason we have developed a storage scheme that compresses the VOF data and show an example of its implementation in Fig. 4.9. The figure shows a coarse level with 24×16 cells, and the sparse color function matrices at the two levels $f = c + 2$ and $f = c + 4$. At the intermediate level of refinement each cell of the coarse grid is divided in 16 subcells with the interface clearly marked on the grid. At the highest level of refinement each coarse cell is subdivided into 256 smaller cells, with matrix entries about 16 times those of the coarse grid. The VOF interface at this level can be compared in resolution to a front tracking representation with markers.

The format used for data storage can be compared to the Compressed Row Storage (CRS) [24] and we show a two-dimensional example in Fig. 4.10. We consider a 5×5 stencil of cells and the associated color function data. For each row we memorize only the number of entries n_c , the C data and their column number. All empty cells are discarded, while a sequence of n consecutive full cells is stored as a single one, with its

4. Two-phase flow and interface capturing

color function value equal to n in the first position, e.g. the third row of Fig. 4.10 where we memorize in the second position the integer 3 to represent the sequence of full cells. With this technique we can use a large number of refinement levels while keeping the storage requirements proportional to the length of the interface divided by the fine grid spacing. This representation requires an efficient numerical algorithm to extract and compress the relevant data.

To further improve the performances when we consider high-resolution fine grids, we do not perform the reconstruction cell by cell, extracting the block of cells needed for every mixed cell. The implementation includes an algorithm that extracts a stripe of $3 \times n$ cells, where n is the length of one full line, and computes the new normals and the fluxes simultaneously for the whole block. Afterwards, one line of cells is updated and the procedure is repeated. The normal values are stored in the same compressed way of the VOF data.

4.3.2. Velocity refinement with an optimal control approach

When dealing with a multilevel VOF method, we need to project the velocity field from the coarse grid to the fine one, since the advection of the color function needs a velocity value on each node of the fine grid. Since the divergence-free constraint is imposed on the coarse grid, we want to preserve it on the fine grid. If we consider a simple approach in which the velocity on the added points are calculated as a simple averaging of the value on the coarse grid, we can easily see that the fine velocity field is not divergence-free. We stress again that in this context the mass conservation constraint is satisfied only in an integral fashion.

We introduce an optimal control problem to get a divergence-free preserving projection operator for the velocity field. We rewrite the refined velocities as a combination of all the velocities on the coarse grid, and impose the divergence-free constraint with a Lagrangian multiplier approach. The set of fine velocity values that satisfies these hypotheses is not unique, so we can also impose, as the target of the optimal approach, that the resulting velocities are the closest to the averaged values.

Let us consider the coarse cell shown in Fig. 4.11, where we already know the velocities in the points 0, 1, 2 and 3. The midpoint refinement puts in the five new nodes 4, 5, 6, 7 and 8. Therefore we need to compute 10 velocity components, number that clearly outnumbers the constraints of mass conservation, given by

$$u_4 + u_8 - u_7 - u_0 + v_8 + v_7 - v_4 - v_0 = 0, \quad (4.39a)$$

$$u_1 + u_5 - u_8 - u_4 + v_5 + v_8 - v_1 - v_4 = 0, \quad (4.39b)$$

$$u_8 + u_6 - u_3 - u_7 + v_6 + v_3 - v_8 - v_7 = 0, \quad (4.39c)$$

$$u_5 + u_2 - u_8 - u_6 + v_2 + v_6 - v_5 - v_8 = 0, \quad (4.39d)$$

that represent the divergence values in each of the fine cells created from the given coarse

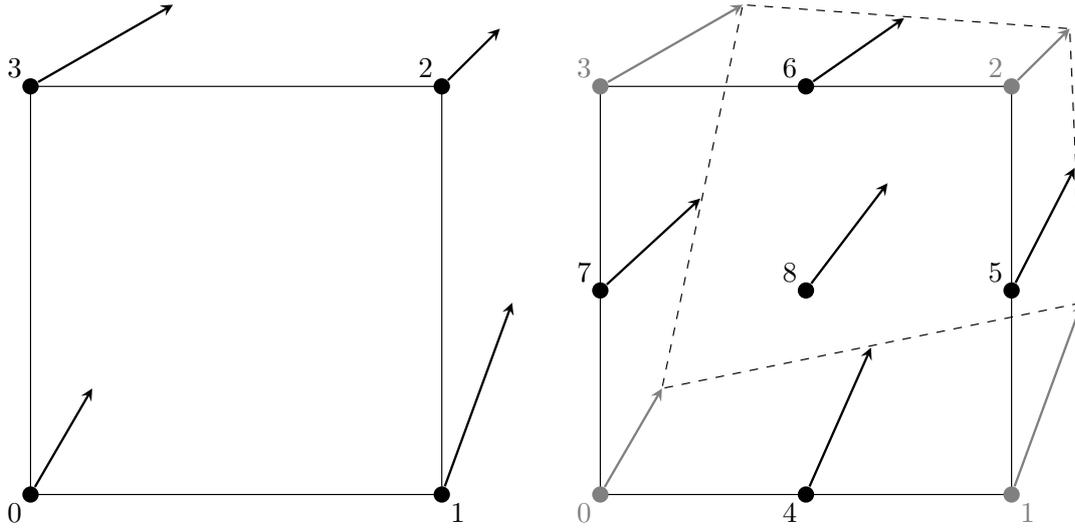


Figure 4.11.: The coarse element with four nodes (left) and the refined one with nine nodes (right).

cell. therefore, we set some of them to the averaged value

$$\begin{aligned}
 v_4 &= (v_0 + v_1)/2, & u_5 &= (u_1 + u_2)/2, \\
 v_6 &= (v_2 + v_3)/2, & u_7 &= (u_3 + u_0)/2, \\
 u_8 &= (u_0 + u_1 + u_2 + u_3)/4, & v_8 &= (v_0 + v_1 + v_2 + v_3)/4.
 \end{aligned} \tag{4.40}$$

We are left with u_4 , v_5 , u_6 and v_7 as unknowns.

We can now introduce our functional J as

$$J = \frac{1}{2}(u_4 - \tilde{u}_4)^2 + \frac{1}{2}(v_5 - \tilde{v}_5)^2 + \frac{1}{2}(u_6 - \tilde{u}_6)^2 + \frac{1}{2}(v_7 - \tilde{v}_7)^2, \tag{4.41}$$

where the target values are indicated by a tilde. We choose them to be the averaged values,

$$\tilde{u}_4 = (u_0 + u_1)/2, \quad \tilde{v}_5 = (v_1 + v_2)/2, \tag{4.42a}$$

$$\tilde{u}_6 = (u_2 + u_3)/2, \quad \tilde{v}_7 = (v_3 + v_0)/2. \tag{4.42b}$$

We now build the augmented Lagrangian functional P as

$$P = J + \sum_{i=0}^3 \gamma_i D_i, \tag{4.43}$$

where γ_i are the Lagrangian multipliers associated with the four discrete divergence on the l.h.s. of (4.39) and indicated here as D_i .

4. Two-phase flow and interface capturing

We can now solve our problem by determining the minimum of P . We set to zero its first variation δP

$$\begin{aligned} \delta P = & (u_4 - \tilde{u}_4)\delta u_4 + (v_5 - \tilde{v}_5)\delta v_5 + (u_6 - \tilde{u}_6)\delta u_6 + (v_7 - \tilde{v}_7)\delta v_7 + \gamma_0(\delta u_4 + \delta v_7) + \\ & + \gamma_1(\delta u_4 + \delta v_5) + \gamma_2(\delta u_6 + \delta v_7) + \gamma_3(\delta u_6 + \delta v_5) + \sum_{i=0}^3 \delta \gamma_i D_i. \end{aligned} \quad (4.44)$$

All variations in (4.44) are independent from each other, so we can put to zero each of them singularly to get a minimum of P . We obtain a linear system of eight equations in the eight variables u_4, v_5, u_6, v_7 and the four Lagrangian multipliers. Since the four relations (4.39) are not linearly independent, there are an infinite number of solutions. If we take a 7 by 7 minor with full rank, we get the solution desired by leaving one of the Lagrangian multipliers as undefined. In this way, however, we would get that 3 of the (4.39) are satisfied, while the fourth is not. In general, the value of the divergence on the coarse grid is not zero, since we integrate an iterative solver. If we proceed as described, the whole error on divergence would be transferred to one fine cell, the one corresponding to the divergence constraint that we have removed. Alternatively, we can try to split beforehand the divergence error on all four subcells, in order to get a balanced solution on the fine cells.

The previous derivation leads to

$$u_4 = \frac{2u_0 + 2u_1 + v_0 - v_1 + v_2 - v_3}{4}, \quad (4.45a)$$

$$v_5 = \frac{u_0 - u_1 + u_2 - u_3 + 2v_1 + 2v_2}{4}, \quad (4.45b)$$

$$u_6 = \frac{2u_2 + 2u_3 + v_0 - v_1 + v_2 - v_3}{4}, \quad (4.45c)$$

$$v_7 = \frac{u_0 - u_1 + u_2 - u_3 + 2v_3 + 2v_0}{4}. \quad (4.45d)$$

We remark that now the optimized velocities will depend on both u and v components on the coarse grid. The extension to the three-dimensional case is straightforward and does not present any difficulty. We note that, if in three dimensions we start from a two-dimensional coarse grid velocity field, the resulting fine field will be fully three-dimensional.

4.4. Surface tension modeling

As we have seen in the previous section, the capillary force is computed on the fine grid from the sequence of segments approximating the interface line inside a coarse cell. Recalling (4.7), we write it in its variational form

$$(\mathbf{f}_s^f, \mathbf{v}^c) = \int_{\Omega} \sigma \kappa \delta_s(\mathbf{x}) \mathbf{n} \cdot \mathbf{v}^c dS, \quad (4.46)$$

that becomes

$$(\mathbf{f}_s^f, \mathbf{v}^c) = \int_{\Gamma_s} \sigma \kappa \mathbf{n} \cdot \mathbf{v}^c dS. \quad (4.47)$$

The vector $\kappa \mathbf{n}$ can be written as the second derivative along the interface of the position vector

$$\kappa \mathbf{n} = \nabla_s^2 \mathbf{x}_s, \quad (4.48)$$

where $\nabla_s = (I - \mathbf{n}\mathbf{n}) \cdot \nabla$ is the gradient operator over the surface Γ_s and is called *Laplace-Beltrami operator*. Using (4.48) we rewrite (4.47) with an integration by parts as

$$(\mathbf{f}_s^f, \mathbf{v}^c) = \int_{\Gamma_s} \sigma \nabla_s^2 \mathbf{x}_s \cdot \mathbf{v}^c dS = \sigma \int_{\Gamma_s} \nabla_s \mathbf{x}_s : \nabla_s \mathbf{v}^c dS. \quad (4.49)$$

This formulation is useful since it involves only first derivatives of the interface.

Alternatively, the surface tension term can be rewritten in another form, called the *volumetric approach*. Going back to (4.47), we can write

$$\begin{aligned} \int_{\Gamma_s} \sigma \kappa \mathbf{v}^c \cdot \mathbf{n} dS &= \int_{\Omega_l} \sigma \nabla \cdot (\hat{\kappa} \mathbf{v}^c) dV = \int_{\Omega} \chi \sigma \nabla \cdot (\hat{\kappa} \mathbf{v}^c) dV = \\ &= \int_{\Omega} \chi \sigma \hat{\kappa} \nabla \cdot \mathbf{v}^c dV + \int_{\Omega} \chi \sigma \mathbf{v}^c \cdot \nabla \hat{\kappa} dV, \end{aligned} \quad (4.50)$$

using the divergence theorem. Here $\hat{\kappa}$ is an extension of the curvature κ to the whole domain. Any extension is allowed provided $\hat{\kappa}|_{\Gamma_s} = \kappa$. This formulation can be particularly useful when the curvature is known analytically and the extension $\hat{\kappa}$ can be computed easily. At equilibrium the first term of (4.50) balances the pressure jump at the interface, while the second term is the source of spurious currents. We now discuss briefly the numerical implementation of these two approaches.

4.4.1. Numerical implementation of the capillary force

In axisymmetric geometry it is useful to split the curvature in its two main components

$$\kappa = \kappa_1 + \kappa_2, \quad (4.51)$$

which are called principal curvatures. If we indicate with r , θ , and z the cylindrical coordinates, we can easily compute the curvature of the interface Γ_s that is the revolution around the z -axis of the interface line L_s of equation $r = f(z)$. Each meridian plane through the z -axis contains the line L_s and is also perpendicular to the local tangent plane in each point of Γ_s . The local curvature κ_1 on L_s is given by

$$\kappa_1 = -\frac{f''(z)}{(1 + f'^2(z))^{3/2}}. \quad (4.52)$$

The other curvature κ_2 is computed on a plane that is perpendicular to both the tangent and the meridian planes and contains the local normal vector to the line L_s . If we denote with φ the angle between the outgoing normal vector and the z -axis, then

$$\kappa_2 = -\frac{\cos \varphi}{r} = -\frac{1}{f(z)(1 + f'^2(z))}. \quad (4.53)$$

4. Two-phase flow and interface capturing

We can split (4.47) in the two contributions

$$(\mathbf{f}_s^f, \mathbf{v}^c) = \int_{\Gamma_s} \sigma \kappa_1 \mathbf{n} \cdot \mathbf{v}^c dS + \int_{\Gamma_s} \sigma \kappa_2 \mathbf{n} \cdot \mathbf{v}^c dS. \quad (4.54)$$

Since $\kappa_2 = -\cos \varphi/r$ is known analytically we compute this term with the volumetric approach. The term in κ_1 , instead, is computed with the Laplace-Beltrami approach. We note that the VOF reconstruction computes a segment inside each cell, so the first derivative of \mathbf{x}_s inside each fine cell is the tangent vector and is a constant. An extensive mathematical analysis of the Laplace-Beltrami operator for curvature computations in two-phase flow applied to finite element methods can be found in [25, 26]. For the second term, if we know an approximated distribution $\hat{\chi}_h$ from the C data values, (4.50) becomes

$$\int_{\Gamma_s} \sigma \kappa_2 \mathbf{v}^c \cdot \mathbf{n} dS = \int_{\Omega} \hat{\chi}_h \sigma \hat{\kappa}_2 \nabla \cdot \mathbf{v}^c dV + \int_{\Omega} \hat{\chi}_h \sigma \mathbf{v}^c \cdot \nabla \hat{\kappa}_2 dV. \quad (4.55)$$

The distribution $\hat{\chi}_h$ can be calculated by interpolating the C values on the cell vertices, for example. With an extension of κ_2 constant in the radial direction we get a momentum conservation in the form

$$d(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) = b(\hat{\chi}_h \sigma \hat{\kappa}_2, \mathbf{v}) + (\hat{\chi}_h \sigma \nabla \hat{\kappa}_2, \mathbf{v}^c), \quad (4.56)$$

which gives exactly the equilibrium solution $p = \chi_h \sigma \hat{\kappa}_2$ and $\mathbf{u} = 0$ with no spurious currents if the pressure and χ_h are defined over the same finite dimensional space S_h .

When the evaluation of the capillary force is not accurate, the lack of force balance generates the so-called *spurious currents* can appear. They are particularly evident in the static case, where the analytical solution is given by a zero velocity field and a sharp jump at the interface for the pressure. Since in the discretization the capillary force and the pressure gradient are not exactly parallel, the resulting force can generate small currents. It is important that, even if these currents are present at the initial time, they should tend to zero as the time goes on rather than deforming the interface position.

5. Numerical simulations

The code has been tested with some two-phase applications of general interest. First of all, some basic simulation of a column of liquid in a steady gas has been performed to assure that the model is suitable for these applications. Afterwards, some simulations of jets are performed, with the aim to investigate the breakup of the fluid column [27].

5.1. Linear dispersion equation for jets

As a first test we consider the evolution of small perturbations of an axisymmetric jet, when nonlinear effects are still very small. Since the evolution of the interface is driven by surface tension, it is important that the spurious currents are negligible.

5.1.1. Analytical dispersion equation

We follow here the analytical derivation of [28] for the dispersion equation. This formulation shows clearly that asymmetric modes can be excited when the Weber number is greater than a critical value We_c . This value is a function of the wavelength of the disturbance and the density ratio between the gas and liquid phases. The simulations described in the next sections are only axisymmetric, so we must enforce that We is smaller than We_c in order to obtain physically meaningful simulations.

The derivation of the dispersion equation considers the stability of a wave of very small amplitude on a cylindrical jet of infinite length. The liquid jet with density ρ_l and radius R has a uniform relative velocity U with respect to the external gas with density ρ_g . The two fluids are assumed to be incompressible and inviscid. We linearize the Navier-Stokes equations and consider a small perturbation such that

$$\mathbf{u}_j = \mathbf{u}_{0j} + \mathbf{u}'_j, \quad p_j = p_{0j} + p'_j, \quad (5.1)$$

where $j = l, g$ designates the phase and the 0 subscript the stationary solution with $\mathbf{u}_{0j} = 0$ and a constant pressure jump along the interface. The linear perturbed quantities can be expressed

$$p'_j(r, \theta, z, t) = p_j(r) \exp(i(kz + m\theta) + \alpha t), \quad (5.2a)$$

$$\mathbf{u}'_j(r, \theta, z, t) = \mathbf{u}_j(r) \exp(i(kz + m\theta) + \alpha t), \quad (5.2b)$$

where α is the complex growth rate with respect to time. For the azimuthal mode $m = 0$, also known as *varicose* or *sausage* mode, the cross section of the jet is circular and its radius varies only with the z -coordinate. When $m > 0$ the modes are asymmetric. For the $m = 1$ mode the cross section of the jet is still circular but the axis of the perturbed

5. Numerical simulations

jet describes a spiral around the z -axis. This is usually called *snake* or *kink* mode. For $m = 2$, the cross section of the jet is elliptic, and an originally circular jet deforms progressively into a flat curling sheet. For larger m values, the modes are characterized by m oscillations along the unperturbed circumference.

The dispersion relation (5.2) shows that in presence of an external fluid phase the surface tension has a stabilizing effect on asymmetric modes. We define the critical Weber number $(We_c)_m$ as the Weber number for which the growth rate of the m -th mode becomes greater than zero. in particular, for an inviscid jet

$$(We_c)_m = \frac{(\gamma_m + q\beta_m)(m^2 + (kR)^2 - 1)}{kRq\gamma_m\beta_m}, \quad (5.3)$$

with

$$\gamma_m = k \frac{I_m(kR)}{I'_m(kR)}, \quad \beta_m = -k \frac{K_m(kR)}{K'_m(kR)}, \quad (5.4)$$

where I_m and K_m are the m -th order modified Bessel functions of the first and second kind, I'_m and K'_m their first derivative and $q = \rho_g/\rho_l$. We can simplify this relation in the approximation of short or long wavelengths.

Long wavelengths. In this case the disturbance has $m = 1$ is the first one to be excited when $We > (1 + q)/q$.

Short wavelengths. Also in this case the first asymmetric mode to be excited is with $m = 1$ when $We > (1 + q)kR/q$.

We note that the critical Weber number We_c is always proportional to the inverse of the density ratio q . When $q \rightarrow 0$, we recover the physical situation of a single fluid surrounded by vacuum as in Rayleigh's theory [29], $We_c \rightarrow \infty$ and the mode is always axisymmetric. If we consider a mixture of water and air with $q = 1.3 \times 10^{-3}$, the lowest critical Weber number is $We_c \approx 769$ for $m = 1$.

Below this critical value only the $m = 0$ mode is present and the growth rate α is computed from the characteristic equation

$$(\rho_l \gamma_0 + \rho_g \beta_0)\alpha^2 + 2ik\alpha(\rho_g \beta_0 u_g + \rho_l \gamma_0 u_l) - k^2(\rho_g \beta_0 u_g^2 + \rho_l \gamma_0 u_l^2) - k \frac{\sigma}{R^2}[1 - (kR)^2] = 0. \quad (5.5)$$

We split the complex growth rate α in its real and complex parts α_r and α_i respectively. We get for the real part

$$\alpha_r = \pm \frac{U}{R} \sqrt{\frac{\gamma_0 \beta_0 q (kR)^2}{(\gamma_0 + \beta_0 q)^2} + \frac{1}{We} \frac{(kR)[1 - (kR)^2]}{(\gamma_0 + \beta_0 q)}}, \quad (5.6)$$

with $We = (RU^2 \rho_l)/\sigma$.

Let us consider now three limiting cases of (5.6).

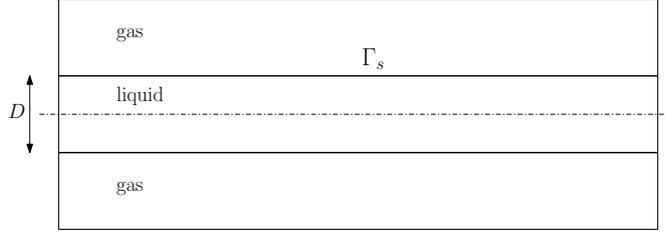


Figure 5.1.: Initial axisymmetric geometry.

- **Rayleigh's equation.** This is the limit for $q \rightarrow 0$, representative of a liquid jet in vacuum. In this situation we get

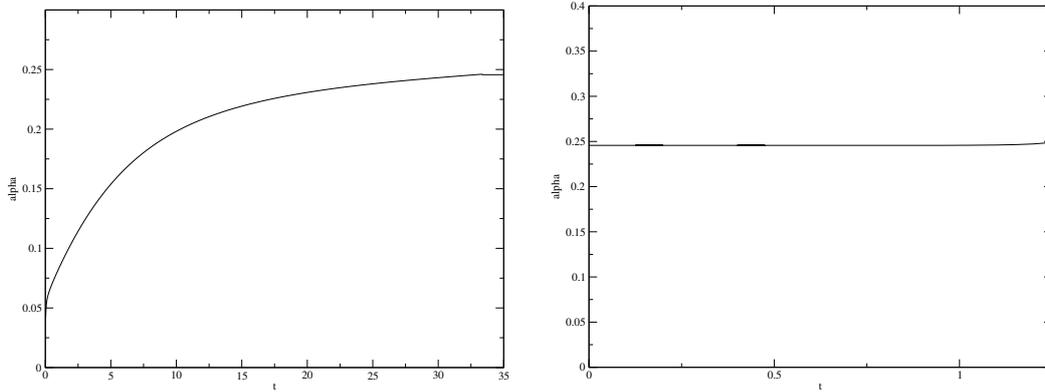
$$\alpha_r = \pm \frac{U}{R} \sqrt{\frac{1}{We} \frac{(kR) [1 - (kR)^2]}{\gamma_0}}. \quad (5.7)$$

- **Short wavelength limit.** This case corresponds to $kR \rightarrow \infty$ that means $\beta_0 = \gamma_0 = 1$ and

$$\alpha_r = \pm \frac{U}{R} \sqrt{\frac{(kR)^2 q}{(1+q)^2} + \frac{1}{We} \frac{(kR) [1 - (kR)^2]}{1+q}}. \quad (5.8)$$

- **Long wavelength limit.** This case corresponds to $k \rightarrow 0$, $\beta_0 \rightarrow -kR \ln(kR)$ and $\gamma_0 \rightarrow 2/(kR)$, so we get

$$\alpha_r = \pm \frac{U}{R} \sqrt{-\frac{2(kR)^3 \ln(kR) q}{[2 - q(kR)^2 \ln(kR)]^2} + \frac{1}{We} \frac{(kR)^2 [1 - (kR)^2]}{[2 - q(kR)^2 \ln(kR)]}}. \quad (5.9)$$


 Figure 5.2.: Time evolution of the growth rate α_r for the $m = 0$ mode during the initial transient (left) and after the restart of the computation (right).

5. Numerical simulations

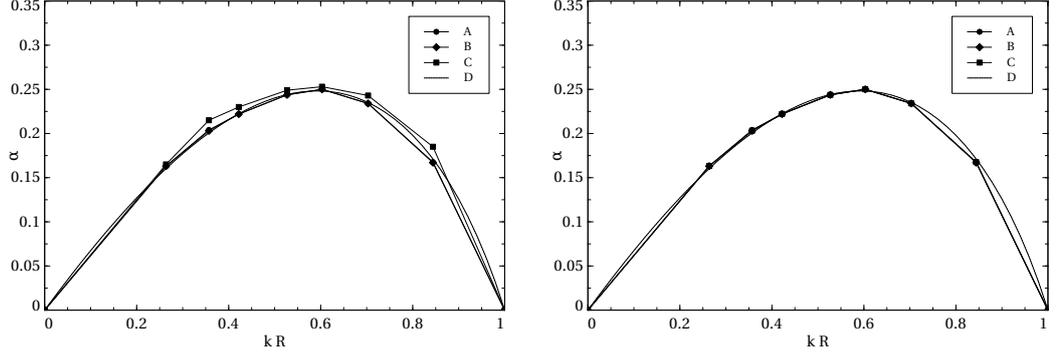


Figure 5.3.: Numerical growth rate α for $\mu_l/\mu_g = 100$ for different mesh resolutions on the left (32 (A) , 64 (B), 128 (C)) and for a different number of refinement levels on the right ($f = c + 2$ (A), $f = c + 3$ (B), $f = c + 4$ (C)). In both graphs it is also plotted the inviscid theoretical curve of (5.7) (D).

5.1.2. Numerical results

We now try to reproduce the evolution of (5.7) with our code. We consider the axisymmetric domain shown in Fig. 5.1. A liquid of density ρ_l and viscosity μ_l is located within a cylindrical region Ω_l of radius R . Outside the liquid there is a gas of density ρ_g and viscosity μ_g . In order to reproduce the infinite cylinder of (5.7) we apply periodic boundary conditions on the two bases and assume free flow conditions on its lateral surface. We impose an initial perturbation only in the position r_I of the interface of the form

$$r_I = r_I(z) = R + \delta_0 \cos\left(\frac{2\pi z}{\lambda}\right), \quad (5.10)$$

where δ_0 is the initial amplitude and λ the wavelength of the perturbation.

Since we analyze the linear case, we need to use very small values of δ_0 and of the velocity field. If the capillary force (4.7) is not carefully discretized, for δ_0 that goes to zero and with a small initial velocity, the evolution could be dominated by the spurious currents. With reference to Sec. 4.4.1, in this situation we get $\kappa_1 \simeq 0$, so the spurious currents at the initial time can be generated only by the surface tension term with κ_2 . The steady state solution provides $\kappa_2 = 1/R$, and with this value we can positively use the volumetric approach to get negligible spurious currents and a correct pressure jump.

In order to reproduce the initially linear instability evolution we use the linearized version of (4.1) to evolve in time the velocity and pressure fields. Since in the applied initial conditions the velocity and pressure fields are not perturbed, we divide the procedure in two steps, as shown in Fig. 5.2. We first start at time $t = 0$ with the velocity and pressure profiles of the steady-state solution but with the perturbed interface (5.10) and evolve the system with very small time steps. When the growth rate α_r approaches a stationary value, the velocity and pressure fields are then consistent with the perturbed interface and we assume these profiles as the initial configuration for the second part of the simulation. We then monitor the time evolution of the value of α_r that should

remain almost constant, as shown on the right of Fig. 5.2.

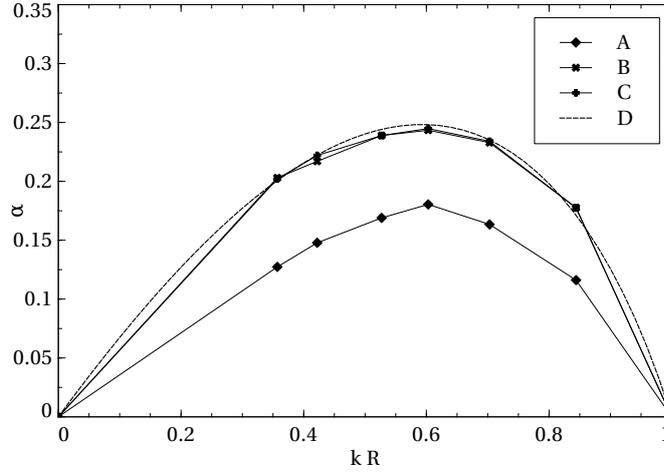


Figure 5.4.: Numerical growth rate α for different viscosity ratios μ_l/μ_g (10 (A), 100 (B) and 1000 (C)). In the graph it is also plotted the inviscid theoretical curve of (5.7) (D).

In Fig. 5.3 the dispersion relation of the growth rate α_r is plotted as a function of the non-dimensional wave number kR , for different resolutions of the coarse mesh and different levels of refinement. The geometric setup of the simulation sets a radius of the liquid cylinder $R = 0.328125 \times 10^{-3} m$ with a perturbation amplitude $\delta_0 = 0.01 R$. The physical properties of the fluid are density $\rho_l = 1000 kg/m^3$, viscosity $\mu_l = 0.0316 kg/m s$ and surface tension coefficient $\sigma = 0.0725 J/m^2$. The density ratio is $\rho_l/\rho_g = 1000$, and the viscosity ratio $\mu_l/\mu_g = 100$. The non-dimensional numbers we get for these parameters are $Re = 10.38$, $We = 4.53$ and $Oh = 0.205$, where the reference velocity is computed from the static pressure jump, $U = \sqrt{\sigma\kappa_2/\rho_l}$. The curves on the left of Fig. 5.3 are obtained with 32, 64, 128 cells in the radial direction and two refinement levels for the color function, $f = c + 2$. The curves on the right consider 64 cells in the radial direction and 2, 3 and 4 refinement levels. The inviscid theoretical curve of (5.7) is also plotted in each figure. We note that the numerical results differ only for the case with the lowest resolution of 32 cells, while the number of levels of refinement do not change the solution appreciably.

In Fig. 5.4 we focus on the dependence of the results on the viscosity ratio μ_l/μ_g for the three values 10, 100 and 1000. Since the density ratio is rather high, the results are already close to the asymptotic line already at viscosity ratios μ_l/μ_g of about 100.

We also consider the dependence of the dispersion equation on the viscosity μ_l of the liquid phase and show the results in Fig. 5.5. The left image is obtained with the previous physical parameters, while on the right the liquid viscosity is set equal to $0.01 kg/m s$, while the viscosity ratio is unchanged. In this case $Oh = 0.065$, and the whole curve changes, with an increased maximum. In this case, the continuous lines in Fig. 5.5 represent the analytical dispersion relation derived by Weber [30], that is valid for a

5. Numerical simulations

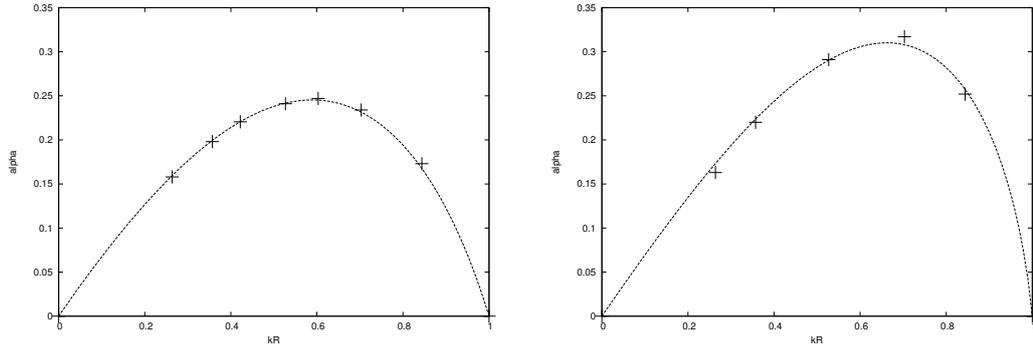


Figure 5.5.: Numerical (+) and theoretical dispersion for the growth rate for $Oh = 0.205$ (left) and $Oh = 0.065$ (right).

liquid surrounded by the vacuum.

In general the agreement between our numerical simulations and theoretical results is rather good.

5.2. Axisymmetric pulsating jets

The next case we have analyzed is a liquid jet that enters in a volume occupied by a gas from a circular nozzle. This phenomenon has a strong relation with the jet instability seen in the previous section and has been studied extensively theoretically, numerically and experimentally [31]. Real cases are however difficult to reproduce, since the instability is generated by small perturbations that are always present at the nozzle, or even upstream. These perturbations are mostly of stochastic nature, related to turbulence and it is not easy to quantify them. For this reason we impose a known sinusoidal oscillation to the constant inlet velocity that generates some elementary and well-defined effects of interest, such as the breakup of the whole column and the formation of a single droplet and thread. The inlet velocity is given by the expression

$$U = U_0 (1 + A \sin(\omega t)) , \quad (5.11)$$

with $\omega k = U_0$. We only deal with small diameter jets, where the gravity term can be neglected. The parameters that govern the jet decay are: the driving amplitude A , the non-dimensional or reduced wave-number kR , the Weber number We and the Ohnesorge number Oh . We restrict our analysis to three different wavelengths, around the maximum of Fig. 5.4, and three inlet velocities up to a Weber number greater than the critical value.

Convergence test

As seen in Sec. 4.3 the magnitude of the transfer operator S^{fc} determines when the approximations introduced with the multilevel VOF algorithm do not influence the

simulations. Therefore we first consider a test case to determine a consistent set of fine and coarse meshes and verify that S^{fc} is always small during the simulation. We consider a jet of water with density $\rho_l = 1000 \text{ kg/m}^3$, viscosity $\mu_l = 0.003 \text{ kg/m s}$ and surface tension coefficient $\sigma = 0.0725 \text{ J/m}^2$, injected with velocity $U_0 = 1 \text{ m/s}$ from a hole of radius $R = 0.3164 \times 10^{-3} \text{ m}$. The non-dimensional numbers for this case are $Re = 210.93$, $We = 8.73$ and $Oh = 0.014$. The gas is assumed to be air, then the density ratio between the liquid and gas phases is 1000 and the viscosity ratio 100. The amplitude of the sinusoidal oscillation of the inlet velocity is $A = 0.5$, its wavelength $\lambda = 3.5 \times 10^{-3} \text{ m}$, with a reduced wave-number $kR = 0.57$. For this set of physical parameters the critical Weber number is $We_c \approx 1000$, well above the Weber number of this simulation, and the jet evolution is fully axisymmetric.

We first consider no mesh refinements. When the resolution is too low, several small droplets do indeed appear, but they totally generated by numerics. The minimal number of cells that produces a single stable drop is 128 in the radial direction and 768 in the axial one.

To test the multilevel algorithm, we reduce the velocity and pressure grid to 32×192 cells and consider different refinement levels ($f = c + l$, $l = 1, 2, 3, 4$), all of them obtained with mid-point refinement. The most refined mesh will have 512×3072 cells. For $l = 1$ the interface breaks as in the single-level simulations, while for $l = 2, 3, 4$ the color function is well resolved. However, the norm of the transfer operator S^{fc} is not always negligible. The post-processing computation of the operator S_c^{fc} in the coarse grid can be performed with the projection P^{cf} and the restriction R^{fc} operators that are already used in the multigrid solver of the Navier-Stokes equations. We first compute (4.38) for all basis functions \mathbf{v}^f at the fine level

$$(S_f^{fc}, \mathbf{v}^f) = (S^{fc}(P^{cf}(\mathbf{u}^c, p^c)), \mathbf{v}^f). \quad (5.12)$$

We then use the restriction operator R^{fc} to compute the projection S_c^{fc} over the coarse grid

$$(S_c^{fc}, \mathbf{v}^c) = R^{fc}((S_f^{fc}, \mathbf{v}^f)), \quad (5.13)$$

for all basis functions \mathbf{v}^f at the coarse level. In Tab. 5.1 we report the maximum value

level	1	2	3	4
cells	64×384	128×768	256×1536	512×3072
error	8.2×10^{-2}	1.2×10^{-2}	5.1×10^{-4}	7.1×10^{-4}

Table 5.1.: The maximum value of $\|S_c^{fc}\|_\infty / \|\mathbf{f}_s\|_\infty$ for the coarse grid with 32×192 cells, four different refinement levels, in the non-dimensional time interval $[0, 2]$.

during the time interval $[0, 2]$ of the ratio between the transfer operator S^{fc} and the capillary force term. The table shows clearly that we can neglect the transfer operator when $l = 3, 4$, and in this case the computation of the local normals and curvatures can be considered accurate.

5. Numerical simulations

We remark that this does not imply that the pressure and velocity fields at the coarse resolution 32×192 are fully resolved, but only that the interface representation at the two highest refinement levels is accurate for that coarse grid. therefore, we need to test that the solution obtained is mesh independent by refining the coarse grid and analyzing the differences in the solutions. In Fig. 5.6 case *A* has 32×192 cells on the coarse grid, case *B* 64×384 cells and case *C* 128×768 cells. The non-dimensional time step Δt is fixed at 1×10^{-3} . We evaluate the transfer operator with respect to the norm of the

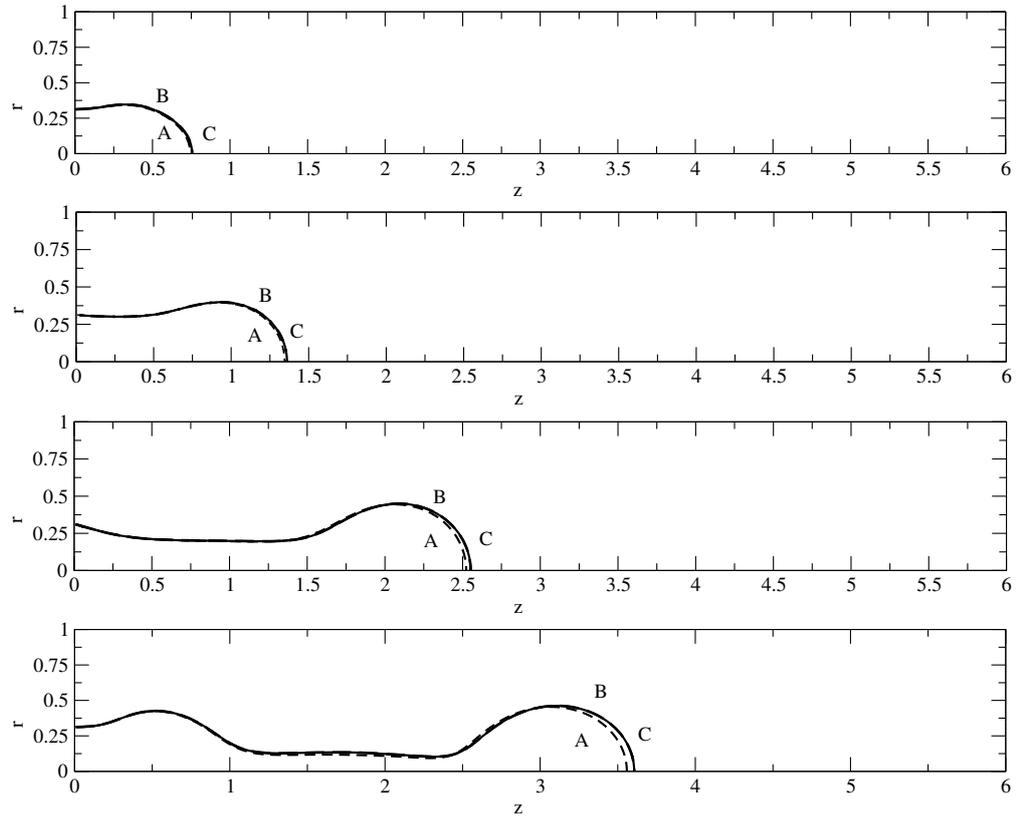


Figure 5.6.: Evolution of the interface line for the three different grid resolutions: 32×192 (*A*), 64×384 (*B*) and 128×768 (*C*), at the non-dimensional times $t = 0.5, 1, 2, 3$.

capillary force and get a small value in all situations. In particular, we get 8.03×10^{-4} for case *A*, 3.63×10^{-5} for case *B* and 1.04×10^{-5} for case *C*. The interface at the lowest resolution is quite different from that obtained in the two other cases, so we infer that the intermediate resolution with 64×384 cells on the coarse grid and three levels of refinement can be used as the reference discretization in all the following simulations.

As a result of the optimized memorization and computation of the interface geometrical properties, the use of different refinement levels does not affect the CPU times and memory needs significantly. Without them, the costs of a simulation with the same

resolution in VOF grid would have required substantial increments in both of them.

Fig. 5.7 shows the details of the evolution of the physical quantities of this simulation. Complete profiles for pressure, velocity and color function on the axis are plotted. The modulated velocity at the inlet generates a bulge followed by a neck, that becomes thinner and thinner as the simulation advances in time. Eventually the head separates forming a droplet and the neck becomes a thread, that detaches again from the following new bulge, evolving in a smaller droplet. While at the beginning the pressure and velocity fields are set by the applied inlet conditions, when we approach the change in topology the capillary force becomes the dominant term. After the detachment, the capillary force acts strongly on the droplet, that oscillates and quickly settles down to a spherical shape. This process is very fast given the small dimensions of the jet. The position of the satellite droplet depends on the timing of the two pinches. If the first breakup occurs in the front of the thread, the neck retracts backwards and it is absorbed by the incoming bulge, giving no time for a second breakup event. When the timing between the anterior and the posterior breakup is very short, the satellite can settle down in-between the big droplet and the jet body. The velocity of the droplets after the separation approaches the mean velocity at the inlet U_0 .

The transfer operator monitoring demonstrates that the results shown in Fig. 5.7 are accurate up to the breakup. Since there is no physical model of the pinching event, the instant in which the change in topology occurs is dependent on the fine grid resolution. As we have seen, the small satellite existence and position are deeply influenced by this timing. Further analysis, with refined coarse grid or an adaptive mesh refinement technique, can be carried on to assure that the fine scale is suitable to reproduce the jet evolution with accuracy [32].

Pulsating jets with different wavelengths

This section is devoted to the analysis of the jet evolution when we vary the wavelength of the applied disturbance, since this parameter appears to be the driving parameter of the evolution. With reference to Fig. 5.4, we see that the reduced wave number $kR = 2\pi R/\lambda$ that generates the maximum growth rate is between 0.6 and 0.7. For this case, the distance where the first drop detaches is the minimal achievable and is usually called *breakup length*. We consider three different wavelengths: $\lambda_2 = 3.5 \times 10^{-3} m$, near the maximum growth rate, $\lambda_1 = 2.5 \times 10^{-3} m$, and $\lambda_3 = 7.5 \times 10^{-3} m$, at opposite sides with respect to the maximum. All other physical parameters are kept constant. In Fig. 5.8 we follow the evolution of the jet with wavelength λ_2 , from its formation up to the breakup of the main droplet and the satellite. This small droplet relaxes to a spherical shape and is eventually absorbed by the incoming jet.

Fig. 5.9 shows the development of the jet with wavelength λ_1 . The first part of the evolution is similar to the previous case, with the formation of a bulge of spherical shape that detaches from the main body at a later time. There is a second breakup behind the filament, also in this case, and the satellite quickly reaches a spherical shape under the action of surface tension. In this case $kR = 0.825$, and Fig. 5.4 clearly indicates that the growth rate is substantially smaller.

5. Numerical simulations

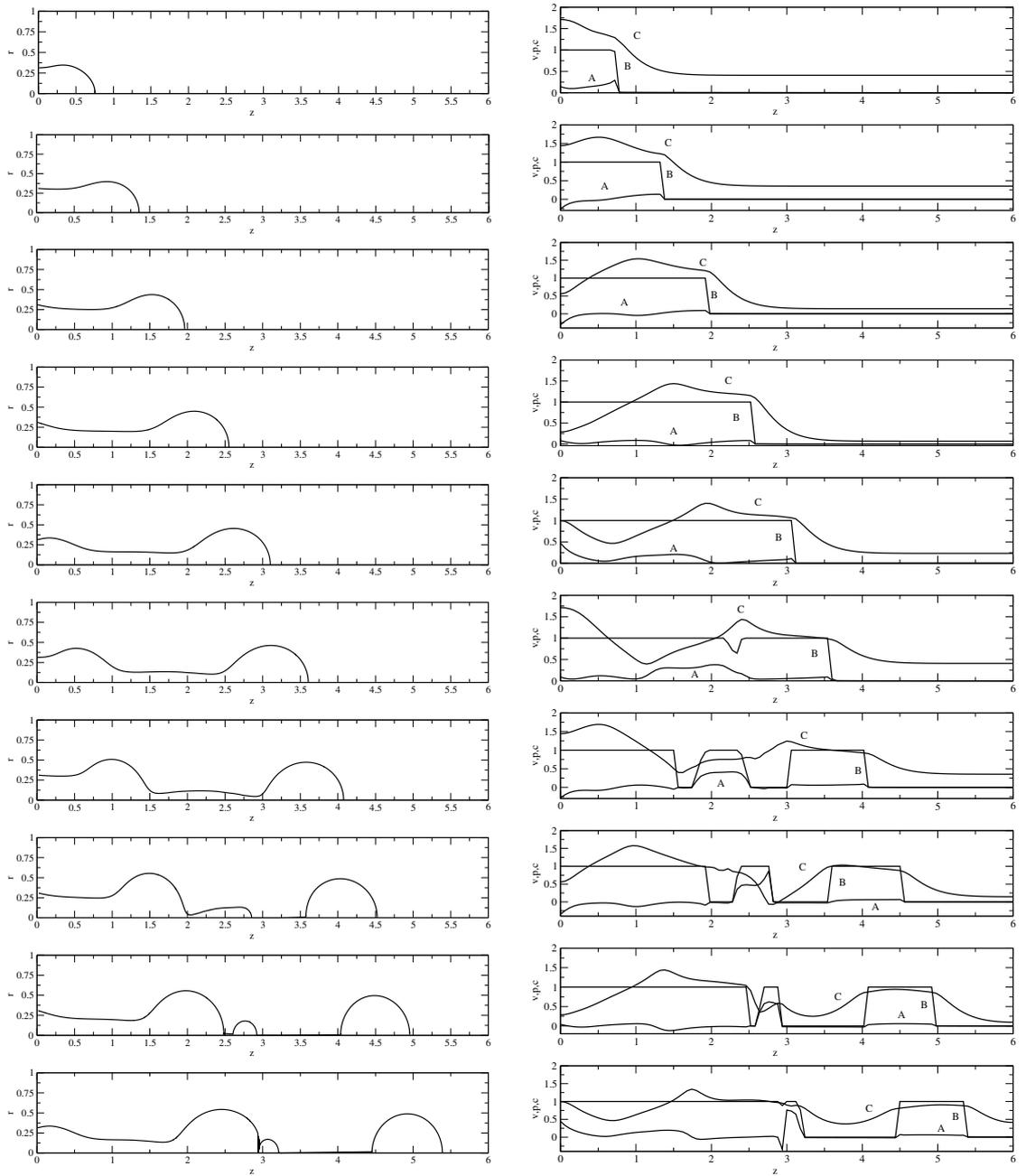


Figure 5.7.: Interface evolution (left) and pressure, color function and velocity profiles along the jet axis (right, lines A, B and C, respectively) at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (top to bottom).

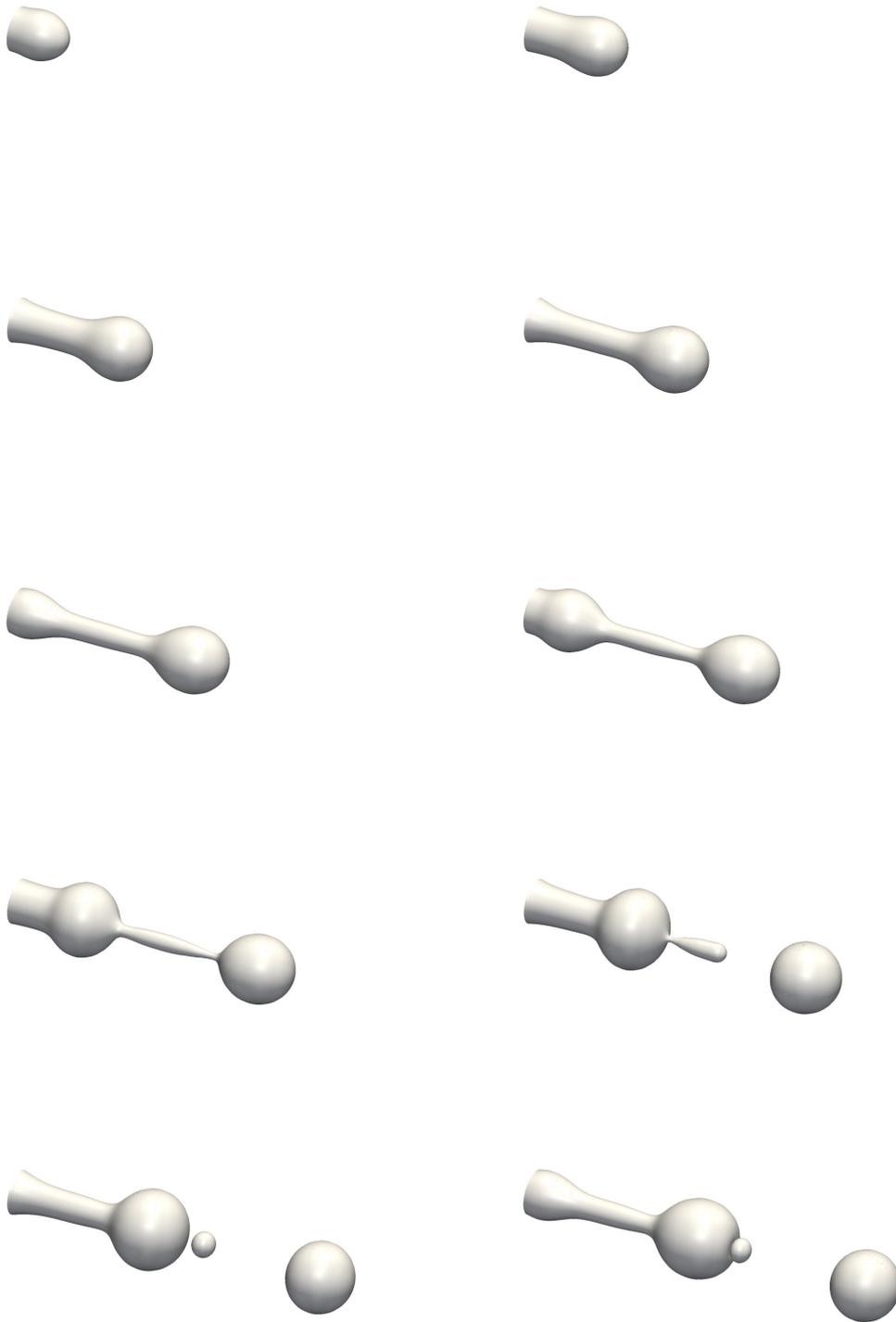


Figure 5.8.: Jet evolution for the intermediate wave length λ_2 at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (left to right and top to bottom).

5. Numerical simulations

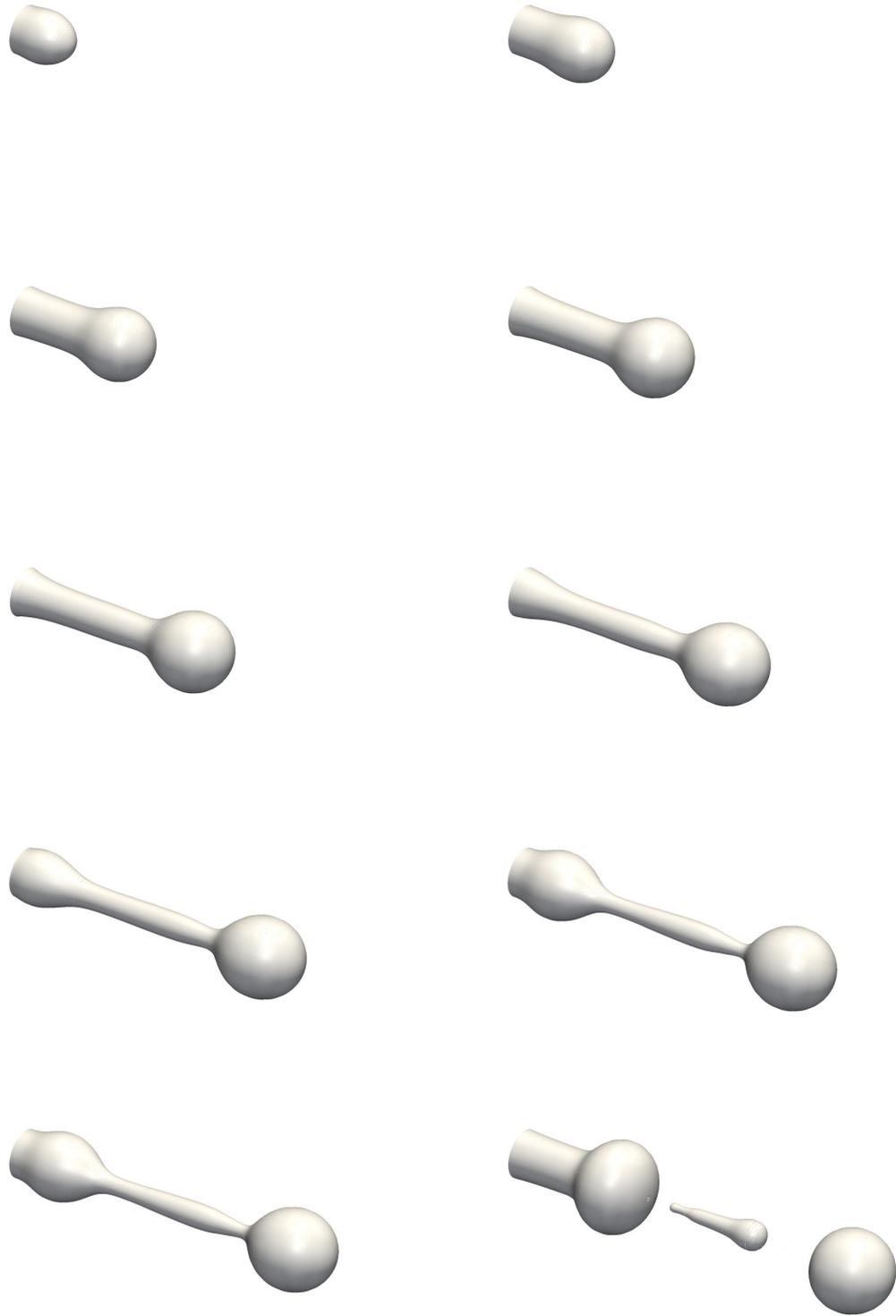


Figure 5.9.: Jet evolution for the shortest wave length λ_1 at times 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (left to right and top to bottom).

5.2. Axisymmetric pulsating jets

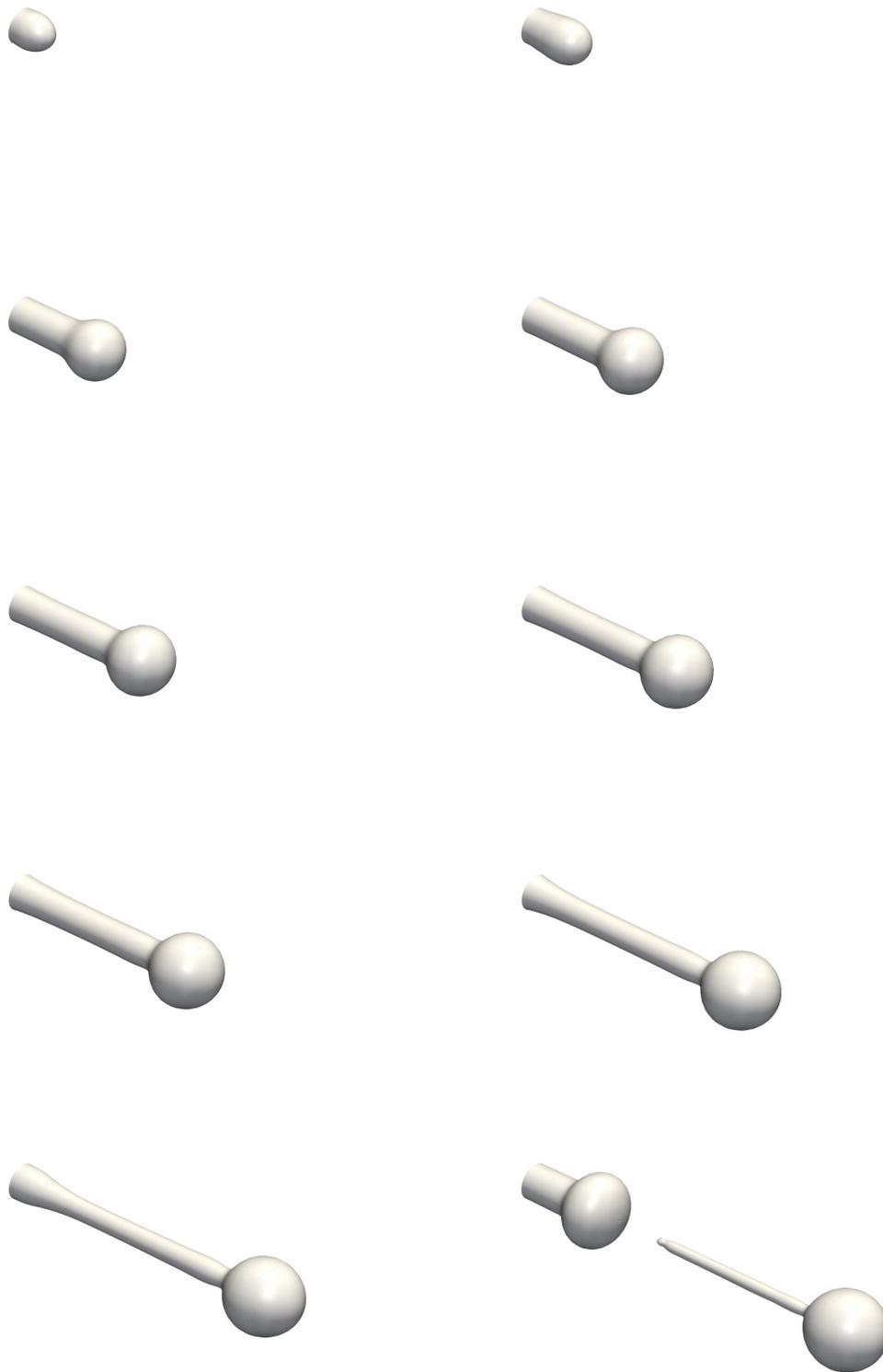


Figure 5.10.: Jet evolution for the longest wave length λ_3 at times 0.5, 1, 2, 2.5, 3, 3.5, 4, 5, 6, 7.5 (left to right and top to bottom).

5. Numerical simulations

Fig. 5.10 shows the evolution for the longest wavelength λ_3 . Since $kR = 0.275$, the growth rate is very small and the bulge develops in a more remarkable way. The connecting neck stretches and the first breakup occurs now at the end of the filament. Surface tension pulls the filament towards the leading bulge.

We have to remark that the jet evolution is accurately reproduced up to the breakup, that can still be dependent on the adopted resolution, especially in the longest wavelength case, where the filament reaches the greater length. At the breakup the value of the transfer operator S^{fc} is at the maximum value during its time evolution.

Pulsating jets with different inlet velocities

Another parameter that modifies significantly the jet behavior is the inlet velocity. As it is increased, turbulence starts to appear and soon becomes the dominant mechanism affecting the interface evolution and breakup. In order to be able to perform direct numerical simulations we stay away from situations in which a subgrid modeling of the turbulence becomes necessary. Turbulence can influence also the velocity profile at the inlet and it can be generated upstream of the nozzle position. A profile can be extracted by a single phase simulation of the region before the nozzle and used as input boundary condition for the two-phase simulation. In this case we monitor again the value of the transfer operator and check that it remains small.

The most important parameter to take into account however is the critical Weber number We_c . The hypothesis of axisymmetry is no more valid when the characteristic Weber number of the jet exceeds this critical value. We will show some simulation where this condition is no longer satisfied to verify the theoretical results.

We consider two cases, one with $We < We_c$ and another with $We > We_c$. The wavelength in the first case is equal to $2.5 \times 10^{-3} m$ and the mean inlet velocity U_0 is $10 m/s$, with an amplitude $A = 0.25$. The liquid jet has density $\rho_l = 1000 kg/m^3$, viscosity $\mu_l = 0.01 kg/m s$ and surface tension coefficient $\sigma = 0.0725 J/m^2$. The geometrical configuration has again $D = 0.328125 \times 10^{-3} m$ and the gas phase is defined by a density ratio of 1000 and a viscosity ratio of 100, typical values of water/air systems. With these parameters we find $Re = 328$, $We = 452$ and $Oh = 0.065$. The critical Weber number depends only on the density ratio of the fluids, so it remains the same of the previous cases, $We_c \approx 1000$. This case is again completely axisymmetric. The coarse grid has 64×384 cells and we use four levels of grid refinement for the color function. The increased velocity requires a smaller timestep then before and we use $\Delta t = 1 \times 10^{-4}$. When $t > 4$ the transfer operator S^{fc} becomes appreciable in magnitude, when the fluid structures near the jet head become thin, as shown in Fig. 5.11. The formation of the bulge is now influenced by the relative velocity that the jet faces, leading to the development of a cylindrical sheet that surrounds the main liquid core that is similar to the Rayleigh-Taylor instability. The use of finer grids does not bring more accurate results as no breaking event appears.

The case with $U_0 = 100 m/s$ is shown in Fig. 5.12. The physical parameters of the simulation do not change, except for the surface tension σ that now is $1 J/m^2$. The Weber number is now 3281 and it is bigger then the critical value $We_c \approx 1000$. Furthermore,

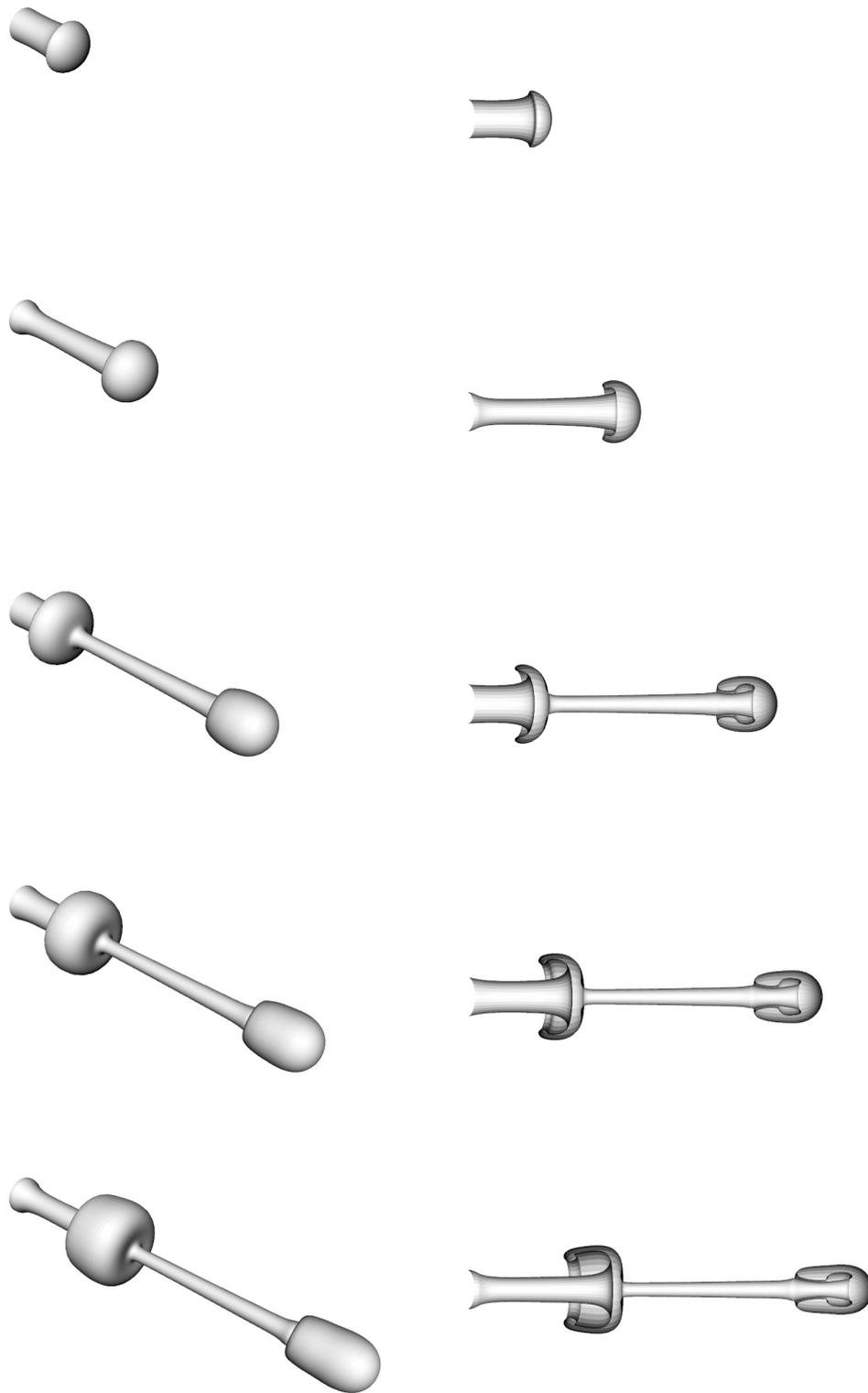


Figure 5.11.: Jet evolution with inlet velocity $U_0 = 10 \text{ m/s}$ at times 1, 2, 3.5, 4, 4.5 (two different views in each row).

5. Numerical simulations

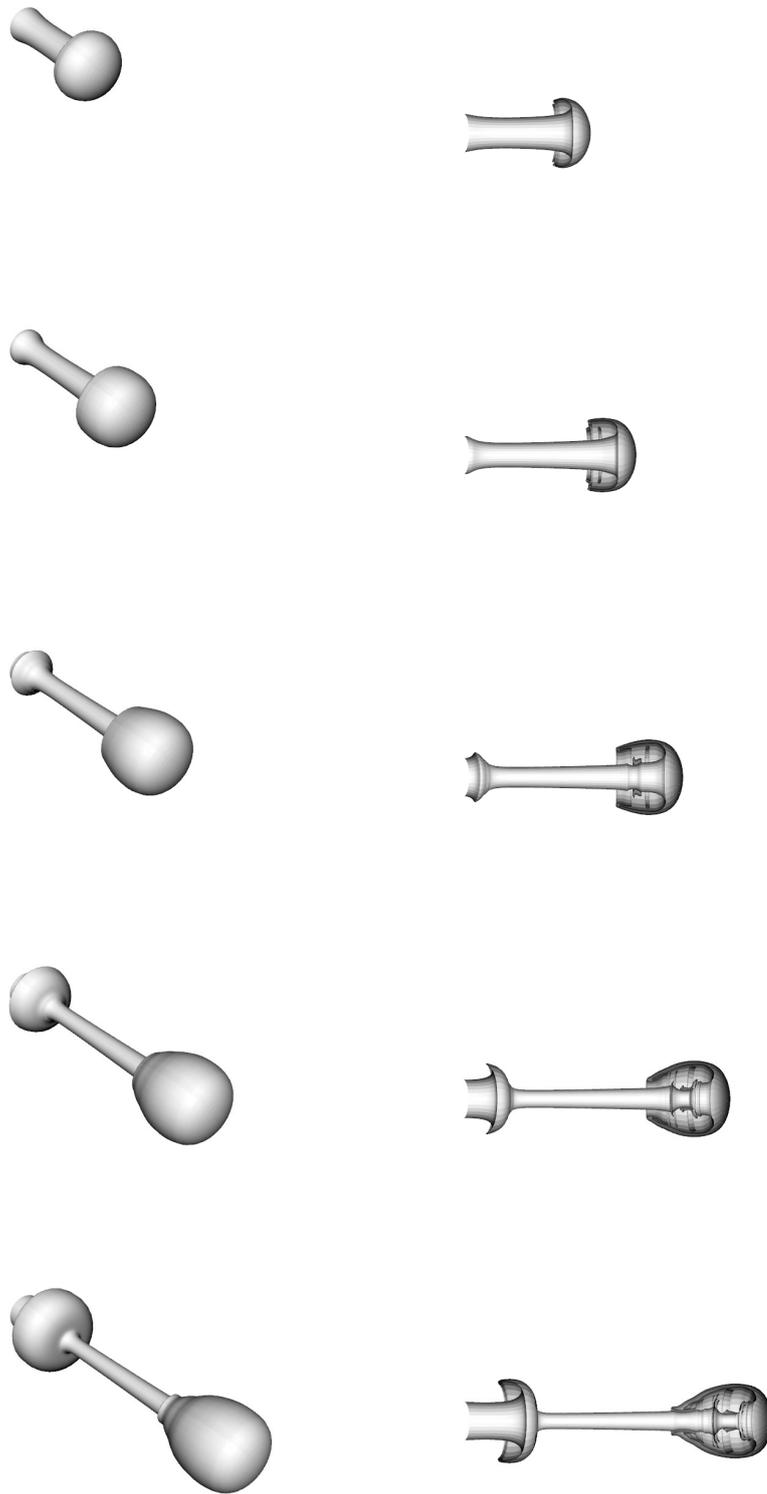


Figure 5.12.: Jet evolution with inlet velocity $U_0 = 100 \text{ m/s}$ at times 1.5, 2, 2.5, 3, 3.5
88 (two different views in each row).

5.2. Axisymmetric pulsating jets

the higher Reynolds number requires the use of a turbulence model, and we choose a simple Smagorinsky LES model together with an upwind scheme to cope with turbulence. The longitudinal section of the jet shows rings of fluid that appear around the jet body. As predicted by the theory, in this situation the axisymmetry hypothesis is no longer valid and the jet evolution is characterized by unrealistic structures. In these conditions real jets would produce fingers and break in several small droplets. The transfer operator becomes very large when $t > 4$, giving another indication that the simulation is not accurate. The transfer operator should be modeled as a macroscopic term because its importance can not be neglected any more.

Part III.

**Nuclear Reactors
Thermal-Hydraulics**

6. Porous media approach for nuclear reactor cores

6.1. Energy equation

In order to describe a model for a core of a nuclear reactor we need to introduce a heat exchange model. The energy balance equation, written as a function of the temperature T , can be introduced in the following form

$$\frac{\partial}{\partial t}(\rho C_p T) + \nabla \cdot (\rho \mathbf{u} C_p T) = \Phi + \nabla \cdot (k \nabla T) + \dot{Q}, \quad (6.1)$$

where ρ is the density, C_p is the pressure specific heat and k the heat conductivity. All these physical properties are assumed to be slightly variable as a function of the temperature. \dot{Q} is the volumetric heat source and Φ the dissipative heat term. The domain in analysis is again denoted by Ω , with $\Gamma = \partial\Omega$ its boundary. Suitable initial and boundary conditions must integrate (6.1).

The derivation of the weak form and its discretization follow the same steps already analyzed for the Navier-Stokes equation. The energy equation is a pure diffusion/advection problem for the scalar variable T . The ellipticity of the diffusive operator allows the existence of a weak solution, while the uniqueness is linked to the advection operator just like we have seen for the momentum equation. Let $H(\Omega)$ be the space of the weak solutions of (6.1) and φ be its test function. After integrating by parts the diffusive term on the whole domain we get

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial t}(\rho C_p T) \varphi dV + \int_{\Omega} \nabla \cdot (\rho \mathbf{u} C_p T) \varphi dV &= \int_{\Omega} \Phi \varphi dV - \int_{\Omega} k \nabla T \cdot \nabla \varphi dV + \\ &+ \int_{\Omega} \dot{Q} \varphi dV + \int_{\Gamma} (k \nabla T \cdot \mathbf{n}) \varphi dS, \quad \forall \varphi \in H(\Omega). \end{aligned} \quad (6.2)$$

The boundary conditions allows the computation of the last term in (6.2), as seen many times in the previous sections. We introduce the bilinear form $a(\cdot, \cdot)$ and the trilinear form $c(\cdot, \cdot, \cdot)$ defined by

$$a(v, v) = \int_{\Omega} k \nabla v \cdot \nabla v dV, \quad \forall v \in H(\Omega), \quad (6.3)$$

$$c(\mathbf{u}, v, v) = \int_{\Omega} \nabla \cdot (\rho \mathbf{u} C_p v) v dV, \quad \forall \mathbf{u} \in \mathbf{V}(\Omega), \forall v \in H(\Omega). \quad (6.4)$$

We also define an inner product in $H(\Omega)$ designed by (\cdot, \cdot) and the inner product on the boundary Γ designed by $(\cdot, \cdot)_{\Gamma}$. We can rewrite (6.2) as

$$\left(\frac{\partial}{\partial t}(\rho C_p T), \varphi\right) + c(\mathbf{u}, T, \varphi) = (\Phi, \varphi) - a(T, \varphi) + (\dot{Q}, \varphi) + (k \nabla T, \varphi)_{\Gamma}. \quad (6.5)$$

6. Porous media approach for nuclear reactor cores

The discretization of (6.2) projects the equation from the infinite dimensional $H(\Omega)$ to the finite dimensional $H_h(\Omega)$, and the corresponding solution T to T_h . If $H_h(\Omega)$ is a polynomial space of order p the finite element version of (6.2) is written as

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial t} (\rho C_p T_h) \varphi_h dV + \int_{\Omega} \nabla \cdot (\rho \mathbf{u} C_p T_h) \varphi_h dV = \int_{\Omega} \Phi_h \varphi_h dV - \int_{\Omega} k \nabla T_h \cdot \nabla \varphi_h dV + \\ + \int_{\Omega} \dot{Q}_h \varphi_h dV + \int_{\Gamma} (k \nabla T_h \cdot \mathbf{n}) \varphi_h dS \quad \forall \varphi_h \in H_h(\Omega). \end{aligned} \quad (6.6)$$

The parameter $h > 0$ identifies the typical dimension of the finite element. \dot{Q}_h and Φ_h are the projection through i_h^T of \dot{Q} and Φ in the same way as in (3.11). The velocity field \mathbf{u} is considered a known field obtained by the solution of the Navier-Stokes equation. The Galerkin approximation can be used to solve the energy equation with $\{\varphi_h\}_i$ as a basis of $H_h(\Omega)$. With this setting we recover a linear system of equations that corresponds to (6.1) and we can solve it with some of the methods analyzed in Chap. 3. We remark that, since the velocity is taken as an external parameter, the solution of (6.6) is independent from the resolution of the velocity and pressure fields. For the physical properties that depend on the temperature, we will use the temperature value of the previous time step in order to keep the system linear. If the velocity field is also computed as part of the algorithm, the two variables are not strictly coupled, but the value at the previous iteration is used in the coupling terms.

6.2. Porous media approach

Since the reactor core is a very complex system, made up of thousands of fuel pins, arranged in fuel elements, control bars and structural components, it is unlikely to perform a simulation with a detailed representation of all these parts. The approach followed in our situations is to introduce two scales, in a similar manner to what may be done for turbulence subgrid or neutron flux analysis inside the same complex reactor core. The fine scale represents all the geometrical entities, with fine spatial extension and all the details. The coarse scale simulates instead the core as a whole, where the details of the geometry are not taken into account and each coarse grid cell is made up of a lot of sub-scale components. Just like the VOF multilevel approach, we try to use the fine scale data to get some characteristic parameters for the coarse scale cell that is here seen as an homogenization of its components. This approach is generally called *porous media approach*, since it is also typical in the modeling of this kind of materials.

6.2.1. Two-level finite element Navier-Stokes system

In this section we will consider the porous media approximation for all the equations of the Navier-Stokes system, since this modeling will influence the coarse scale formulation of all of them. Let us consider a two level solution scheme where a fine level and a coarse level solution can be defined. We will denote the fine grid quantities without any sign, while the averaged values on the coarse grid will have a cap to identify them, so the

set of solution on the coarse grid will be $\{\widehat{p}_h, \widehat{\mathbf{u}}_h, \widehat{T}_h\}$ with corresponding basis functions $\{\widehat{q}_h\}_i$, $\{\widehat{\mathbf{v}}_h\}_i$ and $\{\widehat{\varphi}_h\}_i$ in $\widehat{Q}_h(\Omega)$, $\widehat{\mathbf{V}}_h(\Omega)$ and $\widehat{H}_h(\Omega)$, respectively. We drop again the h subscript from all equations after this point.

The continuity equation on the coarse grid is determined by the fine grid just like we have seen in Sec. 4.3, so (4.32) holds.

For the momentum conservation equations, we recall (3.11) and follow the same procedure seen in Sec. 4.3. We therefore get again the fine-to-coarse momentum transfer operator P^{fc} of (4.35) and the fine-to-coarse turbulent transfer operator T^{fc} of (4.36). When there is a stress applied on the boundary we must take into account the operator $K^{fc}(\mathbf{u})$ for which

$$\int_{\Omega} K^{fc}(\mathbf{u}) \cdot \widehat{\mathbf{v}} dV = \int_{\Gamma} (\bar{\boldsymbol{\tau}} \cdot \mathbf{n}) \cdot \widehat{\mathbf{v}} dS, \quad (6.7)$$

called *fine-to-coarse friction dissipation operator*.

We now apply the same procedure to the energy equation. The equation of the fine level is (6.5) that we rewrite as

$$(E(T, \mathbf{u}), \varphi) = 0. \quad (6.8)$$

We substitute into it the coarse level solutions $\widehat{\mathbf{u}}$ and \widehat{T} to get

$$\begin{aligned} \left(\frac{\partial}{\partial t} (\rho C_p \widehat{T}), \varphi \right) + c(\widehat{\mathbf{u}}, \widehat{T}, \varphi) - (\Phi, \varphi) + a(\widehat{T}, \varphi) - (\dot{Q}, \varphi) = \\ = (P_e^{fc}(\widehat{T} - T, \widehat{\mathbf{u}} - \mathbf{u}), \varphi) + (T_e^{fc}(\widehat{T}, T, \widehat{\mathbf{u}}, \mathbf{u}), \varphi) + (S_e^{fc}(T), \varphi). \end{aligned} \quad (6.9)$$

The three new transfer operators are denoted by an e since they appear in the energy equation. The first one is the *fine-to-coarse energy transfer operator* P_e^{fc} defined by

$$P_e^{fc}(\widehat{T} - T, \widehat{\mathbf{u}} - \mathbf{u}) = E(\widehat{T} - T, \widehat{\mathbf{u}} - \mathbf{u}). \quad (6.10)$$

T_e^{fc} is the *turbulent energy transfer operator* defined by

$$T_e^{fc}(\widehat{T}, T, \widehat{\mathbf{u}}, \mathbf{u}) = \nabla \cdot (\rho C_p \widehat{\mathbf{u}} \widehat{T}) - \nabla \cdot (\rho C_p \mathbf{u} T) - \nabla \cdot (\rho C_p (\widehat{\mathbf{u}} - \mathbf{u}) (\widehat{T} - T)). \quad (6.11)$$

The last one, $S_e^{fc}(T)$, is related to the boundary conditions and defined by

$$\int_{\Omega} S_e^{fc}(T) \widehat{\varphi} dV = \int_{\Gamma} (k \nabla T \cdot \mathbf{n}) \cdot \widehat{\varphi} dS. \quad (6.12)$$

The operator above is called *fine-to-coarse flux source operator*.

6.2.2. Transfer operator modeling

Differently from the two-phase case, where we have used a coarse grid that can describe accurately the evolution of the system, when dealing with nuclear reactor modeling we adopt a coarse mesh that is not reproducing the actual geometry of the core. For this reason some of the transfer operators must not be set to zero.

6. Porous media approach for nuclear reactor cores

We want to determine the averaged value of the unknowns \hat{p} , $\hat{\mathbf{u}}$ and \hat{T} that describe the general behavior of the fluid inside the core. We can assume that the incompressibility constraint is satisfied on both grids, so there is no additional terms to take into account for the continuity equation.

The turbulent transfer operator T^{fc} in the momentum equation is modeled following the *Reynolds' hypothesis*, namely

$$T^{fc}(\mathbf{u}, \hat{\mathbf{u}}) = \nabla \cdot \hat{\tau}_t, \quad (6.13)$$

where the turbulent tensor $\hat{\tau}_t$ is

$$\hat{\tau}_t = \mu_t(\nabla \hat{\mathbf{u}} : \nabla \hat{\mathbf{u}}^T), \quad (6.14)$$

with μ_t the turbulent viscosity. In order to determine the value of μ_t we can use several turbulence models. In this application the value of μ_t is determined with a simple Smagorinsky model for Large Eddy Simulation (LES) [33].

The modeling of the momentum transfer operator P^{fc} is similar and leads to

$$P^{fc}(\hat{p}-p, \hat{\mathbf{u}}-\mathbf{u}) = \zeta(\mathbf{x}) \left(\left(\frac{\partial \rho \hat{\mathbf{u}}}{\partial t}, \hat{\mathbf{v}} \right) + a(\hat{\mathbf{u}}, \hat{\mathbf{v}}) + b(\hat{p}, \hat{\mathbf{v}}) - (\rho \mathbf{g}, \hat{\mathbf{v}}) - (\hat{\tau}_{\text{eff}}, \nabla \hat{\mathbf{v}}) \right), \quad (6.15)$$

where $\zeta(\mathbf{x})$ is the fraction of fuel and structural material in the total volume. The tensor $\hat{\tau}_{\text{eff}}$ is defined as

$$\hat{\tau}_{\text{eff}} = \mu_{\text{eff}}(\nabla \hat{\mathbf{u}} : \nabla \hat{\mathbf{u}}^T). \quad (6.16)$$

The value of μ_{eff} must be calculated with a sub-channel simulation or determined by experimental data.

We model the friction energy that is dissipated at the fine level with the operator $K^{fc}(\mathbf{u})$. We assume for this term a simple model that recalls the pressure drops in a pipe and write

$$K^{fc}(\mathbf{u}) = \zeta(\mathbf{x}) \frac{2\rho \hat{\mathbf{u}} \|\hat{\mathbf{u}}\|}{D_{eq}} \lambda \quad (6.17)$$

where D_{eq} is the equivalent diameter of the channel and λ is a friction coefficient. The determination of this parameter should also come from experimental or numerical data of sub-channel simulations.

Regarding the additional terms in the energy equation (6.9), the turbulent energy transfer operator is modeled following the *Reynolds' analogy* for the turbulent Prandtl number Pr_t so that

$$T_e^{fc}(\hat{T}, T, \hat{\mathbf{u}}, \mathbf{u}) = \nabla \cdot \left(\frac{\mu_t}{Pr_t} \nabla \hat{T} \right), \quad (6.18)$$

where μ_t is the turbulent viscosity previously defined. The energy exchange operator P_e^{fc} is defined on the coarse grid as

$$P_e^{fc}(T - \hat{T}) = \zeta(\mathbf{x}) \left(\frac{\partial}{\partial t} (\rho C_p \hat{T}) + \nabla \cdot (\rho C_p \hat{\mathbf{u}} \hat{T}) - \Phi - \dot{Q} - \nabla \cdot (k_{\text{eff}} \nabla \hat{T}) \right). \quad (6.19)$$

Once again, the new parameter k_{eff} can be defined with direct simulations of the channel or sub-channel configurations or by experiment. The operator $S_e^{fc}(T)$ represents the heat source that is generated through the fuel pin surfaces. For the heat production in the core we may assume

$$S_e^{fc}(T) = W_0 \sin\left(\pi \frac{z - H_{in}}{H_{out} - H_{in}}\right), \quad (6.20)$$

where H_{in} and H_{out} are the heights where the heat generation starts and ends. Outside this interval there is no generation. The quantity W_0 is assumed to be a known function of space which is defined by the power distribution factor.

6.3. Numerical simulations

In this section we show some results obtained with the model previously described. These simulations are meant to be an auxiliary tool in the reactor design process, so they must provide fast results for an always evolving configuration. The extracted data do not need to be accurate but must represent the general behavior and react coherently to system modifications. The use of a fully three-dimensional code allows the possibility to analyze asymmetric effects that a traditional mono-dimensional tool, such as the ones in use in reactor design from long time, are not able to capture.

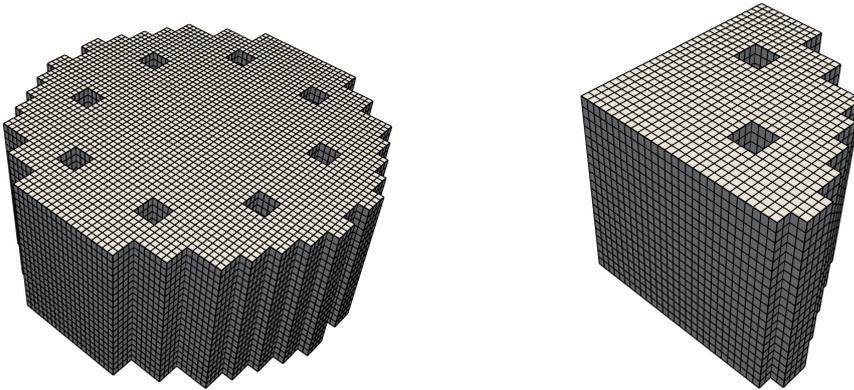


Figure 6.1.: Full reactor (left) and computational domain (right).

The simulation shown in this chapter are for a generic configuration of an open core liquid metal reactor with square fuel elements arranged as shown in Fig. 6.1. Thanks to symmetry we can consider only one quarter of the reactor. The holes correspond to control rods that are not crossed by the coolant.

6.3.1. Boundary conditions

The boundary conditions used for the simulation consider known values for p_{in} and T_{in} on the inlet surface that corresponds to the bottom of the core. The tangential component

6. Porous media approach for nuclear reactor cores

of the velocity is set to zero. The planes at $x = 0$ and $y = 0$ are planes of symmetry. At the external surface of the reactor free slip boundary conditions are set, since we are calculating a mean value for velocity on a whole element, and no heat flux. On the top we set outflow boundary conditions and a fixed profile of pressure $p = p_0(x, y)$.

6.3.2. Thermophysical properties of liquid metals

The simulations have been performed in the perspective of innovative nuclear reactor design, so the coolant we consider is liquid lead, that is one of the possible choices for IV generation nuclear reactors. In general, lead-bismuth eutectic (LBE) provides better characteristics from many points of view, such as cross sections, radiation damage, activation, and melting point. Bismuth is however expensive and therefore reactor design that uses only lead are becoming popular. From the thermal-hydraulics point of view, the thermophysical properties involved in the simulations, that are density and viscosity, do not present significant differences between lead and LBE.

The lead density is assumed to be a function of temperature

$$\rho = (11367 - 1.1944T) \frac{Kg}{m^3}, \quad (6.21)$$

for lead in the range $600 K < T < 1700 K$. The viscosity μ comes from the relation

$$\mu = 4.55 \times 10^{-04} e^{(1069/T)} Pa s, \quad (6.22)$$

for lead in the range $600 K < T < 1500 K$. the thermal conductivity κ is given by

$$\kappa = 15.8 + 108 \times 10^{-4} (T - 600.4) \frac{W}{m K}, \quad (6.23)$$

and the specific heat capacity at constant pressure C_p is fixed in the range of temperatures considered and it is

$$C_p = 147.3 \frac{J}{Kg K}. \quad (6.24)$$

6.3.3. Working conditions

Property	Value
ρ	10562
μ	0.0022
κ	16.58
C_p	147.3
D_{eq}	0.0129

Table 6.1.: Geometrical and physical properties at $T=673.15 K$ in SI units.

The working conditions for pressure are $p_{in} = 36 KPa$ and for temperature $T_{in} = 673.13 K$. We show the value of each of the thermophysical properties at this temperature in Tab. 6.1, together with the geometrical properties from the subgrid modeling.

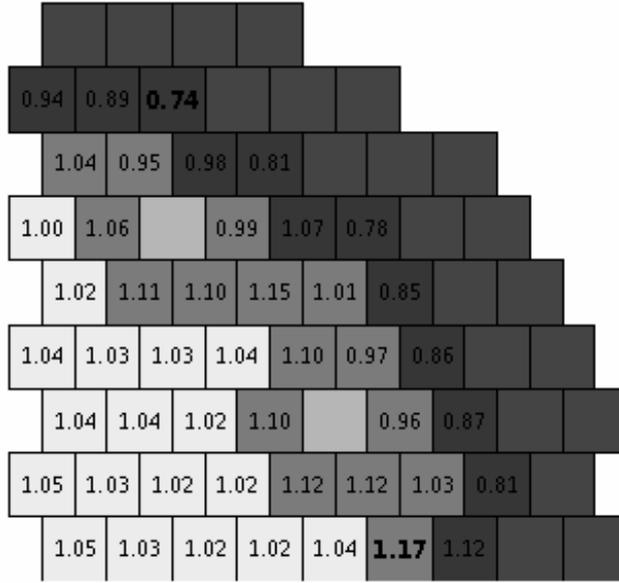


Figure 6.2.: Power distribution. Each generating element is identified by its peak factor, all others are non generating elements.

The power distribution assumed in the simulations is shown in Fig. 6.2. On the axial direction, the total length is set to 2 m and the generation acts only between $H_{in} = 0.9\text{ m}$ and $H_{out} = 1.8\text{ m}$. We assume a sinusoidal shape in the axial direction for the power source, as seen already in (6.20), that we rewrite slightly differently as

$$\dot{q}''' = W_0 \frac{\pi}{2(H_{out} - H_{in})} \sin\left(\pi \frac{z - H_{in}}{H_{out} - H_{in}}\right), \quad (6.25)$$

with W_0 constant over all the domain. We recall that no heat generation is present outside the interval $[H_{in}, H_{out}]$. The choice for W_0 can be determined from the total power production per unit volume and in our simulations is set to $120.25\text{ MW}/\text{m}^3$, that corresponds roughly to a reactor power of 1500 MW_{th} .

6.3.4. Simulations of an open core reactor

In this section we show some results of the simulations performed. We have considered the same configuration and compared the results when the physical properties of the coolant are kept constant at the inlet temperature value and when they depend on temperature. A constant value is set for the pressure at the outlet surface and it is assumed to be the reference pressure $p_r = 0$. The maps for velocity, pressure, density and temperature in the latter case are shown in Figs. 6.3 and 6.4. We can see that the velocity field is quite regular on the outlet surface, with an oscillation between the maximum and the minimum below 3% around the mean value of 1.58 m/s . The pressure drop is modeled

6. Porous media approach for nuclear reactor cores

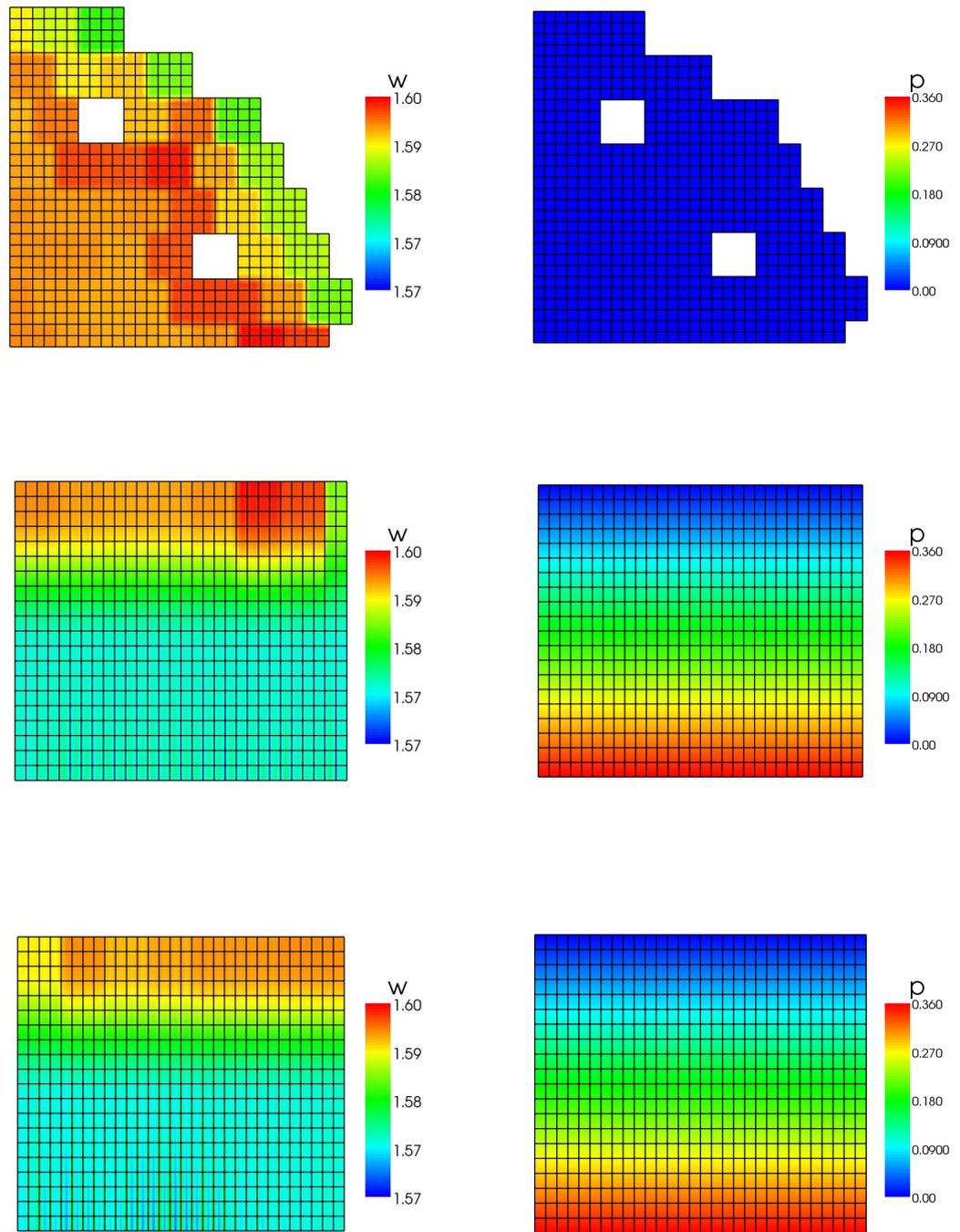


Figure 6.3.: Velocity (left) and pressure (right) distribution on different planes: $z = 2$ (top), $y = 0$ (middle), $x = 0$ (bottom).

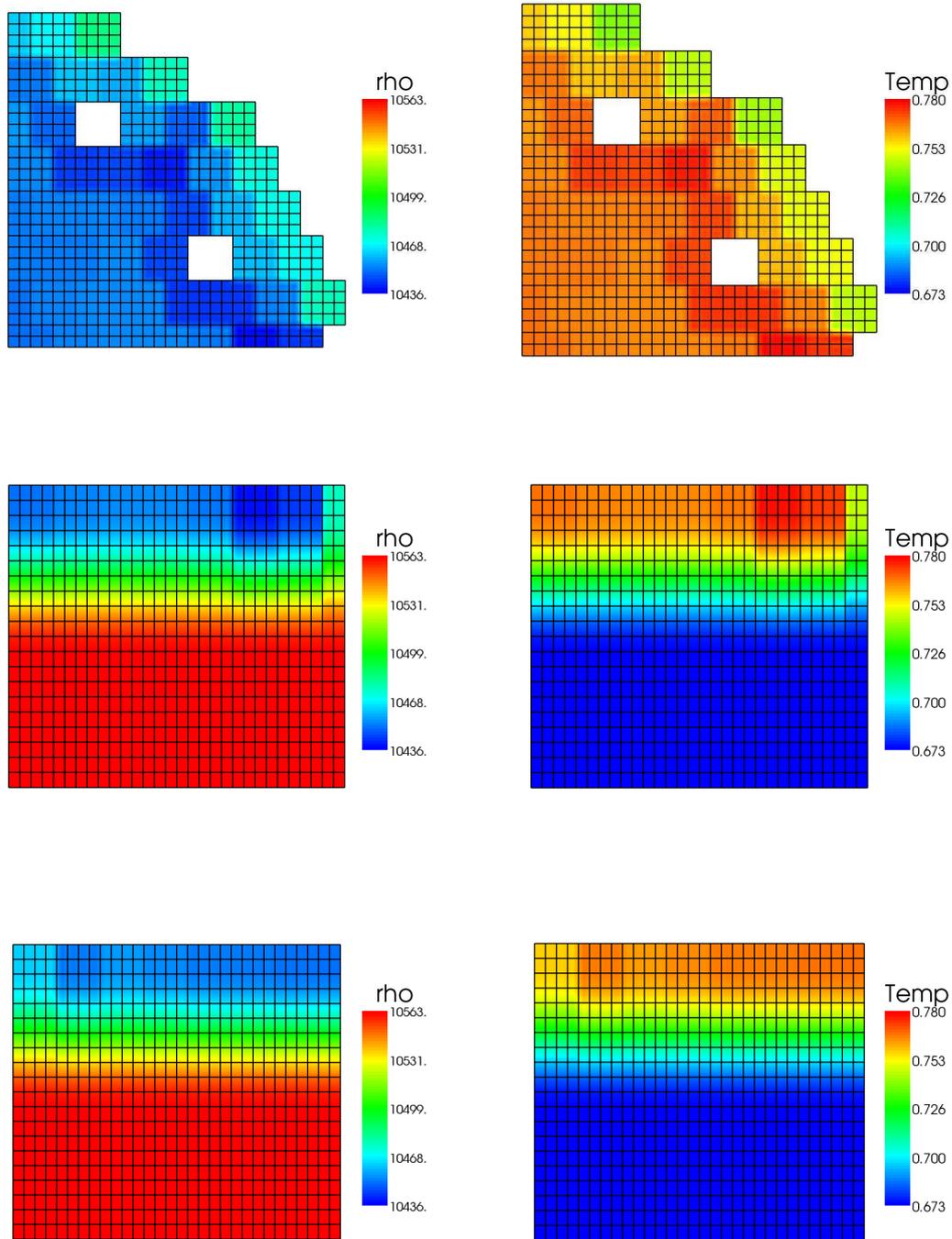


Figure 6.4.: Density (left) and temperature (right) distribution on different planes: $z = 2$ (top), $y = 0$ (middle), $x = 0$ (bottom).

6. Porous media approach for nuclear reactor cores

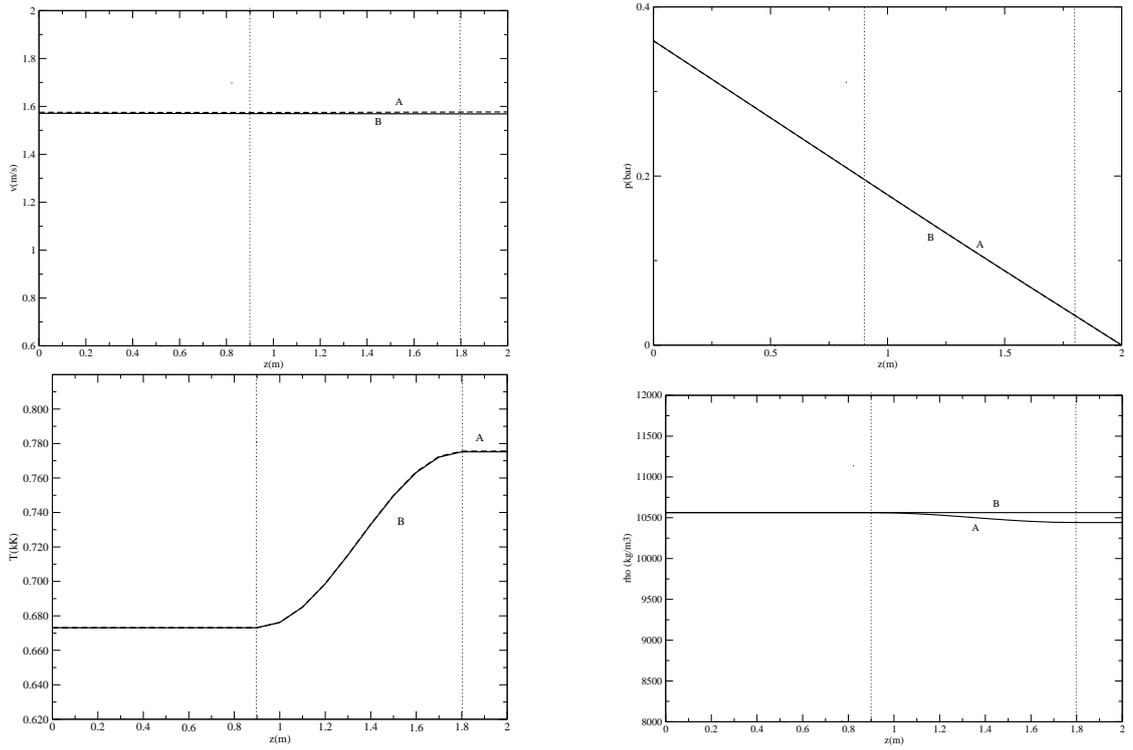


Figure 6.5.: Comparison between variable (A) and constant (B) physical properties with respect to temperature: velocity and pressure (top), density and temperature (bottom) on the line $x = y = 1.1$.

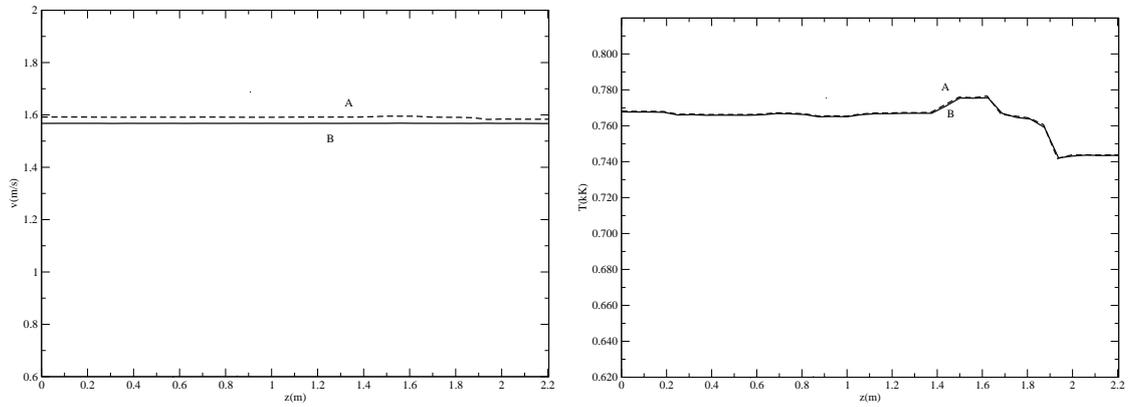


Figure 6.6.: Comparison between variable (A) and constant (B) physical properties with respect to temperature: velocity and temperature profiles on the line $y = x, z = 2$.

only with the term (6.17) and the friction coefficient is determined classically from the Moody diagram, in absence of experimental or computational data from the fine grid. The case illustrated does not report additional localized pressure drop generated by the presence of spacing grids, so the drop is constant in each section and the pressure profile is linear. The temperature profile follows almost exactly the imposed power distribution, meaning that there is a very low heat exchange and mixing among fuel assemblies. The temperature jump is around 80 K and this value was the target assumed for the core. In Fig. 6.5 we can analyze the behavior along the z -axis at a particular point. We have reported both curves for the case with and without temperature dependence. The only variable which is modified by temperature dependence is the density, that decreases significantly in the upper part of the core. The temperature profile is not quite influenced by this and the jump between the inlet and the exit remains around 80 K . In Fig. 6.6 we can see the velocity and temperature variations along a line on the top of the core. A small variation in velocity between these two cases can be seen even if the temperature profile remains substantially the same.

Conclusions

The focus of this thesis has been the development of a new finite element code for engineering application in the field of thermal fluid dynamics and the analysis of its performances and capabilities. The first three chapters have illustrated the mathematical foundations of the method and described the numerical approach for the resolution of the system of equations. Afterwards, two very different fields of applications have been considered to test the validity of the code. In the fourth chapter a VOF technique for interface capturing has been described for direct numerical simulations of two-phase flows. In the fifth chapter the simulation results are presented, starting with a comparison with the analytical linear dispersion equation of a perturbed liquid column and then by showing the evolution of the jet leading to the breakup of the whole column for different physical parameters. The sixth chapter deals with the second topic, that is the simulation of the thermal-hydraulics of a reactor core with a porous media approach. This model is applied to a new generation of liquid metal-cooled nuclear reactors.

The code has been completely redesigned and rewritten, and most of the time of this three-year project has been devoted to the development and testing of the main components of the code, rather than the simulation of different applications of interest. The code has been also coupled to several standard data format, such as HDF, and graphical open-source software, such as Paraview. However, a few important features are still missing, such as parallelism, either for shared and distributed memory systems, and the support for different open-source math libraries. Nevertheless, the modular structure is a key feature of this code, and it allows the introduction of new software blocks in a relatively straightforward way. A few new libraries are actually being developed and they will help to create a general purpose computational tool. The benefits of working with an in-house code reside in the fact that each component, starting from the smallest one up to the main framework, is completely under control of the developers and of experienced users.

The two-phase simulations of liquid jets have shown a good agreement with the experimental and numerical data available in the literature. The multilevel VOF technique has proven to be a powerful tool for an optimal use of the computer resources, by considering a very fine grid for the interface evolution and a coarser grid for the dynamics. This has been particularly helpful for a smoother and better resolved representation of the interface. The simulations discussed in this thesis are only two-dimensional, but the code is now ready to perform fully three-dimensional cases.

The simulations have shown good results also in the field of thermal-hydraulics for nuclear reactors. The porous approach has proved to be suitable and effective when the real geometrical configuration can not be reproduced with the available computational resources. A weak point remains the set of parameters that must come from subchannel

Conclusions

simulations. Here they have been calculated classically in mono-dimensional fashion, but the code can now simulate the subchannel configuration and this will give a significant improvement on the quantitative aspects of the results. Another great improvement that will be introduced is the coupling with the in-development neutronics module, that will give an evolving power source directly derived from the thermal flux.

Bibliography

- [1] R. Temam. *Navier-Stokes Equations and Nonlinear Functional Analysis*. SIAM, Philadelphia, 1995. 7, 41
- [2] R. Adams. *Sobolev Spaces*. Academic Press, New York, 1975. 7, 12, 13
- [3] V. Girault and P. Raviart. *Finite Element Method for Navier-Stokes Equations. Theory and Algorithms*. Springer, New York, 1986. 7
- [4] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*. Springer Verlag, 2002. 7
- [5] J.R. Whiteman. The mathematics of finite elements and applications. *IMA Journal of Applied Mathematics*. 17
- [6] A.M. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer, 2008. 17
- [7] C. Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46:224–243, 1901. 20
- [8] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1-2):1–88, 1997. 27
- [9] P.G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, 1978. 29
- [10] J.L. Guermond and L. Quartapelle. Calculation of incompressible viscous flows by an unconditionally stable projection FEM. *Journal of Computational Physics*, 132(1):12–33, 1997. 31
- [11] B.A. Szabó and I. Babuška. *Finite element analysis*. Wiley-Interscience, 1991. 32
- [12] S. Manservigi. Numerical Analysis of Vanka-Type Solvers for Steady Stokes and Navier–Stokes Flows. *SIAM Journal on Numerical Analysis*, 44:2025, 2006. 36
- [13] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial Mathematics, 2003. 39
- [14] J.L. Guermond and L. Quartapelle. On stability and convergence of projection methods based on pressure Poisson equation. *International Journal for Numerical Methods in Fluids*, 26(9):1039–1053, 1998. 40

Bibliography

- [15] J.L. Guermond and J. Shen. A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics*, 192(1):262–276, 2003. 41
- [16] L. Quartapelle. *Numerical solution of the incompressible Navier-Stokes equations*. Springer, 1993. 43
- [17] P. Wesseling. *Introduction to multigrid methods*. Storming Media, 1995. 44
- [18] V.V. Shaidurov and J. Xu. *Multigrid methods for finite elements*. Kluwer Academic Publishers, 1995. 47
- [19] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31(1):567–603, 1999. 57
- [20] J.E. Pilliod and E.G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004. 57, 58
- [21] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152(2):423–456, 1999. 59
- [22] E. Aulisa, S. Manservigi, R. Scardovelli, and S. Zaleski. A geometrical area-preserving volume-of-fluid advection method. *J. Comput. Phys.*, 192:355–364, 2003. 61
- [23] A. Cervone, S. Manservigi, R. Scardovelli, and S. Zaleski. A geometrical predictor–corrector advection scheme and its application to the volume fraction function. *Journal of Computational Physics*, 228(2):406–419, 2009. 63
- [24] R. Barrett and C. Romine. *Templates for the solution of linear systems: building blocks for iterative methods*. Society for Industrial Mathematics, 1994. 67
- [25] E. Aulisa, S. Manservigi, and R. Scardovelli. A novel representation of the surface tension force for two-phase flows with reduced spurious currents. *Comput. Method. Appl. M.*, 195:6239–6257, 2006. 72
- [26] S. Gross and A. Reusken. Finite element discretization error analysis of a surface tension force in two-phase incompressible flows. *Siam J. Numer. Anal.*, 45(4):1679–1700, 2007. 72
- [27] D. Fuster, A. Bague, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, S. Scardovelli, and S. Zaleski. Simulation of primary atomization with an octree adaptive mesh refinement and vof method. *Int. J. Multiphas. Flow*, 35(6):550–565, 2009. 73
- [28] H.Q. Yang. Asymmetric instability of a liquid jet. *Phys. Fluids*, 4(4):681–689, 1992. 73
- [29] Lord Rayleigh. *On the instability of jets*. Proc. London Math. Soc., 1878. 74

- [30] C. Weber. Zum zerfall eines flussigkeitsstrahles. *Z. Angew. Math. Mech.*, 11:136–154, 1931. 77
- [31] H. Chaves, F. Obermeier, and T. Seidel. Fundamental investigations of the disintegration of a sinusoidally forced liquid jet. In *Eighth international conference on liquid atomization and spray systems*, pages 1018–1025, Pasadena, 2000. Carnegie Mellon University. 78
- [32] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228(16):5838–5866, 2009. 81
- [33] O.C. Zienkiewicz and R.L. Taylor. *The finite element method for solid and structural mechanics*. Butterworth-Heinemann, 2005. 96