Università degli Studi di Padova
Department of Industrial Engineering

Ph.D. degree in Industrial Engineering

A high-resolution fully compressible Navier-Stokes solver for analysis of moving objects at high Mach numbers

Ph.D. candidate:
Francesco De Vanna

Thesis advisor:
Prof. Ernesto Benini
Thesis co-advisor:
Prof. Francesco Picano

A.A. 2018/19

Two things fill the mind with ever new and increasing admiration and awe, the more often and steadily we reflect upon them: the starry heavens above me and the moral law within me.

IMMANUEL KANT, *Critique of Practical Reason* (1788)

To my family

Contents

Nomenclature

Acronyms

AMR	Adaptive Mesh Refinement
BFGM	Body Fitted Grid Method
CAD	Computer Aided Design
CFL	Courant-Friedrichs-Lewy
CPU	Computer Processing Unit
DDES	Delayed Detached Eddy Simulation
DES	Detached Eddy Simulation
DFM	Direct Forcing Method
DNS	Direct Numerical Simulation
eDFM	explicit Direct Forcing Method
ENO	Essentially Non-Oscillatory
FSI	Fluid Structure Interaction
GPFM	Ghost-Point Forcing Method
IBM	Immersed Boundary Method
IDDES	Improved Delayed Detached Eddy Simulation
iDFM	implicit Direct Forcing Method
KGP6	Kennedy-Grüber-Pirozzoli (sixth-order)
LES	Large Eddy Simulation
LLF	Local Lax Friedrichs
MPI	Message Passing Interface
NS	Navier–Stokes
NSCBC	Navier–Stokes Characteristic Boundary Conditions
ODE	Ordinary Differential Equation
PDE	Partial Differential equation
RANS	Reynolds Averaged Navier–Stokes
RH	Rankine Hugoniot
RK	Runge-Kutta
RK3	Runge-Kutta (third-order)
RTT	Reynolds Transport Theorem
SGS	Subgrid-scale
SRS	Scale Resolving Simulation
TVD	Total Variation Diminishing
WALE	Wall-adaptive Large-Eddy
WENO	Weighted Essentially Non-Oscillatory
WENO5	Weighted Essentially Non-Oscillatory fifth-order

1 Introduction

Nothing great in the world was accomplished without passion.

Georg Wilhelm Friedrich Hegel

Contents

1.1 Introduction

Πάντα ῥεῖ. This famous aphorism sums up the thought of the grecian philosopher Heraclitus of Ephesus (535 – 475 BC) and it is rendered in english with *«Everything flows»*. With this sentence Heraclitus wants to convey the idea that everything is in a continuous change. In the Heraclitus' philosophy, Nature is too complex to repeat itself and the fluid motion is the most representative example of how things continue to move without rest.

> Πάντα χωρεῖ καὶ οὐδὲν μένει, δὶς ἐς τὸν αὐτὸν ποταμὸν οὐκ ἂν ἐμβαίης.
>
> *«Everything changes and nothing rests, you cannot step twice into the same stream.»*

Today Science has done many advances from Heraclitus' times, and nowadays we know that the fluid motion admits a statistical description. However, we are still far from a deep knowledge of fluid phenomena and the fluid dynamics represent a challenging task for future experimental and numerical activities. The latter in particular is increasingly important in all the scientific fields and nowadays the mathematical modelling is required to be extremely accurate in order to deep understand the complexity of the physics behind lots of practical problems. Specifically in the field of compressible viscous flows, the mechanic of the fluids is a challenging task. Such flows, due to their unique hyperbolic dynamics, exhibit a complex and non-linear behaviour, thus ad-hoc numerical methods are required for their discrete treatment. Moreover, especially applying such models to turbulent and fully separated flows, the computational power becomes demanding, forcing to develop massive parallel and highly scalable tools.

1.2 The importance of being *«compressible»*

From low-speed regimes up to high-speed conditions, compressible flows range a wide variety of problems. According to many studies, the aerodynamics of both civil and military aircrafts [3, 2] or the modern engines performances [73, 62, 63, 57, 9] are strongly linked to flow compressible phenomena and dynamics. Today, far from the most popular applications of internal and external aerodynamics, compressible fluid dynamics represents an increasingly exciting issue in the field of medical and environmental applications. Variability of pressure and density influences different sciences. For example, the human voice [131, 112] consists in a challenging topic whose physics has not been completely understood yet; as well as the noise control and the aero-acoustics related problems [41, 14, 83, 29] requires even more accurate, practical and predictive solutions.

Despite its complexity, the numerical description of compressible flows - and in particular the numerical methods for Euler equations of Gas Dynamics - go back to the dawn of the computer era and several works and textbooks have appeared over the years (e.g., Courant et al. (1952), Godunov (1959), LeVeque (1998), Hirsch (2007), Toro (2009)). In the last two decades the formulation of high-order high-resolution shock-capturing numerical schemes in the path of Essentially non-Oscillatory (ENO) and Weighted-Essentially non-Oscillatory (WENO) schemes [53, 59, 113, 1, 5, 16, 4, 126] as far as the development of ad-hoc total-variation-diminishing (TVD) time-integration methods [43] have significally improved both the comprehension of the convective terms of Navier-Stokes equations and the capabilities of the numerical solvers to accurately and stably evolve strong compressible flow conditions. The extension of the previous schemes to a hybrid-compact/hybrid-central formulations [90, 21, 91, 92] combined with shock-detection techniques [30, 55] improved the numerical resolution and reduced the computation efforts in the discretisation process.

1.3 Moving objects

Due to the high-fidelity of the actual numerical models and the continuous increase of the computational power, today even more difficult simulations and complex dynamical systems can be treated with a computational approach. In this path, the fluid-structure interaction (FSI) represents one of the most challenging tasks for future fluid dynamics modelling. Historically speaking, the problem has been initially formulated in the filed of incompressible flows over civil structures (see, e.g., Green and Unruh (2006)) and it has been extended to a wide variety of practical problems (see, e.g., [121, 76, 13]). Dealing with the high-speed flows, the FSI lacks an accurate and stable numerical approach. The topic includes a wide variety of applications and the numerical simulation of compressible viscous flows around moving objects is still a crucial target in many engineering fields. The latter range from the control of aero-elastic structures [50] up to the study of particulate compressible flows [122] and surely today the success of landing on dense atmosphere planets [10, 34] or the efficiency improvement of high-bypass turbofan engines [36] are even more affected by the non-linear dynamics between the structural components and the supersonic flow field.

1.4 Turbulence modelling

Shocks and discontinuities are not the only problems related to compressible flows and the turbulent phenomena play a crucial role in the accurate prediction and simulation of such systems. Turbulence is a chaotic, time-dependent and multi-scale physical phenomenon inherent in the fluid motion and governed by the Navier-Stokes equations. The main feature of turbulent flows consists in a continuous cascade of mechanical energy from the bigger scales to the smaller ones. Coherent structures with different sizes characterise all these scales. Thus, the bigger vortices are comparable to a macro physical length of the problem while the smaller could be even micrometrical. The bigger structures, in particular, represent a tiny portion of the mechanics associated to a turbulent flow and going along the turbulent cascade and travelling the turbulence granularity, the finest vortices are the ones that play the final and fundamental role in the dissipation mechanism, destroying the energy of the system in a small portion of space. In 1941 the essential work of A. N. Kolmogorov $(1903-1987)$ firstly advanced a theoretical description about the role of the dissipative scales, that thanks to his contribution captured his name.

Today three main approaches are available in order to describe the complex dynamics associated to a turbulent flow: the *Direct Numerical Simulation* (DNS), the *Large Eddy Simulation* (LES) and the *Reynolds-Averaged Navier-Stokes* (RANS) approach [98].

The DNS method consists in directly discretising the Navier-Stokes system of equations without introducing any model for turbulence. This approach is recognised as the most accurate and consistent physic method, but computationally speaking it is very demanding. To resolve all the smallest structures associated with the turbulent energy cascade, the numerical discretisation should be the finest. The basic theory of homogeneous turbulent flows shows the impossibility of a DNS method to be used in applicative situations. Since the dimension of the smallest scales is inversely proportional to the Reynolds number and in applicative problems the latter has a magnitude order of $\sim 10^7$, considered η the Kolmogorov's length and L_0 a characteristics size of a problem these two parameters are related through $L_0/\eta \propto Re^{3/4}$, resulting that a fully resolved three-dimensional DNS requires $(Re^{3/4})^3 \simeq 10^{15\div16}$ number of points at each time-step. The goal is still impossible even for the most modern and powerful supercomputers.

Since the beginning of computation, the turbulence community has made many efforts to address high-Reynolds turbulent flows and thanks to them today many alternatives are at our disposal. To avoid the expensive DNS approach and predict the mechanics of a turbulent flow with a sufficient level of accuracy, the most employed and well-known method for industrial and applicative fluid dynamics problems consists in the Reynolds-Averaged Navier-Stokes (RANS) technique. The RANS approach represents an entirely different philosophy with respect to DNS. In this case, the numerical algorithm solves the mean turbulent fields resulting in a drastically reduction of the number of discretisation elements employed in the numerical simulation. However, the method induces a lot of adverse effects. The mean dynamics introduces six more unknowns variables in the mathematical description of the fluid motion; thus, some modelling techniques for these terms are required. As a consequence, the results are strongly affected by the selection of the turbulence model, often leading to wrong predictions. Moreover, the unsteady dynamics associated to the vortical structures, crucial in many applications, is entirely missed by the method.

The LES represents a good compromise between the DNS method and the RANS technique. The latter directly solves the more significant turbulent structures (*eddies*), while the smaller is modelled and filtered in such a way that the turbulent energy cascade is preserved. Today the LES approach seems to be a very promising strategy (see, e.g., [125, 82, 114, 116, 136]) representing a valid and more accurate alternative to RANS modelling. Modern industrial CFD has already moved from the more consolidated RANS approach to Scale Resolving Simulations (SRS). The latter aims at resolving just a portion of the turbulent spectrum, filtering the rest and saving a large amount of computational power and memory storage (see, e.g., Gritskevich et al. (2012)). In this path the Detached Eddy Simulations (DES), the Delayed Detached Eddy Simulations (DDES) and the Improved Delayed Detached Eddy Simulations (IDDES) consist in a hybrid/zonal RANS and LES models, representing real and actual solutions for practical and applicative problems. According to the actual computational power, even a pure LES seems to be feasible in some applicative problems.

1.5 Motivations

Inside this scenario, the present work takes place. With this PhD project we developed, and we fully validated a high-resolution numerical model suitable for fluid mechanics simulations in a wide range of speeds and regimes. Nowadays, such topic - especially starting the development process from scratch - could seem in contrast with the availability of a large variety of highly-optimised and multi-platform fluid dynamics software. Among these, it is enough to mention the most famous commercial solutions like *Ansys Fluent*, *Star-CCM+* and *Comsol Multiphysics* or the most employed open-source implementations like *OpenFoam* and *NEK5000*. Anyway, facing frontier engineering applications or fundamental topics in physics of fluids still represents a challenging task for all of these solutions. In particular, the aerodynamics problems in the transonic regime or the dynamics associated to the interaction between compressible flow and a moving structure require the researcher to be aware not only on the physics related to the problem but also to the numerics and the numerical implementation behind the model he is employing. This task is very far from the objectives of commercial software which aims to be as general as possible and user-friendly. Being easy to be used makes this software to be often opaque to the user and excessively conservative, implementing strongly diffusive numerical methods and highly-relaxing the numerical solution. Especially in the field of frontier applications, these methods often result in wrong predictions of the flow behaviour. Thus an ad-hoc numerical tool is required. For this reason, we developed URANOS. URANOS is a low-dissipative high-order and high-resolution numerical solver especially developed for fluid simulation in strong-compressible viscous conditions and able to deal with moving objects at high-Mach numbers. The solver name, standing for Unsteady Robust All-around Navier-StOkes Solver, is also evocative, highlighting the general purposes of the tool, thus recalling the term with the starry sky is translated into ancient greek: Οὐρανός.

1.6 Solver structure

URANOS is a DNS solver which implements the full set of the Navier-Stokes system of equations in a time-dependent conservative formulation. This choice represents a natural solution for high-fidelity simulations. Being aware of the impossibility of treating high-Reynolds problems, the development process and the validation of the tool has been focussed on moderate-Reynolds DNS applications. In addition, the solver implements the LES approach dealing with three different turbulence models. This peculiarity allows simulating high-Reynolds problems staying within an accurate and high-performance numerical method. Also, the LES extension of the solver has been successfully tested and validated over well-documented benchmarks. Speaking about the numerics behind the solver, the latter relies on a high-order non-uniform Cartesian finite-difference approach and employs the Immersed Boundary Method (IBM) to deal with embedded geometries in the fluid domain. The tool is able to address a large variety of both fundamental and aerodynamics applications, in term of shock/turbulence interaction and flows over complex (even moving) geometries in strong compressible conditions. Especially in the field of bluff-bodies aerodynamics the solver has shown an excellent level of accuracy in a wide range of Reynolds and Mach numbers.

1.7 Thesis outline

In the present dissertation, the numerical methodology will be described in high level details, and in particular, the thesis is organised in the following 7 chapters:

- §2 «Mathematical model» provides a theoretical description of the model describing the dynamics of a time-dependent compressible flow and derives some suitable formulations for numerical applications;

- §3 «Numerical integration of differential equations» gives a brief overview of the numerical methods for the time-integration of partial differential equations, focusing to the one implemented and to its validation;

- §4 «Numerical treatment of convective fluxes» aims to explain the problems related to shocks and discontinuities and it provides some ad-hoc numerical strategies to face them. The chapter ends with the validation of our methodology, resulting in a considerable number of tests and benchmarks;

- §5 «Numerical treatment of viscous fluxes» provides a detailed description of the diffusive terms of the Navier-Stokes system of equation, presenting a suitable formulation for them and with the related tests suite;

- §6 «Code parallelisation and 3D results» gives an overview of parallelisation strategies for three-dimensional domains resulting in a demanding technique for real applicative problems. The chapter ends presenting central findings in the field of three-dimensional turbulent and fully separated flows;

- §7 «The immersed boundary methods» aims to describe the Immersed Boundary methodology. The chapter discerns between the Direct-Forcing method and the Ghost-Point-Forcing method; the latter is explained with a significant level of details representing our choice;

- §8 «Results with the immersed boundary method» concludes the dissertation presenting a large variety of benchmarks and test cases to validate the entire tool combining the high-order solver with the IBM.

2
Mathematical model

All models are wrong, but some models are useful.

George Edward Pelham Box

Contents

In the present chapter, we will derive the Navier-Stokes system of equations in conservative formulation for a Newtonian ideal gas. A suitable formulation for numerical applications will be pointed out, and an ad-hoc non-dimensional description of the model will be derived. The model can describe a large variety of fluid systems, but remaining inside the *Continuum Mechanic Hypothesis*. Fro this formalism is mandatory to have the Knudsen number Kn, i.e., the ratio between *molecular mean free path* λ and a characteristic length of system L_0, to be much smaller of a unit.

2.1 The conservation principles for a continuum system

Being Σ a *closed system* described by a time-dependent material volume $\mathcal{V}_t$ and being $\mathcal{V}_t$ moving at the same speed of the local velocity field. If $\partial\mathcal{V}_t$ represents the $\mathcal{V}_t$ frontier, the three fundamental laws that describe the system are: The mass conservation equation

$$\frac{d}{dt}\int_{\mathcal{V}_t} \rho \, d\mathcal{V} = 0 \tag{2.1}$$

Newton's second law

$$\frac{d}{dt}\int_{\mathcal{V}_t} \rho u_i \, d\mathcal{V} = F_i \quad i = 1, \ldots, 3 \tag{2.2}$$

and the first law of the thermodynamics

$$\frac{d}{dt}\int_{\mathcal{V}_t} \rho E \, d\mathcal{V} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \tag{2.3}$$

Here ρ is the density of $\mathcal{V}_t$; u_i is its speed in i-th direction; F_i is the resulting force along the the i; E is the total energy per unit mass while $\delta Q/\partial t$ and $\delta W/\partial t$ are respectively the heat flow and the mechanical work temporal variations exchanged through ∂V_t. Being ϕ a general transported variable, the Reynolds Transport Theorem (RTT) applied to ϕ could be indicated as follows

$$\frac{d}{dt}\int_{\mathcal{V}_t} \phi \, d\mathcal{V} = \int_{\mathcal{V}} \frac{\partial \phi}{\partial t} \, d\mathcal{V} + \int_{\partial V} \phi u_j n_j \, dS \qquad ^{1} \tag{2.4}$$

or equivalently, exploiting the divergence theorem

$$\frac{d}{dt}\int_{\mathcal{V}_t} \phi \, d\mathcal{V} = \int_{\mathcal{V}} \frac{\partial \phi}{\partial t} + \frac{\partial \phi u_j}{\partial x_j} \, d\mathcal{V} \tag{2.5}$$

Hence, applying the expression (2.5) to equations (2.1), (2.2) and (2.3) we can write

$$\int_{\mathcal{V}} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} \, d\mathcal{V} = 0 \tag{2.6a}$$

$$\int_{\mathcal{V}} \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} \, d\mathcal{V} = F_i \tag{2.6b}$$

$$\int_{\mathcal{V}} \frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_j}{\partial x_j} \, d\mathcal{V} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \tag{2.6c}$$

[1] In the present dissertation the Einstein's convention for the summation on repeated indices will we employed apart where different stated.

The latter set of equations represents a simple mathematical manipulation describing the conservation principles in terms of mass, momentum and total energy for a general thermo-mechanical system under the hypothesis of the Continuum Mechanics.

2.1.1 Mass conservation

From equation (2.6a) the arbitrariness of the integration volume leads us to conclude that

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0 \tag{2.7}$$

The equation represents a differential formulation for the mass conservation of the system Σ.

2.1.2 Momentum conservation

Working on the right-hand-side of equation (2.6b) we need to describe the external forces F_i using a suitable model. Here two contribution can be pointed out so

$$F_i = \int_{\mathcal{V}} \rho f_i \, d\mathcal{V} + \int_{\partial\mathcal{V}} \sigma_{ij} n_j \, dS \tag{2.8}$$

where f_i is the resulting volume force (i.e., gravitational forces, electro-magnetical effects...) in i-th direction per unit mass and σ_{ij} is the stress tensor due to internal shear effects. So, inserting equation (2.8) in (2.6b) we can write

$$\int_{\mathcal{V}} \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} \, d\mathcal{V} = \int_{\mathcal{V}} \rho f_i \, d\mathcal{V} + \int_{\partial\mathcal{V}} \sigma_{ij} n_j \, dS \tag{2.9}$$

Applying the divergence theorem to the right-hand-side

$$\int_{\mathcal{V}} \left[\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} - \rho f_i - \frac{\partial \sigma_{ij}}{\partial x_j} \right] d\mathcal{V} = 0 \tag{2.10}$$

so for the arbitrariness of the integration volume, we conclude as follows

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i \tag{2.11}$$

The latter equation represents a differential formulation for the momentum conservation of the system Σ.

2.1.3 Energy conservation

Working on the right-hand-side of equation (2.6c) the temporal variation of the heat flow can be expressed as

$$\frac{\delta Q}{\delta t} = - \int_{\partial\mathcal{V}} q_i n_i \, dS \tag{2.12}$$

where q_i, the heat flux in i-th direction. q_i can be modelled using Fourier's law

$$q_i = -\lambda \frac{\partial T}{\partial x_i} \tag{2.13}$$

where $\lambda = \lambda(T)$ is the *heat transport coefficient* of the system and $\partial T/\partial x_i$ is the temperature gradient. The temporal variation of the mechanical work can be expressed by the scalar product between the acting forces and the local speed

$$\frac{\delta W}{\delta t} = -\int_{\partial\mathcal{V}} \sigma_{ij} n_j u_i d\mathcal{V} - \int_{\partial\mathcal{V}} \rho f_i u_i d\mathcal{V} \tag{2.14}$$

Thus, introducing (2.12) and (2.14) in (2.6c), and taking into account (2.13) we can write

$$\int_{\mathcal{V}} \frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_j}{\partial x_j} d\mathcal{V} = -\int_{\partial\mathcal{V}} q_j n_j \, dS + \int_{\partial\mathcal{V}} \sigma_{ij} n_j u_i d\mathcal{V} + \int_{\partial\mathcal{V}} \rho f_i u_i d\mathcal{V} \tag{2.15}$$

equation that, exploiting the divergence theorem and the arbitrariness of the integration volume, results

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j}\right) + \frac{\partial \sigma_{ij} u_i}{\partial x_j} + \rho f_i u_i \tag{2.16}$$

The last relation represents a differential formulation for the total energy conservation of the system Σ.

2.2 The Navier-Stokes system of equations

2.2.1 Newtonian fluids

The conservation principles in differential formulation expressed by equations (2.7) (2.11) (2.16) represent the mathematical model for describing the conservation properties for a general continuum medium. In the present paragraph, we want to characterise the model for a fluid with *Newtonian rheology*. In particular, being σ_{ij} the stress tensor of a Newtonian fluid, this can be decomposed in two terms: a pressure contribution, existing both in static and in dynamic conditions, and a dynamical contribution (d_{ij}), due to the relative motion between elementary fluid volumes, thus

$$\sigma_{ij} = -p\delta_{ij} + d_{ij} \tag{2.17}$$

here p denotes the thermodynamical pressure while δ_{ij} is the Kronecker's delta. A fluid is said to be *Newtonian* if the dynamical contribution of the stress tensor is a linear combination of the velocity gradient

$$d_{ij} = A_{ijkl} \frac{\partial u_k}{\partial x_l} \tag{2.18}$$

here A_{ijkl} is a fourth-order tensor collecting the combination coefficients. For the symmetrical properties of stress tensor, it is possible to prove that d_{ij} can be expressed by

$$d_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \mu_v \frac{\partial u_s}{\partial x_s} \delta_{ij} \tag{2.19}$$

where μ and μ_v are respectively the dynamical and the volumetric viscosities. Following the *Stokes' hypothesis*

$$\mu_v = -\frac{2}{3}\mu \tag{2.20}$$

so the dynamical contribution of the stress tensor for a Newtonian fluid can be expressed as

$$d_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_s}{\partial x_s}\delta_{ij} \right) \tag{2.21}$$

2.2.2 Model description

In the previous paragraphs, we derived the conservation principles in differential equations for a continuum medium and we introduced the model of a fluid with *Newtonian* rheology. Combining the expression of the stress tensor (2.17) into (2.11) and (2.16) and taking into account (2.21) we obtain that

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u_j}{\partial x_j} \tag{2.22a}$$

$$\frac{\partial \rho u_i}{\partial t} = -\frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij}) + \frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_s}{\partial x_s} \right) \right) + \rho f_i \tag{2.22b}$$

$$\frac{\partial \rho E}{\partial t} = -\frac{\partial}{\partial x_j}((\rho E + p)u_j) + \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j} \right) + \frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_s}{\partial x_s} \right) u_i \right) + \rho f_i u_i \tag{2.22c}$$

The previous set of partial differential equations (PDE) represent the Navier-Stokes (NS) system of equations in conservative formulation for a fluid with Newtonian rheology. The system can be cast in the vectorial form

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}_j(\mathbf{U})}{\partial x_j} + \frac{\partial \mathbf{F}_{vj}(\mathbf{U})}{\partial x_j} + \mathbf{S}(\mathbf{U}), \quad j = 1, \dots, 3 \tag{2.23}$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}, \mathbf{F}_j = \begin{pmatrix} \rho u_i \\ \rho u_i u_j + p\delta_{ij} \\ (\rho E + p)u_i \end{pmatrix}, \mathbf{F}_{vj} = \begin{pmatrix} 0 \\ d_{ij} \\ d_{ij}u_j + \lambda\frac{\partial T}{\partial x_i} \end{pmatrix}, \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ \rho f_i \\ \rho f_i u_i \end{pmatrix} \tag{2.24}$$

respectively, the vector of conservative variables, the vector of convective and viscous fluxes in the jth direction and the vector of source terms. The latter formulation is the most suitable in case of numerical applications. The model (2.23) describes the dynamic of compressible flow with Newtonian rheology and it consists in five equations with seven unknown variables $(\rho, \rho u_i, \rho E, p, T)$. The system must be completed using an equation of state for the fluid $\rho(p,T) = 0$ and a constitutive equation for the total energy. Our model considers the fluid to obey the ideal gas equation

$$\rho = \frac{p}{RT} \tag{2.25}$$

where R is the specific gas constant[2] while the total energy can expressed as

$$E = e + \frac{1}{2}(u_i u_i) \tag{2.27}$$

[2] We define the specific gas constant as

Here $1/2(u_i u_i)$ is the specific kinetic energy while e is the internal energy per unit mass. The latter for an ideal gas follows

$$e = c_v T = \frac{pR}{(\gamma - 1)} \tag{2.28}$$

Here c_v is the heat coefficient at constant volume while c_p is the heat coefficient at constant pressure of the gas and $\gamma = c_p/c_v$ is their ratio. Finally, the viscosity μ is assumed to obey to the Sutherland's law

$$\mu(T) = T^{3/2}\left(\frac{T_0 + S}{T + S}\right) \tag{2.29}$$

where $T_0 = 273.15K$ and $S = 110.4K$ are empirical parameters and λ, the thermal conductivity, is related to μ via the Prandtl number definition

$$\lambda(T) = \frac{\mu(T)c_p}{Pr} \tag{2.30}$$

where Pr is approximatively equal to 0.71.

2.3 Non-dimensional formulation

Dimensional Navier-Stokes equations, derived in the previous paragraphs, are not suitable for numerical modelling for two main reasons:

1. Using non-dimensional equations it is possible to prescribe the minimum number of independent parameters that control the system. Hence the results will correspond to a class of different physical experiments characterised by the same non-dimensional groups, e.g. Mach and Reynolds numbers;

2. computer processing units (CPUs) work inefficiently with numbers belonging the highly different orders of magnitude; thus non-dimensional formulation of numerical models, normalising the relevant quantities to be order one, tends to improve the computational speed and the accuracy.

2.3.1 The Π theorem and dimensional analysis

From the Π theorem and the dimensional analysis, we know that if a physical system is governed by N_p dimensional parameters, N_d fundamental dimensions and N_r relations, the number of independent fundamental groups that describe the system is

$$N_g = N_p - (N_d + N_r) \tag{2.31}$$

Let us perform the analysis for the Navier-Stokes system of equations. Here $N_p = 15$ is the number of the reference parameters describing the problem: t_0, L_0, ρ_0, u_0, p_0, T_0, E_0, μ_0,

$$R = \frac{\mathcal{R}}{\mathcal{M}} \tag{2.26}$$

where $\mathcal{R} = 8.314\, J \cdot mol^{-1} \cdot K^{-1}$ is the universal gas constant and $\mathcal{M}\,[kg \cdot mol^{-1}]$ is the molar mass of the gas.

λ_0, R_0, c_{p0}, c_{v0}, e_0, Pr_0, γ_0. Those parameter can be expressed by a combination of $N_d = 4$ fundamental dimensions: time $[s]$, length $[m]$, mass $[kg]$ and temperature $[K]$ and they are related by $N_r = 7$ equations:

$$p_0 = \rho_0 R_0 T_0, \quad \mu_0 = T_0^{3/2}\left(\frac{T_0+S}{T_0+S}\right) \tag{2.32a}$$

$$\lambda_0 = \frac{\mu_0 c_{p0}}{Pr_0}, \qquad e_0 = c_{v0} T_0 \tag{2.32b}$$

$$E_0 = e_0 + \frac{1}{2}u_0^2, \quad c_{v0} = \frac{R_0}{\gamma_0 - 1}, \quad c_{p0} = \frac{\gamma R_0}{\gamma_0 - 1} \tag{2.32c}$$

so the number of fundamental groups that describe the system is $N_g = 4$. The choice of the reference variables is arbitrary and in this work we used L_0 as the reference length, ρ_0 as the reference density, p_0 as the reference pressure and $u_0 = \sqrt{p_0/\rho_0}$ as the reference speed (consisting in the speed of sound apart $\sqrt{\gamma_0}$), thus we can introduce the following change of variables:

$$x = L_0 x', \quad u_i = u_0 u_i', \quad t = \frac{L_0}{u_0}t', \quad \rho = \rho' \rho_0, \quad T = T_0 T' \tag{2.33a}$$

$$e = \frac{p_0}{\rho_0}e', \quad E_{tot} = \frac{p_0}{\rho_0}E'_{tot}, \quad \mu = \mu_0 \mu', \quad \lambda = \lambda_0 \lambda' \tag{2.33b}$$

where $(\cdot)'$ stands for a *non-dimensional quantity.* Now we can recast the Navier-Stokes equations using the above definitions. Starting from the mass conservation equation, we can write

$$\frac{\partial \rho'}{\partial t'}\frac{\rho_0 u_0}{L_0} = -\frac{\partial \rho' u_j'}{\partial x_j'}\frac{\rho_0 u_0}{L_0} \tag{2.34}$$

obtaining that the non-dimensional equation for mass conservation fits with as the dimensional one:

$$\frac{\partial \rho'}{\partial t'} = -\frac{\partial \rho' u_j'}{\partial x_j'} \tag{2.35}$$

For the momentum conservation equation we can write

$$\frac{\partial \rho u_i}{\partial t} = \frac{\partial \rho' u_i'}{\partial t'}\frac{\rho_0 u_0^2}{L_0} \tag{2.36}$$

$$\frac{\partial}{\partial x_j}\left(\rho u_i u_j\right) = \frac{\partial}{\partial x_j'}\left(\rho' u_i' u_j'\right)\frac{\rho_0 u_0^2}{L_0} \tag{2.37}$$

$$\frac{\partial}{\partial x_j}\left(p_i \delta_{ij}\right) = \frac{\partial}{\partial x_j'}\left(p_i' \delta_{ij}\right)\frac{p_0}{L_0} \tag{2.38}$$

$$\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_s}{\partial x_s}\right)\right) = \frac{\partial}{\partial x_j'}\left(\mu'\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'} - \frac{2}{3}\frac{\partial u_s'}{\partial x_s'}\right)\right)\frac{\mu_0 u_0}{L_0^2} \tag{2.39}$$

thus

$$\frac{\rho_0 u_0^2}{L_0}(\cdot)' = -\frac{\rho_0 u_0^2}{L_0}(\cdot)' - \frac{p_0}{L_0}(\cdot)' + \frac{\mu_0 u_0}{L_0^2}(\cdot)' \tag{2.40}$$

Now, dividing by $\rho_0 u_0^2/L_0$

$$(\cdot)' = -(\cdot)' - \frac{p_0}{\rho_0 u_0^2}(\cdot)' + \frac{\mu_0}{\rho_0 L_0 u_0}(\cdot)' \tag{2.41}$$

and keeping in mind $u_0 = \sqrt{p_0/\rho_0}$ we obtain

$$(\cdot)' = -(\cdot)' - (\cdot)' + \frac{\mu_0}{\rho_0 L_0 u_0}(\cdot)'. \tag{2.42}$$

Defining the heat capacity ratio

$$\gamma = \frac{c_{p0}}{c_{v0}} \tag{2.43}$$

the free-stream Mach number

$$M_\infty = \frac{u_\infty}{\sqrt{\gamma p_0/\rho_0}} \tag{2.44}$$

and the Reynolds number

$$Re = \frac{\rho_0 u_\infty L_0}{\mu_\infty} \tag{2.45}$$

the symbolic equation (2.42) allows us to write the non-dimensional formulation of the momentum conservation equation as

$$\frac{\partial \rho' u_i'}{\partial t'} = -\frac{\partial}{\partial x_j'}(\rho' u_i' u_j' + p_i' \delta_{ij}) + \frac{\sqrt{\gamma} M_\infty}{Re}\frac{\partial}{\partial x_j'}\left(\mu'\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'} - \frac{2}{3}\frac{\partial u_s'}{\partial x_s'}\right)\right) \tag{2.46}$$

For the total energy conservation equation follows that

$$\frac{\partial \rho E}{\partial t} = \frac{\partial \rho' E'}{\partial t'}\frac{\rho_0 u_0}{L_0}\frac{p_0}{\rho_0} \tag{2.47}$$

$$\frac{\partial}{\partial x_j}((\rho E + p)u_j) = \frac{\partial}{\partial x_j'}((\rho' E' + p')u_j')\frac{\rho_0 u_0}{L_0}\frac{p_0}{\rho_0} \tag{2.48}$$

$$\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j}\right) = \frac{\partial}{\partial x_j'}\left(\lambda' \frac{\partial T'}{\partial x_j'}\right)\frac{\lambda_0 T_0}{L_0^2} \tag{2.49}$$

$$\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_s}{\partial x_s}\right)u_i\right) = \frac{\partial}{\partial x_j'}\left(\mu'\left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'} - \frac{2}{3}\frac{\partial u_s'}{\partial x_s'}\right)u_i'\right)\frac{\mu_0 u_0^2}{L_0^2} \tag{2.50}$$

this, in a symbolic way, becomes

$$\frac{\rho_0 u_0}{L_0}\frac{p_0}{\rho_0}(\cdot)' = -\frac{\rho_0 u_0}{L_0}\frac{p_0}{\rho_0}(\cdot)' + \frac{\lambda_0 T_0}{L_0^2}(\cdot)' + \frac{\mu_0 u_0^2}{L_0^2}(\cdot)'. \tag{2.51}$$

Dividing by $\frac{\rho_0 u_0}{L_0}\frac{p_0}{\rho_0}$ we obtain

$$(\cdot)' = -(\cdot)' + \frac{\lambda_0 T_0}{L_0^2}\frac{L_0 \rho_0}{\rho_0 u_0 p_0}(\cdot)' + \frac{\mu_0 u_0^2}{L_0^2}\frac{L_0 \rho_0}{\rho_0 u_0 p_0}(\cdot)' \tag{2.52}$$

so recalling the definition of Prandtl number

$$Pr = \frac{\gamma R_0}{\gamma - 1}\frac{\mu_\infty}{\lambda_\infty} \tag{2.53}$$

and the definition of M_∞ and Re we can write

$$(\cdot)' = -(\cdot)' + \frac{\gamma R_0}{\gamma - 1}\frac{1}{Pr}\frac{\sqrt{\gamma}M_\infty}{Re}(\cdot)' + \frac{\sqrt{\gamma}M_\infty}{Re}(\cdot)' \tag{2.54}$$

a symbolic equation that allows as to write the non-dimensional formulation of the energy conservation equation as

$$\begin{aligned}\frac{\partial \rho' E'}{\partial t'} = &-\frac{\partial}{\partial x'_j}((\rho' E' + p')u'_j)\\ &+ \frac{\sqrt{\gamma}M_\infty}{Re}\left(\frac{\gamma R_0}{\gamma - 1}\frac{1}{Pr}\frac{\partial}{\partial x'_j}\left(k'\frac{\partial T'}{\partial x'_j}\right) + \frac{\partial}{\partial x'_j}\left(\mu'\left(\frac{\partial u'_i}{\partial x'_j} + \frac{\partial u'_j}{\partial x'_i} - \frac{2}{3}\frac{\partial u'_s}{\partial x'_s}\right)u'_i\right)\right)\end{aligned} \tag{2.55}$$

Equations (2.35), (2.46) and (2.55) represent a non-dimensional formulation of the Navier-Stokes system of equations suitable for numerical applications. Until the non-dimensional groups: γ, M_∞, Re and Pr are imposed, the reference variables may assume any value. The most convenient choice is to assume $L_0 = 1$, $\rho_0 = 1$, $p_0 = 1$ and $T_0 = 1$ (so $R_0 = p_0/(\rho_0 T_0) = 1$) and computing all the other parameters as a consequence, so γ is given

$$\gamma = \frac{c_{p0}}{c_{v0}} \tag{2.56}$$

and no other parameters appear in its definition. The Mach number can be used to compute the free-stream speed u_∞ so

$$u_\infty = \sqrt{\gamma}M_\infty \tag{2.57}$$

Once u_∞ is known, the Reynolds number can be employed in the computation of the free-stream dynamic viscosity

$$\mu_\infty = \frac{\sqrt{\gamma}M_\infty}{Re} \tag{2.58}$$

and finally, the Prandtl number is used in the computation of the free-stream thermal conductivity

$$\lambda_\infty = \frac{\gamma}{\gamma - 1}\frac{1}{Pr}\frac{\sqrt{\gamma}M_\infty}{Re} \tag{2.59}$$

Finally, we need to provide a non-dimensional form to the gas equation of state, the constitutive energy equation, and Sutherland's law. Thus being

$$p' p_0 = \rho' \rho_0 R' R_0 T' T_0 \tag{2.60}$$

we obtain

$$p' = \rho' T' \tag{2.61}$$

Dealing with the total energy constitutive equation

$$E'\frac{p_0}{\rho_0} = c_{v0} T' T_0 + \frac{1}{2}(u'_i u'_i)^2 u_0^2 \tag{2.62}$$

and remembering $c_{v0} = \frac{1}{\gamma_0 - 1}$ and the definition of u_0 we can write

$$E' = \frac{1}{(\gamma - 1)} \frac{p'}{\rho'} + \frac{1}{2}(u'_i u'_i)^2 \tag{2.63}$$

Finally, the non-dimensional Sutherland's law can be derived as

$$\mu'(T') = T'^{3/2} \left(\frac{1 + S/T_0}{T' + S/T_0} \right) \tag{2.64}$$

2.3.2 The Non-dimensional Navier-Stokes equations for an ideal gas

Recasting the equations

$$\frac{\partial \rho'}{\partial t'} = -\frac{\partial \rho' u'_j}{\partial x'_j} \tag{2.65a}$$

$$\frac{\partial \rho' u'_i}{\partial t'} = -\frac{\partial}{\partial x'_j}(\rho' u'_i u'_j + p'_i \delta_{ij}) + \frac{\sqrt{\gamma} M_\infty}{Re} \frac{\partial}{\partial x'_j} \left(\mu' d'_{ij} \right) \tag{2.65b}$$

$$\frac{\partial \rho' E'}{\partial t'} = -\frac{\partial}{\partial x'_j}((\rho' E' + p')u'_j) + \frac{\sqrt{\gamma} M_\infty}{Re} \left(\frac{\gamma}{\gamma - 1} \frac{1}{Pr} \frac{\partial}{\partial x'_j} \left(k' \frac{\partial T'}{\partial x'_j} \right) + \frac{\partial}{\partial x'_j} \left(\mu' d'_{ij} u'_i \right) \right) \tag{2.65c}$$

in the vectorial formulation and dropping the $(\cdot)'$ superscript we can write

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}_j(\mathbf{U})}{\partial x_j} + \frac{\partial \mathbf{F}_{vj}(\mathbf{U})}{\partial x_j}, \quad j = 1, \ldots, 3 \tag{2.66}$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \\ \rho u_i \\ \\ \rho E \end{pmatrix}, \mathbf{F}_j = \begin{pmatrix} \rho u_i \\ \\ \rho u_i u_j + p\delta_{ij} \\ \\ (\rho E + p) u_i \end{pmatrix}, \mathbf{F}_{vj} = \frac{\sqrt{\gamma} M_\infty}{Re} \begin{pmatrix} 0 \\ \\ d_{ij} \\ \\ d_{ij} u_j + \frac{1}{Pr} \frac{\gamma}{\gamma - 1} \lambda \frac{\partial T}{\partial x_i} \end{pmatrix}, \tag{2.67}$$

represent the non-dimensional vectors of the conservative variables, advection and viscous fluxes, respectively. The non-dimensional equation of state ends the model

$$p = \rho T \tag{2.68}$$

the non-dimensional expression for the total energy

$$E = \frac{1}{(\gamma - 1)} \frac{p}{\rho} + \frac{1}{2} u^2 \tag{2.69}$$

and the non-dimensional Sutherland's law

$$\mu(T) = T^{3/2} \left(\frac{1 + S/T_0}{T + S/T_0} \right) \tag{2.70}$$

To conclude the paragraph, we want to underline some points:

1. The system (2.66) can be seen as a summation of (at least) three contributions: a temporal contribution due to the temporal variation of conservative variables; a convective contribution, physically related to the transport phenomena associated to the velocity and the pressure fields; and a diffusive contribution, due to the thermodynamical behaviour of the gas and the mutual forces exchanged by elementary fluid volumes

$$\underbrace{\frac{\partial \mathbf{U}}{\partial t}}_{\text{temporal contribution}} = \underbrace{-\frac{\partial \mathbf{F}_j(\mathbf{U})}{\partial x_j}}_{\text{convective contribution}} + \underbrace{\frac{\partial \mathbf{F}_{vj}(\mathbf{U})}{\partial x_j}}_{\text{diffusive contribution}} \tag{2.71}$$

   The three terms of the Navier-Stokes equations need their numerical treatment. The convective term, among them, due to the compressibility of the flow, is the one that requires the higher coding effort.

2. The role of viscous effects is highlighted in our non-dimensional formulation. From (2.66) we can see that dropping the viscous terms, the Navier-Stokes equations reduce to the Euler equations of Gas Dynamics. In particular, our formulation keeps the same mathematical structure for the non-dimensional and the dimensional Euler equations. The fact is highly recommended to improve the maintainability of numerical applications based on this model.

3. From equations (2.66) it seems that the Reynolds number and the free-stream Mach number always appear inside the same ratio, making pointless the definition of one of them. However, because the M_∞ must be separately enforced at the boundary condition on free stream velocity, the two groups maintain a separate role.

2.4 Compressible Large-Eddy Simulation formulation

As we mentioned in the introduction, dealing with the full set of the Navier-Stokes system of equations and employing a DNS approach is still too computational demanding in the case of applicative high-Reynolds fluid dynamics problems; thus some model for turbulence must be employed. In this work, the LES approach has been implemented and in the present section, a brief theoretical description of the filtered compressible Navier-Stokes equations will be derived and discussed. For the details of the approach, the interested reader is addressed to look at Garnier et al. (2009).

2.4.1 Filtered Navier-Stokes equations

Being $\overline{(\cdot)}$ a spatial filter and $\bar{\phi}$ a filtered flow variable, we define $\tilde{\phi}$ the Favre filter of ϕ as

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \tag{2.72}$$

Applying $\overline{(\cdot)}$ to the set of the compressible non-dimensional Navier-Stokes equations and taking into account the definition (2.72) we obtain

$$\frac{\partial\bar{\rho}}{\partial t} = -\frac{\partial\bar{\rho}\tilde{u}_j}{\partial x_j} \tag{2.73a}$$

$$\frac{\partial\bar{\rho}\tilde{u}_i}{\partial t} = -\frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_i\tilde{u}_j + \bar{p}_i\delta_{ij}) + \frac{\sqrt{\gamma}M_\infty}{Re}\frac{\partial}{\partial x_j}\left(\bar{\mu}\bar{d}_{ij}\right) - \frac{\partial}{\partial x_j}\left(\overline{\rho u_i u_j} - \bar{\rho}\tilde{u}_i\tilde{u}_j\right) \tag{2.73b}$$

$$\begin{aligned}\frac{\partial\bar{\rho}\tilde{E}}{\partial t} &= -\frac{\partial}{\partial x_j}((\bar{\rho}\tilde{E} + \tilde{p})\tilde{u}_j) \\ &+ \frac{\sqrt{\gamma}M_\infty}{Re}\left(\frac{\gamma}{\gamma-1}\frac{1}{Pr}\frac{\partial}{\partial x_j}\left(k\frac{\partial\tilde{T}}{\partial x_j}\right) + \frac{\partial}{\partial x_j}\left(\bar{\mu}\bar{d}_{ij}\tilde{u}_i\right)\right) - \frac{\partial}{\partial x_j}\left(\overline{(\rho E + p)u_i} - (\bar{\rho}\tilde{E} + \bar{p})\tilde{u}_i\right)\end{aligned} \tag{2.73c}$$

Comparing the present system (2.73) to the original non-filtered formulation (2.66) we immediately recognise some additional contributions. In particular, the momentum conservation equation is modified taking into account the divergence of the subgrid-scale (SGS) stress tensor, expressed by

$$\bar{T}_{ij}^{SGS} = \overline{\rho u_i u_j} - \bar{\rho}\tilde{u}_i\tilde{u}_j \tag{2.74}$$

while the energy conservation equation accounts for the divergence of subgrid-scale energy term defined as

$$\bar{E}_j^{SGS} = \overline{(\rho E + p)u_i} - (\bar{\rho}\tilde{E} + \bar{p})\tilde{u}_i \tag{2.75a}$$

$$= \overline{\rho c_p T u_j} - \bar{\rho}\bar{c}_p\tilde{T}\tilde{u}_j + \frac{1}{2}\bar{T}_{ij}^{SGS}\tilde{u}_j - \frac{1}{2}\bar{T}_{kk}^{SGS}\tilde{u}_j \tag{2.75b}$$

The equations (2.74) and (2.75) represent the under resolved turbulent contributions due to the filtering process and the LES approach aims to define a suitable model for them. Following the Boussinesq's hypothesis, the deviatoric part of the subgrid-scale stress tensor $\bar{T}_{ij}^{SGS}$ can be evaluated as

$$\bar{T}_{ij}^{SGS} - \frac{1}{3}\bar{T}_{kk}^{SGS}\delta_{ij} = -2\mu_{SGS}\left(\tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right) \tag{2.76}$$

where μ_{SGS} is the subgrid *turbulent eddy viscosity*, T_{kk}^{SGS} is the isotropic contribution to the subgrid stress tensor and $\tilde{S}_{ij}$ denotes the resolved strain-rate tensor

$$\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{2.77}$$

Similarly, the energy subgrid-scale terms are modelled through

$$\bar{E}_j^{SGS} = \overline{\rho c_p T u_j} - \bar{\rho}\bar{c}_p\tilde{T}\tilde{u}_j + \frac{1}{2}\bar{T}_{ij}^{SGS}\tilde{u}_j - \frac{1}{2}\bar{T}_{kk}^{SGS}\tilde{u}_j \tag{2.78a}$$

$$= -\lambda_{SGS}\frac{\partial \tilde{T}}{\partial x_j} + \frac{1}{2}\bar{T}_{ij}^{SGS}\tilde{u}_j - \frac{1}{2}\bar{T}_{kk}^{SGS}\tilde{u}_j \tag{2.78b}$$

where λ_{SGS} is the subgrid-scale turbulent diffusivity expressed by

$$\lambda_{SGS} = \mu_{SGS}\frac{\bar{c}_p}{Pr_T} \tag{2.79}$$

Here Pr_T is the turbulent Prandtl number in our computation kept constant and fixed equal to 0.9. Plugging the Boudssineq's framework into the filtered Navier-Stokes equations (2.73) and dropping the isotropic contribution of the subgrid-scale stress tensor (see, e.g. [118, 119]) results in the following non-dimensional set of conservative equations

$$\frac{\partial \bar{\rho}}{\partial t} = -\frac{\partial \bar{\rho}\tilde{u}_j}{\partial x_j} \tag{2.80a}$$

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} = -\frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_i\tilde{u}_j + \bar{p}_i\delta_{ij}) + \frac{\sqrt{\gamma}M_\infty}{Re}\frac{\partial}{\partial x_j}\left(\bar{\mu}_{tot}\bar{d}_{ij}\right) \tag{2.80b}$$

$$\frac{\partial \bar{\rho}\tilde{E}}{\partial t} = -\frac{\partial}{\partial x_j}((\bar{\rho}\tilde{E} + \bar{p})\tilde{u}_j) + \frac{\sqrt{\gamma}M_\infty}{Re}\left(\bar{\lambda}_{tot}\frac{\partial}{\partial x_j}\left(k\frac{\partial \tilde{T}}{\partial x_j}\right) + \frac{\partial}{\partial x_j}\left(\bar{\mu}_{tot}\bar{d}_{ij}\tilde{u}_i\right)\right) \tag{2.80c}$$

where

$$\bar{\mu}_{tot} = \mu(\tilde{T}) + \frac{Re}{\sqrt{\gamma}M}\mu_{SGS} \tag{2.81}$$

end

$$\bar{\lambda}_{tot} = \frac{\gamma}{\gamma - 1}\left(\frac{\mu(\tilde{T})}{Pr} + \frac{Re}{\sqrt{\gamma}M}\frac{\mu_{SGS}}{Pr_T}\right) \tag{2.82}$$

denotes respectively the total dynamical viscosity (sum of the non-dimensional resolved molecular viscosity $\mu(\tilde{T})$ and the non-dimensional subgrid-scale viscosity) and the total diffusivity of the flow. From the set of equations (2.73) we conclude that the filtering process, in combination with the Bossinesq's hypothesis, introduces a further unknown. The latter consists in the subgrid-scale eddy viscosity μ_{SGS}. Thus a suitable model for μ_{SGS} is required. Differently from the RANS approach, because the LES framework aims to resolve the bigger turbulent structures, a model accounting for μ_{SGS} should be universal and exploitable in a wide range of flow situations. In this path, many efforts have been made by the turbulence community trying to couple the eddy viscosity with the resolved flow quantities. In particular, the most employed model for turbulent eddy viscosities are purely algebraic and reduce to the expression

$$\mu_{SGS} = \bar{\rho}(C_m\Delta)^2\mathcal{D}_m[\bar{\mathbf{U}}] \tag{2.83}$$

Here C_m is a tuning parameter of the model, $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$ is the subgrid characteristic length expressing the size of the filter and $\mathcal{D}_m[\cdot]$ is the non-linear differential operator of the model applied to the resolved flow quantities $\bar{\mathbf{U}}$. In the present work, among the various formulation for the expression (2.83) three models have been taken into account: the *classical Smagorinsky* model, the *Wall-Adaptive Large-Eddy* (WALE) viscosity model and the *Sigma* model.

2.4.2 The classical Smagorinsky model

The classical Smagorinsky model represents the simplest model for μ_{SGS}. The model defines the differential operator $\mathcal{D}_m[\cdot]$ as

$$\mathcal{D}_S[\bar{\mathbf{U}}] = \sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}} \tag{2.84}$$

while the tuning parameter C_m, using the local equilibrium hypothesis and assuming a Kolmogorov spectrum for the turbulent energy cascade, can be evaluated as

$$C_S = \frac{1}{\pi}\left(\frac{3K_0}{2}\right)^{-3/4} \simeq 0.18 \tag{2.85}$$

Here K_0 is the Kolmogorov constant equal to 1.4. However, fixing a uniform value for C_S results in a wrong description of the turbulent subgrid-scale contributions. No corrections are taken into account for the laminar regions of the flow or the rotating portions. Moreover, the expression (2.84) exhibits an asymptotic behaviour in the near-wall regions approximatively equal to $\mu_{SGS} \sim \mathcal{O}(1)$. This trend is not physically consistent to the wall-turbulence theory which predicts a cubic behaviour in the wall-inner layer $\mu_{SGS} \sim \mathcal{O}(y^+)^3$ [60], where y^+ is the wall distance expressed in Kolmogorov units. To overcame the problem Germano et al. (1991) proposed to dynamically compute the Smagorinsky constants C_S from the resolved flow quantities, resulting in a space and time dependent formulation $C_S = C_S(\mathbf{x}, t)$. The method proposed by Germano et al. (1991) is addressed as *dynamical Smagorinsky*. The latter, even if it is quite computationally demanding, is very general and can be applied for all the turbulence models whose mathematical structure is expressed by the equation (2.83).

2.4.3 The wall-adaptive Large-Eddy viscosity model

A solution to solve the near-wall region problem was provided by Nicoud and Ducros (1999) who found a combination of the resolved velocity spatial derivatives that exhibit the asymptotic behaviour $\mu_{SGS} \sim \mathcal{O}(y^+)^3$ in the near-wall region. The method, known as *Wall-Adaptive Large-Eddy* (WALE) viscosity model defines $\mathcal{D}_m[\cdot]$ as

$$\mathcal{D}_W[\mathbf{U}] = \frac{(S^d_{ij}S^d_{ij})^{3/2}}{(S_{ij}S_{ij})^{5/2} + (S^d_{ij}S^d_{ij})^{5/4}} \tag{2.86}$$

where S^d_{ij} is the traceless symmetric part of the square of the resolved velocity gradient tensor defined as

$$S^d_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_l}\frac{\partial \tilde{u}_l}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_l}\frac{\partial \tilde{u}_l}{\partial x_i}\right) - \frac{1}{3}\frac{\partial \tilde{u}_m}{\partial x_l}\frac{\partial \tilde{u}_l}{\partial x_m}\delta_{ij} \tag{2.87}$$

Statical correlations based on DNS benchmarks want the constant of the model to be approximatively equal to $C_W = \sqrt{10.6}C_S$, where C_S is the Smagorinsky constant. The WALE model also possesses the interesting property that the subgrid viscosity vanishes if the flow is two-dimensional, in agreement with the physical behaviour (see [82] for details).

2.4.4 The σ-model

Many properties are required to improved the differential operator $\mathcal{D}_m[\cdot]$ of a LES turbulence model. Similarly to the WALE model, $\mathcal{D}_m[\cdot]$ should asymptotically vanish in near-wall regions in order to mimic the turbulence behaviour due to the zero-velocity condition. In particular, from the analytical description provided by Kuhn (1986), the third power law for μ_{SGS} consists of a strong requirement for any model. Let us call this *property 1*. At the same time, the turbulent viscosity should vanish in the case of a solid rotating flow and in the case of pure shear, like the WALE model. More generally, an improved differential operator should vanish for any 2D flow and this is resumed in the *property 2*. Indeed, such a situation for the resolved scales is not compatible with subgrid-scale behaviour, which is presumably three-dimensional. The same reasoning leads to the conclusion that the SGS viscosity should be zero in the case where the resolved scales are either in pure axisymmetric or isotropic contraction/expansion, consisting in a third fundamental property (*property 3*). The σ-model aims to target these three properties defining the non-linear differential operator $\mathcal{D}_m[\cdot]$ as

$$\mathcal{D}_\sigma[\cdot] = \frac{\sigma_3(\sigma_1 - \sigma_2)(\sigma_2 - \sigma_3)}{\sigma_1^2} \tag{2.88}$$

Here $\sigma_1 > \sigma_2 > \sigma_3 \geq 0$ denotes the singular values of the velocity gradient, thus the square root of the eigenvalues of the G_{ij} tensor. The latter is expressed by

$$G_{ij} = \frac{\partial \tilde{u}_i}{\partial x_j}\frac{\partial \tilde{u}_j}{\partial x_i} \tag{2.89}$$

It can be proved that G_{ij} is symmetric and semi-positive definite, so it has three real non-negative eigenvalue $\lambda_1 > \lambda_2 > \lambda_3$, concluding that $\sigma_i = \sqrt{\lambda_i}$, $i = 1, \ldots ,3$ can be ever computed. As shown by Toda et al. (2010), this arrangement theoretically preserves all the properties mentioned above. The method has been recently applied for both canonical and applicative flow simulations and the interested reader is addressed to look at Rieth et al. (2014) for a comparison with the other models.

2.5 Boundary conditions for Navier Stokes equations

In the Navier-Stokes system of equations an initial condition for the conservative variables and a set of conditions at the limits must be specified. Thus, we implemented a solver that supports the most common boundary conditions for Navier-Stokes equations both standard (periodic conditions, extrapolation, adiabatic/isothermal no-slip wall, adiabatic/isothermal slip-wall) and characteristic (subsonic/supersonic inflow and outflow). For conventional boundaries, the author suggests to analyse [35, 23, 56] while for *Navier-Stokes characteristic boundary condition* (NSCBC) a brief overview will be pointed out in the following paragraphs. Some notions will also be given in respect of a travelling shock wave initialisation, introducing the unsteady Rankine-Hugoniot conditions.

2.5.1 Navier-Stokes characteristic boundary conditions

Navier-Stokes characteristic boundary conditions are still a challenging problem for compressible flow simulations. The problem consists in the determination of a proper mathematical model able to describe accurately time-dependent boundaries for compressible flow dynamics. Because of the compressibility, the acoustics effects are inherent to the system's dynamics; thus, the flow variables tend to reflect at the boundary locations, producing unphysical waves and oscillations. The problem has been widely studied in the literature (see, e.g., Poinsot (2004)), and in this paragraph, we want to briefly overview the method. Following the idea of Pirozzoli and Colonius (2013), here we will present just one-dimensional inviscid approximation methodology. The latter was found to be applicable in all our computations. A more detailed presentation, taking into account transverse therms and viscous effects can be found in Lodato (2011).

Considering a one-dimensional inviscid approximation of the Navier-Stokes system equations (2.66) in the form of

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} \tag{2.90}$$

From the theory of hyperbolic PDE[3] we can recast the system as following

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{R}_x \boldsymbol{\Lambda}_x \mathbf{L}_x \frac{\partial \mathbf{U}}{\partial x} \tag{2.95}$$

Now defining

[3] Being $\mathbf{U} = \{u_k\}_{k=1}^{n}$ a set of variables for the system Σ and $\{\mathbf{F}^{(j)}(\mathbf{U})\}_{j=1}^{3}$ their fluxes. From a purely mathematical point of view a non-linear system of partial differential equations (PDE) in the form of

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}^{(j)}(\mathbf{U})}{\partial x_j} \tag{2.91}$$

is said to be *hyperbolic* if the Jacobian matrixes associated to the fluxes $\mathbf{F}^{(j)}(\mathbf{U})$

$$J_{ik}^{(j)} = \frac{\partial F_i^{(j)}(\mathbf{U})}{\partial u_k} \tag{2.92}$$

has real eigenvalues $\{\lambda_k^{(j)}\}_{k=1}^{n}$. Without being too detailed in respect of the theory of hyperbolic PDE, it can be proved that the Euler equations of Gas Dynamics represent a hyperbolic system of equations (see for instance [66, 117]) and their eigenvalues are

$$\lambda_k^{(j)} = \{u_j - c, u_j, u_j, u_j, u_j + c\} \quad j = 1, \ldots, 3 \tag{2.93}$$

For a hyperbolic model the eigenvalues of the Jacobian matrix J_{ik} represent the speed with which the information travels in the system while the eigenvectors represent the directions along which the information propagates. In particular, if a system is hyperbolic, the spectral theorem grants the Jacobian matrix J_{ik} to be diagonalised, thus

$$J_{ik} = L_{im} \Lambda_{mn} R_{nk} \tag{2.94}$$

here L_{im} and R_{nk} represent the left and right eigenvector associated to J_{ik} and $\Lambda_{mn} = \delta_{mn}\lambda_n$ is a diagonal matrix collecting in the $m = n$ positions the eigenvalues λ_n. The numerical method for Euler equations, so that the methods suitable for the discretisation of the convective terms of the Navier-Stokes system of equations, often exploit the eigenvalues and the eigenvectors of the Jacobian matrices associated with the flux components. In this field the directions described by the eigenvectors of Euler equation are addressed as *characteristics*.

$$\mathcal{L}_x(\mathbf{U}) = \mathbf{\Lambda}_x \mathbf{L}_x \frac{\partial \mathbf{U}}{\partial x} = \begin{cases} \lambda_1 \left(\frac{\partial p}{\partial x} - \rho c \frac{\partial u}{\partial x} \right) \\ \lambda_2 \left(c^2 \frac{\partial \rho}{\partial x} - \frac{\partial p}{\partial x} \right) \\ \lambda_3 \frac{\partial v}{\partial x} \\ \lambda_4 \frac{\partial w}{\partial x} \\ \lambda_5 \left(\frac{\partial p}{\partial x} + \rho c \frac{\partial u}{\partial x} \right) \end{cases} \tag{2.96}$$

the five characteristics waves associated to the flux component $\mathbf{F}(\mathbf{U})$ we can write

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{R}_x \mathcal{L}_x(\mathbf{U}) \tag{2.97}$$

that expanded reads

$$\begin{cases} \dfrac{\partial \rho}{\partial t} = -d_1 & (2.98\text{a}) \\ \dfrac{\partial \rho u_1}{\partial t} = -u_1 d_1 - \rho d_3 & (2.98\text{b}) \\ \dfrac{\partial \rho u_2}{\partial t} = -u_2 d_2 - \rho d_4 & (2.98\text{c}) \\ \dfrac{\partial \rho u_3}{\partial t} = -u_3 d_3 - \rho d_3 & (2.98\text{d}) \\ \dfrac{\partial \rho E}{\partial t} = -\dfrac{1}{2} u_i u_i d_1 - \dfrac{d_1}{\gamma - 1} - \rho u_1 d_3 - \rho u_2 d_4 - \rho u_3 d_5 & (2.98\text{e}) \end{cases}$$

where

$$\mathbf{d} = \begin{cases} \frac{1}{c^2} \left(\mathcal{L}_2 + \frac{1}{2} (\mathcal{L}_5 + \mathcal{L}_1) \right) \\ \frac{1}{2} \left(\mathcal{L}_5 + \mathcal{L}_1 \right) \\ \frac{1}{2\rho c} \left(\mathcal{L}_5 - \mathcal{L}_1 \right) \\ \mathcal{L}_3 \\ \mathcal{L}_4 \end{cases} \tag{2.99}$$

The equation (2.97) represents a suitable (one-dimensional) model for Navier-Stokes boundary conditions. In fact, to improve the boundary reflection behaviour, some assumptions can be made in respect of the five characteristic waves $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_5\}^T$ concerning the boundary location. Here two cases will be presented: subsonic inflow and subsonic outflow modelling.

2.5.2 Subsonic inflow

In case of a subsonic inflow, the method proposed by Yoo et al. (2007) was employed. We suppose the bound located at $x_b = 0$ thus $(u - c) < 0$ and $\mathcal{L}_1$ is outgoing the domain. On the other hand, $\mathcal{L}_2, \ldots, \mathcal{L}_4$ are incoming. In this case, four target variables are already known at the boundary location, i.e. u_t, v_t, w_t and T_t. Thus the equations (2.98b) (2.98b) (2.98d) and (2.98e) seem to not play any role and just an equation for density is required. Following such assumption, some situations may admit an acoustic reflection from the interior, resulting in a preferable strategy to relax the boundary condition thus defining

$$\mathcal{L}_2 = \eta_2 \frac{\rho c}{L_x}(T_t - T) \tag{2.100a}$$

$$\mathcal{L}_3 = \eta_3 \frac{c}{L_x}(v - v_t) \tag{2.100b}$$

$$\mathcal{L}_4 = \eta_4 \frac{c}{L_x}(w - w_t) \tag{2.100c}$$

$$\mathcal{L}_5 = \eta_5 \frac{\rho c^2(1 - M_{max}^2)}{L_x}(u - u_t) \tag{2.100d}$$

Here M_{max} is the maximum Mach number in the whole domain, L_x is the length in x-direction and η_i are coefficients whose role is to modulate the boundary relaxation. Here a value in the range of $\eta_i = 0.4 \div 0.6$ was found a good compromise in all our computations. Once $\mathcal{L}_2, \dots, \mathcal{L}_5$ have been modelled, the equations (2.98) can be employed to ensure the boundary. The last information required consists in computing a proper value for the outgoing wave $\mathcal{L}_1$. From a numerical point of view, a one-side derivation process can be carried out.

2.5.3 Subsonic outflow

Moving through subsonic outflow boundary condition the approach proposed by Pirozzoli and Colonius (2013) was employed. We suppose the boundary located at $x_b = L_x$. For the elliptical behaviour of the pressure field in the subsonic regime, differently from the subsonic inflow case, the only information we can exploit is the far-field pressure value, p_∞. On the other hand, from the characteristic analysis, the four waves $\mathcal{L}_2, \dots, \mathcal{L}_5$ are leaving the domain, while $\mathcal{L}_1$ is incoming. From a numerical point of view, the outgoing waves can be discretised by a one-side derivative, while $\mathcal{L}_1$ must be modelled. Two main approaches are possible:

1. The most simple approach consists of setting $\mathcal{L}_1 = 0$ supposing that no reflections can arise from the outflow boundary. The method, even if very straightforward, admits a progressive decrease of the mean pressure in the overall domain and sometimes it could result in wrong predictions;

2. On the other hand, the most common approach is the one proposed by Poinsot (2004). Here $\mathcal{L}_1$ is defined as

$$\mathcal{L}_1 = K(p_\infty - p) \tag{2.101}$$

where K is a relaxation constant, function of the Mach number

$$K = \sigma(1 - M_{max}^2)\frac{c}{L_x} \tag{2.102}$$

With this position, the $\mathcal{L}_1$ wave is modelled as dumping term base on the far-field pressure value. Here, M_{max} is the maximum Mach number in the whole domain, L_x is the domain length in x-direction and $\sigma \simeq 0.25 \div 0.27$ is a relaxation parameter. The latter method is strongly recommended, even if the presence of the σ parameter is a source of uncertainty. Thus Pirozzoli and Colonius (2013) proposed to directly discretise $\mathcal{L}_1$ with a one-side derivative approximation

$$\mathcal{L}_1 = \frac{1}{\Delta x_b}(u - c)[(p_\infty - p) - \rho c(u_\infty - u)] \tag{2.103}$$

In the present work, both of the expressions (2.101) and (2.103) were employed depending on the problem. In particular, as found by Pirozzoli and Colonius (2013), we confirm that the expression (2.103), dropping the velocity deficit $(u_\infty - u)$, performs better in ejecting aerodynamics wakes from the domain. Finally, once the characteristic waves $\mathcal{L}_1, \ldots, \mathcal{L}_5$ have been computed, the equations (2.98) can be employed to evolve the flow variables at the boundary location.

2.6 Summary

In the present chapter, the mathematical model describing the dynamics of an ideal Newtonian gas has been derived. Starting from the conservation principles for a closed physical system, the set of PDEs representing the mass conservation, the momentum conservation, and the total energy conservation have been derived. The model consisted in the well known Navier-Stokes system of equations. A brief overview of the dimensional analysis and the Π theorem by Buckingham have been shown and applied to the model; thus a non-dimensional vectorial formulation based on this theory and suitable for numerical applications has been presented. The chapters follows with a brief description of the filtered Navier-Stokes equation in the path of LES modelling. In the second part, an overall presentation of the boundary conditions for Navier-Stokes equations, mainly focused on characteristic boundaries has been shown.

3
Numerical integration of differential equations

God does not care about our mathematical difficulties; He integrates empirically.

Albert Einstein

Contents

In the previous chapter, we analysed and discussed a huge number of time-dependent differential models. Those models require a suitable numerical strategy to be evolved; thus, in this chapter, we will discuss the methods employed in the numerical integration of a time-dependent partial differential equation. These methods will be applied in the evolution of the Navier-Stokes system of equations. In particular, we will start with a brief introduction regarding the general theory behind the numerical integration, and secondly, we will move to high-order integration schemes focusing on the Runge-Kutta methods.

3.1 Introduction to numerical methods

3.1.1 Numerical methods for ODEs

Being $y : \mathbb{R} \to \mathbb{R}$ a real function with $y(t) \in C^1(\mathbb{R})$. A first order *ordinary initial value problem* for y reads

$$\begin{cases} y'(t) = f(t, y(t)), \quad t > t_0 \\ y(t_0) = y_0 \end{cases} \tag{3.1}$$

The latter expression is structurally similar to the Navier-Stokes system of equations, in fact, defining $\mathcal{N}[\cdot]$ the non-linear differential operator collecting a discrete formulation of the spatial derivatives, the equations (2.66) can be recast like

$$\frac{d\mathbf{U}(t)}{dt} = \mathcal{N}[t, \mathbf{U}] \tag{3.2}$$

resulting in the *canonical formulation* of ordinary differential problem. During the last century, many methods have been developed to solve the problem (3.1) and in this brief overview, we will limit to present them qualitatively. A detailed description, characterized by a very rigorous mathematical point of view, can be found in [103, 102], while a dissertation, closer to fluid dynamics problems, can be evinced in [84]. Starting our brief survey, we can claim that two are the main categories defining the numerical methods for ordinary differential equations (ODEs): the one-stage methods and the multi-stage methods.

The **one-stage methods** aim to determine the solution $\mathbf{U}^{n+1}$ at the time level t^{n+1} based just on the solution at the previous time-step $\mathbf{U}^n$. The process can be carried out explicitly in the form of

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \mathcal{N}\left[t^n, \mathbf{U}^n\right] \tag{3.3}$$

or implicitly

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \mathcal{N}\left[t^{n+1}, \mathbf{U}^{n+1}\right] \tag{3.4}$$

expressing the solution at the time level t^{n+1} as a function of both $\mathbf{U}^n$ and $\mathbf{U}^{(n+1)}$.

The **multi-stage methods** instead build the solution at time level t^{n+1} combining the previous $n + 1 - s$ solutions where s is the number of stages of the method, thus a general formulation of a linear multi-stages methods reads

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \sum_{k=s_0}^{s} \alpha_k \mathcal{N}[t^{n+1-k}, \mathbf{U}^{n+1-k}] \tag{3.5}$$

where s_0 denotes the starting stage and α_k are the coefficients of the linear combination. From (3.5), if $s_0 = 0$ the method is implicit, otherwise if $s_0 > 0$ the method is fully explicit.

Generally speaking, the one-step methods are rarely employed for time-accurate numerical applications. Those methods, computing $\mathbf{U}^{n+1}$ employing the previous computation $\mathbf{U}^n$, lack in term of accuracy, an unmissable quality in case of high-fidelity numerical modelling. On the other hand, the multi-step methods, taking into account the previous $n + 1 - s$ computations, can sharply increase the results accuracy, becoming suitable for computational fluid dynamics. Here some notes must be pointed out between explicit and implicit methods. Despite resulting in high-order discretisation strategies, explicit methods consist in very straightforward and flexible numerical algorithms suitable to evolve complex dynamical systems. However, because those methods are not unconditionally stable, the time-step between two consequents solutions could become very small, increasing the overall time of the simulation. That is, for instance, the case of wall-bounded flows. On the other hand, the implicit methods do not require a particular limitation time-stepping and the full stability is recovered for every Δt, but, to determine the implicitly defined $\mathbf{U}^{n+1}$ solution, this strength is counterbalanced by solving a non-linear set of equations at each time-step.

3.1.2 Runge-Kutta methods

As we mentioned in the previous paragraph, the only way to increase the accuracy of a numerical scheme for ODEs is to expand the number of prior solutions retained in the computation, thus employing a multi-stage method. These methods, even if popular in the past, nowadays have been overcome by most performing schemes. The main problem of a linear multi-stages methods consists in their initialisation. Because at time $t = 0$ just the initial condition is at our disposal, the firsts s evolutionary steps need to employ a low-order method to advance the solution. Another way approaching a completely different philosophy relies on Runge-Kutta (RK) methods. To maintain a one-step structure, the RK methods sacrifice the linearity of the scheme resulting in the following s stages formulation:

$$\mathbf{U}^{(i)} = \sum_{k=1}^{i-1} \left(\alpha_{ik} \mathbf{U}^{(k)} + \beta_{ik} \Delta t \mathcal{N} \left[t^{(k)} + \gamma_{ik} \Delta t, \mathbf{U}^{(k)} \right] \right), \quad i = 1, \dots, m \tag{3.6a}$$

$$\mathbf{U}^{(1)} = \mathbf{U}^{n} \tag{3.6b}$$

$$\mathbf{U}^{(m)} = \mathbf{U}^{n+1} \tag{3.6c}$$

Here we can notice that the discrete operator $\mathcal{N}[\cdot]$ is recursively computed over the solution obtained during s fractional steps. α_{ik}, β_{ik} and γ_{ik} are the coefficient defining the scheme. During the last decades, lots of Runge-Kutta methods have been proposed and presented to the scientific community and the reader will be guided to examine Quarteroni and Valli (2008) for a complete description of these methods. In the present work, the third-order explicit Runge-Kutta method (RK3) of Gottlieb and Shu (1998) was employed. The method represents one of the most common choice for the integration of the Navier-Stokes system of equations in compressible conditions and reads

$$\mathbf{U}^{(1)} = \mathbf{U}^{n} + \Delta t \mathcal{N} \left[t^{(n)}, \mathbf{U}^{(n)} \right] \tag{3.7a}$$

$$\mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^{n} + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t \mathcal{N} \left[t^{(1)} + \frac{1}{2}\Delta t, \mathbf{U}^{(1)} \right] \tag{3.7b}$$

$$\mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^{n} + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t \mathcal{N} \left[t^{(2)} + \frac{1}{2}\Delta t, \mathbf{U}^{(2)} \right] \tag{3.7c}$$

In particular, the latter RK scheme was proved to be the optimal third-order method with the non-trivial peculiarity of being Total Variation Diminishing (TVD). This property, defined as the *total variation* of a function $f(x)$

$$TV[f] = \int_{-\infty}^{\infty} \left| \frac{df}{dx} \right| dx \tag{3.8}$$

consists in granting

$$TV[f(x,t_2)] \leq TV[f(x,t_1)] \quad \forall t_2 \geq t_1 \tag{3.9}$$

The TVD methods represent handy tools in case of hyperbolic problems because they practically grant that the total amount of oscillations can not increase during the time. Moreover, the method was proved to be able to keep stability in whatever norm (see Gottlieb and Shu (1998) for details). Besides, as far as code optimisation issues, the method (3.7) can be cast in the so-called *low-storage* formulation, using a number of memory allocations per variable equal to the number of stages.

3.2 Stability criteria for flow simulations

As we pointed out introducing the numerical methods for ODEs, employing an explicit scheme, the time-step is required to be evaluated in such a way that the stability of the scheme is kept over the whole simulation. For fluid dynamics applications, the most tighten criteria consist in the Courant-Friedrichs-Lewy (CFL) stability condition for convective fluxes and the Fourier criterion (FO) for diffusion phenomena. The CFL-condition requires the time-step to be sufficiently small to reproduce the time-variation of the characteristic wave propagation associated with the Euler equation. Among the variety of formulations employed in the definition of a CFL-time-step, in the present work we used a modification of the one proposed by Pirozzoli (2002)

$$\Delta t_{CFL} = CFL \min_{j=1}^{3} \left(\frac{\min(\Delta x_j)}{\max(|u_j| + c)} \right). \tag{3.10}$$

Here Δx_j and u_j are respectively the grid-step and the velocity component in jth direction, c is the local speed of sound and CFL denotes the Courant-Friedrichs-Lewy number. The linear analysis applied to the method of Gottlieb and Shu (1998) proved that the stability is recovered up to $CFL = 1$. A proper choice for practical applications consist in setting the CFL number in the range of $0.5 \div 0.8$. Sometimes, especially in the case of wall turbulence, the diffusive phenomena act the major role in the stability of an explicit numerical method. In this field, the Fourier criterion establishes that the time-step must be set equals to

$$\Delta t_{Fo} = Fo \min \left(\frac{\min_j (\Delta x_j)^2}{\mu_\infty \max(\mu)}, \frac{\gamma/(\gamma-1)\min(\rho)\min_j(\Delta x_j)^2}{\lambda_\infty \max(\mu)} \right) \tag{3.11}$$

here μ is the viscosity, ρ the density, μ_∞ denotes the reference viscosity equal to $\sqrt{\gamma}M/Re$, λ_∞ is the reference diffusivity equal to $\mu_\infty \gamma/(Pr(\gamma-1))$ and Fo denotes the Fourier number, in our computation set equal to 0.1. During a simulation, both the criteria expressed by the equations (3.10) (3.11) must be computed at each time-step, and the minimum between them must be selected as the one to advance the solution, thus

$$\Delta t = \min(\Delta t_{CFL}, \Delta t_{FO}) \tag{3.12}$$

Finally, the numerical methods for ordinary differential equations suggest a suitable algorithm to solve a general time-dependent differential model. In the present work we employed a standard code structure consisting in computing the spatial contribution of the Navier-Stokes system of equations before updating the solution with the RK3 method. A detailed description of how structure a CFD solver can be found in Salvadore et al. (2013).

3.3 Validations and results

An error analysis must be performed to evaluate the error scaling of the Runge-Kutta method. Here, to have an analytical benchmark to compare with, we tested the method over a simple linear ordinary differential equation, thus solving

$$y' + y = 0 \tag{3.13}$$

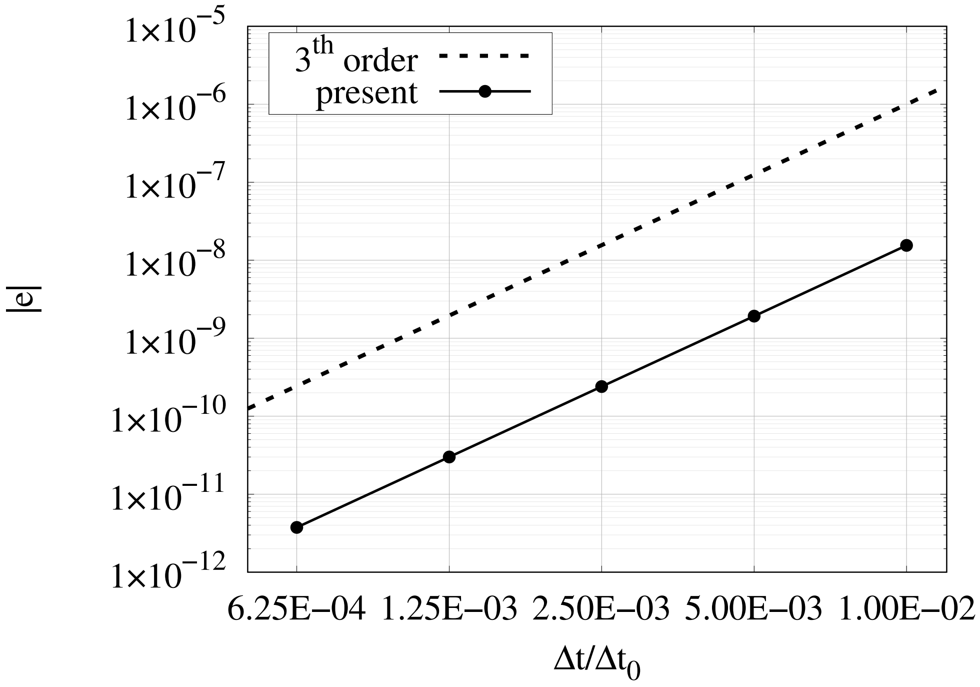


Figure 3.1: Error scaling on $y' + y = 0$ for the RK3 method.

The problem is defined by an initial condition $y(0) = 1$ and its analytical solution is represented by $y(t) = e^{-t}$. Being $y^*(t)$ the numerical result at t, we can compare it with the exact solution looking at the error norm $|e| = |y - y^*|$. Because the RK3 scheme is third-order accurate, the error should scale consequently. In Figure 3.1 the error scaling is reported as a function of the time-step. The latter was fixed equal $\Delta t/\Delta t_0 = \{0.01, 0.005, 0.0025, 0.00125, 0.000625\}^T$ and the computation was evolved up to $t/t_0 = 1$. From the results, we can evince an excellent performance in terms of error scaling with respect to the theoretical predictions, concluding that the scheme is properly implemented. The same routine tested in the present paragraph was employed to integrate the Navier-Stokes system of equations. In Table 3.1 the data corresponding to Figure 3.1 are reported.

	$\Delta t/\Delta t_0$	$\lvert e\rvert$	error order
RK3	$1.000E-02$	$1.54515E-08$	-
	$5.000E-03$	$1.92373E-09$	3.006
	$2.500E-03$	$2.40000E-10$	3.003
	$1.250E-03$	$3.00000E-11$	3.000
	$6.125E-02$	$3.75378E-11$	2.998

Table 3.1: Error scaling on $y' + y = 0$ for the RK3 method.

3.4 Summary

In the present chapter, the numerical method employed in the time-discretisation of a system of ordinary differential equations has been presented. In particular, after a brief introduction regarding the numerical integration, we moved to the Runge-Kutta methods for compressible flow dynamics. Here the explicit RK3 method by Gottlieb and Shu (1998) was presented, and some useful relations to determine a stable time step have been pointed out. The chapter ends with a quantitative result in the way of validating scheme and an overview of the whole numerical method employed in the evolution of the Navier-Stokes system of equations.

4

Numerical treatment of convective fluxes

We cannot solve our problems with the same thinking we used to create them.

Albert Einstein

Contents

In this chapter, we will discuss the numerical methods employed in the discretisation of the convective fluxes of the Navier-Stokes system of equations. For a compressible solver, this contribution is the one that needs the higher coding effort in terms of implementation difficulties and validation. In particular, here the convective fluxes have been discretised using a fifth-order weighted-essentially non-oscillatory scheme (WENO5) hybridised with the sixth-order fully split-convective energy-preserving scheme by Kennedy, Grüber and Pirozzoli (KGP6). The following sections will explained these techniques in details.

4.1 Spatial discretisation of convective fluxes

4.1.1 Weighted essentially non-oscillatory schemes

In the field of compressible flows, strong gradients and shockwaves are core topics. Due to the hyperbolic nature of Euler equations, those phenomena consist in sharp discontinuities of the fluid variables (i.e., density, pressure, temperature, and velocity). From the basics of spectral theory, the discrete reconstruction of a discontinuous function manifests the so-called *Gibbs phenomenon* that consists in spurious oscillations around the discontinuities. In this case, the dynamic of the Navier-Stokes equations propagates the oscillations progressively destroying the numerical solution. For these reasons, in the case of compressible flows with shockwaves, to stably propagate strong gradients and discontinuities, a suitable numerical method must be employed. In the present work, a 5th order weighted-essentially non-oscillatory scheme (WENO5) has been used. The method, originally designed by Jiang and Shu (1996) improves the fundamental suggestion of the essentially non-oscillatory (ENO) schemes, which were introduced by Harten et al. (1987). The key idea behind ENO is to employ the smoothest stencil among several candidates to discretise the convective fluxes at a cell interface. The process grants both a high-order of accuracy and avoids spurious oscillations near shocks. After its first implementation, in order to improve the scheme resolution and its efficiency, a large variety of WENO versions have been developed over the years (see e.g. Shu (1999), Acker et al. (2016), Balsara and Shu (2000), Balsara et al. (2016),Wang et al. (2018a)). In the present work, the so-called WENO-Z method by Castro et al. (2011) was employed.

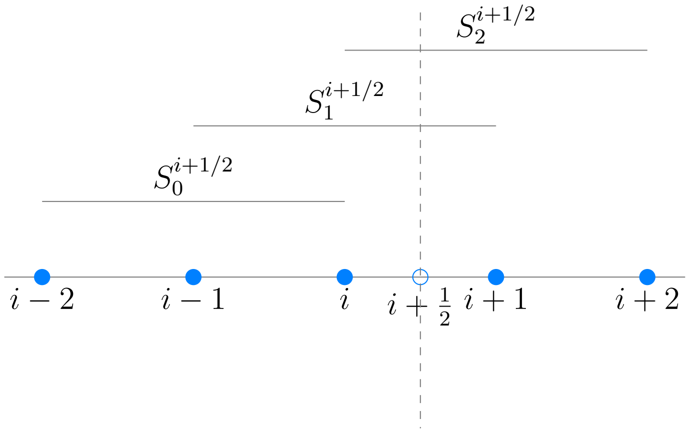


Figure 4.1: Computational stencils employed in WENO5 reconstruction.

The fundamental idea behind the WENO5 procedure is sketched in Figure 4.1. The figure sketches the three computational stencils employed in the WENO reconstruction. The WENO procedure consists in building a high-order polynomial function, centred around the cell interface $i + 1/2$, and based on the three sub-stencils $S_1^{i+1/2}, S_2^{i+1/2}, S_3^{i+1/2}$. Once the three polynomial functions have been computed, a non-linear combination of them, weighted by a smoothness indicator of the function in the actual stencil, is performed. In that way, if a strong gradient or a shockwave appears in the stencil $S^{i+1/2} = \{i-2, \dots, i+2\}$, an automatic upwinding or downwinding procedure is activated, resulting in a proper way to evolve a shock. Here we will describe the implementation steps behind the WENO procedure we will point out the method focussing on a single component of the convective fluxes $\mathbf{F}_j$.

Considering a conservative approximation[1] of a convective flux in the form of

$$\frac{\partial \mathbf{F}_j}{\partial x_j} \simeq \frac{1}{\Delta x_j}\left(\hat{\mathbf{f}}_{i+1/2} - \hat{\mathbf{f}}_{i-1/2}\right)_j \tag{4.1}$$

the WENO's goal is the computation of a high-order approximation of the numerical flux $\hat{\mathbf{f}}_{i+1/2}$ at the cell face $i+1/2$. The procedure is composed by the following sequence:

1. The first step computes the Roe's [23] averaged state at the cell face $i+1/2$ and the determination of the left and right eigenmatrices ($\mathbf{L}_{i+1/2}$, $\mathbf{R}_{i+1/2}$) associated to the Jacobian of flux component [107].

2. The second step consists in the application of the local Lax-Friedrichs (LLF) flux splitting [117] in order to determine the flux component associated with each s-characteristic direction $f^+_{s,i+1/2}$ and $f^-_{s,i+1/2}$ according to

$$\mathbf{f}^{\pm}_{i+1/2} = \frac{1}{2}\mathbf{L}_{i+1/2}\left(\mathbf{F}_i \pm |\lambda_{max}|\mathbf{U}_i\right) \tag{4.2}$$

   where λ_{max} denotes the maximum eigenvalue of the Jacobian matrix associated with the flux component $\mathbf{F}_j$.

3. For every s-direction the WENO interpolation procedure is performed following the one proposed by Jiang and Shu (1996). Here we will explain how $f^+_{s,i+1/2}$ is computed and for the sake of clarity, we drop the $+$ sign in the superscript. The formulas for the negative part of the split flux are symmetric in respect of the $i+1/2$. So, being

$$\mathbf{q}^{(0)}_{i+1/2} = +\frac{1}{3}\mathbf{f}_i + \frac{5}{6}\mathbf{f}_{i+1} - \frac{1}{6}\mathbf{f}_{i+2} \tag{4.3a}$$

$$\mathbf{q}^{(1)}_{i+1/2} = -\frac{1}{6}\mathbf{f}_{i-1} + \frac{5}{6}\mathbf{f}_i + \frac{1}{3}\mathbf{f}_{i+1} \tag{4.3b}$$

$$\mathbf{q}^{(2)}_{i+1/2} = +\frac{1}{3}\mathbf{f}_{i-2} - \frac{7}{6}\mathbf{f}_{i-1} + \frac{11}{6}\mathbf{f}_i \tag{4.3c}$$

   the three third-order interpolation polynomials on three different stencils, $\mathbf{f}^+_{i+1/2}$, can be expressed using a non-linear combination of them, thus

$$\mathbf{f}_{i+1/2} = \sum_{j=0}^{2} \mathbf{w}_j \mathbf{q}^{(j)}_{i+1/2} \tag{4.4}$$

[1] From the general theory of hyperbolic partial differential equations, a numerical discretisation of the flux $\mathbf{F}(\mathbf{U})$ associated with the conservative variables $\mathbf{U}$ is said *conservative* if it can be written

$$\frac{\partial \mathbf{F}_j}{\partial x_j} \simeq \frac{1}{\Delta x_j}\left(\hat{\mathbf{f}}_{i+1/2} - \hat{\mathbf{f}}_{i-1/2}\right)_j$$

The importance of a conservative numerical method is crucial because of two fundamental theorems:

- *A conservative numerical scheme, if convergent, converge to a weak solution of the problem* (Lax and Wendroff, 1960);
- *Non-conservative schemes do not converge to the correct solution if a shock wave is present in the flow* (Hou and LeFloch, 1994)

where $\{\mathbf{w}_j\}_{j=0}^2$ denotes the *non-linear weights* expressed by

$$\mathbf{w}_j = \frac{\alpha_j}{\sum_j \alpha_j}, \quad \text{with} \quad \alpha_j = \gamma_j \left(1 + \frac{|\beta_0 - \beta_2|^2}{(\beta_j + \varepsilon)^2}\right) \tag{4.5}$$

Here ε is a small coefficient, in our computations set to 10^{-10}, whose role is to avoid the vanishment of the denominator, while $\{\beta_j\}_{j=0}^2$ are the smoothness indicators of the functions $\mathbf{f}$ defined by

$$\beta_1 = \frac{13}{12}(\mathbf{f}_i - 2\mathbf{f}_{i+1} + \mathbf{f}_{i+2})^2 + \frac{1}{4}\left(3\mathbf{f}_i - 4\mathbf{f}_{i+1} + \mathbf{f}_{i+2}\right)^2 \tag{4.6a}$$

$$\beta_2 = \frac{13}{12}(\mathbf{f}_{i-1} - 2\mathbf{f}_i + \mathbf{f}_{i+1})^2 + \frac{1}{4}\left(\mathbf{f}_{i-1} - \mathbf{f}_{i+1}\right)^2 \tag{4.6b}$$

$$\beta_3 = \frac{13}{12}(\mathbf{f}_{i-2} - 2\mathbf{f}_{i-1} + \mathbf{f}_i)^2 + \frac{1}{4}\left(\mathbf{f}_{i-2} - 4\mathbf{f}_{i-1} + 3\mathbf{f}_i\right)^2 \tag{4.6c}$$

finally the coefficient $\{\gamma_j\}_{j=0}^2$ are the *linear weights* of the fifth-order WENO scheme and their values are $\gamma_j = \frac{1}{10}\{3,6,1\}^T$. In respect of the original procedure proposed by Jiang and Shu (1996), a little modification of the non-linear weights have been employed, following the idea Castro et al. (2011). In particular, expressing the α_j coefficients as a function of the overall smooth indicator $\tau = |\beta_0 - \beta_2|^2$ slightly improved the resolution properties of the scheme. The modification is known in the literature as WENO-Z.

4. Once the two flux $\mathbf{f}^{\pm}_{i+/2}$ components have been computed, the procedure ends with their recombination in physical space following

$$\hat{\mathbf{f}}_{i+1/2} = \mathbf{R}_{i+1/2}(\mathbf{f}^{+}_{i+1/2} + \mathbf{f}^{-}_{i+1/2}) \tag{4.7}$$

which represents a fifth-order approximation of the numerical flux at the left cell-bound. The reconstruction grants the 5th order of accuracy on smooth flow regions.

4.1.2 Fully split convective energy preserving methods

The WENO's procedure is well known to be robust and stable, but in the regions where discontinuities do not characterise the flow, it does not represent the most suitable choice in terms of accuracy and computational cost. In particular, in these regions, we want to avoid the introduction of artificial viscosity, preserving, as much as possible, the turbulent energy cascade; thus, a central scheme is required. Unfortunately, in the case of compressible flows, a standard central scheme shows bad behaviour even in smooth regions. In these part of the flow field, we employed the sixth-order central fully split approximation of convective flux developed by Kennedy-Grüber-Pirozzoli. An extensive description of the method can be found in Pirozzoli (2010) and Coppola et al. (2019). Here we will overview the essential steps of the method. Being

$$\left.\frac{\partial \rho u_j \phi}{\partial x_j}\right|_i \simeq \frac{1}{\Delta x_j}\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right) \tag{4.8}$$

a convective derivative associated to a generic transported scalar quantity (i.e. unity for the continuity equation, $\{u_i\}_{i=1}^3$ for the momentum equation and $H = \gamma/(\gamma-1)p/\rho + u^2/2$ for the total energy equation) a fully split approximation of equation (4.8) is expressed by

$$\frac{\partial \rho u_j \phi}{\partial x_j} = k_1 \frac{\partial \rho u_j \phi}{\partial x_j} + k_2 \left(u_j \frac{\partial \rho \phi}{\partial x_j} + \rho \frac{\partial u_j \phi}{\partial x_j} + \phi \frac{\partial \rho u_j}{\partial x_j} \right) + (1 - k_1 - k_2) \left(\rho u_j \frac{\partial \phi}{\partial x_j} + \rho \phi \frac{\partial u_j}{\partial x_j} + u_j \phi \frac{\partial \rho}{\partial x_j} \right) \tag{4.9}$$

where a conservative approximation is granted if $k_1 = k_2 = 1/4$. The expansion (4.9) of the convective derivative into a generalised form is able to stabilise the central scheme granting a semi-discrete preservation of kinetic energy in the limit of inviscid, incompressible flow. As shown by Pirozzoli (2010) the numerical flux associated with the split formulation can be written as

$$\hat{f}_{i+1/2} = 2 \sum_{l=1}^{L} a_l \sum_{m=0}^{l-1} \widetilde{(\rho, u, \phi)}_{i-m,l} \tag{4.10}$$

where

$$\widetilde{(\rho, u, \phi)}_{i,l} = \frac{1}{8} (\rho_i + \rho_{i+l})(u_i + u_{i+l})(\phi_i + \phi_{i+l}) \tag{4.11}$$

is the two-point, three-variables discrete averaging operator associated to the transported variable ϕ. The coefficient a_l maximise the formal order of accuracy of a central approximation of a $2L$-size stencil, thus $a_1 = 1/60$, $a_2 = -3/20$, $a_3 = 3/4$.

4.1.3 Convective scheme hybridisation

Finally, to hybridise the WENO5 scheme and the KGP6 method, our implementation makes use of shock detection techniques. The procedure consists in detecting *a priori* the regions where the shock waves are located to selectively employing the WENO scheme limiting the method in these portion of the flow. Since a unique expression able to carry out the job in every flow conditions has not been yet formulated, the present solver carries out three different implementations, each one optimised for specific a requirement.

The most conservative formulation makes use of the absolute value of the density gradient computed at cell interfaces, thus the shock detector reads

$$\theta = \max_{j=1}^{3} \left| \frac{\partial \rho}{\partial x_j} \right| \tag{4.12}$$

A bit tighten is the density jump, defined as

$$\theta = \max_{j=1}^{3} \left| \delta \rho_{j+1/2} \right| \tag{4.13}$$

where $\delta \rho_{j+1/2} = \rho_{j+1} - \rho_j$ is the density jump at the cell interface. Finally, in the case of aerodynamics problems, an improved version of the Ducros sensor [30] has shown excellent properties in the detection of both shocks and slip lines. This shocks detector relies on the expression:

$$\theta = \max \left(- \frac{div(\mathbf{u})}{\sqrt{(div(\mathbf{u}))^2 + rot(\mathbf{u})^2 + \varepsilon}}, 0 \right) \tag{4.14}$$

where $div(\mathbf{u})$ is the velocity divergence, $rot(\mathbf{u})$ is the vorticity and $\varepsilon = (u_\infty / c)^2$. Once the proper shock detector has been selected, a threshold $\bar{\theta}$ to distinguish shocked and smooth regions must be specified, thus if $\theta > \bar{\theta}$ the flow is considered shocked, and the WENO reconstruction procedure is activated in the actual cell and all the three cells in the immediate circumstances, otherwise the KGP scheme is employed. Again the threshold is strongly dependent by the flow in itself and some preliminary settings need to be employed.

4.1.4 Numerical treatment on non-uniform grids

The discretised formulations of the convective fluxes, presented in the previous paragraphs, are suitable for uniform Cartesian grids. The fact could be a substantial limitation in many applicative situations where most the non-linear phenomena are concentrated in small fluid regions, often known as *a priori*. For this reason, employing non-uniform grids is a mandatory target to improve the accuracy of the simulation and reduce the elapsed time. For convective derivative the extension is quite straightforward, thus being $\partial\phi/\partial x$ a spatial derivative for ϕ we can write

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi}\left(\frac{\partial x}{\partial \xi}\right)^{-1} \tag{4.15}$$

where ξ is a new coordinate variable describing the so-called *computational space*. The latter consist of a uniform Cartesian grid described by

$$\xi_i = \frac{(i-1/2)}{N} \qquad i = 1, \dots, N \tag{4.16}$$

and $\partial x/\partial\xi$ is the Jacobian of the transformation between the physical coordinates and the computational space. Knowing a mapping function $x(\xi)$ describing the nodal distribution as a function of the computational grid coordinate, the expression (4.15) allows us to compute the first derivative of the variable ϕ in a simple uniform computational space, employing the metric of $x(\xi)$ for the non-uniform extension. A considerable number of examples in therms of mapping functions $x(\xi)$ can be found in Orlandi (2012).

4.2 Validations and results

4.2.1 Error scaling

The first test that we have considered for the validation of the convective fluxes consists in the advection of a single harmonic density signal with periodic boundary conditions. The problem initialisation reads:

$$(\rho, u, p)_{t=0} = (1 + A\sin(\pi x), M, 1) \tag{4.17}$$

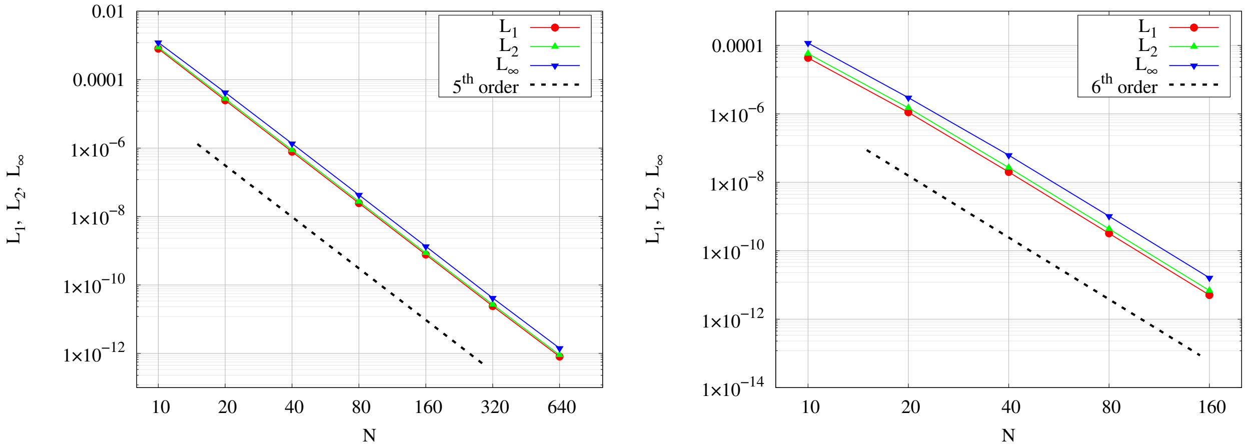
Here M was set equals to 0.05, while $A = 0.2$. For this problem, the Euler system of equations admits an analytical solution expressed by

$$(\rho, u, p) = (1 + A\sin(\pi x - Mt), M, 1) \tag{4.18}$$

consisting in a linear advection of the initial signal. Being $\{\rho_i^*\}_{i=1}^N$ a collection of numerical results for the problem computed in the $i = 1, \dots, N$ nodal positions, we can compare them with the analytical solution by the definition of the $p-norm$ and the $\infty-norm$ of a vector, that reads:

$$L_p = \left(\frac{1}{N} \sum_{i=1}^{N} |\rho_i - \rho_i^*|^p \right)^{1/p} \tag{4.19}$$

$$L_\infty = \max_i |\rho_i - \rho_i^*| \tag{4.20}$$



(a) WENO5 error scaling

(b) KGP6 error scaling

Figure 4.2: Error scaling on smooth solution for WENO5 (a) and KGP6 (b) methods.

	N	L_1	L_2	L_∞	L_1 order	L_2 order	L_∞ order
WENO5	10	0.000805525	0.000883539	0.00122389	-	-	-
	20	2.48468E-05	2.77722E-05	4.22504E-05	5.019	4.992	4.856
	40	7.84571E-07	8.75621E-07	1.35970E-06	4.985	4.987	4.958
	80	2.46497E-08	2.74776E-08	4.28481E-08	4.992	4.994	4.988
	160	7.71033E-10	8.59657E-10	1.34167E-09	4.999	4.998	4.997
	320	2.41230E-11	2.68817E-11	4.18253E-11	4.998	4.999	5.004
	640	8.06707E-13	9.00148E-13	1.42131E-12	4.902	4.9	4.879
KGP6	10	4.34000E-05	5.76996E-05	0.000117319	-	-	-
	20	1.10911E-06	1.49683E-06	2.98040E-06	5.291	5.269	5.299
	40	2.00245E-08	2.69216E-08	6.18570E-08	5.791	5.797	5.590
	80	3.21839E-10	4.36456E-10	1.02255E-09	5.959	5.947	5.919
	160	5.07592E-12	6.88361E-12	1.61948E-11	5.987	5.987	5.980

Table 4.1: Error scaling on smooth solution for WENO5 and KGP6.

Because the WENO5 and the KGP6 schemes are respectively fifth-order and $sixth-order$ accurate on smooth solutions, we expect that the error scales consequently. In Figure 4.2 the error scaling of both schemes, in therm of L_1, L_2 and L_∞ norms is reported as a function of the number of points employed in the discretisation. In particular a set of $N = \{10,20,40,80,160,320,640\}$ points has been employed to test the WENO5 scheme, while $N = \{10,20,40,80,160\}$ has been used for the KGP6 method. In order to fully decouple the spatial and the time errors, the time-step was evaluated setting a CFL number equals to 0.1. The data referring to Figure 4.2 are reported in Table 4.1.

4.2.2 One-dimensional Riemann problem for Euler equations

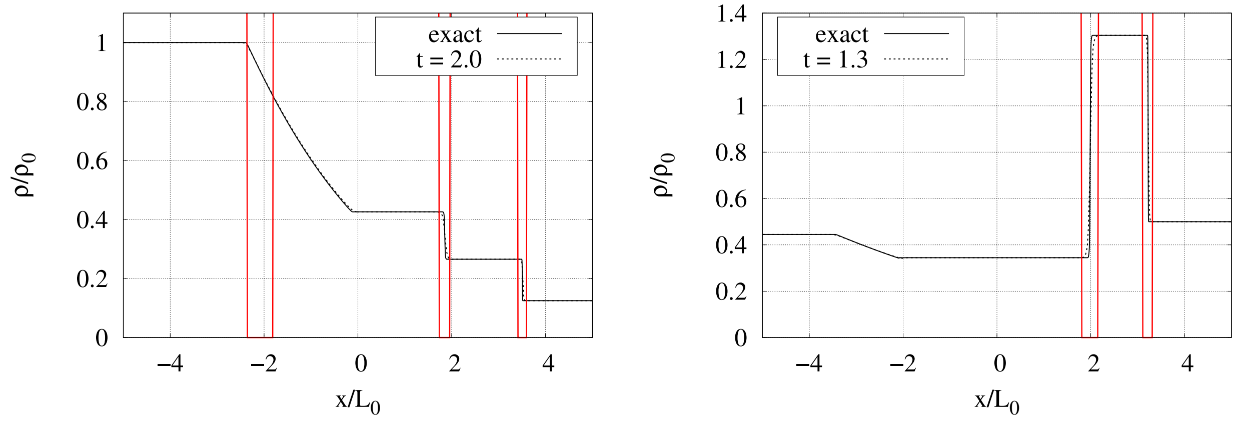
In the previous paragraphs, the performance of WENO5 and the KGP6 methods have been tested over a smooth solution and the error scaling was pointed out. To better show the stability and the robustness of our method, the results obtained over various one-dimensional Riemann problems for the Euler equation will be presented. The Riemann problem, often known as *shock tube*, consists in initialising a discontinuity inside a fluid domain at $t = 0$ and drawing the solution over time. Three cases will be considered: the Sod's shock tube, the Lax's shock tube and the Shu-Osher problem whose initial conditions are expressed respectively

$$(\rho, u, p)_{t=0,Sod} = \begin{cases} (1.000, 0.0, 1.0) & \text{if } x < 0 \\ (0.125, 0.0, 0.1) & \text{if } x > 0 \end{cases} \tag{4.21a}$$

$$(\rho, u, p)_{t=0,Lax} = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x < 0 \\ (0.500, 0.000, 0.571) & \text{if } x > 0 \end{cases} \tag{4.21b}$$

$$(\rho, u, p)_{t=0,Shu-Osher} = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x < 0 \\ (1.0 + 0.2\sin(5.0x), 0.0, 1.0) & \text{if } x > 0 \end{cases} \tag{4.21c}$$

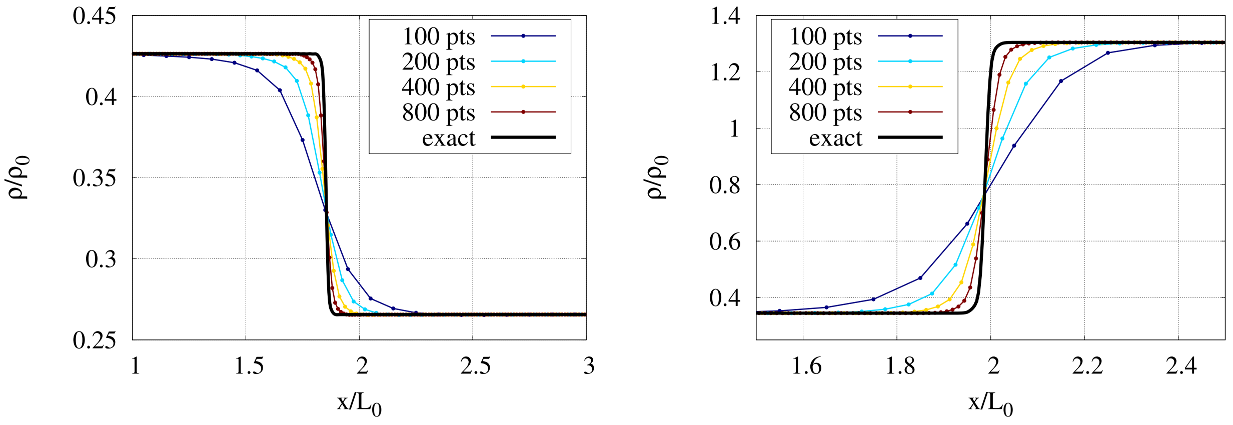
The problem will be solved with the hybrid WENO5+KGP6 scheme employing the *density gradient* shock detector over a uniform grid. The threshold for the density gradient shock sensor was set equals to 0.3. In Figure 4.3, a comparison between the *analytical* and the numerical solutions for Sod's and Lax's shock tubes are reported in terms of the density profile. The *analytical solution* was determined using 3000 points up to $t = 2$ and $t = 1.3$ time units respectively, while the *numerical* calculation consisted in evolving 500 computational nodes. Extrapolation boundaries were ensured at both limits, while the time-step was evaluated from the Courant-Friedrichs-Lewy stability condition with a CFL number equals to 0.5. Red lines represent the shock regions of WENO5 scheme. As we notice, the latter consist in the $\simeq 5 \div 10\%$ of the whole domain and for this reason, employing the WENO5 scheme everywhere it is both pointless and computationally expensive. Figure 4.4 reports an enlargement of the contact discontinuities characterising the two shock tube problems as a function of the grid resolution.



(a) Sod's shock tube

(b) Lax's shock tube

Figure 4.3: Density profiles for the Sod's shock tube (a) and Lax's shock tube (b). Here a comparison between the analytical and the numerical solution is presented. Shocks regions are underlined between red lines.

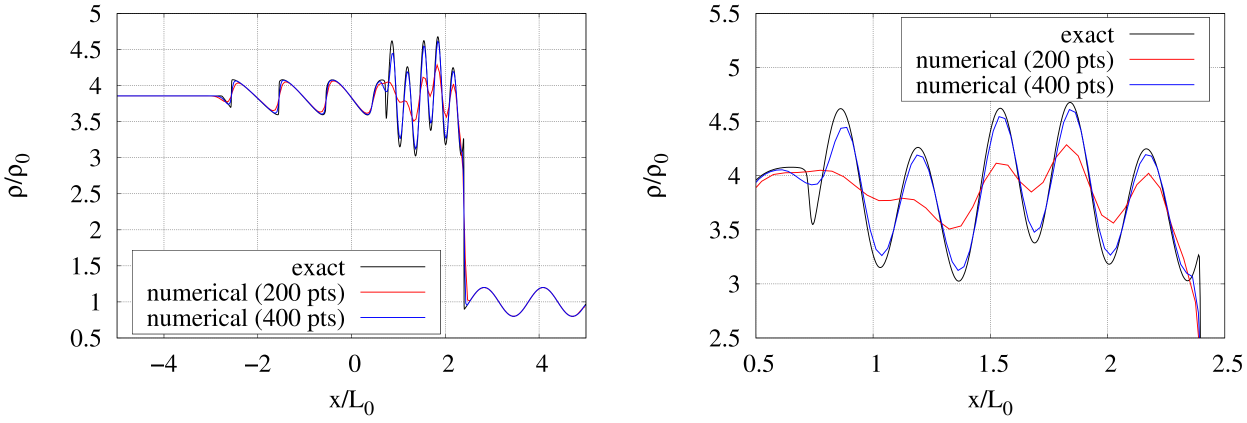


(a) Sod's shock tube

(b) Lax's shock tube

Figure 4.4: Enlargement of the Sod's (a) and Lax's (b) shock tube forward-facing and backward-facing contact discontinuity. Here a comparison between the *analytical* and the numerical solution is presented as a function of the grid resolution.

Moving through the Shu-Osher's test, the problem consists of a moving density-wave interacting with a shock. The problem is considered a very challenging goal for shock-capturing numerical methods. The physical oscillations, in fact, due to the shock-density-wave-interaction, can be easily confused as shocks by the upwinding WENO procedure and consequently diffused. In Figure 4.5 the results in terms of density profiles at various grid resolutions are reported at a reduce time t equals to 1.8. As fas as the simulation set up, the same settings as the Sod's and Lax's shock tube were employed. From Figure 4.5b we can see that, even without increasing too much the number of points employed in the simulation, good resolution properties can be ascribed to our method, resulting in an excellent agreement with the analytical solution.



(a) Shu-Osher problem density profile at $t = 1.8$

(b) Enlargement of the shock-density-wave-interaction

Figure 4.5: Shu-Osher problem for Euler equations. Here the density profile is reported at different grid resolutions. The *exact* solution was obtained over 3000 points discretisation. An enlargement of the shock-density wave interaction is reported in panal (b).

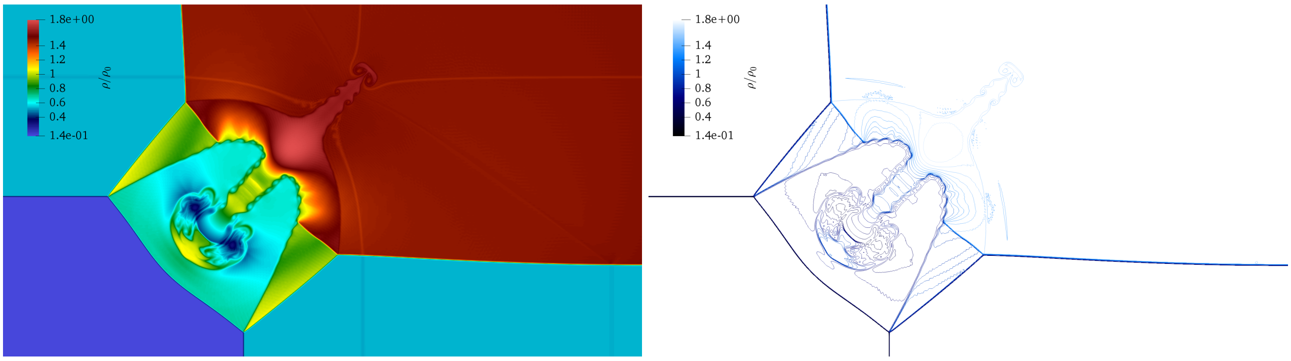
4.2.3 Two-dimensional Riemann problem for Euler equations

Moving through two-dimensional tests, in the present work, the 2D-Riemann problem for Euler equations has been addressed. The problem, similarly to the one-dimensional version, consists in initialising a discontinuous solution to the Euler system of equations and drawing the evolution of the system in time. Introducing a second dimension, a vast number of combinations and configuration of the problem are possible. In Guo and Shi (2018) a detailed description of the most common is reported. Here we will refer to the so-called *four quadrants* problem, and we will practically present the two following initialisations:

$$(\rho,u,v,p)_{t=0,A} = \begin{cases} (1.5000, 0.000, 0.000, 1.500) & \text{if } x>0,\ y>0 \\ (0.5323, 1.206, 0.000, 0.300) & \text{if } x<0,\ y>0 \\ (0.1380, 1.206, 1.206, 0.029) & \text{if } x<0,\ y<0 \\ (0.5323, 0.000, 1.206, 0.300) & \text{if } x>0,\ y<0 \end{cases} \tag{4.22a}$$

$$(\rho,u,v,p)_{t=0,B} = \begin{cases} (1.00, 1.00, +0.75, -0.50) & \text{if } x>0,\ y>0 \\ (2.00, 1.00, +0.75, +0.50) & \text{if } x<0,\ y>0 \\ (1.00, 1.00, -0.75, +0.50) & \text{if } x<0,\ y<0 \\ (3.00, 1.00, -0.75, -0.50) & \text{if } x>0,\ y<0 \end{cases} \tag{4.22b}$$

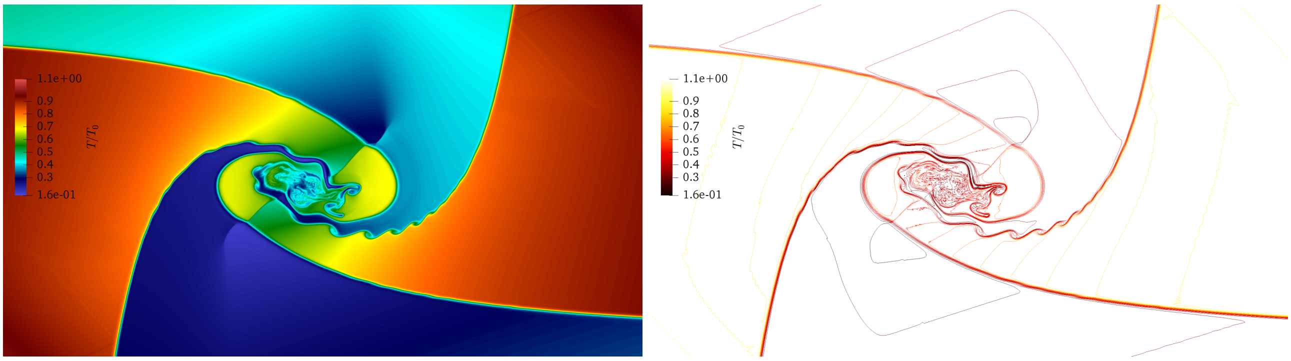
Despite previous validations, the results of the two-dimensional Riemann problem for Euler equations are purely qualitative in term of contours of fluid variables. In Figure 4.6 and Figure 4.7 the results of the computation, in terms of density and temperature fields are reported. Both simulations have been performed on a uniform Cartesian grid sizes $L_x \times L_y = 1 \times 1$ length units and featuring in $N_x \times N_y = 1600 \times 1600 = 2560000$ computational nodes. Extrapolation boundaries have been employed in all the four edges of the domain, and the solution was advanced in time up $t/t_0 = 0.3$ fixing a CFL equals to 0.5. Compared to other qualitative results available in the literature, our solution to the four-quadrants problem is in a complete agreement. Moreover, our computation denotes very high quality and cleanliness and no particular spurious oscillations seam to appear in the flow fields.



(a) Non-dimensional density field

(b) Non-dimensional density contours

Figure 4.6: Non-dimensional density field (a) and twenty level contours (b) for the four quadrant problem in configuration A.

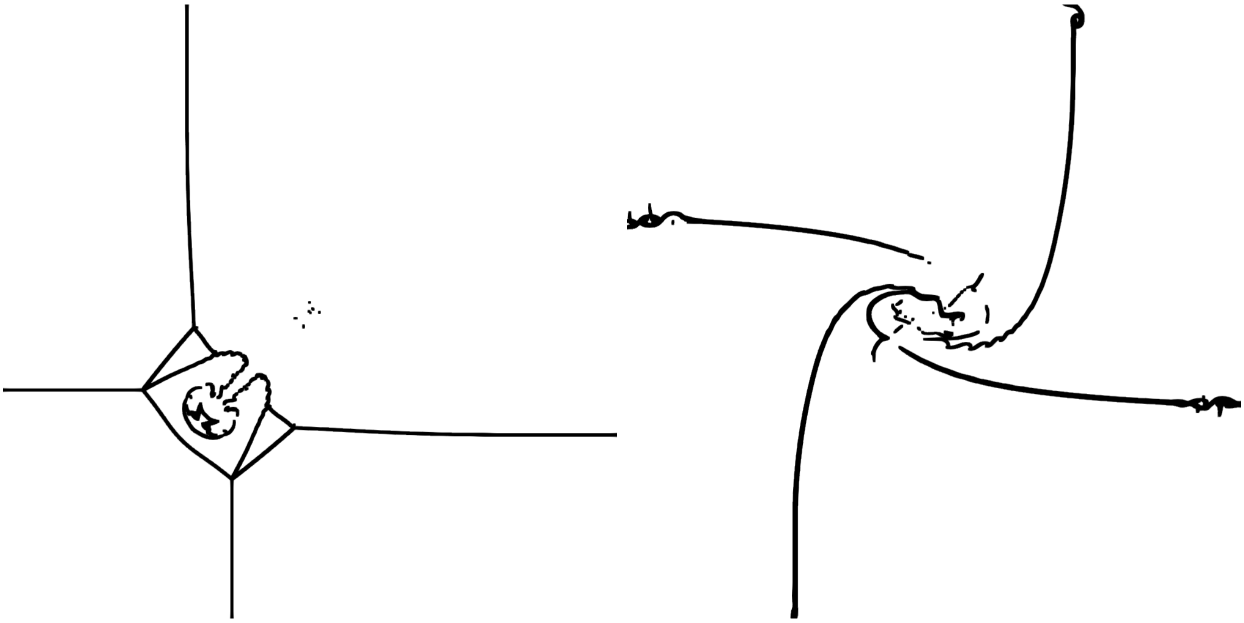


(a) Non-dimensional temperature field

(b) Non-dimensional temperature contours

Figure 4.7: Non-dimensional temperature field (a) and twenty levels contours (b) for the four quadrant problem in configuration B.

The four-quadrants problem gives us the possibility also to test the shock detection algorithm. Here the *density jump* shock detector was employed setting a threshold $\bar{\theta}$ equals to 0.05. In figure Figure 4.8, the shock detector field is reported in all the computational domain for both the two configurations. As we can notice, just the thin regions around the shocks are flagged by the algorithm and only in this little portions of the domain, the WENO reconstruction procedure was activated. This allows the computation to be fast and efficient, minimising the non-linear operation per time-step.



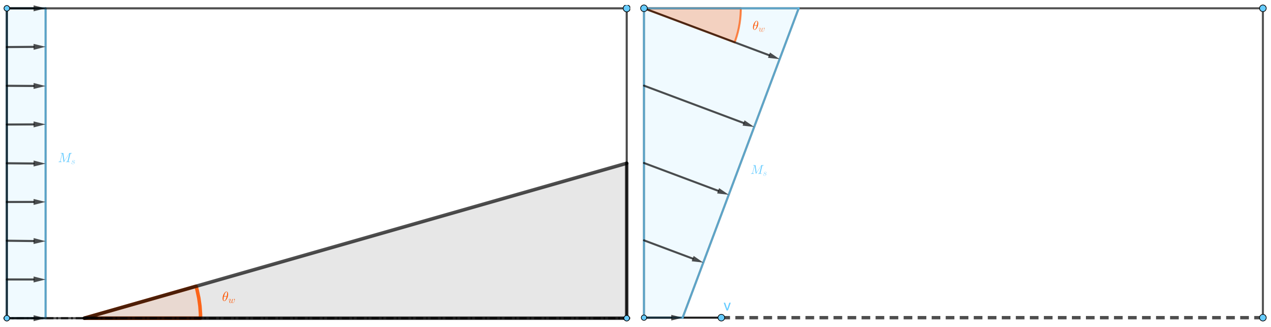
(a) Configuration A

(b) Configuration B

Figure 4.8: WENO-activated cells for the four-quadrants problems. Here the *density jump* detection technique has been employed fixing $\bar{\theta} = 0.05$.

4.2.4 Double-Mach reflection

The last test we considered in the validation of the convective fluxes consists of the *double Mach reflection problem*. The problem always belongs to the 2D-Riemann problems' category for Euler equations, but for its peculiarities, it is often treated separately. Woodward and Colella (1984) suggested the test and a detailed analytical description can be found in Li and Ben-Dor (1995). The problem consists in studying the complexity of the interaction between a moving shock wave and an infinite wedge. The most common configuration sees the shock travelling at $M_s = 10$ and a wedge characterised by an angle $\theta_w = \pi/6$. When the shock runs up the wedge, a self-similar shock structure, characterised by two triple points, evolves. As usual, to adapt the problem for a Cartesian domain, a rigid clockwise rotation is applied to the physical problem. Thus the shock is inclined by an angle $\theta_s = \pi/2 - \theta_w$ in respect of $x - axis$, while the wedge overlaps the bottom boundary of the domain. In Figure 4.9, a sketch of both the physical and the computational domain is reported.



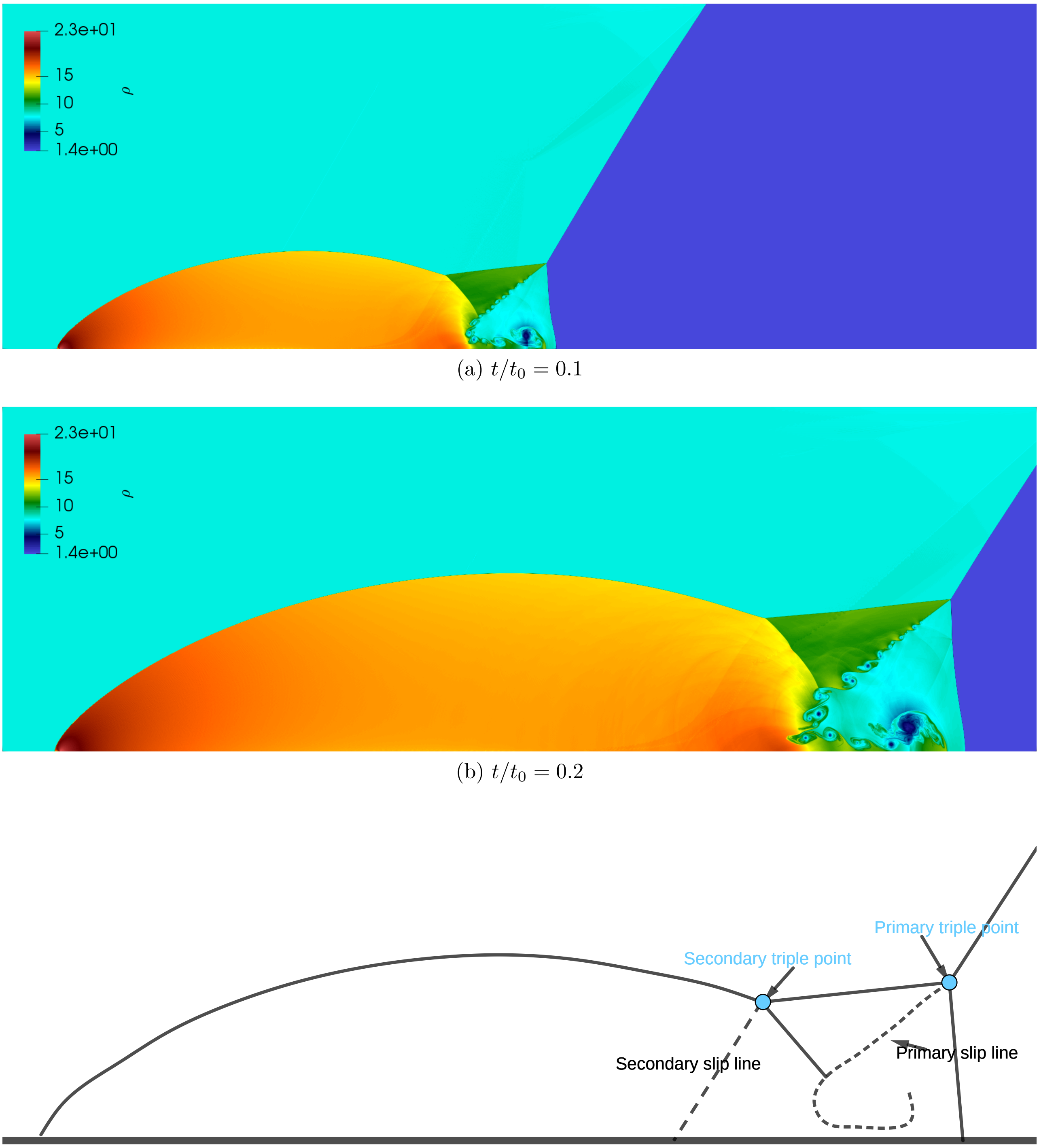
(a) Physical domain

(b) Adaptation for a Cartesian domain

Figure 4.9: Sketch of the physical (a) and the computational (b) domain employed in the simulation of the double Mach reflection problem. On the left panel a shock wave is travelling at M_s against a θ_w-wedge, on the right panel the same shock is rotated clockwisely by the θ_w angle and is travelling against a slip horizontal wall.

Initial condition consists in a $M_s = 10$ travelling oblique shock wave was enforced to reproduce the problem numerically (see appendix A for details) . At the upper bound the analytical solution of the moving shock was enforced, while the bottom side of the domain consisted in two different boundary conditions: for $x < x_0$ a post-shocked field, exactly equals to the one firstly initialised, was continuously enforced during time, while for $x > x_0$ a slip-wall condition was employed. Here x_0 is the initial shock position, in our computation equals $1/6$. The simulation was performed on a uniform Cartesian grid sizes $L_x \times L_y = 3 \times 1$ and featuring $N_x \times N_y = 6000 \times 2000$ computational grid points. The CFL number was usually set equals 0.5. In Figure 4.10 the results at $t/t_0 = 0.2$ in terms of density field are reported, while in Figure 4.11 an enlargement of the complex interaction due to the double Mach reflection is shown.

Similarly to previous conclusions, the results regarding the double Mach reflection are all qualitative and comparable just in term of field contours. However, the present result confirm the trend of our methodology in being particularly clean and robust, and Figure 4.11 denotes that very good resolution properties were recovered the schemes we implemented.

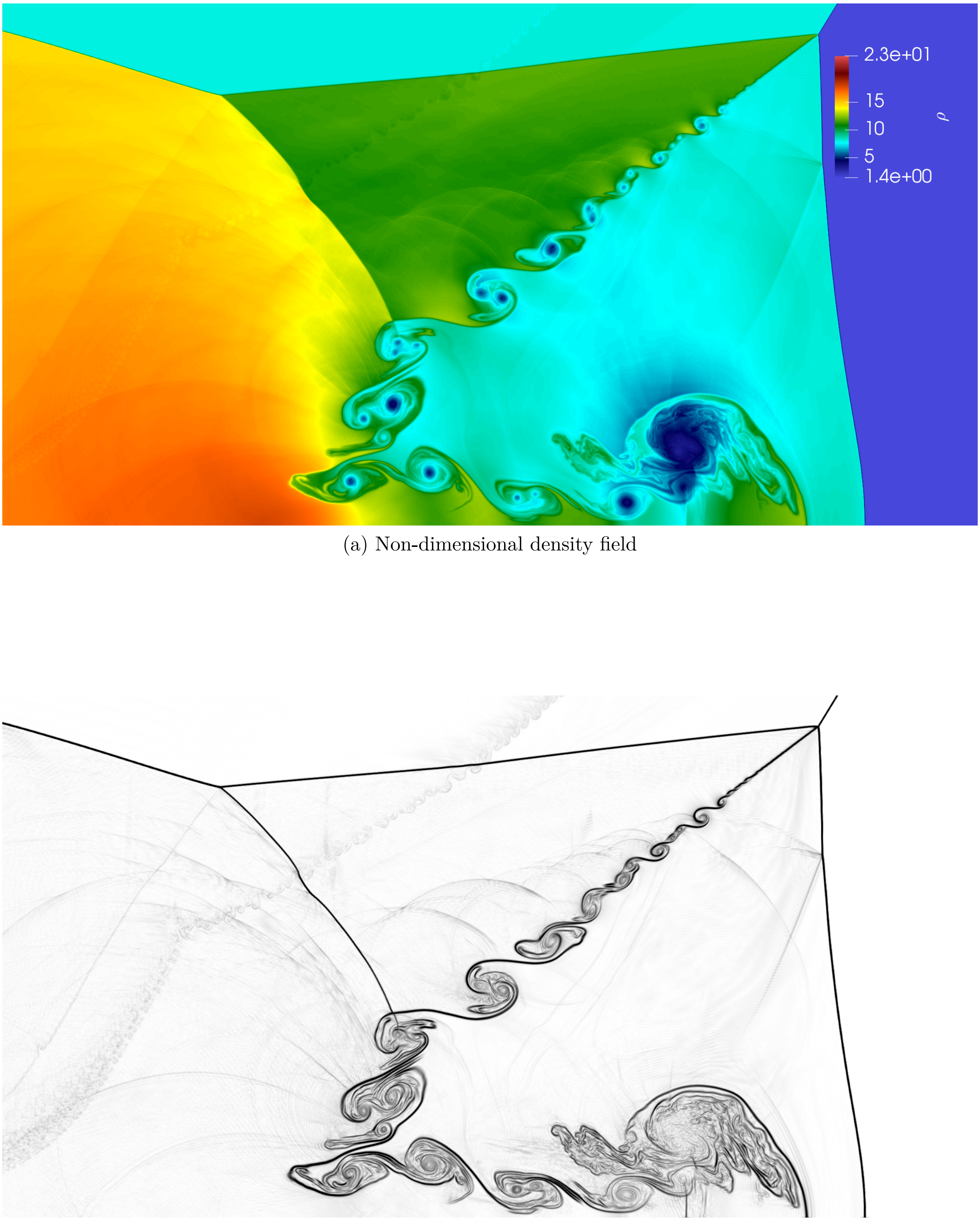


(a) $t/t_0 = 0.1$

(b) $t/t_0 = 0.2$

(c) sketch to the computational geometry

Figure 4.10: Self similar structure of the non-dimensional density field for the double Mach reflection problem at $t/t_0 = 0.1$ (a) and $t/t_0 = 0.2$ (b) and sketch to the computational geometry (c).



(a) Non-dimensional density field

(b) Schlieren density

Figure 4.11: Non-dimensional density field (a) and Schlieren density (b) enlargement for the double Mach reflection problem.

4.3 Summary

In the present chapter, the numerical methods employed in the discretisation of the Navier-Stokes convective fluxes have been presented, in particular, both the WENO5 and KGP6 schemes. The first consisted in one the most suitable shock-capturing method available in the literature; the second consists of a stabler extension of a central scheme, particularly recommended for smooth and turbulent regions. Some details regarding of the hybridisation strategy have been pointed out. Finally, to test the numerical method, a considerable number of results and validations have been reported.

5

Numerical treatment of viscous fluxes

> *I have had my results for a long time: but I do not yet know how I am to arrive at them.*
>
> Carl Friedrich Gauss

Contents

In this chapter, we will discuss the numerical methods employed in the discretisation of the Navier-Stokes system of equations. Those terms have been discretised with a standard sixth-order central finite-difference. However, a suitable formulation has been employed to improve the resolution property of the scheme and its stability. At the end of the chapter, a conspicuous amount of numerical results and validations will be presented.

5.1 Spatial discretisation of viscous fluxes

5.1.1 Fully expanded formulation of viscous fluxes

Numerical treatment of viscous fluxes could be computational demanding for a co-located finite-difference solvers. Recalling the expression of the dynamical contribution of the stress tensor

$$\frac{\partial d_{ij}}{\partial x_j} = \frac{\sqrt{\gamma} M_\infty}{Re} \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_s}{\partial x_s} \delta_{ij} \right) \right) \tag{5.1}$$

and the diffusive contribution to the energy equation

$$\frac{-\partial q}{\partial x_j} + \frac{\partial u_i d_{ij}}{\partial x_j} = \frac{\sqrt{\gamma} M_\infty}{Re} \left(\frac{\gamma}{\gamma - 1} \frac{1}{Pr} \frac{\partial}{\partial x_j} \left(\lambda \frac{\partial T}{\partial x_j} \right) + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_s}{\partial x_s} \right) u_i \right) \right) \tag{5.2}$$

we immediately recognise the complexity behind the viscous terms discretisation. Before going into the details of the numerical discretisation of the Navier-Stokes viscous terms, a suitable formulation of viscous fluxes for a co-locate finite difference solver will be pointed out. Following the suggestion of Pirozzoli (2011a) a *viscous derivative*, exploiting the product role, can be expanded as follows

$$\frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} \right) = \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial \mu}{\partial x_j} \frac{\partial u_i}{\partial x_j} \tag{5.3}$$

The process can be carried out for all the viscous terms resulting in a *fully expanded formulation*, improving both the resolution and the stability of the central scheme. In the final formulation we will see that the number of mixed derivatives is much less reduced achieving a considerable gain in term of computational efficiency. Focusing on d_{ij} we can expand its divergence as

$$\frac{\partial d_{ij}}{\partial x_j} = \mu \frac{\partial \tau_{ij}}{\partial x_j} + \tau_{ij} \frac{\partial \mu}{\partial x_j} \tag{5.4}$$

where τ_{ij} represents the differential contribution. Expanding (5.4) we obtain

$$\frac{\partial \tau_{ij}}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_s}{\partial x_s} \delta_{ij} \right) \tag{5.5a}$$

$$= \frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_j} \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial}{\partial x_j} \frac{\partial u_s}{\partial x_s} \delta_{ij}. \tag{5.5b}$$

Recognising that

$$\frac{\partial}{\partial x_j} \frac{\partial u_j}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\partial u_s}{\partial x_s} \right) \delta_{ij} \tag{5.6}$$

we can write

$$\frac{\partial \tau_{ij}}{\partial x_j} = \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{1}{3} \frac{\partial}{\partial x_i} \left(\frac{\partial u_s}{\partial x_s} \right) \delta_{ij} \tag{5.7}$$

Thus (5.4) becomes

$$\frac{\partial d_{ij}}{\partial x_j} = \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{1}{3}\mu \frac{\partial}{\partial x_i}\left(\frac{\partial u_s}{\partial x_s}\right)\delta_{ij} + \tau_{ij}\frac{\partial \mu}{\partial x_j} \tag{5.8}$$

The latter expression highlights some salient points. Fist of all, in the limits of an incompressible flow, equation (5.8) reduces to the well-known

$$\frac{\partial d_{ij}}{\partial x_j} = \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{5.9}$$

concluding that the viscous compressible effects can be seen as a high-order series expansion of incompressible phenomena. Secondly - except for the gradient of the velocity divergence - no mixed derivatives appear in this formulation. Finally, recalling tensorial calculus identities, the computation of the momentum viscous terms can be re-used into the energy equation

$$\frac{\partial u_i d_{ij}}{\partial x_j} = \frac{\partial d_{ij}}{\partial x_j} u_i + \mu \tau_{ij} \frac{\partial u_i}{\partial x_j} \tag{5.10}$$

5.1.2 Numerical method and algorithm

The discretisation of the viscous terms has been carried out into two sub-steps.

1. To speed up the memory accesses, the primitives variable (i.e., u, v, w, p, T) and the viscosity field μ have been extracted from the conservative fields. The process is carried out once per iteration in all the computational domain;

2. when the primitive fields have been computed, a second computational kernel was employed to compute their gradients. All the spatial derivative in the fully expanded formulation of the viscous fluxes have been calculated via a standard sixth-order central scheme and in particular both the first and the second derivative approximation have been employed. Thus, being $\partial\phi/\partial x$ and $\partial^2\phi/\partial x^2$ respectively a first and a second derivative for the flow variable ϕ and $S = \{i - L/2, \dots, i + L/2\}$ the computational stencil ($L = 3$) around the ith computational node, the scheme reads

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{1}{\Delta x}\sum_{s=1}^{L/2} \alpha_s^{(1)}(\phi_{i+s} - \phi_{i-s}) + \mathcal{O}(\Delta x^L) \tag{5.11}$$

$$\left(\frac{\partial^2 \phi}{\partial^2 x}\right)_i = \frac{1}{\Delta x^2}\sum_{s=-L/2}^{L/2} \alpha_s^{(2)}\phi_{i+s} + \mathcal{O}(\Delta x^L) \tag{5.12}$$

   here the coefficients $\alpha_s^{(1)}$ and $\alpha_s^{(2)}$ maximise the formal order of accuracy in the actual stencil, thus $a_s^{(1)} = \{1/60, -3/20, 3/4\}$ and $a_s^{(2)} = \{1/90, -3/20, 3/2, -49/18, 3/2, -3/20, 1/90\}$.

5.1.3 Numerical treatment on non-uniform grids

The method for the discretisation of viscous derivatives works in case of a uniform Cartesian grid. As we already mentioned in § 4, this could be very limit its applications. Like in the case of convective fluxes, also diffusive terms contain a discrete number of first derivatives. For their numerical treatment the reader is remanded to the description proposed in § 4.15 and here we will focus on second derivatives. Being $\partial^2\phi/\partial x^2$ the second derivative of a flow variable ϕ, if the physical space in which ϕ lives is described by a generalised coordinate $x(\xi)$, we can write

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\partial^2\phi}{\partial\xi^2}\left(\frac{\partial\xi}{\partial x}\right)^2 + \frac{\partial\phi}{\partial\xi}\left(\frac{\partial^2\xi}{\partial x^2}\right) \tag{5.13}$$

where $\partial\xi/\partial x$ and $\partial^2\xi/\partial x^2$ are respectively the inverse of the Jacobian and the inverse of the Hessian of the coordinate transformation from the physical to the computational space. Thanks to differential calculus identities, we can express them in respect of $x(\xi)$ derivative as follows

$$\frac{\partial\xi}{\partial x} = \left(\frac{\partial x}{\partial\xi}\right)^{-1} \tag{5.14}$$

$$\frac{\partial^2\xi}{\partial x^2} = -\frac{\partial^2 x}{\partial^2\xi}\left(\frac{\partial x}{\partial\xi}\right)^{-3} \tag{5.15}$$

As for the convective fluxes calculation, the formulation (5.13) computes the ϕ gradients on a uniform grid (expressed by the ξ coordinate), while the extension to non-uniform grids can be done using the metrics of the function $x(\xi)$.

5.2 Validations and results

In this section, we will consider a set of valid benchmarks to test and validate the discretisation of the viscous terms of the Navier-Stokes equations. In particular, some analytical solutions, i.e., the Poiseuille flow, the first and second Stokes problems, will be pointed out and compared with the numerical solutions. A detailed analytical description of this problem can be found in [6] and in [61].

5.2.1 Poiseuille flow

The Poiseuille flow in an infinite planar channel is one of the most popular analytical solution of the Navier-Stokes equations. Considering a laminar flow in an incompressible regime, confined between two parallel isothermal walls at $T/T_0 = 1$. Being h the span of the channel in $y-$direction and x the stream-wise direction of the flow. For such a system, thanks to continuity equation, we can conclude that all the flow variables are uniquely functions of the $y-$coordinate. Moreover, if the flow reaches the stationary condition, the momentum Navier-Stokes equations reduce to

$$\frac{dp}{dx} = \mu_\infty\frac{d^2u}{dy^2} \tag{5.16}$$

where μ_∞ denotes

$$\mu_\infty = \frac{\sqrt{\gamma}M}{Re} \tag{5.17}$$

Now, ending the problem with the boundary conditions for the velocity at the wall position (i.e. $u(y=-h/2) = u(y=h/2) = 0$) and knowing that the pressure gradient does not depend on the $y-$coordinate, the equation (5.16) can be integrated obtaining

$$u(y) = -\frac{1}{2\mu_\infty}\frac{dp}{dx}(h^2 - y^2) \tag{5.18}$$

The formula expresses that the motion of the flow is driven by the pressure gradient dp/dx, which represents the free parameter of the problem. Because the pressure gradient can assume any value it is convenient to express it as a function of the non-dimensional groups and specifically as a function of Mach number. In this way, setting a very low Mach number, we can ensure the incompressibility of the flow resulting in the hypothesis of the Poiseuille flow. Thus defining

$$\frac{dp}{dx} = -\frac{2\gamma M_\infty^2}{Re} \tag{5.19}$$

we obtain

$$\boxed{u(y) = \sqrt{\gamma} M_\infty (h^2 - y^2)} \tag{5.20}$$

The latter expression represents the analytical solution of the Poiseuille flow as a function of the free-stream Mach number. The expression could be suitably compared with our computations. As far as the energy equation, reducing the energy equation accordingly with the Poiseuille flow hypothesis we obtain

$$\lambda_\infty \left(\frac{d^2 T}{dy^2} \right) + \mu_\infty \frac{d}{dy} \left(\frac{du}{dy} u \right) = 0 \tag{5.21}$$

where λ_∞ denotes

$$\lambda_\infty = \frac{\gamma}{\gamma - 1} \frac{\mu_\infty}{Pr} \tag{5.22}$$

In the limit of incompressible flows, the temperature field is fully decoupled by the velocity and $u = u(y)$ is a known function derived from the integration of momentum equation. Integrating the equation (5.21) we can write

$$\frac{\partial T}{\partial y} = -\frac{\mu_\infty}{\lambda_\infty} \int \frac{d}{dy} \left(\frac{du}{dy} u \right) dy + c_1. \tag{5.23}$$

For the fundamental theorem of calculus

$$\frac{dT}{dy} = -\frac{\mu_\infty}{\lambda_\infty} \left(\frac{du}{dy} u \right) + c_1 \tag{5.24}$$

and integrating again, we obtain

$$T(y) = -\frac{\mu_\infty}{\lambda_\infty} \int \left(\frac{du}{dy} u \right) + c_1 y + c_2. \tag{5.25}$$

Now, exploiting the integration by parts we can write

$$\int \left(\frac{du}{dy} u \right) = u(y)^2 - \int \left(u \frac{du}{dy} \right) \tag{5.26}$$

obtaining

$$\int \left(\frac{du}{dy} u \right) = \frac{u(y)^2}{2} \tag{5.27}$$

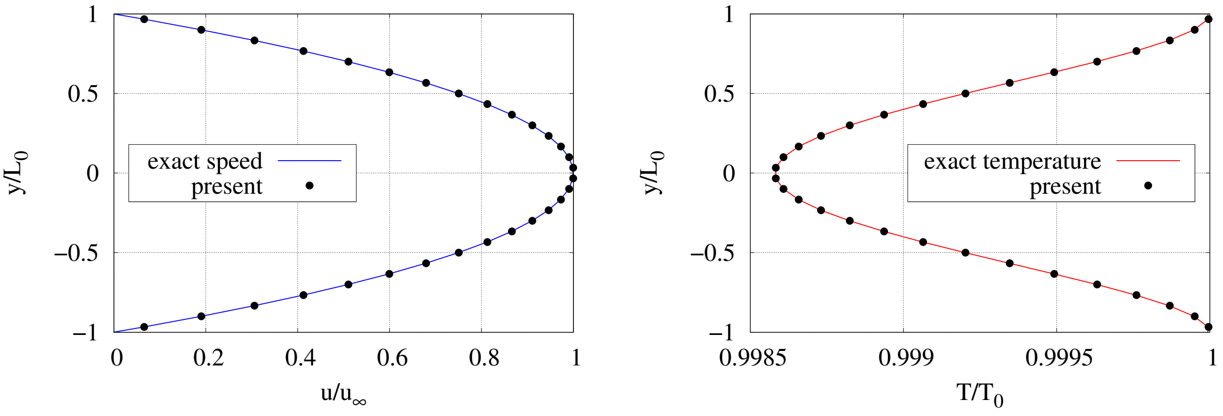
equation that, plugged in (5.25), gives

$$T(y) = -\frac{\mu_\infty}{\lambda_\infty} \frac{u(y)^2}{2} + c_1 y + c_2 \tag{5.28}$$

Finally, applying the boundary conditions $T(h/2)/T_0 = T(-h/2)/T_0 = 1$ and remembering $u(-h/2) = u(h/2) = 0$ we obtain

$$\boxed{T(y) = 1 - \frac{\gamma - 1}{\gamma} Pr \frac{u(y)^2}{2}} \tag{5.29}$$

equation that represents the temperature profile for a Poiseuille flow in an infinite planar channel. In order to reproduce the problem numerically without introducing any compressibility effects we simulated the flow in a periodic channel at $Re = u_\infty h/\nu_\infty = 10$ and $M = u_\infty/c_\infty = 0.1$. u_∞ denoted the pick velocity of the Poiseuille flow; h is the span of the channel and ν_∞ the kinematic viscosity of the fluid. Periodic boundary conditions were enforced along the streamwise direction; the pressure gradient dp/dx was employed as a source term to force the motion. Isothermal no-slip wall boundaries were employed at the upper and lower bounds of the computational domain. A uniform Cartesian mesh featuring $N_x \times N_y = 5 \times 30$ grid points for a domain of $L_x \times L_y = (-2:2) \times (-1:1)$ was used. The solution evolved reaching a steady-state condition. The time-step was evaluated via the CFL condition setting a CFL number equal to 0.5. The results, in term of velocity and temperature profiles are reported in Figure 5.1. From a comparison between our computation and the analytical solutions (5.20) (5.29), the likelihood of the outcome significantly fits.



(a) Velocity profile

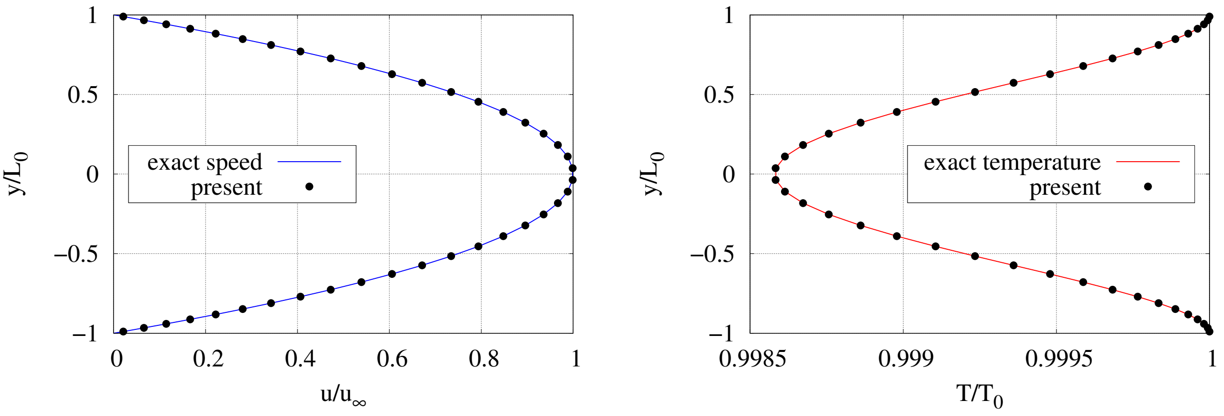
(b) Temperature profile

Figure 5.1: Velocity (a) and temperature (b) profiles for a Poiseuille flow at $M = 0.1$ and $Re = 10$.

To validate the algorithm in the case of non-uniform meshes, the same simulation was performed over a Cartesian grid stretched around the two walls locations and featuring $N_x \times N_y = 5 \times 40$ grid points. The latter was distributed employing the mapping function proposed by Orlandi (2012)

$$x(\xi) = \frac{1}{2} \frac{\tanh\left(\alpha\left(\xi - \frac{1}{2}\right)\right)}{\tanh\left(\frac{1}{2}\alpha\right)} \tag{5.30}$$

where α denotes a stretching parameter, in our computation set to 2.5. The results, in term of velocity and temperature profiles, are reported in Figure 5.2.



(a) Velocity profile

(b) Temperature profile

Figure 5.2: Velocity (a) and temperature (b) profiles on a non-uniform Cartesian grid for a Poiseuille flow at $M = 0.1$ and $Re = 10$.

5.2.2 First Stokes problem

Moving to a more complex validation, in the present paragraphs we want to face the so-called *first Stokes problem*. The problem consists of a semi-infinite flow region bounded by a rigid plate at $y = 0$ and initially at rest. Suddenly, the lower plate gains a steady velocity u_∞. Because of the viscosity of the fluid, the velocity perturbation starts to propagate in the flow domain and, in the limit of an incompressible flows, the u speed depends only on the wall-normal coordinate [6, 61]. The governing differential equations reduce to

$$\frac{\partial u}{\partial t} = \mu_\infty \frac{\partial^2 u}{\partial y^2} \tag{5.31}$$

with the boundary conditions

$$\begin{cases} u(y,0) = 0 \\ u(0,t) = u_\infty, \quad \text{for} \quad t > 0 \end{cases} \tag{5.32}$$

The analytical solution of the problem is represented by

$$\boxed{u(y,t) = u_\infty \left(1 - \text{erf}\left(\frac{y}{\sqrt{4\mu_\infty}}\right)\right)} \tag{5.33}$$

where erf(x) denotes the error function expressed by

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{5.34}$$

To reproduce the problem numerically, we simulated the flow in a semi-infinite configuration bounded by an isothermal no-slip wall condition at the lower side. Periodic boundary conditions were enforced on the left and the right sides of the domain, while an extrapolation condition was set at the upper bound. The Mach number $M = u_\infty/c_\infty$, to avoid any compressibility effect, was set equal to 0.1 and here u_∞ represents the wall speed. The problem was solved referred to a non-uniform Cartesian grid expressed by

$$x(\xi) = 1.0 + \frac{tanh(\alpha(\xi - 1.0))}{tanh(\alpha)} \tag{5.35}$$

ensuring a grid stretching along the wall-normal direction. Here α is the stretching parameter in our computation set to 3.0. The grid featured $N_x \times N_y = 5 \times 30$ computational nodes and covered a domain of $L_x \times L_y = 1 \times 20$ length units. The time-step was computed setting the CFL number equal to 0.5, and the flow variables have been evolved up to $t/t_0 = 600$ units of time. In Figure 5.3, the comparison between our simulation and the analytical solution, at different time levels, is reported. As we notice, the velocity perturbation, due to the wall speed, starts to propagates from the lower to the upper bounds of the domain fitting the analytical solution. In Figure 5.4, to distinguish the point clustering at the wall location an enlargement of that region has been plotted.

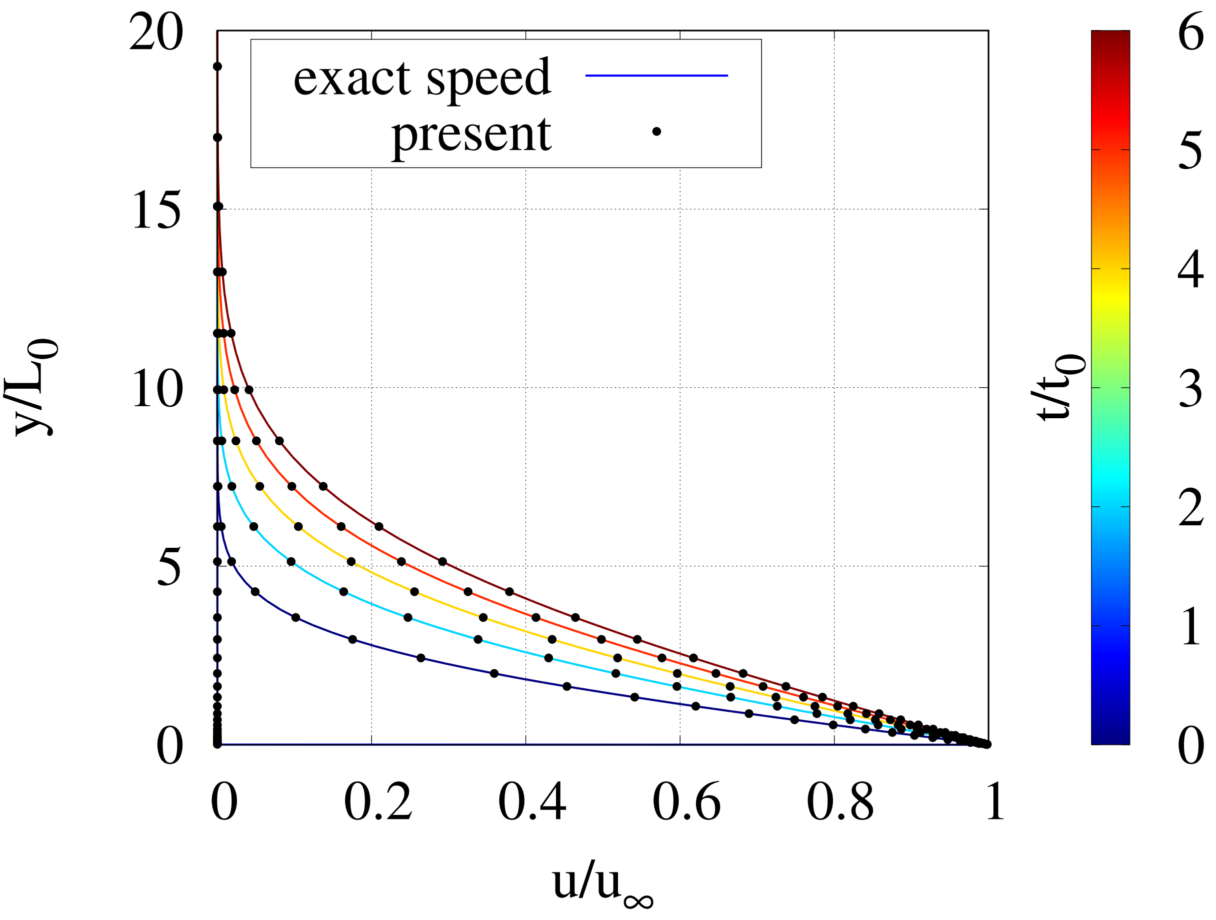


Figure 5.3: Velocity profiles in scale for the *first Stokes problem* at $M = 0.1$. The computation was performed over a non-uniform Cartesian grid stretched along the wall normal direction.

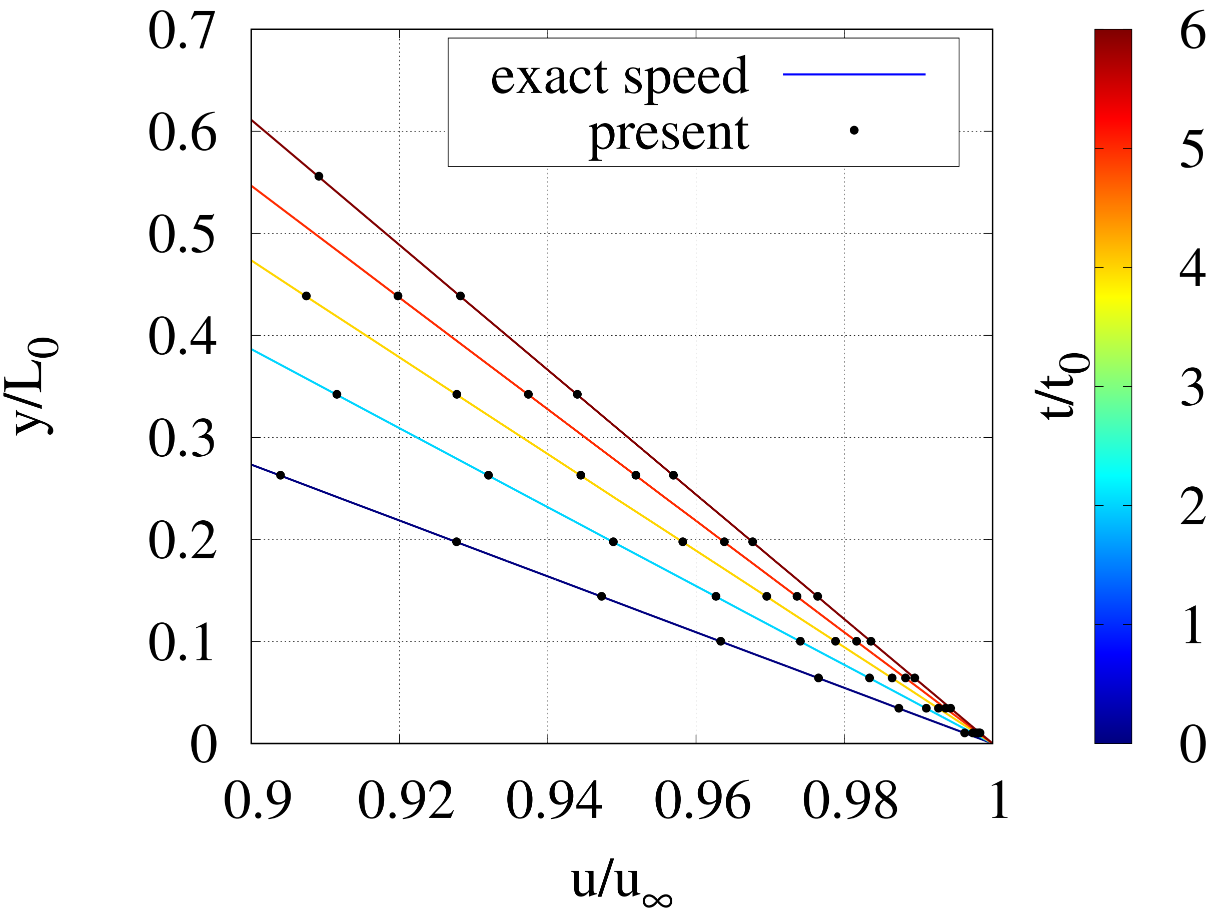


Figure 5.4: Enlargement of the wall region of the velocity profiles in for the *first Stokes problem* at $M = 0.1$.

5.2.3 Second Stokes problem

The last test we analyse in the validation of the viscous fluxes discretisation is the flow triggered by an oscillating plane boundary. The problem, also known as *second Stokes problem*, represents an extension of the problem discussed in the previous paragraph. We suppose the upper half of the $(x, y)-$plane to be occupied by a fluid, while a rigid boundary at $y = 0$ is moving with speed $u_\infty \cos(\omega t)$. The flow, initially at rest, starts to follow the wall speed, inducing the motion of all the domain. In the limit of incompressible flows, the problem is governed by the equation (5.31) and, following the description of Batchelor (2000), the analytical solution of the problem is represented by

$$\boxed{u(y,t) = u_\infty e^{-\omega_c y} \cos(\omega t - \omega_c y)} \tag{5.36}$$

here ω is the wall wave number and ω_c is a reduce frequency expressed by

$$\omega_c = \sqrt{\frac{\omega}{2\mu_\infty}} \tag{5.37}$$

Reproducing the problem we set $\omega = 1/4\pi$ and we simulated a semi-infinite configuration of a flow at $M = u_\infty/c_\infty = 0.01$. u_∞ denotes the pick wall's velocity. Periodic boundaries were enforced on the lateral edges, while at the top the conservative variables have been extrapolated.

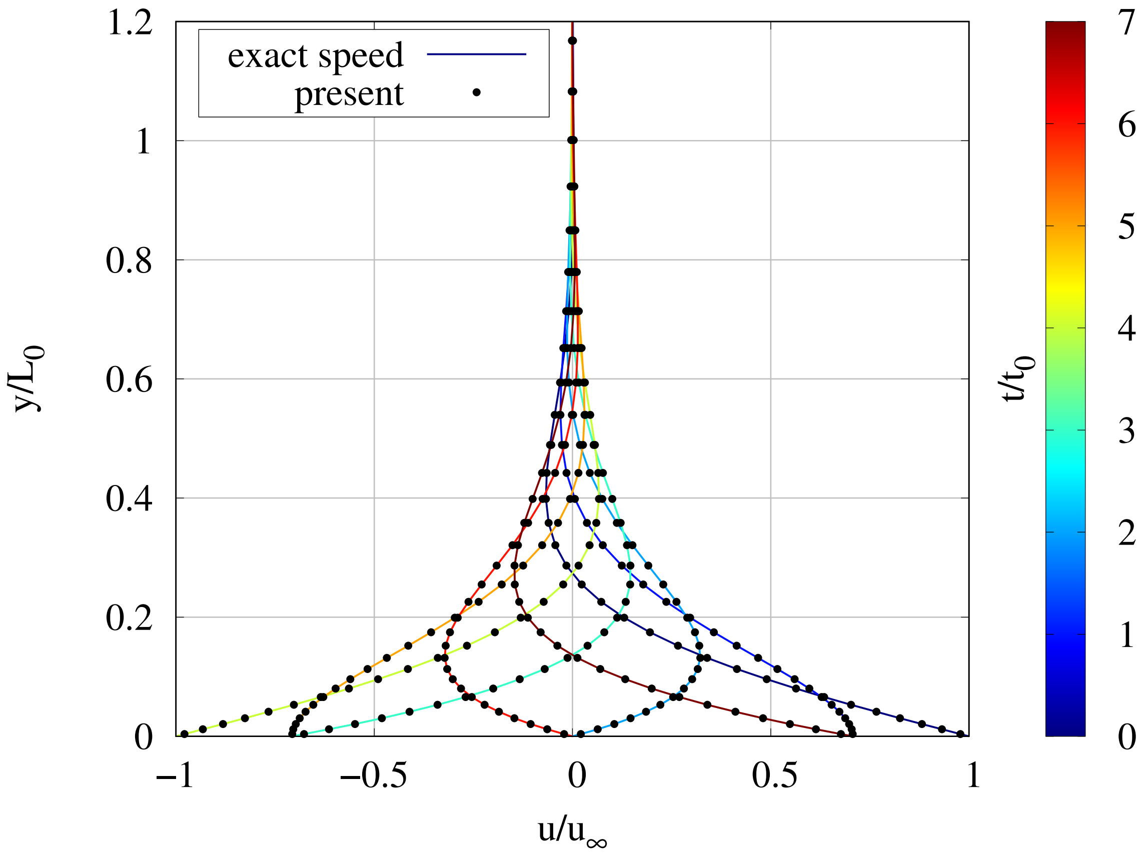


Figure 5.5: Seven velocity profiles for the *second Stokes problem* at $M = 0.01$ and $Re = 1$. The computation was performed over a non-uniform Cartesian grid stretched along the wall normal direction.

The computation was performed over a non-uniform Cartesian grid featuring $N_x \times N_y = 5 \times 40$ computational nodes and covering a domain of $L_x \times L_y = (-1 : 1) \times 2$ units of length. The nodes were analytically distributed by the expression (5.35) and the stretching parameter α was set equal to 2.0. In Figure 5.5, the comparison between our numerical results and the analytical solution are compared for seven different levels of time concluding a full fitting. Similarly to the previous test, an enlargement of the wall region is reported in Figure 5.6 where the velocity profiles have been plotted in logarithmic scale.

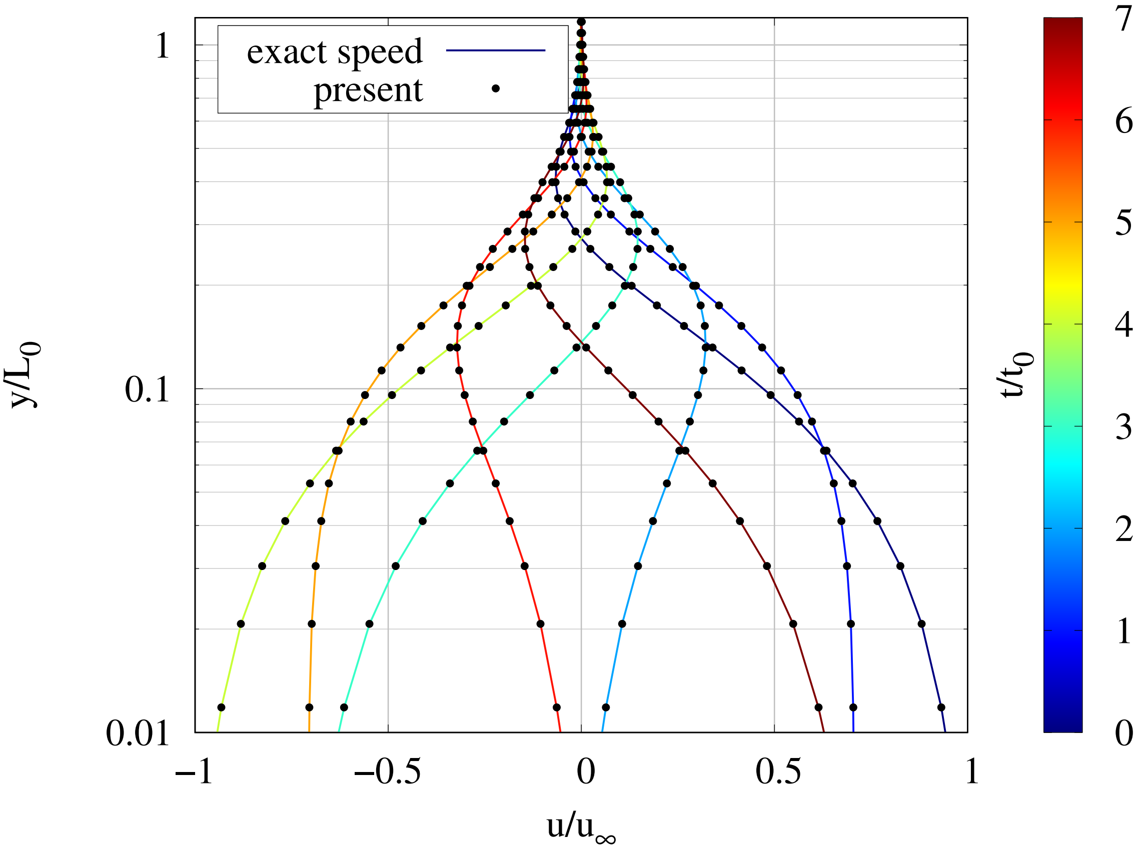


Figure 5.6: Seven velocity profiles for the *second Stokes problem* at $M = 0.01$ and $Re = 1$. Enlargement of the wall region of the velocity profiles.

5.3 Summary

In the present chapter, the numerical methods employed in the discretisation of the viscous fluxes of the Navier-Stokes system of equations have been presented. In particular, a standard sixth-order central scheme has been used in combination with a suitable fully split formulation of the diffusive terms. This formulation is strongly recommended in the case of compressible flows, increasing the spectral properties of the central scheme and its stability. Three tests, of increasing difficulty, have been presented. The latter consisted of well-known analytical solutions of the Navier-Stokes equations in the limit of incompressible flows. A full agreement between the numerical and the analytical solution make us confident in the excellent predictability of our method.

6

Code parallelisation and 3D results

Turbulence is the most important unsolved problem of classical physics.

Richard Feynman

Contents

In the previous chapters, we described the numerical tools implemented in the discretisation of the Navier-Stokes system of equations. Those routines took into account the single contributions of the equations in terms of temporal, convective, and viscous terms. In the present chapter, we would like to show an overall description of our solver, mainly focused on its parallel structure. After this presentation, some validations and comparative results in the field of turbulent and fully separated flows will be pointed out.

6.1 Parallelisation strategy

In the last decades, because of their accurate behaviour, the finite-difference methods have been widely applied to simulate a large variety of fluid dynamics problems. To increase the accuracy of the numerical results, a considerable number of discretisation schemes have been proposed. Thanks to the research activity in this field, nowadays, the direct numerical simulations of fluid flow phenomena have reached a fidelity level comparable to the most accurate and expensive experiments, becoming a real alternative to them. On the other hand, numerical analysis stands that, to increase the accuracy of a discretisation method, the number of information required must increase. This fact makes high-order numerical schemes very computational demanding in terms of the number of operations per time-step and memory requirements. Thus, some strategy to reduce the simulation's time must be employed.

Today, parallel programming and the implementation of parallel algorithms over distributed memory architectures, seem to be the most reliable strategies able to face the problem. In the present work, all the algorithms presented in the previous chapters have been implemented in a distributed memory parallel approach. In particular, our numerical application makes use of a hybrid implementation consisting of employing both the MPI and the OpenMP specifications. MPI - an acronym standing for *message passing interface* - consists in a widely used library for solving difficult computational problems in all the significant fields of science and engineering. The interface makes available several routines to convert a *serial* application into a parallel one. The library - now at its fourth version - completely supports both FORTRAN90 and C/C++. The fundamental idea of MPI consists in splitting the computation over thousands of cores, performing a small number of operations per computational unit. Every single core must be aware of what the other cores are running (or at least it must know what its *neighbours* are computing). The MPI specification manages the information interchange between them. An overall description of the MPI library can be found in [46, 47] while a dissertation, closed to fluid dynamics problems, can be found in [27].

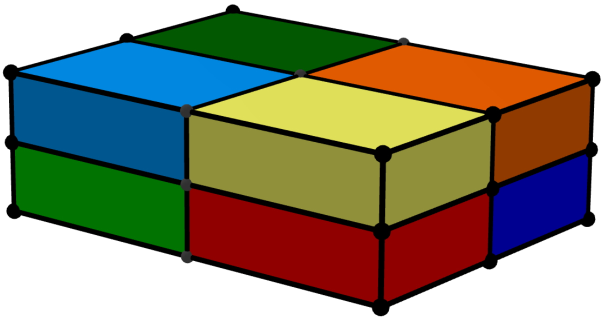


Figure 6.1: Sketch of the 3D MPI domain decomposition employed in the parallelisation splitting

On the other hand, OpenMP consists in an alternative parallelisation paradigm suitable for shared memory architectures. In this case, the processed data must be available on the same memory slot, making this specification quite restrictive for massive computations. In this way the hybridisation of the two paradigms takes advantage of both of them. In a standard distributed memory architecture, the MPI paradigm can be employed for *extra-nodes* computations, splitting the calculus over a considerable number of nodes. The *intra-nodes* (as known as threads) computations, instead, relying on the same memory slot, can be speed-up with the OpenMP paradigm. The latter consists in compiler directives able to distribute the computational load over the threads.

Following the suggestions of Chao et al. (2009) and Salvadore et al. (2013) a fully three-dimensional MPI domain decomposition hybridised with OpenMP compiler directives has been implemented. In particular, as far as the MPI implementation the domain decomposition shares the halo-nodes thought non-blocking communications. The latter process has been efficiently addressed thought point-to-point directives (i.e. `MPI_ISEND/MPI_IRECV`) and has been partially desynchronised concerning the computational kernels. MPI derived data type have been used to collect the shared data and the conservative variables were stored in four-dimensional arrays $\phi(i,j,k,n)$, where $\phi(i,j,k,1:5) = \{\rho, \rho u, \rho v, \rho w, \rho E_{tot}\}^T_{i,j,k}$. The implementation has been tested in terms of scalability performances. In particular, both *strong*- and *weak-scaling* tests have been performed. These two attributes indicate how efficient is an application increasing the number of processing units.

6.1.1 Strong scaling

The strong scaling consists in keeping fixed the size of the problem but increasing the number of cores employed in its computation. This scalability property is used as justification for those models that take a long time to run and a possible way to measure it consists in computing the *scaling factor* ($\mathcal{SF}$)

$$\mathcal{SF} = \left(\frac{t_{ref}}{t_N}\right)_S \tag{6.1}$$

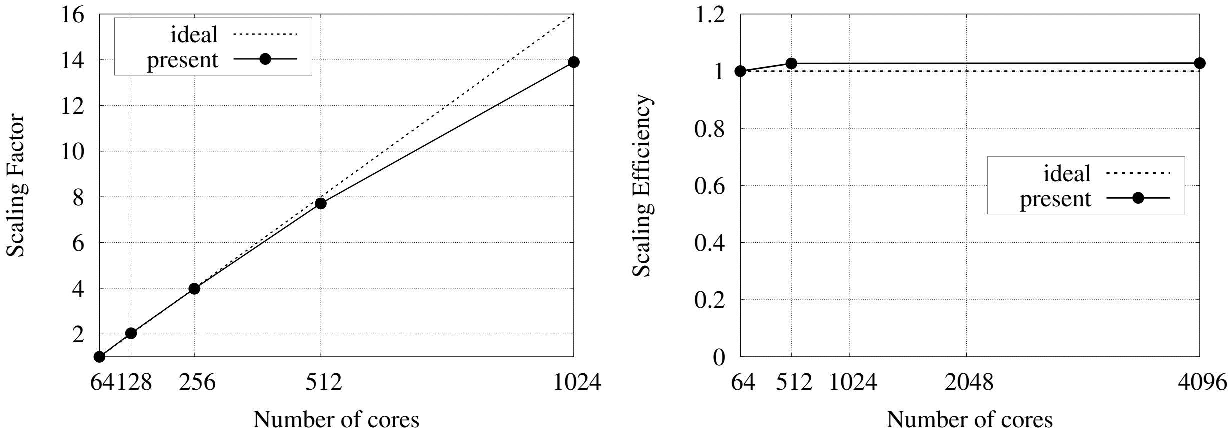
$\mathcal{SF}$ represents the ratio between the elapsed time to complete a job with a reference group of processing elements (t_{ref}) and the elapsed time to complete the same task over N reference groups (t_N) keeping the problem's size (S) fixed. In principle, t_N should be equal to $N \cdot t_{ref}$, thus $\mathcal{SF}$ is theoretically a linear function of the number of processing units. Unfortunately, in real applications, increasing the number of cores, the time to share the information progressively increases, saturating the $\mathcal{SF}$. The scaling factor was computed exploiting the MARCONI supercomputer infrastructure from CINECA, Italy. The test consisted in evolving 10 Navier-Stokes's iterations of an ambient field in a cubic domain. The latter was discretised with a uniform Cartesian grid featuring 512 × 512 × 512 grid points and the test was performed up to 1024 physical cores. To avoid any speed-up due to multi-threading, the number of threads was kept constant during the entire campaign and fixed equal to one. The reference group of precessing elements was selected equal to 64 cores which corresponds to the number of cores in a single MARCONI's node. In Figure 6.2a the $\mathcal{SF}$ is reported as a function of the cores' number.

6.1.2 Weak scaling

The weak scaling, similarly to the strong scaling, gives information about the parallel performances of a numerical application while the number of cores employed in the computation increases. Differently, from the strong scaling, the weak scaling fixed the problem inside a single computational group. This type of attribute is taken as a justification for those programs that consume a lot of memory (or other system resources) during their running and a way to quantify it consists in computing the *scaling efficiency* ($\mathcal{SH}$) defined as

$$\mathcal{SH} = \left(\frac{t_{ref}}{t_N}\right)_{S\times G} \tag{6.2}$$

$\mathcal{SH}$ represents the ratio between the elapsed time to complete a job over a reference group of processing elements and the elapsed time to complete the same task over N reference groups keeping constant the problem's size per each processing group. Fixing the problem's size per each processing unit, the number of floating-point operations per single cores remains the same, thus the scaling efficiency $\mathcal{SH}$ should be independent on the number of processing units and equal to one. In real applications, increasing the size of the problem per single core the memory latency, the miscaches and other system occurrences reduce the performances of the application, affecting the scaling efficiency. In the present work, the scaling efficiency was computed always exploiting the MARCONI's infrastructure. In particular, 100 Navier-Stokes' iterations of an ambient field have been performed over a cubic domain. The latter was discretised by three uniform Cartesian grids featuring $\{32^3, 64^3, 128^3\}^T$ computational points and the computations were respectively run over $64\{1, 8, 64\}^T$ cores. This setting kept the number of grid nodes per processor constant and equal to 512. In Figure 6.2b the $\mathcal{SH}$ is reported as a function of the cores' number. The latter results allows to conclude that our application is well optimised in term of memory accesses and cache exploitation.



(a) Strong scaling

(b) Weak scaling

Figure 6.2: (a) Strong scaling of the numerical method up to 1024 cores in a domain of 512^3 grid points. (b) Weak performance for the numerical application keeping 512 grid point per core. The test evolved the *ambient initial conditions* for the Navier-Stokes system of equation consisting in $(u_i/u_\infty)_{t=0} = 0$ and $(p/p_0)_{t=0} = 1$ and $(T/T_0)_{t=0} = 1$.

	# procs	elapsed time [s]	scaling factor
Strong scaling (512^3 nodes)	2^5	78.236	1.00
	2^6	38.624	2.03
	2^7	19.660	3.98
	2^8	10.142	7.71
	2^9	5.628	13.9
	# procs	elapsed time [s]	scaling efficiency
Weak scaling (512 nodes/procs)	64^1	2.187	1.000
	64^2	2.130	1.027
	64^3	2.128	1.028

Table 6.1: Strong and weak scaling performances for the numerical method.

Thanks to the good parallel performance of our numerical method, three-dimensional simulations become a possibility. In the following section, we are going to show a fully three-dimensional test in the field of turbulent and fully separated flows. The section concludes the validation campaign of our solver.

6.2 Three-dimensional results

6.2.1 Wall turbulence

Since the dawn of the fluid dynamics research, the wall turbulence has been represented as one of the most studied phenomena in the field of turbulent and fully separated flows. Back to the fundamental works of Prandtl (1904), Blasius (1908) and Von Karman (1937) nowadays many approaches have been proposed, both numerical [8, 78, 79] and experimental [115], facing the problem in various applicative configurations. The most common arrangements are the turbulent pipe and the turbulent channel. These two dispositions show similar results in terms of flow statistics and physical deployment of turbulence and recently Luchini (2017) proved the universality of the turbulent statistics in both the two geometries. The current paragraph shows the behaviour of our computational model in respect of three-dimensional fully-separated flows and in particular, because of the large variety of results available, here a planar turbulent channel will be taken into account. The problem will we be faced in the limit of incompressible flow. At first sight, this choice could be seem far from being a good test for a compressible solver, indeed it consists in a real challenging application. The employment of a co-located finite-difference grid makes the decoupling between the pressure and the velocity fields a frequent eventuality, especially in incompressible turbulent conditions and not rarely the blow-up of the simulation follows; thus, the turbulent channel flow in incompressible regime can be considered a test also for the robustness of the code. As usual, before moving thought the results, we will start with a theoretical introduction for whose details the reader is remanded to [61].

6.2.2 Pressure-driven turbulent channel

In the field of turbulent flows, a well-known result is represented by the *Reynolds averaged Navier-Stokes equations* (RANS). The model describes the dynamics of a turbulent flow in terms of its mean behaviour over a considerable amount of observations. Applied to incompressible flows the equations read

$$\frac{\partial \langle u_j \rangle}{\partial x_j} = 0 \tag{6.3a}$$

$$\frac{\partial \langle u_i \rangle}{\partial t} + \langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \langle p \rangle \delta_{ij}}{\partial x_j} + \frac{\mu}{\rho}\frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} - \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j} \tag{6.3b}$$

where $\langle \phi \rangle$ represents the *Reynolds average* of a flow variable $\phi(t)$ expressed by

$$\langle \phi \rangle = \lim_{t\to\infty} \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} \phi(t^*) dt^* \tag{6.4}$$

and ϕ' represents the turbulent fluctuation of ϕ expressed by

$$\phi' = \phi - \langle \phi \rangle \tag{6.5}$$

Our goal is to simplify the equations (6.3) in the hypothesis of channel flow. Considering a flow, bounded by two planar walls, in a Cartesian reference frame and being x, y and z respectively the stream-wise, the wall-normal and the span-wise coordinates of the channel. In this reference the two walls are located at $y = \pm\delta$. Here we supposed the stream-wise and the span-wise dimensions of the system much higher in respect of span of the channel; thus $(L_x/2\delta), (L_z/2\delta) \to \infty$ and we assume the mean flow to be stationary. In this hypothesis, the temporal variations and the stream/span-wise spatial derivatives of the mean flow can be neglected

$$\frac{\partial\langle\phi\rangle}{\partial t} = 0 \tag{6.6}$$

$$\frac{\partial\langle\phi\rangle}{\partial x} = \frac{\partial\langle\phi\rangle}{\partial z} = 0. \tag{6.7}$$

Because of the planarity of the system also $\langle w\rangle = 0$. With these assumptions, the mass conservation for the mean flow reduces to

$$\frac{\partial\langle v\rangle}{\partial y} = 0. \tag{6.8}$$

The above formula, in combination with the boundary conditions at the wall location, makes us able to conclude that $\langle v\rangle = 0$; the momentum conservation equations consequently reduces to

$$-\frac{1}{\rho}\frac{\partial\langle p\rangle}{\partial x} + \frac{\mu}{\rho}\frac{\partial^2\langle u\rangle}{\partial y^2} - \frac{\partial\langle u'v'\rangle}{\partial y} = 0 \tag{6.9a}$$

$$-\frac{1}{\rho}\frac{\partial\langle p\rangle}{\partial y} - \frac{\partial\langle v'^2\rangle}{\partial y} = 0 \tag{6.9b}$$

Let us analyse the equation (6.9b). The latter allows us to conclude that the group

$$\langle p\rangle + \rho\langle v'\rangle = f(x) \tag{6.10}$$

depends just on x; moreover, computing its derivative in respect on the independent variable and evaluating it at the wall location we obtain

$$\frac{df(x)}{dx} = \frac{d}{dx}\left(\langle p\rangle + \rho\langle v'^2\rangle\right)_{y=\pm\delta} = \frac{d\langle p\rangle}{dx} \tag{6.11}$$

The latter equation makes us able to conclude that the mean pressure gradient depends just on x. Thus, inserting this result in the equation (6.9a) we can write

$$\frac{d\langle p\rangle}{dx} = \frac{\partial}{\partial y}\left(\mu\frac{\partial\langle u\rangle}{\partial y} - \rho\langle u'v'\rangle\right) \tag{6.12}$$

Here we notice that the term inside the brackets is dimensionally a shear stress so

$$\frac{d\langle p\rangle}{dx} = \frac{d\tau(y)}{dy} \tag{6.13}$$

The equation (6.13) represents a simple ordinary differential model linking the pressure gradient and the wall-normal shear stress. Integrating it in respect of y we obtain

$$\tau(y) = \frac{d\langle p \rangle}{dx} y + c_1 \tag{6.14}$$

The constants c_1 can be easily determined knowing that, for the symmetry of the problem, $\tau(y = 0)$, so $c_1 = 0$

$$\tau(y) = -\frac{d\langle p \rangle}{dx} \frac{|y|}{\delta} \tag{6.15}$$

This fundamental relation establishes a strong relation between the pressure gradient and the wall shear stress, in fact, computing it at the wall location

$$\frac{\tau_w}{\delta} = -\frac{d\langle p \rangle}{dx} \tag{6.16}$$

we conclude that the mean pressure gradient represents the forcing term that acts in the flow motion, balancing, in term of the mean flow dynamics, the wall shear stress. To reproduce numerically a channel flow, the mean pressure gradient, or equivalently the mean wall shear stress, must be imposed. In the present work, the first choice was employed. Here it is not inessential to be said that the pressure gradient can assume any value and the right choice consists in expressing it in term of non-dimensional reference quantities. So, introducing the bulk velocity as the volume integral of the $u-$momentum divided by the volume integral of the density[1]

$$u_b = \frac{\int_V \rho u \, dV}{\int_V \rho \, dV} \tag{6.17}$$

we can express the mean pressure gradient as a function of the deficit between a reference speed u_∞ and the bulk speed

$$\frac{d\langle p \rangle}{dx} = u_\infty - u_b = \mathscr{F} \tag{6.18}$$

expression that defines suitable forcing terms for the Navier-Stokes system of equations. Hence, defining

$$\mathbf{S}^T = \{0, \rho\mathscr{F}, 0, 0, \rho u\mathscr{F}\}^T \tag{6.19}$$

and adding it to right-hand-side of the Navier-Stokes equations (2.66) we obtain that the bulk speed of the channel fits dynamically the reference speed u_∞. With this strategy, periodic boundary conditions can be applied along both the stream and the span-wise directions avoiding any synthetic turbulence injection at the inflow position. Finally, the reference speed u_∞ can be expressed as a function of the Mach number, defining as usual, $u_\infty = \sqrt{\gamma} M_\infty$ while a suitable choice for the Reynolds number consists in the *Reynolds bulk* defined as

$$Re_b = \frac{2u_b\delta}{\nu_\infty} \tag{6.20}$$

u_b denotes the bulk speed of the flow, 2δ is the wall-normal dimension of the channel and ν_∞ is the kinematic viscosity of the fluid. It is well known that the Reynolds bulk number must be sufficiently large for the channel to be turbulent.

[1] Here the procedure will be explained in respect of a compressible solver, the following relations simplify consequently if the density is constant.

6.2.3 The law of the wall

From the last analysis, we conclude that the pressure gradient drives a turbulent channel, and the Reynolds bulk represents a suitable parameter in the definition of its flow regime. However, because of the relation between the mean pressure gradient and the wall shear stress, the problem exhibits a second (equivalent) non-dimensional formulation. The latter is based on shear quantities at wall location. From the dimensional parameters ρ, ν, δ, τ_w and y another common choice in term reference quantities is the one that employs the definition of the *shear velocity*

$$u_\tau = \sqrt{\frac{\tau_w}{\rho_w}} \tag{6.21}$$

and the *viscous length*

$$\delta_\nu = \frac{\nu_w}{u_\tau} \tag{6.22}$$

The ratio δ/δ_ν

$$Re_\tau = \frac{u_\tau \delta}{\nu_w} \tag{6.23}$$

defines the shear Reynolds number that again must be sufficiently large (in respect of the unity) for a turbulent flow. In addition to the previous results, the definition of these new references makes us able to perform some further analysis. Theoretical considerations (see for instance [75, 98]) suggest that, in the limit of $Re_\tau \to \infty$ and for $y/\delta_\nu \to 0$, the near-wall velocity (*inner layer*) does not depend on Re_τ but it is described by a single variable function

$$\langle u(y) \rangle \sim u_\tau f\left(\frac{y}{\delta_\nu}\right), \quad \frac{y}{\delta_\nu} \to 0 \tag{6.24}$$

On the other hand, for $y/\delta_\nu \to \infty$, the velocity profile becomes independent from the fluid's viscosity, continuing to be independent also on Re_τ. In this region (*outer layer*) the velocity can be expressed as a function of the non-dimensional coordinate y/δ

$$\langle u(y) \rangle \sim u_\tau g\left(\frac{y}{\delta}\right), \quad \frac{y}{\delta_\nu} \to \infty \tag{6.25}$$

Now, deriving the expressions (6.24) and (6.25) we obtain

$$\frac{d\langle u(y) \rangle}{dy} \sim \frac{u_\tau}{y} f'\left(\frac{y}{\delta_\nu}\right) \tag{6.26}$$

$$\frac{d\langle u(y) \rangle}{dy} \sim \frac{u_\tau}{y} g'\left(\frac{y}{\delta}\right) \tag{6.27}$$

Surely, some asymptotic compatibility relations must be recovered for the last two derivatives, and in the limit for $y/\delta \to 0$ and $y/\delta_\nu \to \infty$ their value must be equal to a same constant κ^{-1}. Thus we can write

$$\frac{d\langle u(y) \rangle}{dy} \sim \kappa^{-1} \frac{u_\tau}{y} \quad \text{for} \quad \frac{y}{\delta_\nu} \to \infty \tag{6.28}$$

Integrating the last relation and introducing the wall speed $u^+ = \langle u \rangle / u_\tau$ and the wall unit $y^+ = y/\delta_\nu$ we obtain

$$u^+(y^+) \sim \kappa^{-1}\ln(y^+) + \mathcal{B} \quad \text{for} \quad y^+ \to \infty \quad (\text{or} \quad \frac{y}{\delta} \to 0) \tag{6.29}$$

The latter result represents an asymptotic relation for the *outer layer* velocity profile. On the other hand, by the definition of the mean wall shear stress $(\mu d\langle u\rangle/dy)_{y=\pm\delta} = \tau_w$ we easily conclude that

$$\left(\frac{du^+}{dy^+}\right)_{y=\pm\delta} = \frac{\tau_w}{\mu}\frac{\delta_\nu}{u_\tau} = \frac{\rho u_\tau^2}{\mu}\frac{\delta_\nu}{u_\tau} = 1 \tag{6.30}$$

so, inserting in (6.24) we obtain

$$u^+(y^+) \sim y^+, \quad \text{for} \quad y^+ \to 0 \tag{6.31}$$

Finally, from an applicative point of view, the equations (6.31) and (6.29) approximatively define the mean near-wall streamwise velocity in term of wall unit as follows

$$u^+(y^+) = \begin{cases} y^+ & \text{for} \quad y^+ \lesssim 5 \\ \kappa^{-1}\ln(y^+) + \mathcal{B} & \text{for} \quad y^+ \gtrsim 30 \end{cases} \tag{6.32}$$

κ is the Von-Karman constant, and $\mathcal{B}$ is an experimental parameter tuning the logarithmic part of the equation. Experimental results and numerical simulations suggest $k \simeq 0.41$ while $\mathcal{B} \simeq 5$. The present theory was recently improved by Luchini (2017), who proposed a theoretical extension for the outer layer expression to increase the universality of the law of the wall. In particular, including the pressure gradient as a small contribution in the outer layer region the equation (6.28) becomes

$$\frac{d\langle u(y)\rangle}{dy} \sim \kappa^{-1}\frac{u_\tau}{y} - \frac{A_1}{\tau_w}\frac{dp}{dx} \quad \text{for} \quad \frac{y}{\delta_\nu} \to \infty \tag{6.33}$$

and integrating

$$u^+ = \kappa^{-1}\ln(y^+) + A_1 g Re_\tau^{-1} y^+ + \mathcal{B} \tag{6.34}$$

A_1 is a fitting parameter in the range of $A_1 \simeq 1$ and $g = -(dp/dx)\delta/\tau_w$, so, in the case of a pressure driven planar channel, it is equal to 1. As we can notice, the equation (6.34) is consistent with the classical theory in the case of $Re_\tau \to \infty$. Finally, the experimental results performed in the last century suggest a link between Re_b and Re_τ. The empirical relation (see Pope (2001)) expresses a well-fitting between these two parameters

$$Re_\tau \simeq 0.09 Re_b^{0.88} \tag{6.35}$$

6.2.4 Three-dimensional DNS results

Following the available literature [8, 78], we simulated a turbulent channel at $Re_b = 2u_b\delta/\nu_\infty = \{5790, 10074\}^T$, Reynolds bulk regime that correspond to shear Reynolds number of $Re_\tau = \{180, 300\}^T$. The Mach number, to avoid any compressible effects, was set to 0.1 thus the resulting driven pressure gradient was set equal to $d\langle p\rangle/dx = \sqrt{\gamma}M_\infty - u_b$. Periodic boundary conditions were enforced along the stream-wise and the span-wise directions, while no-slip isothermal wall conditions were employed at the top and the bottom side of the domain. The wall temperature was set equal to $T/T_0 = 1$. The domain was discretised by a uniform Cartesian grid featuring $N_x \times N_y \times N_z = 384 \times 128 \times 192$ grid points. The stream-wise, the wall-normal and the span-wise dimensions of the channel sized respectively $2\pi\delta \times 2\delta \times \pi\delta$. As far as the numerical method, no hybridisation strategies were employed in the discretisation of the convective fluxes, but the purely KGP6 scheme was used. To promote the turbulence transition, the initial conditions for the primitive variables has been imposed following the procedure proposed by Dan and John (1991). Thus at $t/t_0 = 0$, to the analytical solution of the Poiseuille flow, a vortex pair expressed by

$$\begin{cases} \psi = (1-y^2)^2 z e^{-16x^2-4z^2} \\ u = 0 \\ v = \psi_z \\ w = -\psi_y \end{cases} \tag{6.36}$$

has been superimposed. The solution was advanced in time until a statistical convergence of the flow variables has been reached. The time step was evaluated employing the Courant-Friedrichs-Lewy condition and setting a CFL number equal to 0.5. In Figure 6.3, the instantaneous iso-contours of the second invariant of velocity gradient are reported. The latter defined as

$$Q = \frac{1}{2}\left(||\Omega_{ij}||^2 - ||S_{ij}||^2\right) > 0 \tag{6.37}$$

consists of a quantitative definition of the three-dimensional regions characterised by coherent vortical structures. Here Ω_{ij} and S_{ij} are respectively the vorticity tensor and the strain-rate tensor, defined as

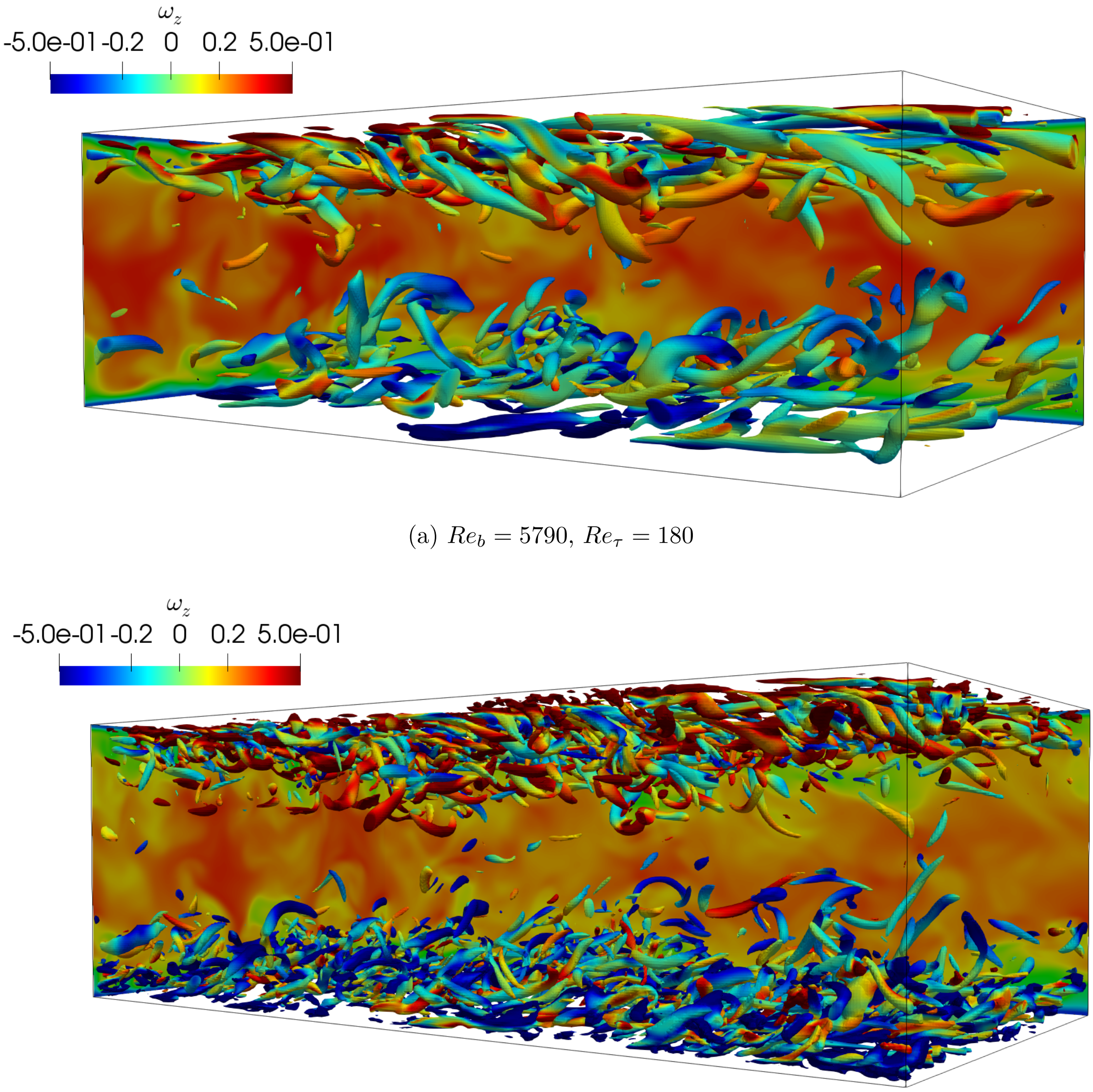
$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right), \quad i,j = 1,\dots,3 \tag{6.38}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i,j = 1,\dots,3 \tag{6.39}$$

A detailed description of vortices visualisation can be found in Haller (2005). From a qualitative point of view, the instantaneous results in Figure 6.3 greatly respect the turbulence theory. Here, we can hightlight the role of the Reynolds number in respect of the turbulence granularity and on the scale of the smallest vortices. Moving through quantitative results, in Figure 6.4 the mean stream-wise velocity profile is reported as a function of the wall distance. Both the two figure use the inner-scaling; thus, the velocity is scaled by the friction speed u_τ and the wall distance with the viscous length δ_ν. Our results are compared both with the one obtained by Bernardini et al. (2014) and Modesti and Pirozzoli (2016) and with the analytical behaviour predicted by Von Karman (1937) and Luchini (2017), concluding an excellent agreement. In Table 6.2 the flow parameters are reported comparing the target values vs the computed ones.

	Parameter	Computed value	Target value
$Re_\tau = 180$	M_b	$0.9996E-01$	$0.1000E+00$
	Re_b	$0.5788E+04$	$0.5790E+04$
	Re_τ	$0.1839E+03$	$0.1842E+03$
$Re_\tau = 300$	M_b	$0.9996E-01$	$0.1000E+00$
	Re_b	$0.1007E+05$	$0.1007E+05$
	Re_τ	$0.3035E+03$	$0.3000E+03$

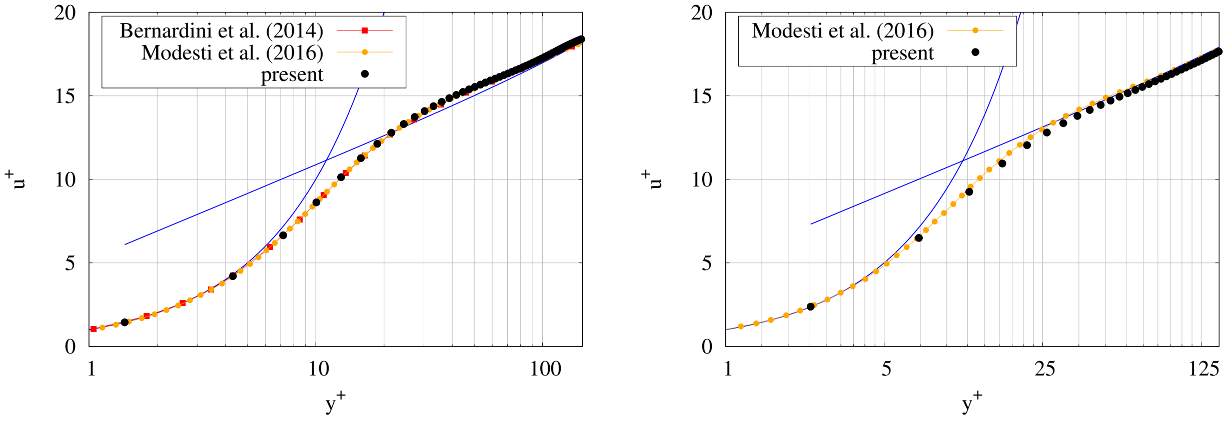
Table 6.2: Table of the flow parameters involved in the simulation of the turbulent channel problem. Here $M_b = u_b/c_b$ denoted the bulk Mach number, $Re_b = u_b\delta/\nu_\infty$ is the bulk Reynolds number and $Re_\tau = u_\tau\delta/\nu_w$ is the shear Reynolds number. The table shows a comparison between the target and the computed values.



(a) $Re_b = 5790$, $Re_\tau = 180$

(b) $Re_b = 10074$, $Re_\tau = 300$

Figure 6.3: Visualisation of the simulations of the turbulent channel at $Re_b = 5790$ and 10074. Here the $Q-$criterion shows the 1.5 iso-contours of the second invariant of the velocity-gradient. The color map refers to the non-dimensional z-vorticity component.



(a) $Re_b = 5790$, $Re_\tau = 180$

(b) $Re_b = 10074$, $Re_\tau = 300$

Figure 6.4: Stream wise velocity statistics for a channel flow at $Re_\tau = 180$ (a) and $Re_\tau = 300$. The results are compared with Bernardini et al. (2014) and Modesti and Pirozzoli (2016). The solid lines denotes the law of the wall expressed by $u^+ = 0.41^{-1}\ln(y^+) + Re_\tau^{-1} y^+ + 5.2$

6.2.5 Three-dimensional LES results

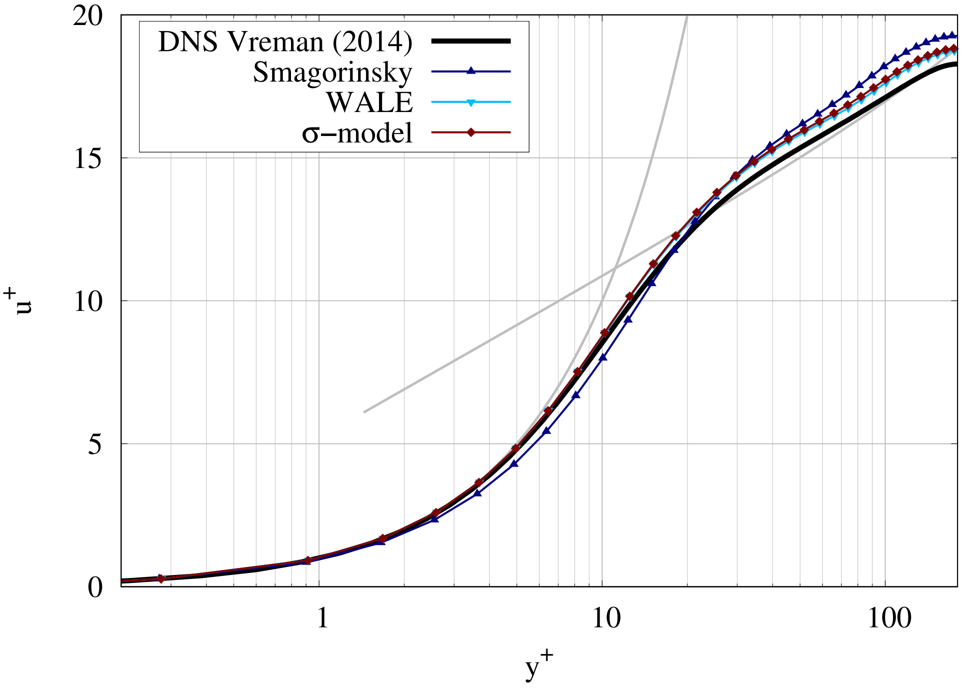
The channel flow at $Re_\tau = 180$ has been addressed as a benchmark also for the LES turbulence models. In particular, here the same setting of the previous section has been employed; thus, a slightly-compressible turbulent channel at $M_\infty = 0.1$ was simulated. Similarly to the previous arrangement, the Mach number has been used to control the bulk speed, so indirectly imposing the pressure gradient $d\langle p\rangle/dx = \sqrt{\gamma}M_\infty - u_b$. The shear Reynolds number Re_τ instead was indirectly imposed setting a bulk Reynolds number equal to $Re_b = 5790$. Periodic boundary conditions were enforced along both the stream-wise and the span-wise directions, while no-slip isothermal wall conditions were employed at the top and at the bottom sides of the domain. The domain was discretised by a non-uniform Cartesian grid clustering the computational points around the wall location. For the job an error mapping function in the form of

$$x(\xi) = \frac{1}{2}\frac{\mathrm{erf}\left(\alpha(\xi - \frac{1}{2}\right)}{\mathrm{erf}\left(\frac{1}{2}\alpha\right)} \tag{6.40}$$

has been employed. Here α, to guarantee a unitary y^+ at the wall location, was set equal to 3.5 while the grid was exactly halved in all the three Cartesian directions fitting $192 \times 64 \times 96$ number of points. As far as the numerical method, no hybridisation strategies were employed for the convective fluxes and just the KGP6 method was used. The simulation was initialised interpolating the DNS obtained in the previous simulation.

In Figure 6.5, the normalised stream-wise velocity component is plot as a function of the normalised wall coordinate and compared among the various models. From these results, we can immediately recognise that the Smagorinsky model fails in fitting both the inner and the outer portion of the flow. On the other hand, the WALE and σ models are quantitatively equivalent recovery a right prediction compared to the DNS benchmark. The same conclusions can be stated looking at the Figure 6.6 and Figure 6.7 were the root-mean-square of the three normalised velocity components have been plotted as a function of the Kolmogorov wall unit. In particular, Figure 6.7b shows the behaviour of the turbulent sub-grid-scale viscosity normalised by the molecular viscosity. As we notice, both the WALE and the σ models vanish the μ_{SGS}

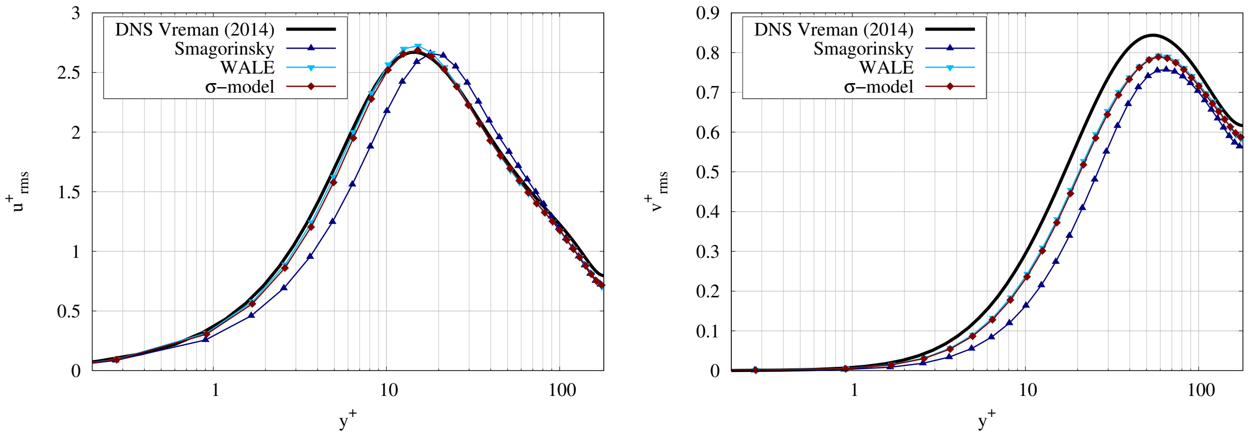
at the wall location, respecting the analytical prediction provided Kuhn (1986) who proved $\mu_{SGS} \sim \mathcal{O}(y^+)^3$ in the inner layer.



(a)

Figure 6.5: Normalised stream-wise velocity component for a channel flow at $Re_\tau = 180$ with various LES turbulence models. The results are compared with the DNS data provided by Vreman and Kuerten (2014). The solid lines denotes the law of the wall expressed by $u^+ = 0.41^{-1}\ln(y^+) + Re_\tau^{-1}y^+ + 5.2$

.

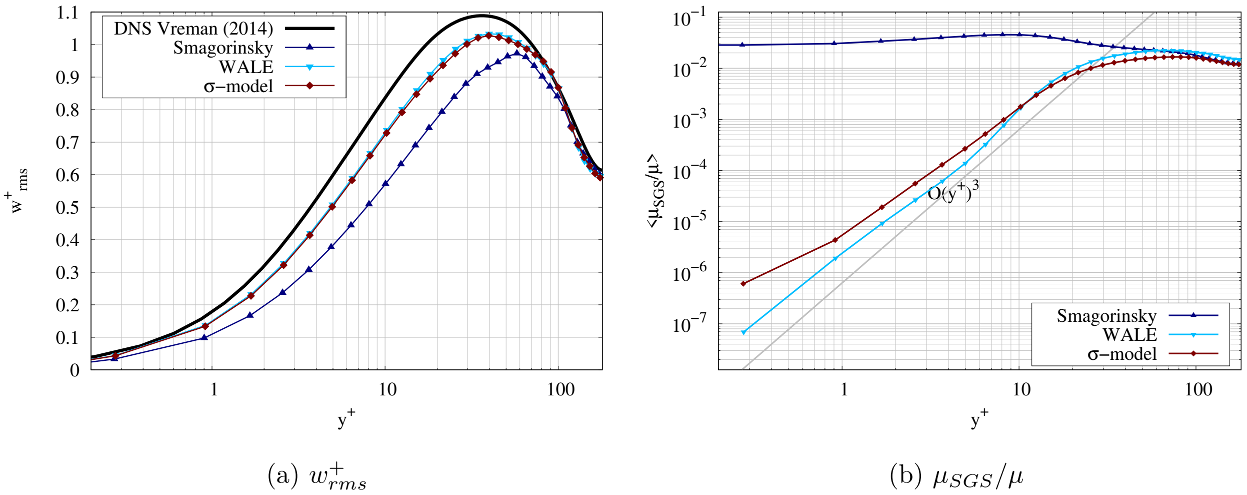


(a) u^+_{rms}

(b) v^+_{rms}

Figure 6.6: Normalised root-mean-square of the velocity components for a channel flow at $Re_\tau = 180$ with various LES turbulence models. The results are compared with the DNS data provided by Vreman and Kuerten (2014)

.



(a) w^+_{rms}

(b) μ_{SGS}/μ

Figure 6.7: Normalised root-mean-square of the velocity components for a channel flow at $Re_\tau = 180$ with various LES turbulence models. The results are compared with the DNS data provided by Vreman and Kuerten (2014)

.

6.3 Summary

The present chapter was divided into two main parts. In the first part, a brief introduction about parallel programming and parallel computing was pointed out. Here, the parallelisation strategy employed in the development of our in-house fluid dynamics solver was presented and tested. From the scaling results, the solver showed good parallel performances up to a competitive amount of cores. This fact made us able to perform three-dimensional simulations in the field of turbulent and fully separated flows. Among the variety of tests and experimental data, we selected the turbulent channel problem as a benchmark for our numerical model. The second part of the chapter was devoted to the wall turbulence; thus, before comparing our numerical results with the most accurate solutions available in the literature, a detailed theoretical description of the problem was presented.

7
The immersed boundary methods

Science may be described as the art of systematic oversimplification.

KARL R. POPPER

Contents

In the previous chapters, we described the numerical methods employed in the discretisation of the Navier-Stokes system of equations, presenting a considerable number of benchmarks and test cases to validate the core of our numerical tool. In the present chapter, we will move to aerodynamics applications discussing the numerical strategies employed in the treatment of embedded boundaries inside the computational domain. Among the variety of numerical techniques suitable to face this target we employed the Immersed Boundary Method (IBM). The chapter aims to present a detailed description of the class of the immersed boundary methods focusing on the most common approaches available in the literature and in particular the Direct Forcing Method (DFM) and the Ghost-Point Forcing Method (GPFM) will be presented and discussed. The latter method is the one implemented in our solver.

7.1 Introduction and literature review

Numerical simulations of compressible viscous flows around complex objects is a crucial target in many engineering applications, and the problem becomes even more complex dealing with moving solids inside the computational domain. These problems range from the control of aeroelastic structures up to the study of particulate flows and surely today the success of landing on dense atmosphere planets [10, 34] or the efficiency improvement of high-bypass turbofan engines [36] are even more affected by the non-linear dynamics between the structural components and the supersonic flow field. Up to now, two main approaches seams to be the most promising: The Body Fitted Grid Method (BFGM) and the Immersed Boundary Method (IBM). Regarding the first, the method was extensively exploited in many commercial and in-house applications, and it was found very accurate in the prediction of the aerodynamic behaviour of complex geometries. Unfortunately, the BFGM does not allow manageable implementations of high-order discretisation schemes and - especially in case of moving objects - it represents a very computational demanding approach requiring the computational grid to be generated and recomputed at each time step.

On the other hand, the IBM does not require a particular coding effort for grid generation, and high-order discretisation schemes can be employed in a simple finite-difference framework. Because of this, the IBM seems to be the most promising numerical strategy able to deal with moving objects in numerical flow simulations. The method provides a coupling interface between the solid and the fluid domain allowing the body surface to cut the computational cells; thus a Cartesian mesh can be employed independently of the geometry complexity or the motion of the boundary. In particular, in the latter case, the grid can be computed once, at the beginning of the simulation, and no other grid generations must be performed during the computation, highly increasing the computational efficiency.

From a historical point of view, the IBM sees Peskin (1972) as its first designer and developer. In his pioneering work, the author applied the method to model complex biological incompressible flows. In this field the system's dynamic is strongly affected by the motion and the deformability of the boundaries, inducing strong non-linear phenomena in the flow field. Thus to accurately simulate the deformation of the biological tissue caused by the fluid interaction, Peskin developed a fully coupled algorithm initiating the immersed boundary studies. Despite its original application, to date the IBM has been widely applied in every kind of fluid dynamics problems. Uhlmann (2005), Breugem (2012), Picano et al. (2015) and Niazi Ardekani et al. (2018) applied the method to deal with fully resolved particulate flow. Fadlun et al. (2000) investigated it for turbulent incompressible flows while Ren et al. (2013) and Wang et al. (2016) applied it in the case of thermal flow problems. More recently De Palma et al. (2006), de Tullio et al. (2007), Merlin et al. (2013), Luo et al. (2015), Piquet et al. (2016), Bernardini et al. (2016) and Boukharfane et al. (2018) applied the method to compressible flows. A clever description of the method can be found in the works of Iaccarino and Verzicco (2003) and Mittal and Iaccarino (2005b).

Even if in literature the IBM is categorised as a single numerical technique, various numerical strategies under the *immersed boundary* name have been proposed and developed following Peskin's original work and, for sure, the two most common and successful approaches are the Direct-Forcing Method (DFM) and the Ghost-Point-Forcing Method (GPFM). In the following paragraphs, we will explain these two methods with a sufficient level of details focussing in particular on the GPFM, which represents the method implemented in our solver.

7.2 The direct forcing method

The DFM consists of directly forcing the Navier-Stokes system of equations by adding a source term that simulates the presence of a solid body. The forcing process can be carried out with various numerical techniques, both explicitly and implicitly and employing various interpolation techniques. Even though the method is easily implemented and accurate, it can deal only with Dirichlet boundary conditions. This fact limits its capabilities and its implementations, especially in case of thermal and compressible flows applications. Let us give a conceptual overview of the method. Considering the Navier-Stokes system of equations forced by a source term $\mathbf{S}(\mathbf{U})$

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j(\mathbf{U})}{\partial x_j} - \frac{\partial \mathbf{F}_{vj}(\mathbf{U})}{\partial x_j} = \mathbf{S}(\mathbf{U}) \tag{7.1}$$

the goal of the DFM consists of designing the source therm in such a way that the presence of the immersed body is taken into account. Thus, being $\{\mathbf{x}_b^l\}_{l=1}^n$ a collection of *Lagrangian points*, the source term $\mathbf{S}(\mathbf{U})$ can be defined as following

$$\mathbf{S}(\mathbf{U}) = \{0, \rho\mathscr{F}_i, \rho\mathscr{F}_i u_i\}^T, \quad i = 1, \ldots, 3 \tag{7.2}$$

where $\mathscr{F}_i$ is an *Eulerian force field* defined as

$$\mathscr{F}_i(\mathbf{x}) = \begin{cases} u_i - \bar{u}_i^l & \text{if} \quad \|\mathbf{x} - \mathbf{x}_b^l\| \to 0 \\ 0 & \text{if} \quad \|\mathbf{x} - \mathbf{x}_b^l\| \to \infty \end{cases} \tag{7.3}$$

Here u_i and $\bar{u}_i^l$ are respectively the velocity component in $i-$th direction and a target speed at the boundary location $\mathbf{x}_b^l$. With these positions, the source term assumes the role of a dumping contribution, forcing the solution of the forced Navier-Stokes system of equations (7.1) to respect the target velocity at the boundary location and enforcing the boundary condition for the momentum conservation equations. From an applicative point of view in case of forced equations, it is quite common to face the integration process with a predictor-corrector strategy. A fist integration step (*predictor step*) reads the equations without the source terms; thus it computes a guess solution $\mathbf{U}^*$ solving the system

$$\frac{\partial \mathbf{U}^*}{\partial t} + \frac{\partial \mathbf{F}_j(\mathbf{U}^*)}{\partial x_j} - \frac{\partial \mathbf{F}_{vj}(\mathbf{U}^*)}{\partial x_j} = 0 \tag{7.4}$$

Once $\mathbf{U}^*$ is known, a *corrector step* is employed to impose the source, thus solving the equation

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{S}(\mathbf{U}^*) \tag{7.5}$$

Avoiding the explanation of how to get the solution of the predictor step - a problem we extensively described in the previous chapters - here, we will focus to the numerics behind the corrector step. As we already mentioned, the problem can be faced both explicitly and implicitly, and in the present dissertation, we will present both the alternatives.

7.2.1 The explicit direct forcing method

Designed initially by Uhlmann (2005) today the explicit Direct Forcing Method (eDFM) was extensively improved and successfully employed in many numerical tools, especially in the incompressible flows regime. Among the various eDFMs, surely the method proposed by Breugem (2012) - consisting of an iterative process able to ensure the desired velocity condition at the boundary location - represents one of the most efficient and accurate procedure. Thus, let us suppose to know the three predictor fields associated to the velocity components $\{u_i^*\}_{i=1}^3$. Fixing a tolerance *toll* and a maximum number of iterations it_{max}, the initialisation of the method wants the u_i^* field to be passed to a temporary variable $\tilde{u}_i$, thus $u_i^* \rightarrow \tilde{u}_i$. Once the process is completed, the following loop, over an integer counter $k = 1, \dots, it_{max}$ and over the set of Lagrangian points $l = 1, \dots, n$ is performed:

1. The first step consists in interpolating the velocity field at the boundary location $\{\mathbf{x}_b^l\}_{l=1}^n$ using the definition of the Dirac delta distribution

$$\left(u_i^{l(k)}\right)_b = \int_{-\infty}^{\infty} u_i^{*(k-1)} \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} \quad l = 1, \dots, n \tag{7.6}$$

2. Once $\left(u_i^{l(k)}\right)_b$ is computed, an acting Lagrangian force per unit mass, structurally similar to the Eulerian force field $\mathscr{F}_i$, can be calculated. The latter is defined as

$$f_i^{l(k)} = f_i^{l(k-1)} + \frac{1}{\Delta t}\left(\bar{u}_i^l - \left(u_i^{l(k)}\right)_b\right) \quad l = 1, \dots, n \tag{7.7}$$

3. Using the Dirac delta definition the Lagrangian force can be spread to the grid obtaining the Eulerian force field $\mathscr{F}_i$ as

$$\mathscr{F}_i = \int_{-\infty}^{\infty} f_i^{l(k)} \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x}_l \quad l = 1, \dots, n \tag{7.8}$$

   ending in this way the loop on Lagrangian points.

4. Once $\mathscr{F}_i$ is computed a forward Euler integration step is performed to update the predictor velocity field u_i^*, thus

$$u_i^{*(k)} = \tilde{u}_i + \Delta t \mathscr{F}_i \tag{7.9}$$

   The loop on k re-start from the point 1. until $k < it_{max}$ and $e > toll$ where e denotes the maximum error related to the boundary target speed defined as

$$e = \max_{l=1}^{n} \left| \left(u_i^{l(k)}\right)_b - \bar{u}_i^l \right| \tag{7.10}$$

   Once the convergence has been reached or the number of iterations exceeds it_{max} the velocity field is updated using the last predictor velocity field, thus $u_i \leftarrow u_i^{*(k)}$.

The Breugem's method is quite easily implemented and straightforward in ensuring the desired velocity at the boundary location, but it is strongly affected by the numerical approximation of the Dirac delta distribution. The latter, in a three-dimensional discrete path, is defined as

$$\delta(\mathbf{x} - \mathbf{x}_0) = \prod_{i=1}^{3} \frac{1}{\Delta x_i} \phi\left(\frac{x_i - x_{i,0}}{\Delta x_i}\right) \tag{7.11}$$

Here ϕ denotes a smooth function whose argument is a non-dimensional radius expressed by $r_i = (x_i - x_{i,0})/\Delta x_i$. In particular being r_{max} the maximum radius of ϕ and $I_r = [-r_{max} : r_{max}]$ its support, the ϕ function must respect the following smooth conditions

$$\phi(r) \in \mathcal{C}^0[I_r] \tag{7.12}$$

$$\phi(r) = 0 \quad \text{if } |r| \geq r_{max} \tag{7.13}$$

$$\int_{-\infty}^{\infty} \phi(r) = 1 \tag{7.14}$$

$$\int_{-\infty}^{\infty} r\phi(r) = 0 \tag{7.15}$$

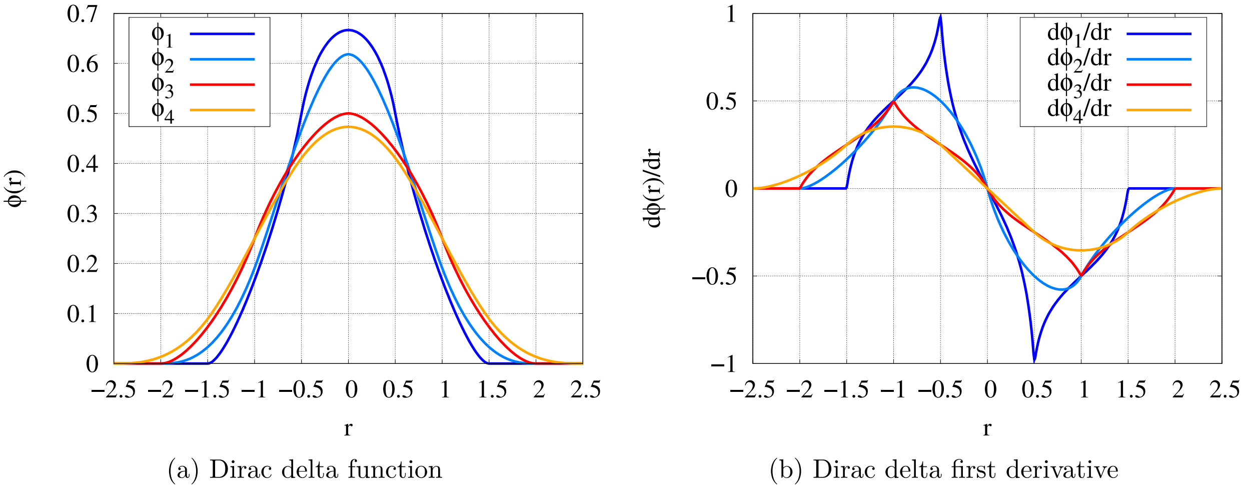
In particular, the first condition expresses the continuity of ϕ, and the second grants the function to have a compact support. The third bound instead ensures a conservation property for the interpolation, and finally, the fourth line grants the first momentum of ϕ to be zero. Many authors, respecting the smoothness conditions, proposed their own approximated formulation of the Dirac delta function and a comprehensive description of the available implementations can be found in Yang et al. (2009). Here the equations (7.16) reports some formulations with an increasing level of regularity, while in Figure 7.1 a comparison between them is shown. In particular from Figure 7.1 we can notice that spreading the Dirac delta over bigger support makes the function smoother and smoother. In this respect, it is not pointless to be said that increasing the size of the Dirac delta support (i.e., selecting a formulation with a higher r_{max}) requires more iterations to reach the convergence of the eDFM. The fact can be easily explained recognising that while the interpolation supports increase, the overlapping interpolation areas also increase, coupling stronger and stronger a boundary point with its neighbours.

$$\phi_1(r) = \begin{cases} \frac{1}{3}\left(1 + \sqrt{-3|r|^2 + 1}\right) & \text{if } |r| \leq 0.5 \\ \frac{1}{6}\left(5 - 3|r| - \sqrt{-3(1 - |r|^2) + 1}\right) & \text{if } 0.5 \leq |r| \leq 1.5 \\ 0 & \text{if } |r| \geq 1.5 \end{cases} \tag{7.16}$$

$$\phi_2(r) = \begin{cases} \frac{17}{48} + \frac{\sqrt{3}\pi}{108} + \frac{|r|}{4} - \frac{r^2}{4} + \frac{1-2|r|}{16}\sqrt{-12r^2 + 12|r| + 1} \\ \qquad -\frac{\sqrt{3}}{12}\arcsin\left(\frac{\sqrt{3}}{2}(2|r| - 1)\right) & \text{if } |r| \leq 1 \\ \frac{55}{48} - \frac{\sqrt{3}\pi}{108} - \frac{13|r|}{12} + \frac{r^2}{4} + \frac{2|r|-3}{48}\sqrt{-12r^2 + 36|r| - 23} \\ +\frac{\sqrt{3}}{36}\arcsin\left(\frac{\sqrt{3}}{2}(2|r| - 3)\right) & \text{if } 1 \leq |r| \leq 2 \\ 0 & \text{if } |r| \geq 2 \end{cases} \tag{7.17}$$

$$\phi_3(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4|r|^2}\right) & \text{if } |r| \leq 1 \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4|r|^2}\right) & \text{if } 1 \leq |r| \leq 2 \\ 0 & \text{if } |r| \geq 2 \end{cases} \tag{7.18}$$

$$\phi_4(r) = \begin{cases} \frac{3}{8} + \frac{\pi}{32} - \frac{r^2}{4} & \text{if } 0 \leq |r| \leq 0.5 \\ \frac{1}{4} + \frac{1-|r|}{8}\sqrt{-2 + 8|r| - 4|r|^2} - \frac{1}{8}\arcsin(\sqrt{2}(|r| - 1)) & \text{if } 0.5 < |r| \leq 1.5 \\ \frac{17}{16} - \frac{\pi}{64} + \frac{|r|-2}{16}\sqrt{-14 + 16|r| - 4r^2} + \frac{1}{16}\arcsin(\sqrt{2}(|r| - 2)) + \frac{r^2}{8} - \frac{3|r|}{4} & \text{if } 1.5 < |r| \leq 2.5 \\ 0 & \text{if } |r| \geq 2.5 \end{cases} \tag{7.19}$$



(a) Dirac delta function

(b) Dirac delta first derivative

Figure 7.1: Comparison between various Dirac delta numerical approximations.

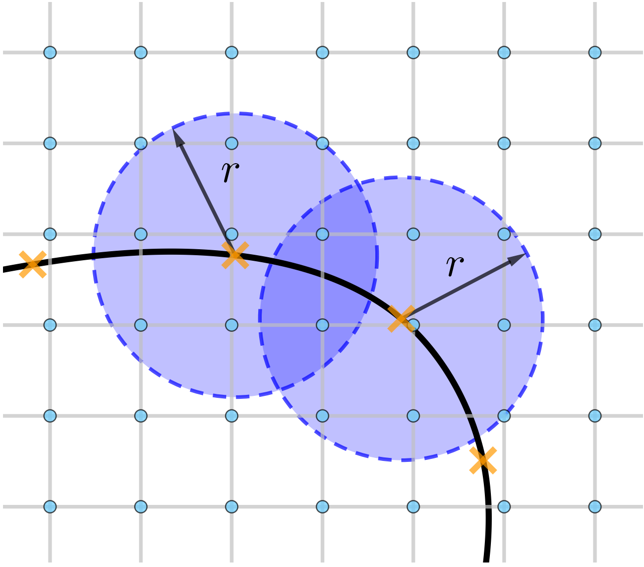


Figure 7.2: Sketch of the overlapping area between the interpolation supports around two consequent boundary points. The black solid line represents the boundary curve, the orange points denote the Lagrangian points while the blue circle represents the support of the numerical Dirac delta.

7.2.2 The implicit direct forcing method

The iteration process related to the eDFM could be very computational demanding, especially in strongly compressible flow conditions. In this field, the discontinuities related to the shocks slow the efficiency of the algorithm requiring even hundreds of iterations to reach the convergence. To overcome the problem, the algorithm can be formulated fully implicitly. In this way, the desired velocity is ensured in just one step in the whole boundary points. The process is carried out solving a linear system whose coefficients take into account all the overlapping interpolation areas between a boundary point and its neighbours. In this paragraph, we will discuss the implicit Direct Forcing Method (iDFM) from a theoretical point of view. A literature reference can be found in Qiu et al. (2016).

Being $\{\mathbf{x}_b^l\}_{l=1}^n$ a set of Lagrangian points describing the boundary surface of a three-dimensional object. By the definition of the Dirac delta distribution, we can express the interpolated value of the velocity field $\{u_i\}_{i=1}^3$ at the boundary location as

$$\left(u_i^l\right)_b = \int_{-\infty}^{\infty} u_i \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} \quad l = 1, \dots, n \tag{7.20}$$

Now writing the velocity field u_i as

$$u_i = u_i^* + \wp u_i \tag{7.21}$$

the summation between a predictor velocity field (u_i^*) and an unknown velocity perturbation $(\wp u_i)$ we can plug the equation (7.21) in (7.20) obtaining

$$\left(u_i^l\right)_b = \int_{-\infty}^{\infty} u_i^* \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} + \int_{-\infty}^{\infty} \wp u_i \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} \quad l = 1, \dots, n \tag{7.22}$$

Now, the perturbations $\wp u_i$ is a scalar field defined in the Eulerian space. As usual in the DFM approach, it is convenient to express it as a function of an unknown perturbation living at the boundary location. This process can be carried out interpolating $\wp u_i$ via the Dirac delta definition

$$\wp u_i = \int_{-\infty}^{\infty} \wp u_i^l \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x}_l \quad l = 1, \dots, n \tag{7.23}$$

thus, inserting in (7.22) we can write

$$\left(u_i^l\right)_b = \int_{-\infty}^{\infty} u_i^* \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \wp u_i^l \delta(\mathbf{x} - \mathbf{x}_b^l) \delta(\mathbf{x} - \mathbf{x}_b^m) d\mathbf{x} d\mathbf{x}_l \quad l, m = 1, \dots, n \tag{7.24}$$

The equation (7.24) expresses a relation between the velocity perturbation $\wp u_i^l$, the desired velocity value $\left(u_i^l\right)_b$ and the interpolated predictor velocity field u_i^* at the boundary location. In particular, we can easily recognise that the term

$$\left(u_i^l\right)_b - \int_{-\infty}^{\infty} u_i^* \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x} \quad l = 1, \dots, n \tag{7.25}$$

defines a Lagrangian force per unit mass f_i^l obtaining that the unknown velocity perturbation can be expressed as

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \wp u_i^l \delta(\mathbf{x} - \mathbf{x}_b^l) \delta(\mathbf{x} - \mathbf{x}_b^m) d\mathbf{x} d\mathbf{x}_l = f_i^l \quad l, m = 1, \dots, n \tag{7.26}$$

Discretising the equation (7.24) we can write

$$\sum_{ijk} \sum_{l=1}^{n} \wp u_i^l D(x_{ijk} - x_b^l) \Delta x_{ijk} D(x_{ijk} - x_b^m) \Delta x_m = u_i^l - \sum_{ijk} (u_i^*)_{ijk} D(x_{ijk} - x_b^l) \Delta x_{ijk} \tag{7.27}$$

Here $D(x_{ijk} - x_b^l)$ represents a discrete formulation of Dirac delta while

$$A_{ml} = \sum_{ijk} \sum_{l=1}^{n} D(x_{ijk} - x_b^l) \Delta x_{ijk} D(x_{ijk} - x_b^m) \Delta x_m \tag{7.28}$$

$$b_l = \left(u_i^l\right)_b - \sum_{ijk} (u_i^*)_{ijk} D(x_{ijk} - x_b^l) \Delta x_{ijk} \tag{7.29}$$

$$x_l = \wp u_i^l \tag{7.30}$$

denotes respectively the matrix collecting the geometrical information related to the interpolation coefficients, the vector of the Lagrangian force at the boundary location and the vector of the unknown Lagrangian perturbations, obtaining that the equation (7.24) is represented by the linear system

$$A_{ml} x_l = b_l \tag{7.31}$$

Solving the equation (7.31), we obtain the values of the unknown velocity perturbations at the boundary location. Once the vector $\{\wp u_i^l\}_{l=1}^n$ have been obtained, we need to distribute the Lagrangian perturbations in the Eulerian grid. The process can be straightforwardly performed employing the equation

$$\wp u_i = \int_{-\infty}^{\infty} \wp u_i^l \delta(\mathbf{x} - \mathbf{x}_b^l) d\mathbf{x}_l \tag{7.32}$$

resulting in the field of the velocity perturbations that can be directly sum up to the predictor velocity field u_i^*.

To conclude, the DFM, in its explicit and implicit versions, represents an easily implemented and accurate numerical strategy able to ensure Dirichlet boundary condition in case of complex steady and moving objects. The selection of the explicit or the implicit version of the method depends on many parameters related both to code design and simulation accuracy. Where the iDFM can ensure more accurately the boundary condition and provides a stabler algorithm, the eDFM is much easier to be implemented, especially in a MPI parallel frame, resulting in a more cost-effective numerical strategy.

7.3 The ghost point forcing method

In the previous section, we showed how the DFM modifies the solution of the Navier-Stokes equations to ensure a desired condition at the boundary location. As we explained, the method represents an accurate and straightforward technology to ensure Dirichlet boundary conditions. This property, in the last decades, has been represented the success of the embedded geometry treatment in many incompressible flow solvers. In the case of incompressible flows, the boundary requirement consists in the setting of the velocity components, making the DFM an extremely powerful tool to deal with complex geometries in the flow domain.

On the other hand, the GPFM, relying on a completely different concept, represents a more general numerical technique able to deal with both Dirichlet and Neumann boundary conditions. The method is more suitable for the applications where the natural boundary conditions are specified as a gradient of some fluid variables as in thermal and compressible flows problems. Despite its greater flexibility, to deal with general geometries, while for DFM is sufficient to known a set of Lagrangian points, for the GPFM is necessary a more complex data management. The method needs to discern if a computational node is a fluid, a ghost, or a solid one and to pursue this target, some automatic solid-detection techniques must be employed [58]. Thus, the boundary can not be represented by a collection of Lagrangian points, but the surface must mesh through elements of some shape, i.e., segments in 2D and triangles in 3D.

Here a note is demanding. Unstructured mesh generation over solid surfaces is well known to be a very challenging task, and a complete description of the problem is far from the objectives of the present work. The reader who wants to become aware of this topic is recommended to look at the manuals of Golias and Dutton (1997) and El-Hamalawi (2003). Today, the flexibility of the most common commercial and open-source computer-aided engineering (CAE) tools, makes the unstructured mesh generation a straightforward process even for very complex geometries. In the present work, among the large variety of tools available in this field, we selected GMSH as the application with the best compromise in term of simplicity and flexibility. Thus, the program was chosen to create proper surface meshes for the GPFM approach and the related solid-detection algorithms.

GMSH is a multi-platform, fully open-source mesh-generator application specially designed for rapidly prototype unstructured mesh over complex two- and three-dimensional geometries. The program is often already installed in the most common UNIX-type operative system distributions, and a complete description of the program can be found at `http://gmsh.info`. Once a geometry is defined - typically importing it from and external CAD - GMSH provides a series of fast and efficient algorithms to compute and generate an unstructured mesh over the object. Then the mesh can be exported with a large variety of data formats and in the present work, to represent both 2D and 3D surfaces, the `.msh` and the `.stl` data formats were respectively employed and coupled with the flow solver.

The `.msh` data format represents a suitable discretisation model for two-dimensional boundary curves. The latter are represented by a series of closed points $\{\mathbf{x}_l\}_{l=1}^{n}$ and edges $\{e_l\}_{l=1}^{n-1}$ resulting in a simple file easily readable by a FORTRAN code. However, after inputting the file, the data set must be reorganised more suitably. The geometry coupling with the Cartesian grid, in fact, wants the spacing between the points (i.e., the length of the edges) to be the same order of magnitude of the local grid spacing. This information can be used to generate the mesh and set the most proper value of n. Thus in the present work, we decided to link the number of points n employed in the solid discretisation to the minimum grid step. In particular - after inputting the `.msh` file - a criterion based on mutual distances is performed. The procedure grants to minimise the memory storage associated with the solid discretisation, saving just the points whose mutual distance is comparable to the minimum grid step. On the other hand the `.stl` data format describes a three-dimensional solid surface via a series of points and triangles. Also in this case, the spacing between the points must respect the grid proportions; thus, the same optimisation process explained for 2D cases can be employed.

Above the flexibility in term of geometry representation of the previous data formats, the segment discretisation offered by the `.msh` and the triangle discretisation offered by `.stl` represent suitable geometrical models for the most common solid-detection algorithms (see e.g., O'Rourke (1998)). In the following sections, we will describe how an object of any shape can be detected on a Cartesian grid, demanding process for good coupling with the GPFM.

7.3.1 Solid detection on a Cartesian grid

As we already mentioned in the previous paragraphs, the GPFM requires to discern if a computational node is a fluid, a ghost or a solid one and to face the problem some automatic solid-detection techniques must be employed. Among the variety of methods in these fields, in the present work, the *ray-tracing algorithm* was applied [85]. The method represents a robust and efficient numerical strategy able to detect where a solid is located inside a grid of any type. Based on the fundamental idea of Jordan's theorem [54], the method translates the problem of the determination if a grid point is inside or outside a solid in counting the number of intersections between the solid bound and a semi-infinite ray. According to this strategy, to exploit the most cost-effective algorithms provided by the Computational Geometry, the solid must be modelled as a 2D-polygon or a 3D-polyhedron, resulting in a perfect coupling with the data formats explained above. The algorithm employs the following procedure:

1. Being Σ a discrete representation of a two-dimensional (three-dimensional) solid surface, thus being Σ represented by a collection of segments $\{S_i\}_{i=1}^{n}$ (triangles $\{T_i\}_{i=1}^{n}$). Consider $\mathbf{x}_0$ a grid point we want to discern if it is inside or outside Σ and fix $\bar{\mathbf{x}}$ a point lying outside the computational domain. If $\bar{\mathbf{x}}$ is sufficiently far from the domain, a semi-infinite ray starting from $\mathbf{x}_0$ can be approximated through the expression

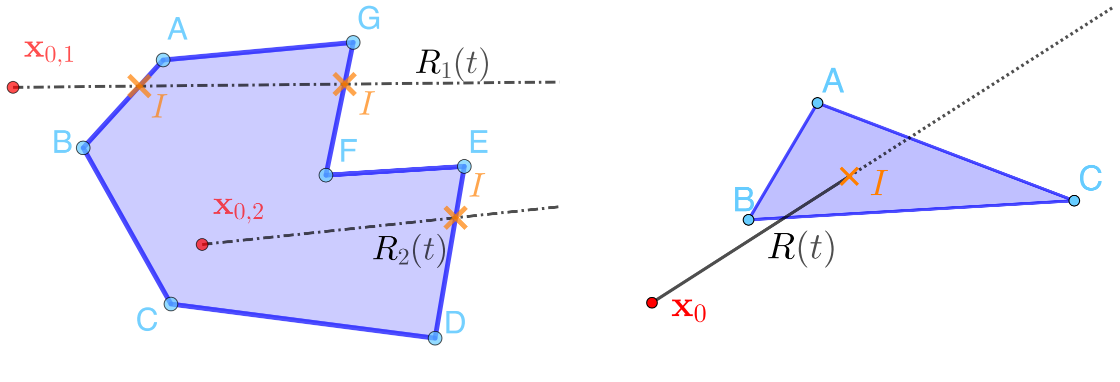
$$\mathbf{R}(t) = \mathbf{x}_0 + (\bar{\mathbf{x}} - \mathbf{x_0})t \quad t \in [0:1] \tag{7.33}$$

2. (a) If the problem is two-dimensional a loop on all the segments $\{S_i\}_{i=1}^{n}$ is performed checking if the ray $\mathbf{R}(t)$ has an intersection with the i-th segment. The problem can be easily faced employing a standard segments-segment intersections algorithm [85].

   (b) On the other hand if the problem is three-dimensional a loop on all the triangles $\{T_i\}_{i=1}^{n}$ is performed and here a test is performed to check if the ray $\mathbf{R}(t)$ has an intersection with it. The problem can be faced employing a standard line-triangle intersections algorithm [80].

   While the geometry is being tested, the number of intersections is saving inside an integer counter.

3. According to Jordan's theorem if the number of intersections is even the point $\mathbf{x}_0$ lies outside the solid surface, otherwise if the number of crossings is odd the point $\mathbf{x}_0$ belongs to the solid.



(a)

(b)

Figure 7.3: Sketch of the ray-tracing algorithm applied to a 2D-polygon (7.3a) and to a line-triangle-intersection problem (7.3b). Here the red points represent the points we want to test while $R(t)$ is the ray. As we notice from figure 7.3a if the number of ray-geometry intersections is even the point lies outside the geometry, on the other hand if the number of intersection is odd the point belongs to the solid.

7.3.2 Data management

The solid detection on a Cartesian grid represents an initialisation step for the GPFM. In particular if the simulated geometry is stationary the ray-tracing algorithm can be invoked one time, at the beginning of the simulation; on the other hand, for dynamic simulations the entire procedure must be performed at each time step. Independently of the problem we are facing the solid detection process can discern if a point is inside, outside or is a ghost point in respect of a solid bound. Being $\mathcal{D}$ the whole computational domain the latter is divided it into three portions: Ω_{solid}, Ω_{fluid} and Ω_{ghost} and the information is stored inside a marker variable ζ defined as

$$\zeta(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \Omega_{solid} \\ 1 & \text{if } \mathbf{x} \in \Omega_{ghost} \\ 2 & \text{if } \mathbf{x} \in \Omega_{fluid} \end{cases} \tag{7.34}$$

where the set of Cartesian points belonging to the solid and fluid regions are respectively defined by

$$\Omega_{solid} = \{\mathbf{x} \in \mathcal{D} \,|\, \zeta(\mathbf{x}) = 0\} \tag{7.35}$$

$$\Omega_{fluid} = \{\mathbf{x} \in \mathcal{D} \,|\, \zeta(\mathbf{x}) = 2\} \tag{7.36}$$

It is essential to highlight that $\Omega_{solid} \cup \Omega_{fluid} = \mathcal{D}$ so the ghost region it is not determined explicitly by the solid-detection process, but it can be computed *a posteriori* as a function of the solid and fluid portions. In particular, Ω_{ghost} must respect the following bounds

$$\begin{aligned}\Omega_{ghost} = \{&(x_i, y_j, z_k) \in \Omega_{solid} \, \text{if} \, \exists (x_l, y_m, z_n) \in \Omega_{fluid} \\ &\text{for}\, l = i-3\ldots, i+3, m = j-3, \ldots, j+3, n = k-3, \ldots, k+3\}\end{aligned} \tag{7.37}$$

resulting in a shell of three layers of points which represents the minimal number of computational nodes needed for the high-order numerical schemes' stencils.

Even if for steady object the data management is the one explain above, dealing with moving solids a fourth value of the marker function ζ must be taken into account considering the so called *fluid emerging points*. In fact being $\mathbf{x}_g(t)$ a ghost point at time level $t = t^n$ (so that $\mathbf{x}_g(t^n) \in \Omega_{ghost}$) the point is flag as *fluid emerging point* if $\mathbf{x}_e(t^{n+1}) \in \Omega_{fluid}$. To avoid spurious oscillations while the solid is moving, those points need a particular treatment in term of boundary interpolation (see §7.3.5).

7.3.3 Geometrical determination of the bound and the image points

After the solid detection and the computation of the marker variable $\zeta = \zeta(\mathbf{x})$ the ghost region Ω_{ghost} provides a set of data suitable for the implementation of the GPFM. For every ghost point $\mathbf{x}_g$ a correspondent boundary point $\mathbf{x}_g$ and a related image point $\mathbf{x}_{ip}$, lying in the flow region, can be associated and computed. After the same interpolation procedures, will be used to ensure the desired boundary condition at the boundary location $\mathbf{x}_b$. In this paragraph, we will explain how the image point can be determined just knowing the ghost node location and a discrete formulation of the boundary curve. Here, for the sake of clarity, the algorithm we will start explaining the algorithm for two-dimensional case, the extension to 3D cases will be straightforward and will be presented later. A sketch of the implementation strategy is reported in Figure 7.4.

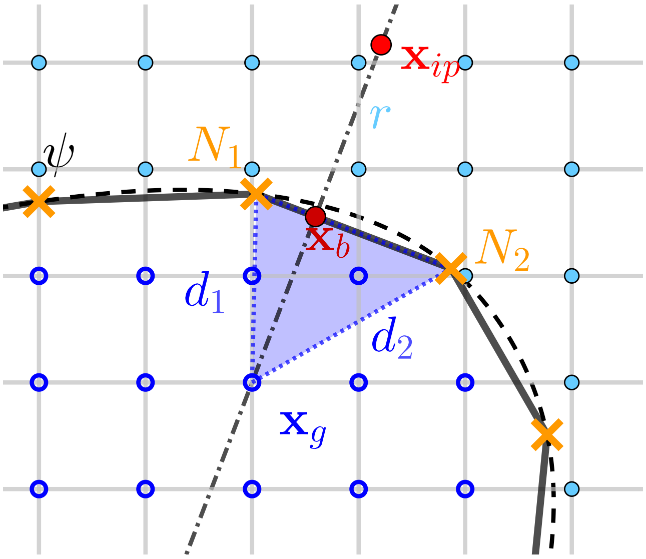


Figure 7.4: Sketch of the determination of the boundary point in respect of a ghost point. Here blue circle represent the ghost node, light blue points denotes the fluid region. Orange crosses represent the Lagrangian points

1. Being $\mathbf{x}_g$ a ghost node, the first step in the determination of the image point location consists in looking at the two Lagrangian points nearest to $\mathbf{x}_g$, thus scrolling the Euclidean distances expressed by

$$d_{g,l} = \sqrt{(\mathbf{x}_g - \mathbf{x}_l) \cdot (\mathbf{x}_g - \mathbf{x}_l)} \quad \forall l = 1, \ldots, n \tag{7.38}$$

2. Once $\mathbf{x}_{N_1}$ and $\mathbf{x}_{N_2}$ have been determined, the line passing through them can be computed as

$$\mathcal{L}: \quad \mathbf{n} \cdot (x, y) + c = 0 \tag{7.39}$$

here $\mathbf{n} = (a, b)$ represents the orthogonal direction in respect of $\mathcal{L}$ and c denotes the line coefficient. The latter are expressed by

$$a = y_{N_1} - y_{N_2} \tag{7.40a}$$

$$b = x_{N_2} - x_{N_1} \tag{7.40b}$$

$$c = x_{N_1} y_{N_2} - x_{N_2} y_{N_1} \tag{7.40c}$$

3. Being $\mathbf{r}(t)$ the orthogonal line in respect of $\mathcal{L}$, a parametric formulation results in

$$\mathbf{r}(t) = \mathbf{x}_g + \mathbf{n}t \quad t \in \mathbb{R} \tag{7.41}$$

so, plugging $\mathbf{r}(t)$ into (7.39) we obtain

$$\mathbf{n} \cdot (\mathbf{x}_g + \mathbf{n}t) + c = 0 \tag{7.42}$$

Solving in t we find

$$t_b = -\frac{c + \mathbf{n} \cdot \mathbf{x}_g}{\mathbf{n} \cdot \mathbf{n}} \tag{7.43}$$

parameter that denotes the location where the line $\mathbf{r}(t)$ cross $\mathcal{L}$. Thus, inserting t_b into equation (7.41) we obtain the nearest point of $\mathbf{x}_g$ lying on the edge $\{N_1 : N_2\}$ resulting in a good approximation for the boundary point $\mathbf{x}_b$. Carrying out the math we obtain

$$\mathbf{x}_b = \left(\frac{b(bx_g - ay_g) - ac}{a^2 + b^2}, \frac{a(ay_g - bx_g) - bc}{a^2 + b^2} \right)^T \tag{7.44}$$

4. A good practice wants the unit vector associated with the normal outer direction at the boundary location to be computed as

$$\mathbf{n}_b = \frac{\mathbf{x}_b - \mathbf{x}_g}{\|\mathbf{x}_b - \mathbf{x}_g\|} \tag{7.45}$$

so finally the location of the image point $\mathbf{x}_{ip}$ can be determined through the expression

$$\mathbf{x}_{ip} = \mathbf{x}_g + 2\mathbf{n}_b \|\mathbf{x}_b - \mathbf{x}_g\| \tag{7.46}$$

Similarly, the process can be performed in three-dimensions. In this case, fixing a ghost node, three nearest points of $\mathbf{x}_g$ must be determined, and the plane passing through them needs to be computed. The latter can be efficiently obtained through the following procedure:

1. Firstly we can get two vectors in the plane by subtracting pairs of points in the plane, thus

$$\mathbf{v}_1 = \mathbf{x}_{N_1} - \mathbf{x}_{N_2} \tag{7.47a}$$
$$\mathbf{v}_2 = \mathbf{x}_{N_1} - \mathbf{x}_{N_3} \tag{7.47b}$$

2. The cross product of these two vectors will be in the unique direction orthogonal to both $\mathbf{v}_1$ and $\mathbf{v}_2$, and hence in the direction of the normal vector to the plane

$$\mathbf{n} = (a, b, c) = \mathbf{v}_1 \times \mathbf{v}_2 \tag{7.48}$$

3. The equation of the plane is then obtained through

$$\mathcal{P} : \quad \mathbf{n} \cdot (x, y, z) + d = 0 \tag{7.49}$$

where d can be determined plugging one of the points $\{x_{N_i}\}_{i=1}^3$ into the equation (7.3.3).

Once the plane coefficients a, b, c, d have been determined, with the same logic of the 2D case, we need to establish the closest point to $\mathbf{x}_b$ lying on the plane $\mathcal{P}$. This represents a good approximation of the boundary point associated with $\mathbf{x}_g$. Expressing parametrically the line passing through $\mathbf{x}_b$ and orthogonal to $\mathcal{P}$

$$\mathbf{r}(t) = \mathbf{x}_g + \mathbf{n}t \quad t \in \mathbb{R} \tag{7.50}$$

so plugging it into

$$\mathbf{n} \cdot (\mathbf{x}_g + \mathbf{n}t) + d = 0 \tag{7.51}$$

we obtain

$$t_b = -\frac{d + \mathbf{n} \cdot \mathbf{x}_g}{\mathbf{n} \cdot \mathbf{n}} \tag{7.52}$$

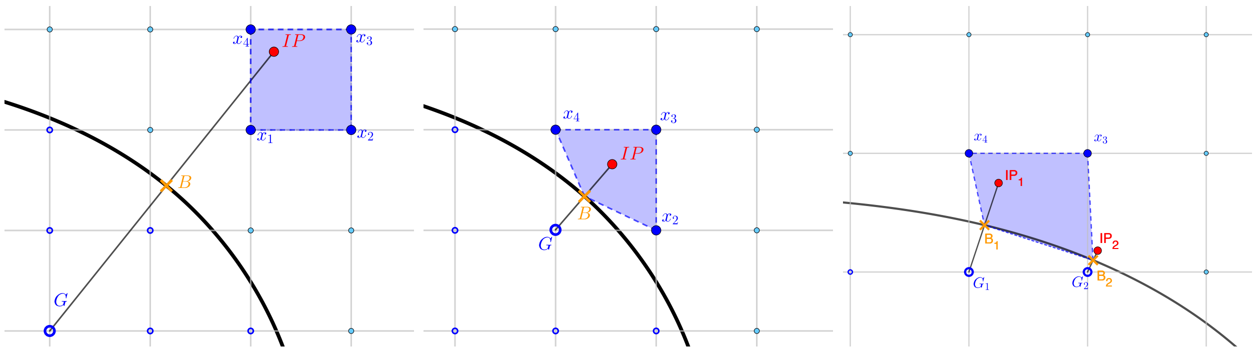
The t_b parameter, inside (7.50), results in the nearest point of $\mathbf{x}_g$ lying on the plane $\mathcal{P}$ point which represents a good approximation of the boundary point $\mathbf{x}_b$. Doubling the distance between $\mathbf{x}_b$ and $\mathbf{x}_g$ along the unit normal direction $\mathbf{n}_b$ we can finally determine the position of the image point associated to $\mathbf{x}_g$.

7.3.4 Boundary interpolation

Once the image point $\mathbf{x}_{ip}$ has been found, the flow-field variables are bilinearly interpolated from its surrenders. As usual, we will start explaining the 2D procedure. Assuming that a generic flow variable ϕ obeys to

$$\phi(x, y) = c_1xy + c_2x + c_3y + c_4 \tag{7.53}$$

the goal is computing the coefficients $\{c_i\}_{i=1}^4$. Therefore, being $\{\phi_i\}_{i=1}^4$ the values of the field ϕ at the surrounding coordinate $\{(x_i, y_i)\}_{i=1}^4$, the coefficients of the interpolation are determined by solving the following linear system:



(a) Interpolation standard procedure (b) Interpolation procedure for cut cells (c) Interpolation procedure for cut cells

Figure 7.5: Sketch of the boundary interpolation procedure in two-dimensions. Here the blue points represent the ghost points, the light blue points represent the fluid nodes while x_1, x_2, x_3, x_4 represent the points retained for the interpolation.

$$\begin{bmatrix} x_1y_1 & x_1 & y_1 & 1 \\ x_2y_2 & x_2 & y_2 & 1 \\ x_3y_3 & x_3 & y_3 & 1 \\ x_4y_4 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} \tag{7.54}$$

Some further considerations must be exploited, and in particular, some exceptions must be pointed out concerning the main procedure. For the case where one (or two) points retained for the interpolation are flagged as ghosts - so that when the boundary curve cuts the interpolation cell - the number of fluid points useful for the variable reconstruction reduces to three or two. In this case, the numerical error drastically drops, and the interpolation procedure becomes a source of numerical instabilities and spurious oscillations. Following the sketch reported in Figure 7.5b, 7.5c we directly superimpose the wall value at the correspondent matrix line using the coordinate of the boundary interception point. Here a distinction between Dirichlet and Neumann boundary condition is needed. For instance, if $(x_2, y_2) \in \Omega_{ghosts}$ and we are dealing with Dirichlet boundary, the correspondent system becomes:

$$\begin{bmatrix} x_1y_1 & x_1 & y_1 & 1 \\ x_by_b & x_b & y_b & 1 \\ x_3y_3 & x_3 & y_3 & 1 \\ x_4y_4 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_b \\ \phi_3 \\ \phi_4 \end{bmatrix} \tag{7.55}$$

otherwise, if we are dealing with Neumann boundary condition the system becomes:

$$\begin{bmatrix} x_1y_1 & x_1 & y_1 & 1 \\ y_bn_x + x_bn_y & n_x & n_y & 0 \\ x_3y_3 & x_3 & y_3 & 1 \\ x_4y_4 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \left(\frac{\partial \phi}{\partial \mathbf{n}}\right)_b \\ \phi_3 \\ \phi_4 \end{bmatrix} \tag{7.56}$$

where the line corresponding to (x_2, y_2) is replaced by $\nabla\phi_b \cdot \mathbf{n}_b$.

Again, three-dimensional boundary approximation can be treated as an extension of two-dimensional problems. In this case, assuming that a generic flow variable ϕ obeys to a three-linear expression

$$\phi(x, y, z) = c_1xyz + c_2xy + c_3xz + c_4yz + c_5x + c_6y + c_7z + c_8 \tag{7.57}$$

the goal is computing the coefficients $\{c_i\}_{i=1}^{8}$. Therefore, being $\{\phi_i\}_{i=1}^{8}$ the values of the field ϕ at the surrounding coordinate $\{(x_i, y_i)\}_{i=1}^{8}$, the coefficients of the interpolation are determined by solving the following linear system:

$$\begin{bmatrix} x_1y_1z_1 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 & 1 \\ x_2y_2z_2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 & 1 \\ x_3y_3z_3 & x_3y_3 & x_3z_3 & y_3z_3 & x_3 & y_3 & z_3 & 1 \\ x_4y_4z_4 & x_4y_4 & x_4z_4 & y_4z_4 & x_4 & y_4 & z_4 & 1 \\ x_5y_5z_5 & x_5y_5 & x_5z_5 & y_5z_5 & x_5 & y_5 & z_5 & 1 \\ x_6y_6z_6 & x_6y_6 & x_6z_6 & y_6z_6 & x_6 & y_6 & z_6 & 1 \\ x_7y_7z_7 & x_7y_7 & x_7z_7 & y_7z_7 & x_7 & y_7 & z_7 & 1 \\ x_8y_8z_8 & x_8y_8 & x_8z_8 & y_8z_8 & x_8 & y_8 & z_8 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \end{bmatrix} \tag{7.58}$$

and similar corrections like the one described above are reserved for cut cells.

7.3.5 Boundary conditions

Once the flow variable ϕ_{ip} is interpolated at the image point position, the wall condition can be finally enforced. The process was carried out reflecting the ϕ_{ip} value in the correspondent ghost point. In particular for Dirichlet boundary condition, being ϕ_b is the desired value of ϕ at the bound location, the value of ϕ_g in the ghost position is expressed by

$$\phi_g = 2\phi_b - \phi_{ip}. \tag{7.59}$$

For inhomogeneous Neumann boundary conditions instead, the value of ϕ_g is given by

$$\phi_g = \phi_{ip} - \Delta l \left(\frac{\partial \phi}{\partial \mathbf{n}}\right)_b , \tag{7.60}$$

Here $\Delta l = \|\mathbf{x}_{ip} - \mathbf{x}_g\|$ while $(\partial\phi/\partial\mathbf{n})_b$ is the desired value of the ϕ gradient at the solid-fluid interface. In our computations, we enforced the boundary conditions on primitive variables $\{u_i\}_{i=1}^3, T$ and p following the method proposed by Piquet et al. [89] so that the equations (7.59) and (7.60) become

$$u_{i,g} = 2u_{i,b} - u_{i,ip}, \quad i = 1, \dots, 3 \tag{7.61}$$

$$T_g = T_{ip} - \Delta l \left(\frac{\partial T}{\partial \mathbf{n}}\right)_b \tag{7.62}$$

$$p_g = p_{ip} - \Delta l \left(\frac{\partial p}{\partial \mathbf{n}}\right)_b \tag{7.63}$$

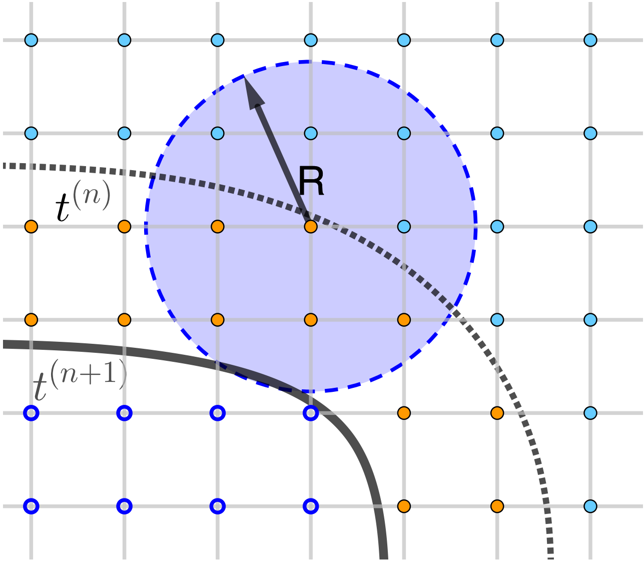


Figure 7.6: Sketch of the interpolation procedure around a *solid emerging point*. Here the dashed black line represents the position of the boundary at time $t^{(n)}$ while the black solid line is the fluid-solid interface at time $t^{(n+1)}$. The orange points are the *solid emerging points* and the blue circle represents the support employed for the interpolation.

In particular, for adiabatic wall $(\partial T/\partial\mathbf{n})_b = 0$ for both steady and moving solids, while the pressure gradient $(\partial p/\partial\mathbf{n})_b$ is expressed by

$$\left(\frac{\partial p}{\partial \mathbf{n}}\right)_b = \begin{cases} 0 & \text{if } \mathbf{u}_b = \mathbf{0} \\ -\left(\rho \frac{D\mathbf{u}_b}{Dt} \cdot \mathbf{n}_b\right) & \text{otherwise} \end{cases} \tag{7.64}$$

To conclude a note is demanding while the boundary is moving and in particular, recalling the concept of *fluid emerging points*. Those Cartesian positions when the boundary has been moved, do not have a previous Runge-Kutta history, resulting in a source of spurious oscillations and instabilities. To get them a more suitable value in respect of their neighbours, in our computation the smoothest Dirac delta (7.19) developed by Yang et al. [134] was employed and in Figure 7.6 a sketch of the interpolation strategy is reported.

7.4 Summary

The present chapter described the numerical methods employed in the treatment of embedded boundaries inside the computational domain. Among the variety of methods available in the literature, the current work discussed the Direct-Forcing-Method (DFM) and the Ghost-Point-Forcing-Method (GPFM). The chapter highlighted the substantial differences between the two methods expressing their pros and cons. While DFM represents a straightforward technology for embedded boundaries, its limitations in the treatment of gradient-type conditions confines the method just to momentum equation enforcement [12]. This circumscribes its feasibility to incompressible flow simulations. On the other hand, the GPFM represents a more flexible numerical strategy, suitable for both Dirichlet and Neumann conditions. For these reasons, the GPFM - in combination with a particular treatment of cut cells and fluid-emerging points - was employed in our computations. In the second part of the chapter, a complete description of the method in term of solid-detection strategies, data management, geometrical reconstruction, and boundary enforcement has been discussed.

8 Results with the immersed boundary method

E come li stornei ne portan l'ali
nel freddo tempo, a schiera larga e piena,
così quel fiato li spiriti mali

di qua, di là, di giù, di sù li mena;
nulla speranza li conforta mai,
non che di posa, ma di minor pena.

The wind of Inferno,
La Divina Commedia - Inferno: C. V, vv. 40-44

Dante Alighieri (1265–1321)

Contents

In the previous chapter, a detailed theoretical description of two discrete immersed boundary formulations was presented and discussed. In particular, the numerical properties and the behaviours of the DFM and the GPFM were highlighted, concluding the latter to be more suitable for compressible aerodynamics over complex geometries. In the present chapter, a large variety of numerical results and validations concerning the coupling between the GPFM and the core of the numerical solver will be analysed. The chapter is organised into two parts. In the first part, the validation campaign concerns steady objects aerodynamics. Hence, starting from the incompressible flow regime, we will move to transitional flows, concluding to shock-obstacle interactions over complex geometries. In the second part, in a similar path, the results obtained concerning moving objects aerodynamics will be detailed. In particular, knowing that sharp-interface IBM for moving solids in compressible viscous flows often exhibit spurious noise propagating from the moving boundary, in the present work, we show that using the KGP6 scheme in combination with the GPFM is able to keep controlled the pressure-velocity spurious noise, stably resolved the simulation without employing artificial dissipation techniques.

8.1 Steady objects aerodynamics

In the present section, the main results obtained concerning the aerodynamics of steady objects will be presented and discussed. In particular, we will detail the cylinder aerodynamics in various configurations, from the incompressible regime, up to strong compressible cases, and we conclude the section dealing with shock interaction against complex steady solids. In Figure 8.1 a sketched of the domain employed in respect to the cylinder test cases is reported, while Table 8.1 reports the domain proportions for all the simulations.

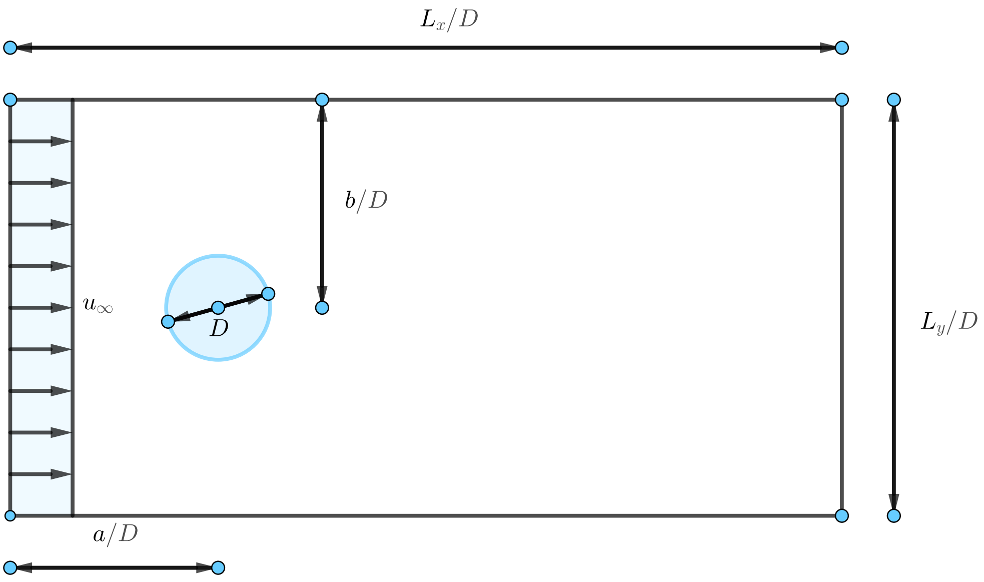


Figure 8.1: Overview of the two-dimensional domain employed for the simulations of the flow past a cylinder.

Simulation	par.	a/D	b/D	L_x/D	L_y/D
Subsonic flow past a confined cylinder	§8.1.1	2	1	8	2
Subsonic flow past a free cylinder	§8.1.2	13	10	50	20
Supersonic flow past a confined cylinder	§8.1.3	2	3	24	6
Supersonic flow past a free cylinder	§8.1.4	10	10	40	20
Shock wave diffraction against a cylinder	§8.1.5	4	4	10	8

Table 8.1: Geometry of the computational domains employed in the cylinder aerodynamics tests suite. The parameters refer to Figure 8.1.

8.1.1 Subsonic flow past a confined cylinder at various Re and D/L_y ratios

The simulation of a flow over a confined cylinder in a low-Mach regime represents a challenging test for a compressible flow solver. Strongly affected by the flow compressibility, to accurately represent a fully subsonic condition, the problem requires the reference Mach number to be very small concerning the unity. As we already mentioned, reducing the Mach number, spurious oscillations due to pressure-velocity decoupling, especially at the boundary location, becomes a frequent eventuality. The usage of the KGP6 method prevents the problem. Here the results obtained over a wide range of Reynolds numbers and for a confined cylinder with various aspect ratios will be presented.

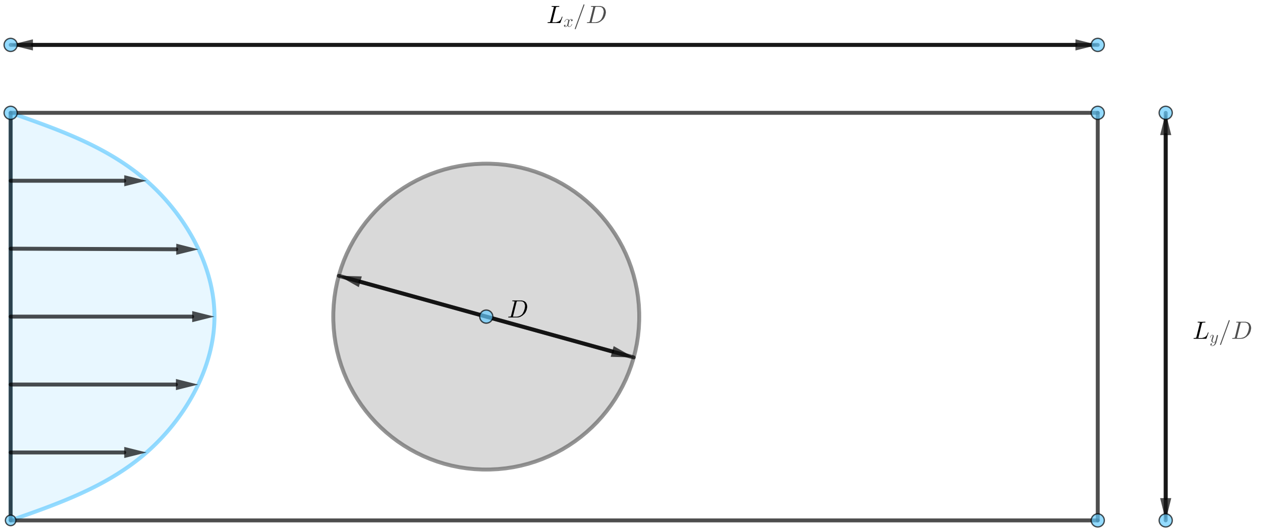


Figure 8.2: Overview of the domain employed for the simulations of the flow past a confined cylinder in low-Mach conditions.

Standard NSCBC [97, 70, 96] were imposed at the inlet and outlet boundary while no-slip wall conditions were enforced at the top and the bottom boundary respectively. The computation have been performed at various Reynolds numbers $Re = u_\infty D/\nu = \{1,10,20,40,100\}^T$ and at $D/L_y = \{0.5, 0.7\}^T$. A parabolic inflow velocity profile was imposed at the left boundary and here u_∞ denotes the pick value of inlet speed, D is the cylinder diameter and ν the kinematic viscosity of the fluid. In Figure 8.2 a sketch of the domain is reported. The initial values for the non-dimensional pressure and temperature were set to $p/p_0 = T/T_0 = 1$ respectively, while initial velocity components are set to zero. To avoid any compressibility effect, the u_∞ value was set in such a way that the free stream Mach number M_∞ equals to 0.01 at the inlet position. The computational grid consisted of a structured uniform grid featuring $N_x \times N_y = 1600 \times 400 = 640000$ computational nodes. The time-step was evaluated from the Courant-Friedrichs-Lewy stability condition with a CFL number equals to 0.5. Interpolating the computed pressure and velocity fields bilinearly around the boundary positions the drag coefficient was calculated. In particular, the following formulation was implemented:

$$c_D = \frac{\oint p_b \mathbf{n}_b \cdot \hat{\mathbf{i}} dS_b + \frac{\sqrt{\gamma} M_\infty}{Re} \oint \mu \sigma_b \mathbf{n}_b \cdot \hat{\mathbf{i}} dS_b}{\frac{1}{2}\gamma p_\infty M_\infty^2} \tag{8.1}$$

here p_b, $\mathbf{n}_b$, σ_b represents respectively the pressure, the normal outer direction and the viscous stress tensor at the boundary position while $\hat{\mathbf{i}}$ is the unit vector associated to the u_∞ flow direction. In Figure 8.3, the computed values of the drag coefficients are compared with the results obtained by Sahin and Owens [108] showing an excellent agreement for all the Reynolds numbers investigated. In Figure ?? and Figure 8.4 the Mach fields in a stationary conditions are reported as a function of the Reynolds number and the D/L_y ratio.

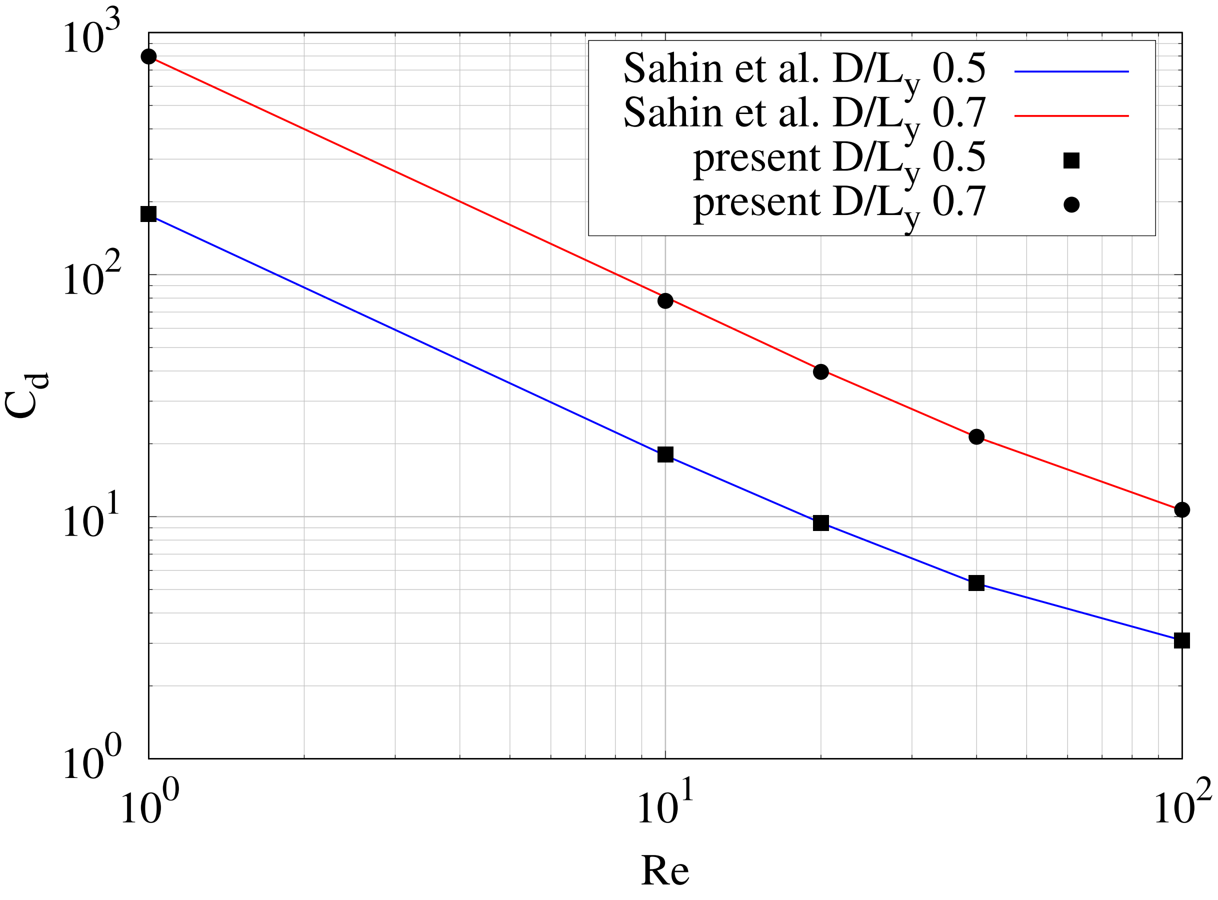
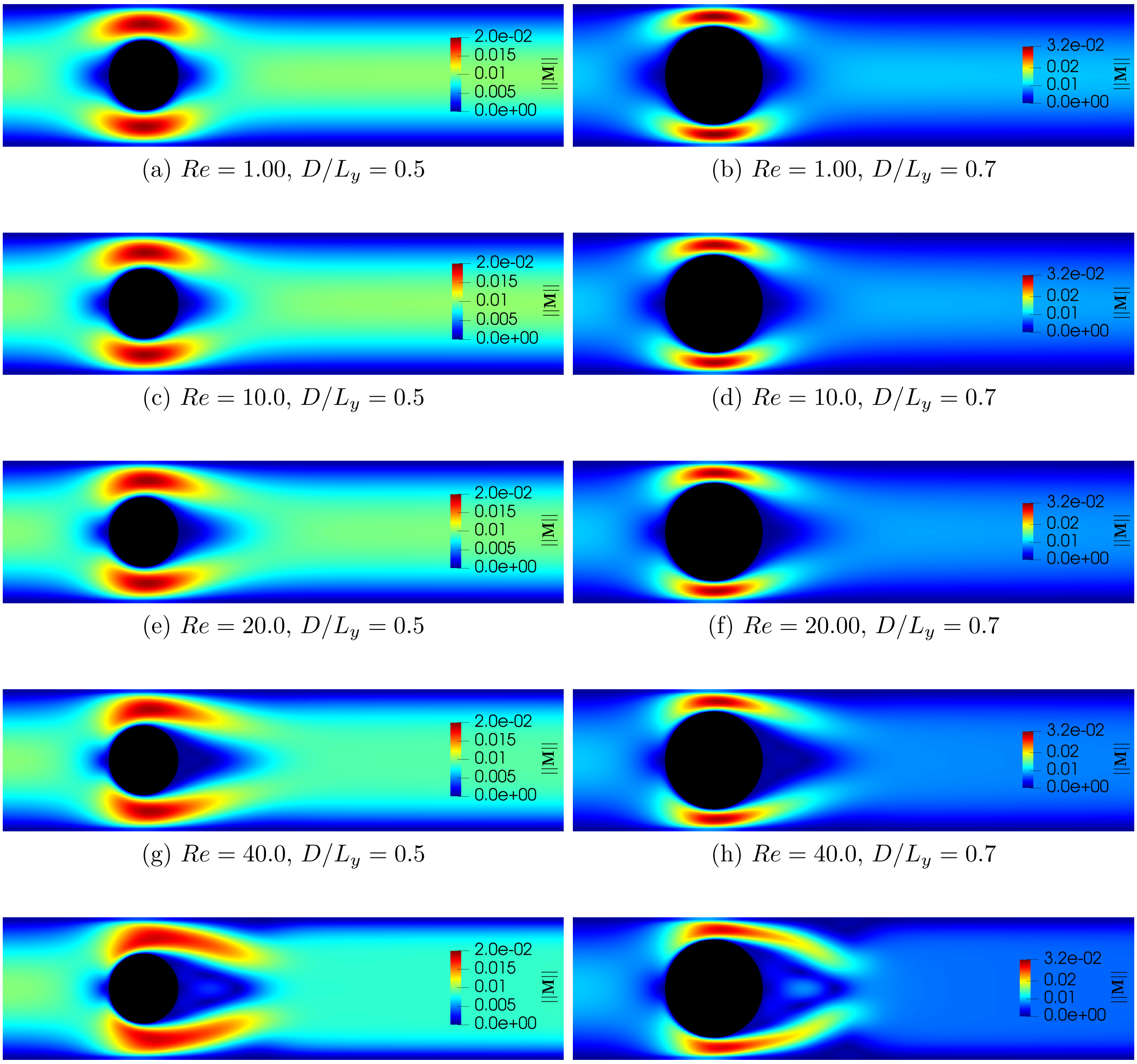


Figure 8.3: Comparison of the computed drag coefficient for a subsonic flow past a confined at $M = 0.01$ with various Reynolds number and D/L_y ratios. In figure Figure 8.3 the drag coefficient obtained with the present method (black points) is reported as a function of the Reynolds number for the cases at $D/L_y = \{0.5, 0.7\}^T$. The results are compared with the simulations of Sahin and Owens (2004)(solid lines). The simulation was performed on a uniform Cartesian grid consisted in $N_x \times N_y = 1600 \times 400$ grid points for a domain of $[8D; 2D]$ and two diameters were reserved upstream the cylinder centre location.



(a) $Re = 1.00$, $D/L_y = 0.5$

(b) $Re = 1.00$, $D/L_y = 0.7$

(c) $Re = 10.0$, $D/L_y = 0.5$

(d) $Re = 10.0$, $D/L_y = 0.7$

(e) $Re = 20.0$, $D/L_y = 0.5$

(f) $Re = 20.00$, $D/L_y = 0.7$

(g) $Re = 40.0$, $D/L_y = 0.5$

(h) $Re = 40.0$, $D/L_y = 0.7$

(i) $Re = 100$, $D/L_y = 0.5$

(j) $Re = 100$, $D/L_y = 0.7$

Figure 8.4: Stationary laminar Mach field at $Re = \{10{,}20{,}40{,}100\}$ and $D/L_y = \{0.5{,}0.7\}$ for a steady confined cylinder at $M_\infty = 0.01$.

8.1.2 Subsonic flow past a free cylinder at various Re numbers

A similar domain configuration, as described in the previous paragraph, was employed to simulate the subsonic flow past a cylinder in an open domain. Again NSCBC were imposed at the inlet and outlet portion of the domain. Symmetry conditions were enforced at the top and the bottom boundary. It should be noticed that, considering that the transverse dimension of the domain $L_y/D = 20$ the choice of the lateral boundary does not produce any significant confinement phenomenon. The stream-wise dimension was set equal to $L_x/D = 50$. Here we simulated the flow at $Re = u_\infty D/\nu = \{10, 20, 40, 70, 100, 300\}^T$. A uniform velocity profile is imposed at the inlet boundary and here u_∞ denotes the value of the inflow speed, D is the circle diameter and ν the kinematic viscosity of the fluid. Following the previous tests, ambient conditions were initialised at $t/t_0 = 0$ while the u_∞ value was set in order to ensure $M = 0.05$

at the inlet flow position. The computational grid consisted of a structured non-uniform grid, stretched around the position of the cylinder and featuring $N_x \times N_y = 800 \times 320 = 256000$ grid points. Grid stretching was applied along both x and y directions but, near the cylinder, a uniform grid spacing, approximately with $D/\Delta x = 50$, was employed. In particular, the grid results in the formulation proposed by Orlandi (2012), so it was expressed by

$$x(\eta) = \begin{cases} \eta \frac{x_c}{\eta_c} & \text{for } |\eta| < \eta_c \\ \frac{\eta}{|\eta|} x_F + \frac{tanh(\alpha_c(\eta - 0.5\frac{\eta}{|\eta|}))}{tanh(\alpha_c(0.5-\eta_c))}(x_F - x_c) & \text{otherwise} \end{cases} \tag{8.2}$$

Here $\eta = \xi - 0.5$ is the computational coordinate, x_F is external boundary location, x_c the size of the inner uniform region and α_c is the stretching parameter. In Figure 8.5 a global view of the computational grid is shown. The CFL number was set equal to 0.5, and the simulation was performed up to the statical-steady state of the flow variables.

A large number of results, both numerical and experimental, have been produced regarding the free cylinder aerodynamics in subsonic flow regime (see e.g., [28, 120, 25, 68, 110, 12]) and here we present our results compared with the most accurate data available in the literature. In particular, because of the inherent unsteadiness of the cases at $Re = 100$ and $Re = 300$, we will treat them separately, dividing the test campaign into two groups.

Focusing on the configurations at $Re = \{10,20,40,70\}^T$, in table Table 8.2 the values of the drag coefficient are compared with the one in literature; in Figure 8.6 the Mach field and the wake recirculation, as far as the z-vorticity contours, are reported. Figure 8.7 instead shows the circular distribution of the pressure coefficient c_p. The latter computed as

$$c_p(\theta) = \frac{p(\theta) - p_\infty}{\frac{1}{2}(\gamma p_\infty M^2)} \tag{8.3}$$

was compared to the one obtained by Dennis and Chang (1970).

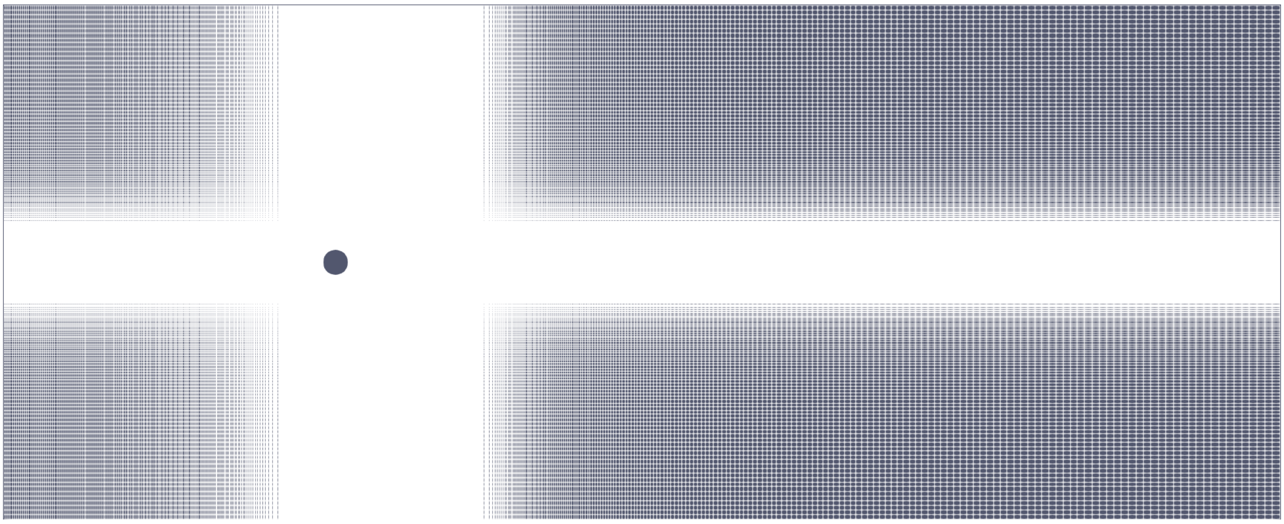
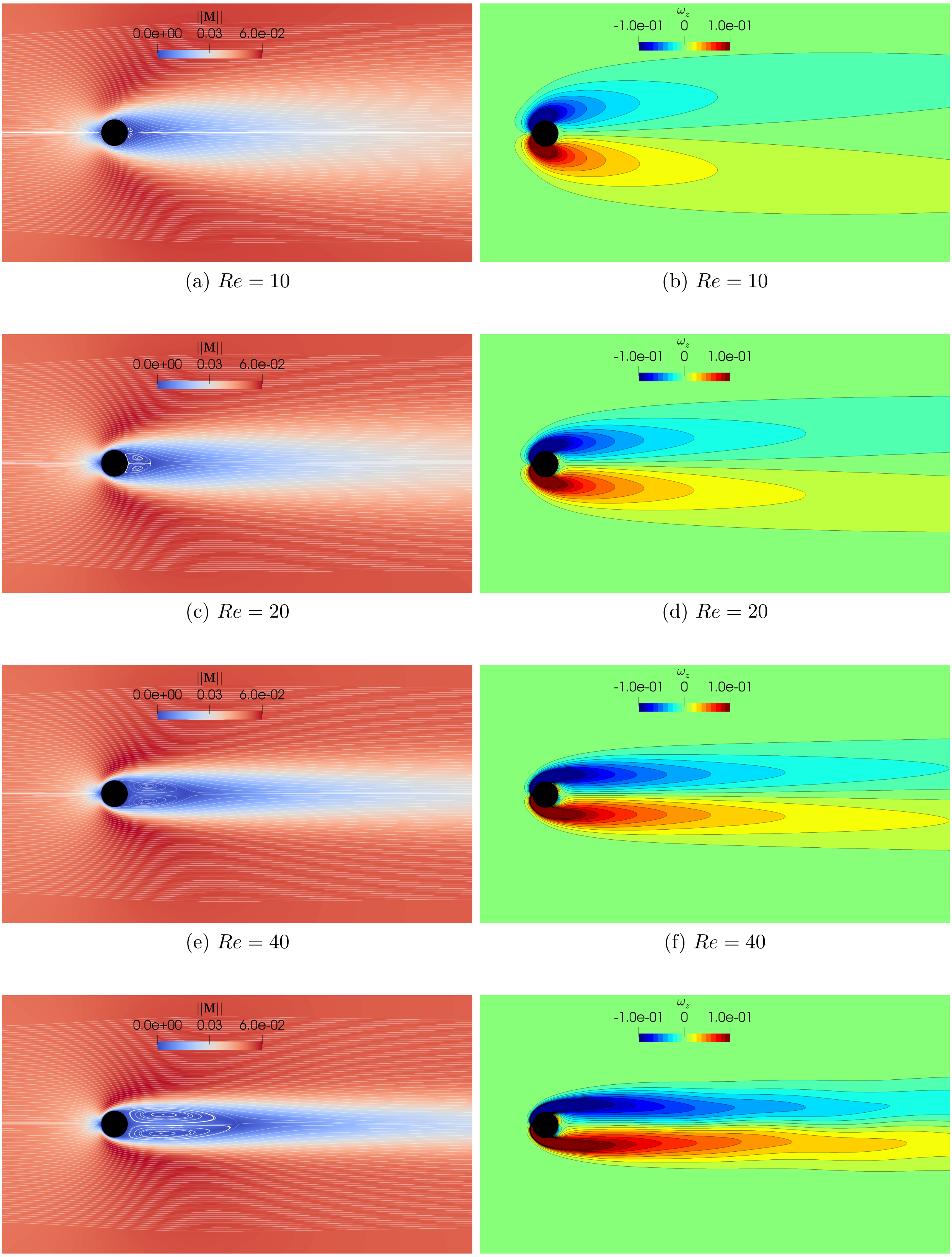


Figure 8.5: Global view of the computational grid employed for the simulation of cylinder in open domain configuration. The latter consisted in a non-uniform Cartesian mesh featuring $N_x \times N_y = 800 \times 320$ computational points for a domain of $[50D; 20D]$; $13D$ were reserved upstream the cylinder centre location and a hyperbolic tangent distribution was employed to cluster the node around the body surface.

	$c_D(Re = 10)$	$c_D(Re = 20)$	$c_D(Re = 40)$	$c_D(Re = 70)$
Dennis and Chang (1970)	2.85	2.05	1.52	1.21
Tritton (2006)	3.07	2.09	1.59	1.30
De Palma et al. (2006)	-	2.05	1.55	-
Linnick and Fasel (2012)	-	2.06	1.54	-
Schneiders et al. (2013a)	2.83	2.15	1.55	-
Boukharfane et al. (2018)	2.99	2.07	1.52	-
Present	2.93	2.09	1.55	1.25

Table 8.2: Drag coefficient literature comparison for a laminar flow past a free cylinder at various Reynolds numbers.



(a) $Re = 10$

(b) $Re = 10$

(c) $Re = 20$

(d) $Re = 20$

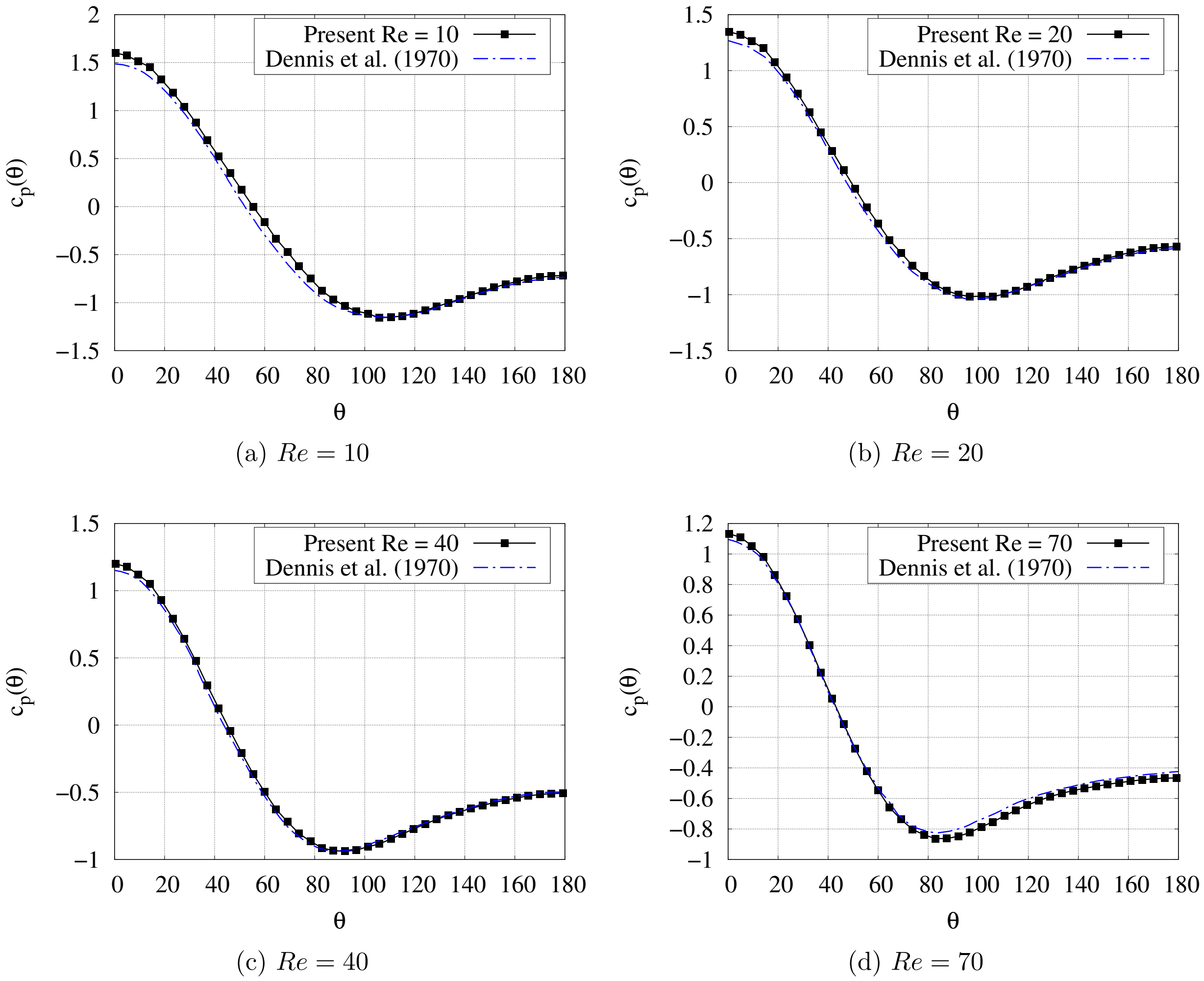
(e) $Re = 40$

(f) $Re = 40$

(g) $Re = 70$

(h) $Re = 70$

Figure 8.6: Cylinder subsonic laminar aerodynamics in open domain configuration. The figure shows the Mach field, the wake recirculation and the vorticity contours (ω_z) for a cylinder at at $Re = \{10{,}20{,}40{,}70\}^T$ and $M = 0.05$.



(a) $Re = 10$

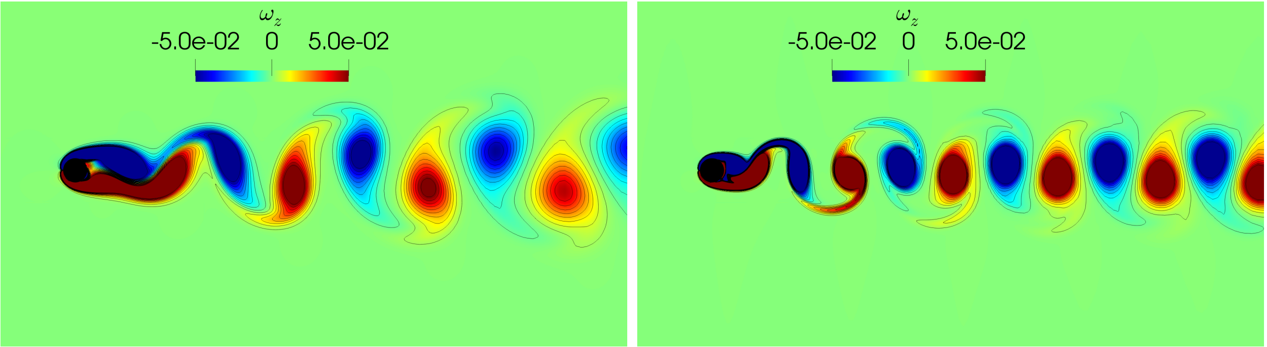
(b) $Re = 20$

(c) $Re = 40$

(d) $Re = 70$

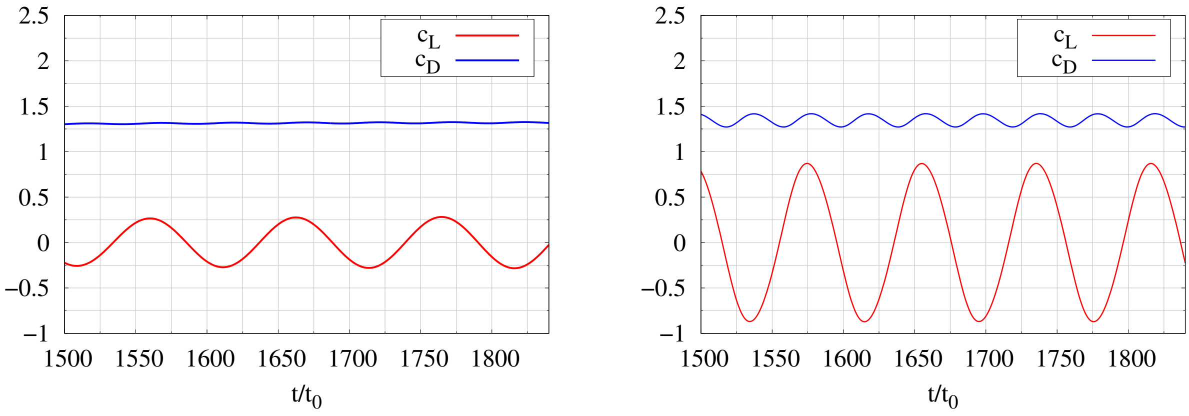
Figure 8.7: Comparison of the pressure coefficient circular distribution around a steady cylinder at $M = 0.05$ and various Reynolds numbers. The figures show the circular distribution of the pressure coefficient for a cylinder in open domain at $Re = \{10, 20, 40, 70\}^T$ and $M = 0.05$. Present results (black points) are compared with the one of Dennis and Chang (1970) (blue dashed lines).

As far as the tests at $Re = \{100, 300\}^T$ are concerned, the inherent unsteadiness of the flow needs to be treated statistically. In particular, we focused our attention on three parameters: the time-averaged drag coefficient, the root mean square of the lift coefficient and the Strouhal number (St) of the wake defined as $St = fD/u_\infty$. Here f denotes the wake frequency, D the cylinder diameter and u_∞ the free stream velocity. In Figure 8.8 the vorticity field is reported while in Figure 8.9 the time history of the lift and drag coefficient is shown. As can be appreciated from the snapshots, the vorticity fields do not show oscillations or noise. The time histories of c_l and c_d are also very clean. Values obtained by present computation always lie in the range reported by other studies. In Table 8.3, the results of the current calculation are compared with the one available in the literature.



(a) $Re = 100$ (b) $Re = 300$

Figure 8.8: Von Karman wake behind a steady cylinder in open domain configuration at $Re = \{100{,}300\}^T$ and $M = 0.05$. Results in term of instantaneous vorticity contours (ω_z).



(a) $Re = 100$ (b) $Re = 300$

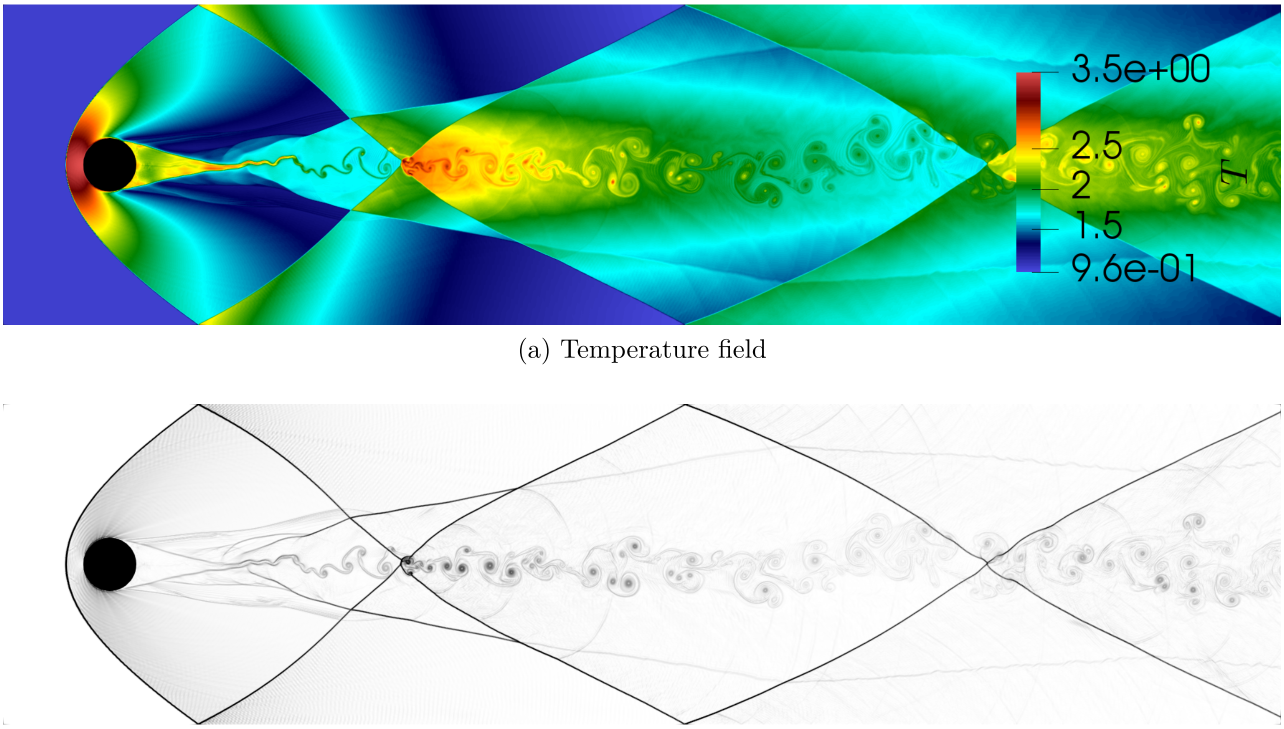
Figure 8.9: Von Karman wake behind a steady cylinder in open domain configuration at $Re = \{100{,}300\}^T$ and $M = 0.05$. Lift and drag history in a statistically-steady range of time.

	c_{Lrms}		$\bar{c}_D$		St	
Re	100	300	100	300	100	300
Ghias et al. (2013)	0.32	0.67	1.36	1.40	0.16	0.21
De Palma et al. (2006)	0.23	-	1.32	-	0.16	-
Rajani et al. (2009)	0.17	0.60	1.33	1.28	0.15	0.21
Boukharfane et al. (2018)	0.25	0.62	1.36	1.26	0.16	0.21
Present	0.22	0.61	1.32	1.34	0.16	0.21

Table 8.3: Literature comparison of force coefficient statistics for a steady cylinder at $Re = 100$ and $Re = 300$ and $M = 0.05$.

8.1.3 Supersonic flow past a confined cylinder

Going on in the validation of the GPFM, here we move the test suite to strong compressible flow conditions. In particular, in this paragraph, the aerodynamics of a confined cylinder in the hypersonic regime will be presented and discussed. The simulation relies on a similar configuration as in Chaudhuri et al. (2011). A *supersonic inflow* boundary condition was enforced at the inlet portion of the domain, while a NSCBC-outflow condition was set at the outlet location; a slip-wall boundary modeled top and bottom sides of the domain. To reproduce as much as possible the inviscid calculation provided by Chaudhuri et al. (2011), the Reynolds number was set equal to $Re = u_\infty D/\nu_a = 10^6$ and a uniform inflow velocity profile was imposed at the left boundary. Because no particular treatments are employed in the near-wall region, such high Reynolds simulation can be considered equivalent to a fully inviscid problem. Here u_∞ denotes the value of the inflow speed, D is the cylinder diameter and ν_a is the ambient kinematic viscosity of the fluid. Ambient conditions were initialised at $t/t_0 = 0$ while the u_∞ value was set in order to ensure a hypersonic Mach number $M = 3.5$ at the inflow position. A grid sensitivity campaign, consisting of three uniform Cartesian meshes, was performed. The meshes feature respectively $N_x \times N_y = [600\times150]; [1200\times300]; [2400\times600]$ grid points for a domain of $[24D; 6D]$ and they are denoted as M_1, M_2, M_3; two diameters were reserved upstream the cylinder centre location. The CFL number was set equal to 0.5, and the simulation was carried out up to a statistical-steady-state.



(a) Temperature field

(b) Schlieren density

Figure 8.10: Cylinder supersonic aerodynamics in a confined configuration. The figure shows the temperature contour 8.10a and the Schlieren density field 8.10b for a confined cylinder at $Re = 10^6$ and $M = 3.5$. The results were obtained on a uniform Cartesian grid featuring $N_x \times N_y = 2400 \times 600$ grid points and for a domains of $[24D; 6D]$.

In Figure 8.10 the temperature instantaneous contour and the Schlieren density field are reported. As we can notice, the flow domain can be divided into two main regions. A portion practically time-independent, immediately after the bow-shock, in between the cylinder location and the curved shock and a region, behind the cylinder, wake-dominated, characterised by a lot of reflected shock-waves interacting with the vortex structures shedded by the cylinder. A quantitative analysis of the computational results can be performed focussing in the first region. In particular, experimental studies suggest the non-dimensional distance between the bow shock location and the cylinder leading edge ($\Delta^* = \Delta/D$) is expressed by

$$\Delta^* = \lambda_1 e^{\frac{\lambda_2}{M^2}} \tag{8.4}$$

where D is the cylinder diameter, while $\lambda_1 = 0.193$ and $\lambda_2 = 4.67$ are fitting parameters. Table 8.4 reports the result obtained with our computation in comparison to the one of Chaudhuri et al. (2011) and Boukharfane et al. (2018) resulting in a good agreement with the literature data.

	$\Delta^*(M_1)$	$\Delta^*(M_2)$	$\Delta^*(M_3)$	equation (8.4)
Chaudhuri et al. (2011)	0.34	0.33	0.31	0.29
Boukharfane et al. (2018)	0.32	0.31	0.30	0.29
Present	0.35	0.33	0.31	0.29

Table 8.4: Bow-shock location Δ^* obtained with different grid resolutions.

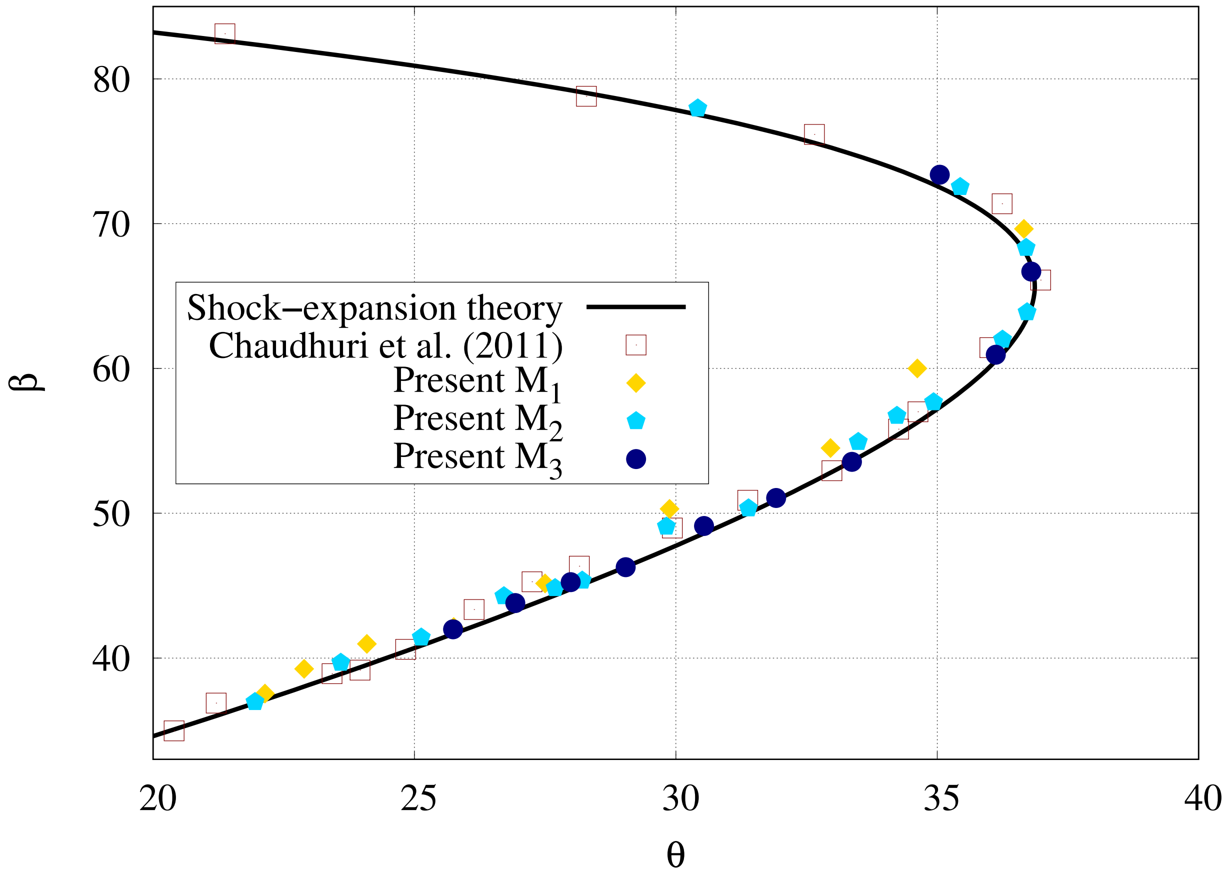


Figure 8.11: Comparison between the shock (β) and the deflection angles (θ) as a function of the grid resolution M_1, M_2, M_3. Solid black line denotes the shock-expansion theory prediction while square points represent the numerical solution obtained by Chaudhuri et al. (2011).

Looking closer to the curved frontal shock, the latter can be studied in light of the supersonic flow theory in stationary conditions. In particular using the equations for oblique shocks, being θ the deflection angle provided by a shock wave and β its physical angles, the shock-expansion theory asserts that θ is a function of β the upstream Mach number through the expression

$$\tan(\theta) = 2\text{cotan}(\beta)\left[\frac{M^2\sin^2(\beta)-1}{M^2(\gamma+\cos(2\beta))+2}\right] \tag{8.5}$$

The latter equation is reported in Figure 8.11 and here, as far as the results in term of $\beta - \theta$ as a function of the grid resolution are compared with the data obtained by Chaudhuri et al. (2011). As we notice, reducing the grid step, the values of the physical and the deflection angles progressively converge to the theoretical solution, proving the phenomenon to be practically stationary and inviscid.

8.1.4 Supersonic flow past a free cylinder

To validate the GPFM in strong compressible and viscous conditions here, similarly to §8.1.2 we simulate the supersonic laminar flow past a cylinder in open domain. NSCBC were imposed at the outlet portion of the domain while extrapolation boundaries were enforced at the top and the bottom boundary. The simulation was performed at $Re = u_\infty D/\nu = 300$. A uniform velocity profile is imposed at the inlet boundary, and here u_∞ denotes the value of the inflow speed, D is the circle diameter and ν the ambient kinematic viscosity of the fluid. Ambient conditions were initialised at $t/t_0 = 0$ while the u_∞ value was set in order to ensure $M = 2.0$ at the inlet location. The computational grid consisted of a structured non-uniform Cartesian grid, stretched around the position of the cylinder and featuring $N_x \times N_y = 1000 \times 500$ grid points for a domain of $[40D; 20D]$. Ten diameters were reserved upstream the cylinder. Grid stretching, following equation (8.2), was applied along both x and y directions and near the cylinder, a uniform grid spacing, approximately with $D/\Delta x = 120$, was employed. The CFL number was set equal to 0.5, and the simulation was carried out up to the steady-state.

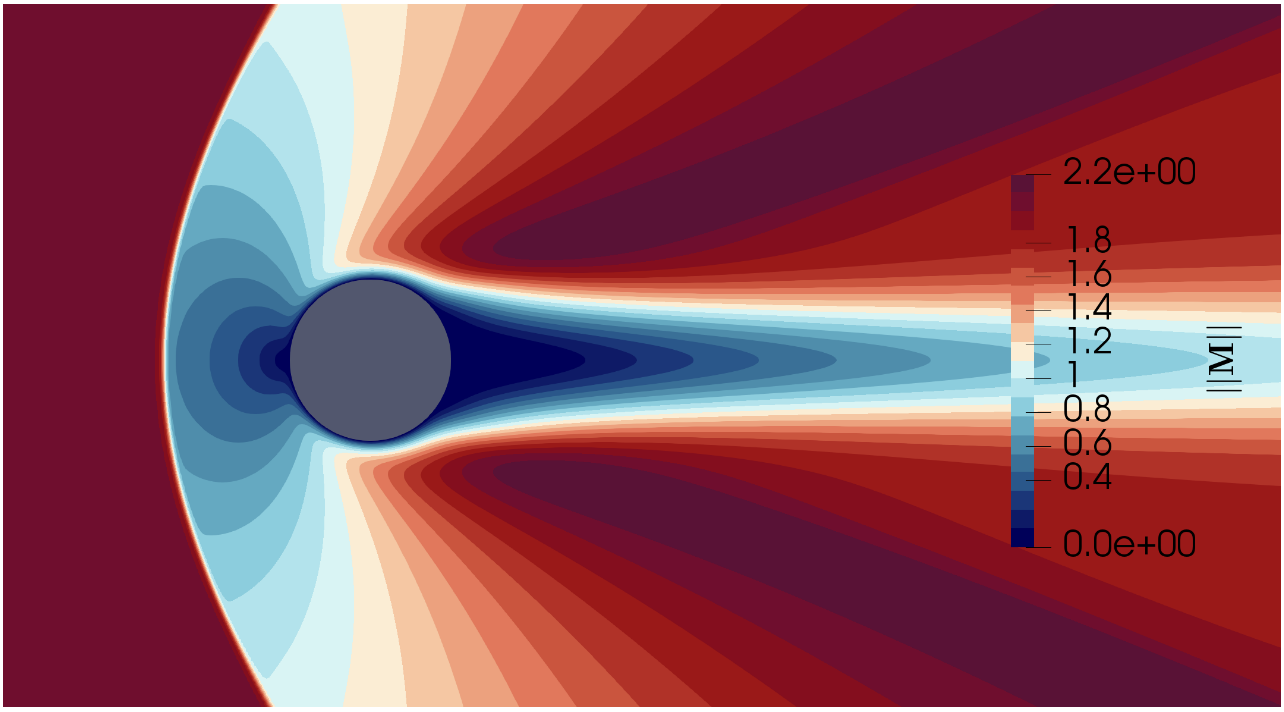


Figure 8.12: Cylinder supersonic aerodynamics in a open domain configuration. The figure shows twenty contours of the Mach field for a cylinder in open domain at $Re = 300$ and $M = 2.0$. The results were obtained on a non-uniform Cartesian grid featuring $N_x \times N_y = 1000 \times 500$ grid points and for a domains of $[40D; 20D]$.

Figure 8.12 shows twenty contours of the Mach field around the cylinder resulting in a very clean and resolved simulation. To quantitive validate the simulation the pressure coefficient was extracted from the flow field and compared with the results of Qu et al. (2018). In particular Figure 8.13 magnifies the pressure coefficient as a function of the y coordinate in the range of $[-0.5 : 0]$ concluding in a very good agreement with literature data.

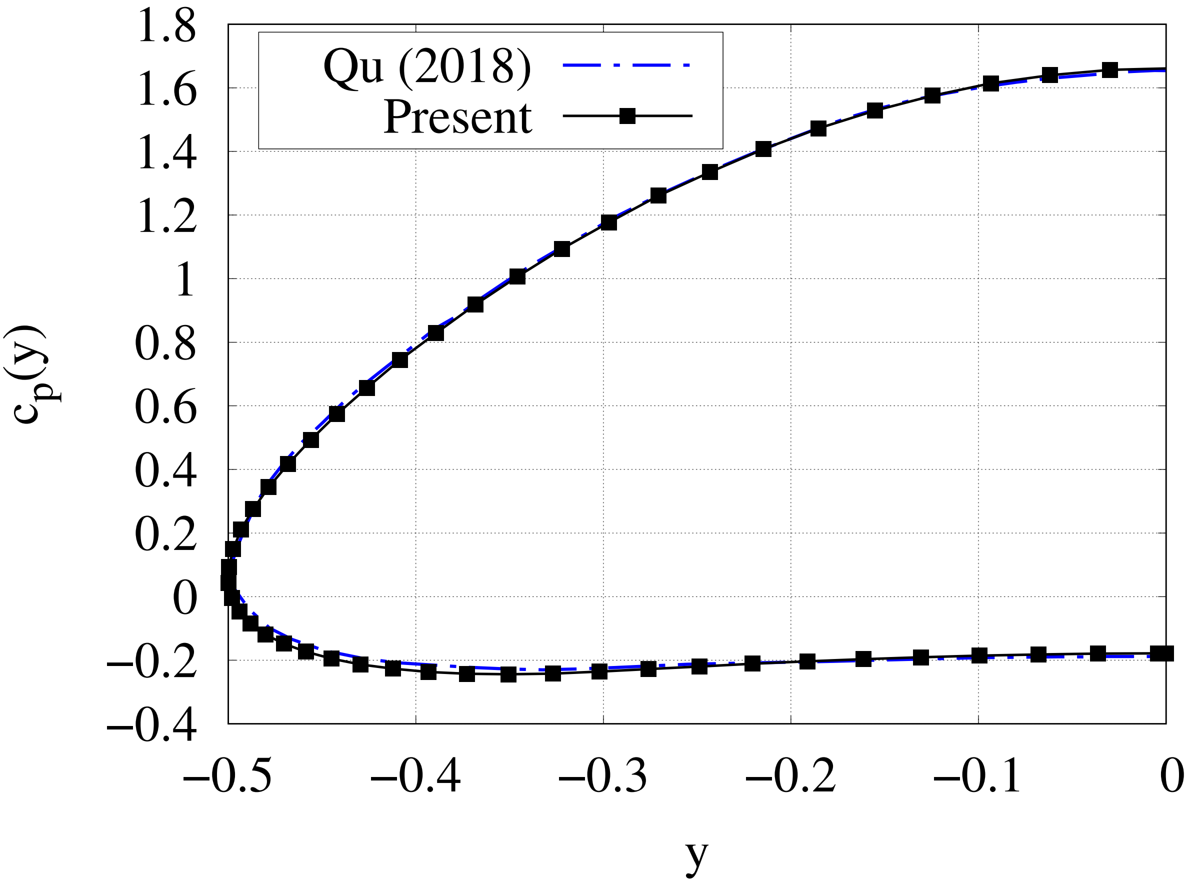
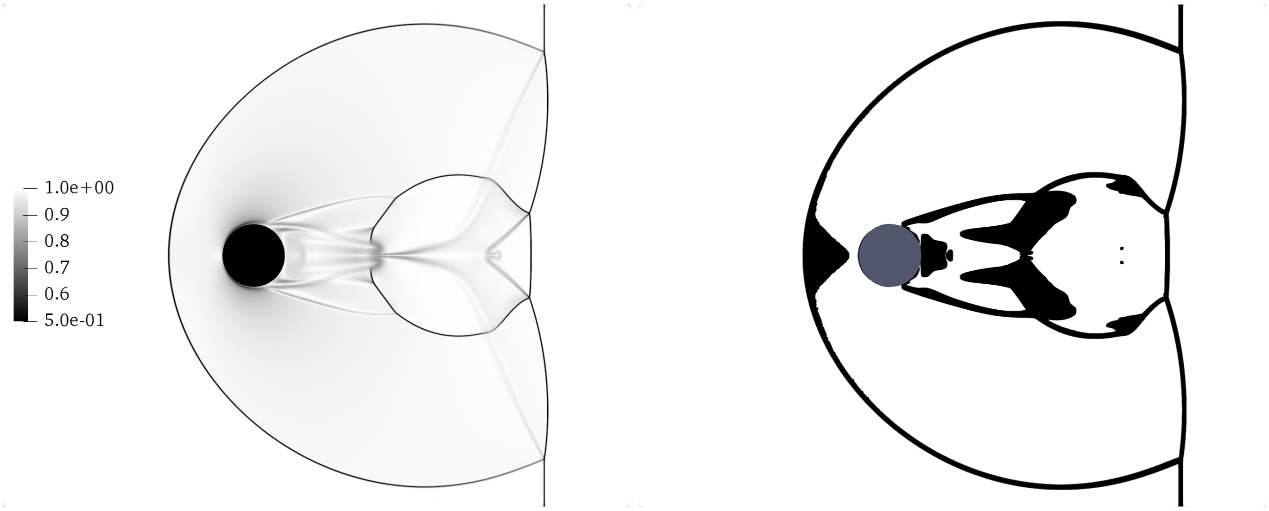


Figure 8.13: Pressure coefficient distribution for a supersonic cylinder in open domain configuration. Present results (black squares) are compared with the simulation of Qu et al. (2018) (blue dashed line).

8.1.5 Shock wave diffraction against a cylinder

In the path of solid objects in strong compressible flow conditions, the present paragraph presents the diffraction of a planar shock wave on a circular cylinder in viscous conditions. The problem has been widely studied in the literature both experimentally and numerically by Bryson and Gross (1961), Chaudhuri et al. (2011), Piquet et al. (2016) and Boukharfane et al. (2018).



(a) Schlieren density

(b) WENO activated cells

Figure 8.14: Shock-wave-diffraction over a steady cylinder at $M_s = 2.81$ and $Re_s = 2000$. In figure 8.14a the shock positions has been highlighted thought their density variation, thus the Schlieren density field is reported. Panel 8.14b shows the WENO-activated-cells. The simulation was performed over a uniform Cartesian grid featuring $N_x \times N_y = 2000 \times 1600$ nodes for a domain of $[10D; 8D]$. Four diameters were reserved upstream the cylinder centre location.

The flow field was initialised in such a way that the Rankine-Hugoniot conditions ensure a traveling shock wave moving from left to right with a Mach number equals to 2.81 (see appendix A for details). In particular, the shock, initially located two diameters upstream the cylinder, divided the flow field into two regions and the left part consisted in a gas initially at rest with $p/p_0 = T/T_0 = 1$. The Reynolds number of the flow $Re = u_s D/\nu_a$ was set equal to 2000. Here u_s denotes the shock speed, D the cylinder diameter and ν_a the ambient kinematic viscosity of the fluid. Extrapolation boundary conditions were ensured in all the four domains edges. The domain size was set equal to $L_x \times L_y = [10D; 8D]$ and the cylinder was located at zero coordinates so that four diameters were reserved upstream the cylinder. The computational grid consisted in a structured uniform mesh featuring $N_x \times N_y = 2000 \times 1600 = 3.2 \cdot 10^6$ nodes. A number of points equal to $D/\Delta x = 200$ was employed around the cylinder. The CFL number was set equal to 0.5.

The complexity of the wave interactions is depicted in Figure 8.14a. Here the numerical Schlieren of the density field is reported at a reduced time equals to $t/t_0 = 2$. The flow field shows a strong symmetry concerning the centre line of the domain, and it evidences some characteristics features such as Mach stems, triple points, shocks, slip lines, and vortices. In particular, the trajectory of the two main triple points was extensively reported by previous studies and here, in Figure 8.15 the results of our computation are compared with the one reported in [15], [19], [89] and [12]. As we notice, in the limit of the uncertainty related to the Bryson and Gross (1961) experiments and looking at the numerical results provided by Chaudhuri et al. (2011), Piquet et al. (2016) and Boukharfane et al. (2018) our simulations show an excellent agreement compared to all the available data, concluding the perfect prediction of our methodology.

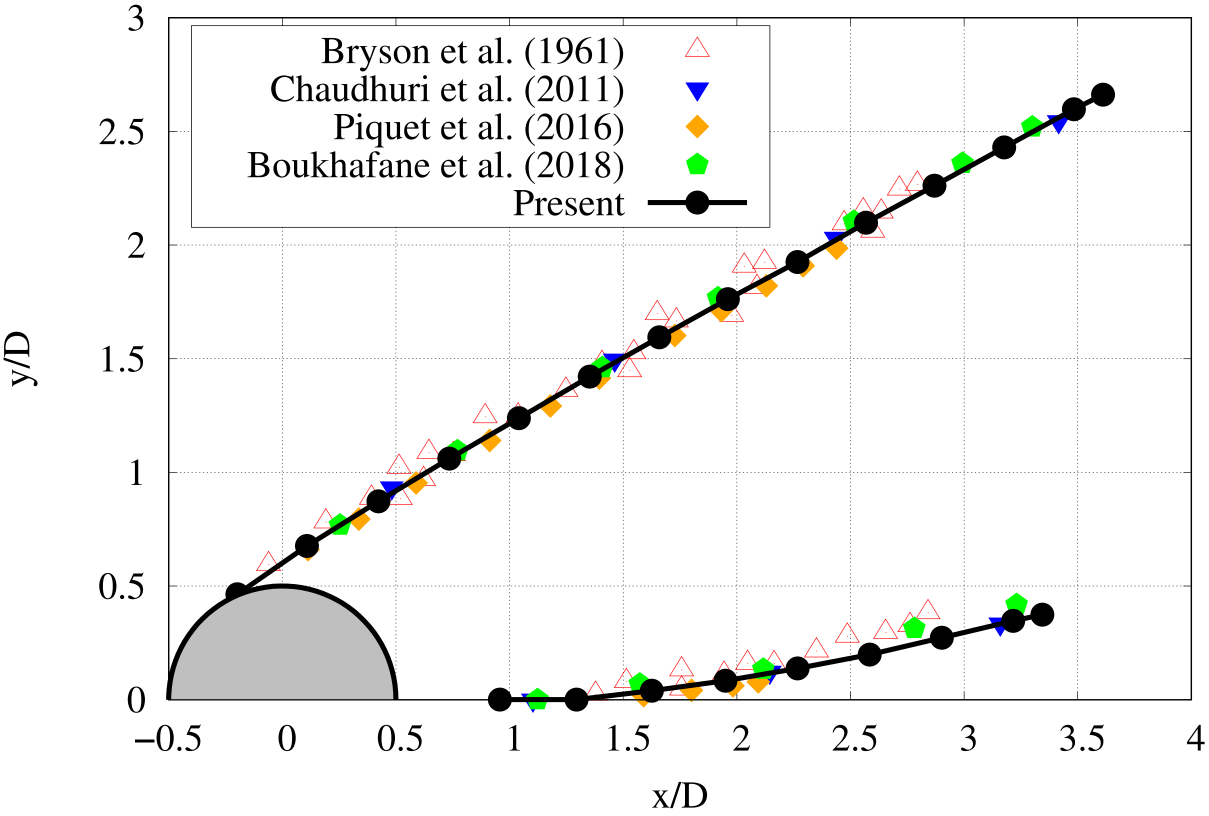
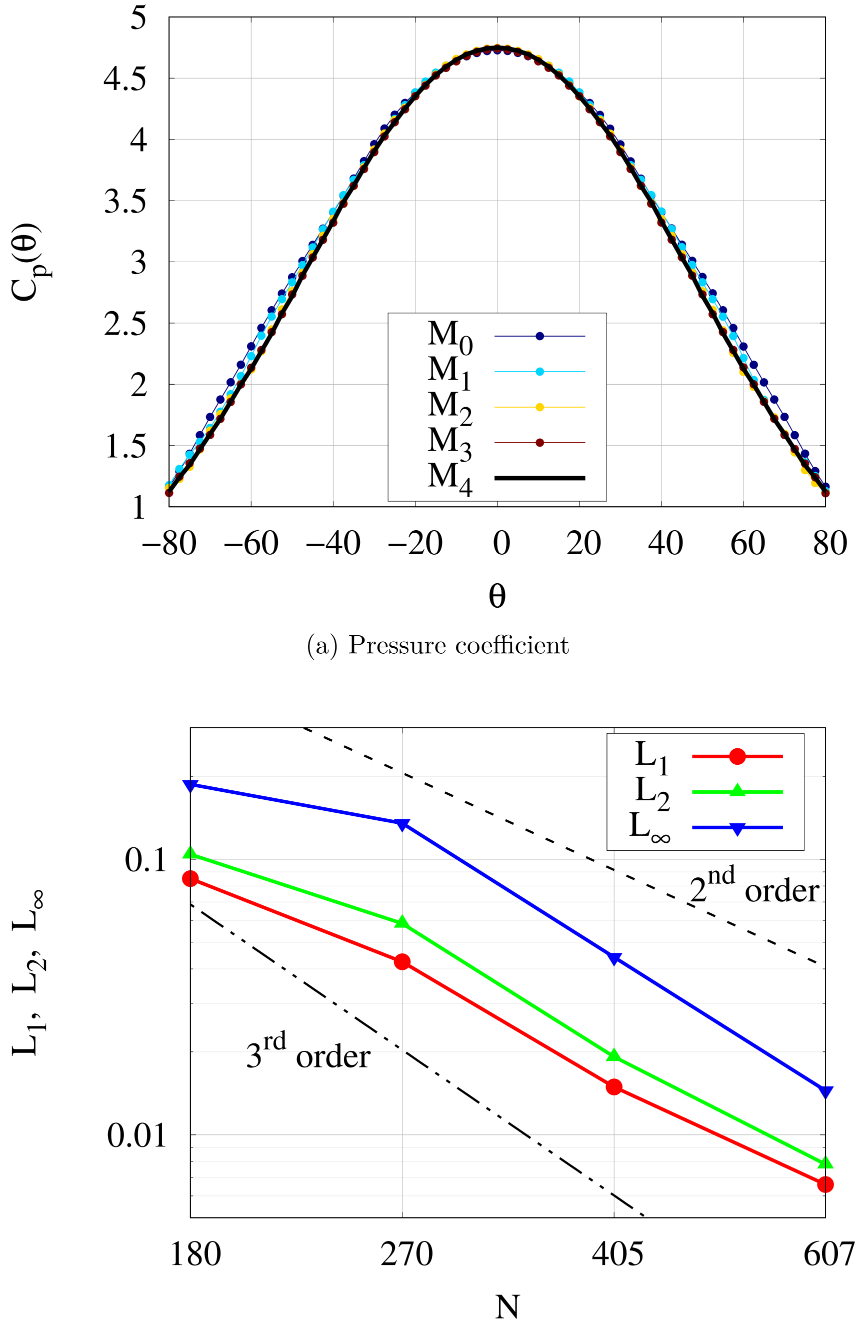


Figure 8.15: Shock-wave-diffraction over a steady cylinder at $M_s = 2.81$ and $Re_s = 2000$. Literature comparison of the triple points trajectory for the problem is reported. Present results are compared with the experimental data of Bryson and Gross (1961) and the numerical simulations of Chaudhuri et al. (2011), Piquet et al. (2016) and Boukharfane et al. (2018).

8.1.6 Immersed boundary error scaling

To quantify the scaling of the numerical error of the overall immersed boundary methodology the diffraction of a planar shock wave against a steady cylinder was found a significative and challenging benchmark. The problem resumes all the ingredients of the present method (i.e. scheme hybridisation, shock-wave automatic detection, viscous effects and time-dependency). The simulation of the previous paragraph has been repeated over a set of five uniform Cartesian meshes $\{M_i\}_{i=0}^{4}$, featuring $N_x \times N_y \simeq [180 \times 180] * \{1.5^i\}_{i=0}^{4}$ grid points for a domain of $L_x \times L_y = [8D \times 8D]$. The shock Mach and Reynolds numbers were imposed equal to $M_s = 2.81$ and $Re_s = 1000$ while the CFL number - to avoid any uncertainty due to the error contribution of the time-integration - was set equal to 0.1. To assess the error scaling the surface pressure coefficient $C_p(\theta)$ has been monitored as a function of the grid resolution and the one computed with the i-th grid-level was compared with M_4 results.



(a) Pressure coefficient

(b) Error scaling

Figure 8.16: Surface pressure coefficient and its error scaling as a function of the grid resolution. Figure 8.16a shows the pressure coefficient for a $M = 2.81$, $Re = 1000$ planar shock wave against a steady cylinder. Panel 8.16b reports the error scaling of the pressure coefficient. The data were collected at $t/t_0 = 1.5$ for four uniform Cartesian meshes whose resolution was set ensuring a grid ratio $r = 1.5$. The case with $N_x \times N_y = 180 \times 180 * \{1.5^4\}$ was considered as the reference solution. From the obtained results the overall accuracy of the entire algorithm is in between of the second and the third-order, consistently with the nominal second order of accuracy of the bilinear interpolation.

The complexity of the flow - which features shocks and acoustics wave behind the cylinder - makes the error evaluation quite difficult. In particular, is well known that the WENO5 scheme reduces to zero-order of accuracy in the region where the flow is not smooth. For this reason, the ±80° shock-free frontal circular sector was selected as the control zone in which to monitor the error behaviour. The comparative process has been carried out over $n = 64$ control points (i.e. every 2.5°) inside the control region and the analysis has been performed looking at the results at $t/t_0 = 1.5$. The planar shock wave has been initialised two diameters upstream of the cylinder centre. The L_p and the L_∞ norms defined as

$$L_p^{M_i} = \left(\frac{1}{n} \sum_{j=1}^{n} \left| C_p^{M_i}(\theta_j) - C_p^{M_4}(\theta_j) \right|^p \right)^{1/p} \tag{8.6}$$

$$L_\infty^{M_i} = \max_{j=1}^{n} \left| C_p^{M_i}(\theta_j) - C_p^{M_4}(\theta_j) \right| \tag{8.7}$$

have been retained as metrics of the error and in particular, the $p = 1$ and $p = 2$ cases have been used in present analysis. The order of accuracy was evaluated with

$$q = \frac{\log\left(L_{(\cdot)}^{M_i} / L_{(\cdot)}^{M_{i-1}} \right)}{\log(1/r)} \tag{8.8}$$

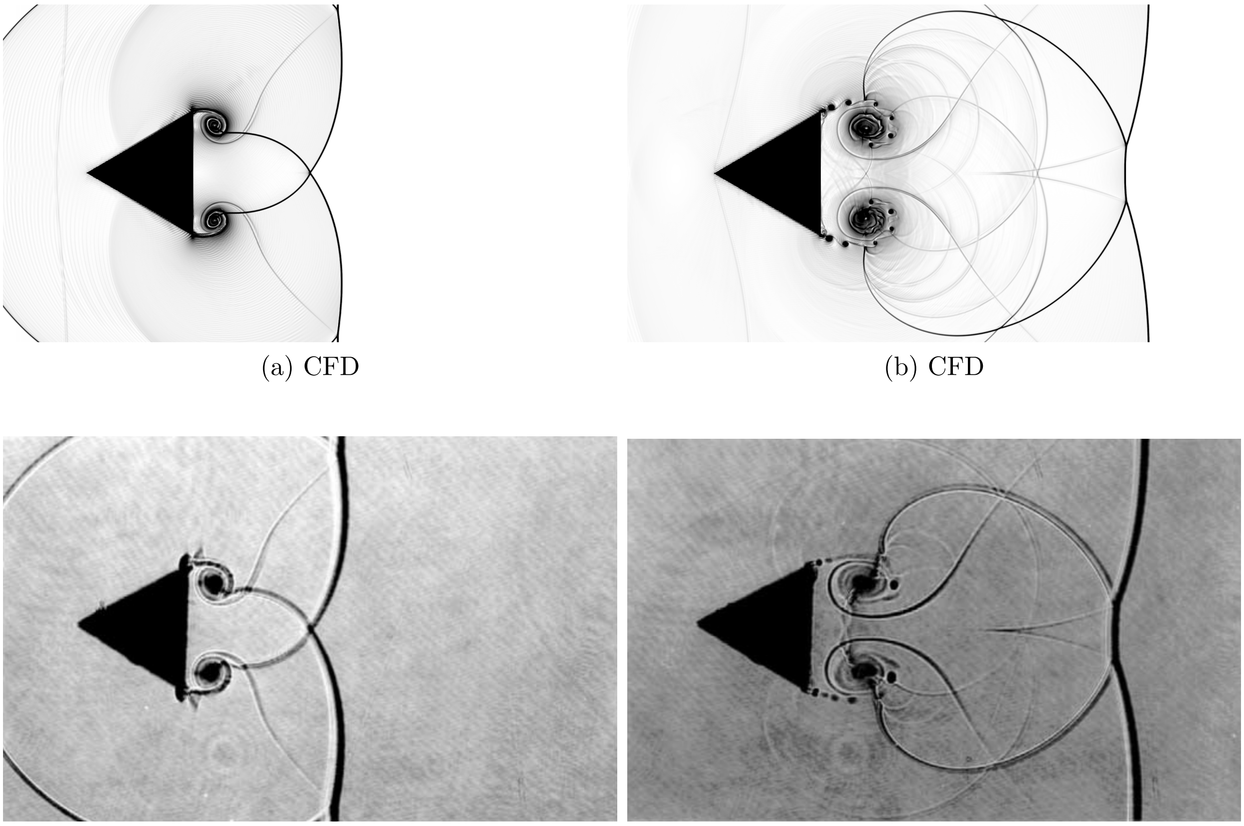
with $r = 1.5$ representing the grid ratio of the present case. In Figure 8.16a the pressure coefficient in the control zone is reported as a function of the grid resolution while Figure 8.16b shows the evolution of the metrics associated with the numerical error. Finally, table 8.16 report the computed errors and the related order od accuracy. For the obtained results we conclude that the present immersed boundary method - in combination with the fluid dynamics solver - keeps the accuracy in between the second and the third-order resulting in an excellent result. In particular, the latter result is fully consistent with the nominal second-order of accuracy of the boundary interpolation method.

N	L_1	L_2	L_∞	L_1 order	L_2 order	L_∞ order
180	0.827335E-01	0.102169E+00	0.190200E+00	-	-	-
270	0.471348E-01	0.617132E-01	0.138030E+00	1.388	1.243	0.791
405	0.144535E-01	0.188992E-01	0.441100E-01	2.915	2.919	2.814
607	0.683969E-02	0.809366E-02	0.146200E-01	1.845	2.092	2.724

Table 8.5: Error scaling of the present methodology for the surface pressure coefficient.

8.1.7 Shock wave diffraction against a wedge

To conclude the validation of the GPFM for steady objects here we present interaction between a planar shock wave and an equilateral wedge. The problem, known also as Schardin's problem, was extensively studied by previous authors both experimentally [17] and numerically [19, 12].



(a) CFD

(b) CFD

(c) Experiments

(d) Experiments

Figure 8.17: Shock-wave-diffraction over a wedge at $M_s = 1.3$ and $Re_s = 10^6$. In figure the numerical Schlieren density field is reported at two different time level and compared with the experimental results of Chang and Chang (2000). The simulation was performed over a uniform Cartesian grid featuring $N_x \times N_y = 2000 \times 2000$ nodes for a domain of $[7L_0; 7L_0]$ where L_0 is the triangle edge height. Three reference lengths were reserved upstream the wedge vertical edge.

To represent the dynamics of the problem a similar configuration to the one explained in §8.1.5 was employed, thus the flow field was initialised in such a way that the Rankine-Hugoniot conditions ensure a traveling shock wave moving from left to right with a Mach number equals to 1.3 (see appendix A), following the experiments of Chang and Chang (2000). The left part of the domain consisted in a gas initially at rest with $p/p_0 = T/T_0 = 1$ while the Reynolds number of the flow $Re = u_s L_0/\nu_a$ was set equal to 10^6. Here u_s denotes the shock speed, L_0 wedge edge height and ν_a the ambient kinematic viscosity of the fluid. Extrapolation boundary conditions were ensured in all the four domains edges. The domain size was set equal to $L_x \times L_y = [7L_0; 7L_0]$ and the vertical edge of the wedge was located in such a way that three reference length would be reserved upstream. The mesh consisted in a structured uniform mesh featuring $N_x \times N_y = 2000 \times 2000$ nodes and the CFL number was set equal to 0.5. The computation was performed up to $t/t_0 = 5$.

In Figure 8.17 the Schlieren density at two different time levels is reported. Here, as far as in the cylinder diffraction case, lots of discontinuities appear behind the wedge, detailing the complexity of the flow. In particular, three main features can be highlighted, consisting of two main triple points and a vortex-pair (a detailed description of the flow field can be found in [17]). Figure 8.18 reports the trajectories of those features compared to the one available in literature resulting in an excellent agreement also for this validation benchmark.

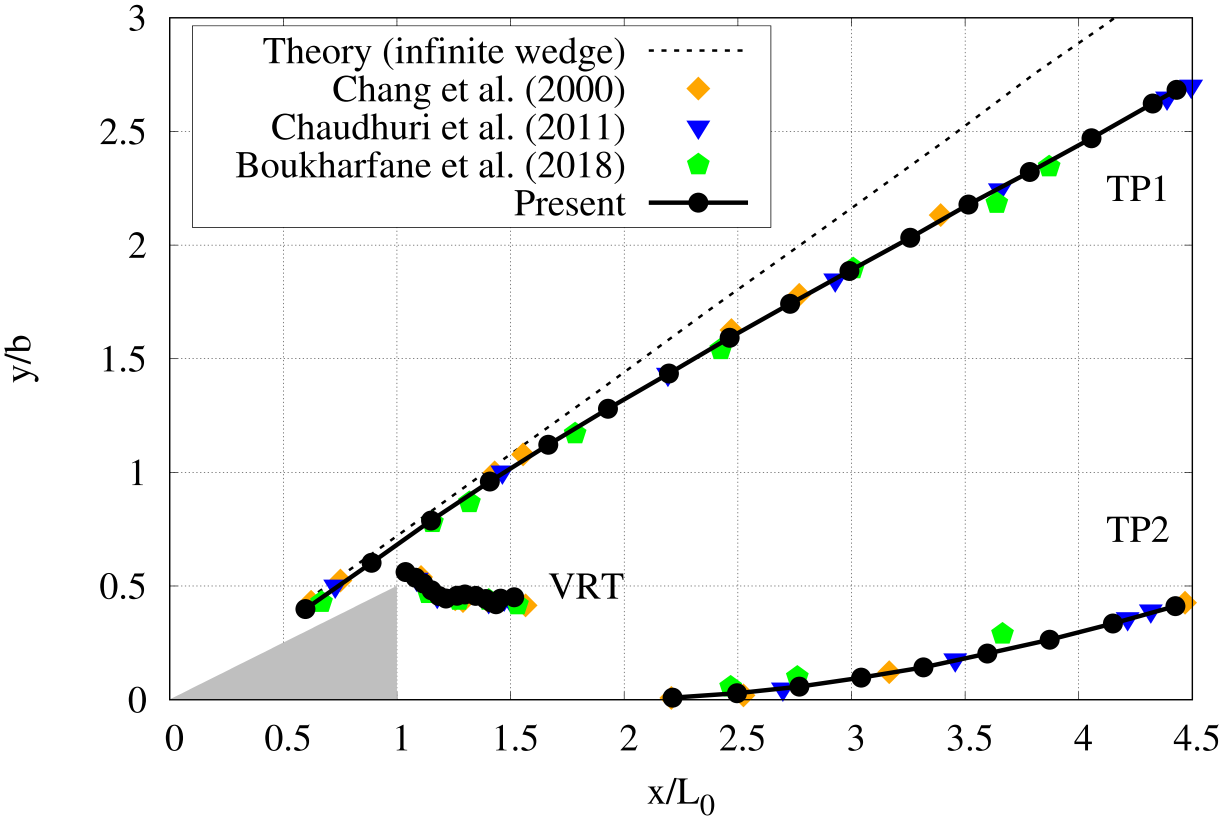


Figure 8.18: Shock-wave-diffraction over a wedge at $M_s = 1.3$ and $Re_s = 10^6$. Literature comparison of the triple points trajectory and vortex centre location for the problem is reported. L_0 is the wedge height while $b = L_0 \tan(\pi/3)$ is the edge length. Present results are compared with the experimental data of Chang and Chang (2000) and the numerical simulations of Chaudhuri et al. (2011) and Boukharfane et al. (2018).

8.2 Objects aerodynamics with moving boundaries

The IBM we developed consists of a general numerical strategy able to accurately simulate both steady and moving boundaries. The present section aims at validating the GPFM for moving objects in compressible viscous flow conditions. A detail description of the test suite applied to nearly incompressible up to strong compressible flows will be presented and discussed.

8.2.1 Subsonic flow past a moving cylinder with prescribed motion

The first test we considered in the validation of the GPFM applied to moving objects consist of a transversely oscillating cylinder in a free-stream configuration and low-Mach conditions. Such case is quite well documented both numerically and experimentally (see e.g.[49, 134, 111, 31]). Here the same domain configuration described in §8.1.2 was employed.

NSCBC were used at the inlet and outlet portion of the domain while non-slip wall conditions were employed at the top and the bottom boundaries. To avoid any confinement effect due to the motion of the cylinder, the transverse dimension of the domain was doubled compared to the steady case so that $L_y/D = 40$; the stream-wise dimension remains the same. We simulated the flow at $Re = u_\infty D/\nu = 185$ where u_∞, D and ν follow the same convections explained in §8.1.2. The value of u_∞ was selected in such a way that a Mach number $M_\infty = 0.25$ is ensured in the free stream conditions following the computation performed by Ehsan Khalili et al. (2018). A grid sensitivity campaign, consisting in three non-uniform Cartesian meshes stretched around the cylinder, was performed. The three meshes feature $N_x \times N_y = [400 \times 320]; [800 \times 640]; [1600 \times 1280]$ grid points are denoted as *coarse*, *medium* and *fine*, respectively. Grid stretching was applied in both stream-wise and span-wise directions but, near the cylinder, a uniform grid spacing was employed. In particular, the three meshes were characterised by a minimum grid step equals to $\Delta x = \Delta y = [D/25]; [D/50]; [D/100]$, respectively. The initial values for the non-dimensional pressure and temperature were set to $p/p_0 = T/T_0 = 1$ respectively, while the velocity components were set equal to the free stream speed in all the domain. The CFL number was set equal to 0.5.

The body motion was prescribed using a single harmonic oscillation in the form of

$$y_c(t) = A \sin(2\pi f_e t) \tag{8.9}$$

and here t is the dimensionless simulation time, $y_c(t)$ is the location of cylinder centre, A is the amplitude associated to the harmonic oscillation and f_e is its frequency. Following the available literature the amplitude was set equals to $A/D = 0.2$ while the frequency f_e was expressed as a function of the cylinder natural shedding frequency $f_0 = St_0 D/u_\infty$. Here St_0 denotes the Strouhal number of a steady cylinder under the same Mach and Reynolds regime and experimental and numerical results suggest that St_0 is in the range of $[1.85 \div 1.95]$. In our computation we set $St_0 = 0.195$. Following the experiments of Gu et al. (1994) and the simulations of Guilmineau and Queutey (2002) and Schneiders et al. (2013b) a spectrum of six frequencies was considered, i.e. $f_e/f_0 = \{0.80, 0.90, 1.00, 1.10, 1.12, 1.20\}^T$. In Table 8.6 the force coefficient statistics for a moving cylinder with a reduced frequency of $f_e/f_0 = 0.8$ are reported in comparison with the available literature and as function of the grid spacing, while in Figure 8.19 the results over the complete frequency spectrum are shown, proving a very good agreement with other studies validating our method for moving objects.

	M	$\bar{c}_D$	c_{Lrms}	c_{Drms}
Guilmineau and Queutey (2002)	0	1.195	0.080	0.036
Yang et al. (2009)	0	1.281	0.076	0.042
Schneiders et al. (2013b)	0.10	1.279	0.082	0.042
Ehsan Khalili et al. (2018)	0.25	1.287	0.079	0.045
Present (coarse)	0.25	1.248	0.063	0.044
Present (medium)	0.25	1.263	0.058	0.043
Present (fine)	0.25	1.272	0.065	0.043

Table 8.6: Literature comparison of force coefficients statistics for a moving cylinder in low Mach regime, $Re = 185$ and with a reduced frequency $f_e/f_0 = 0.8$.

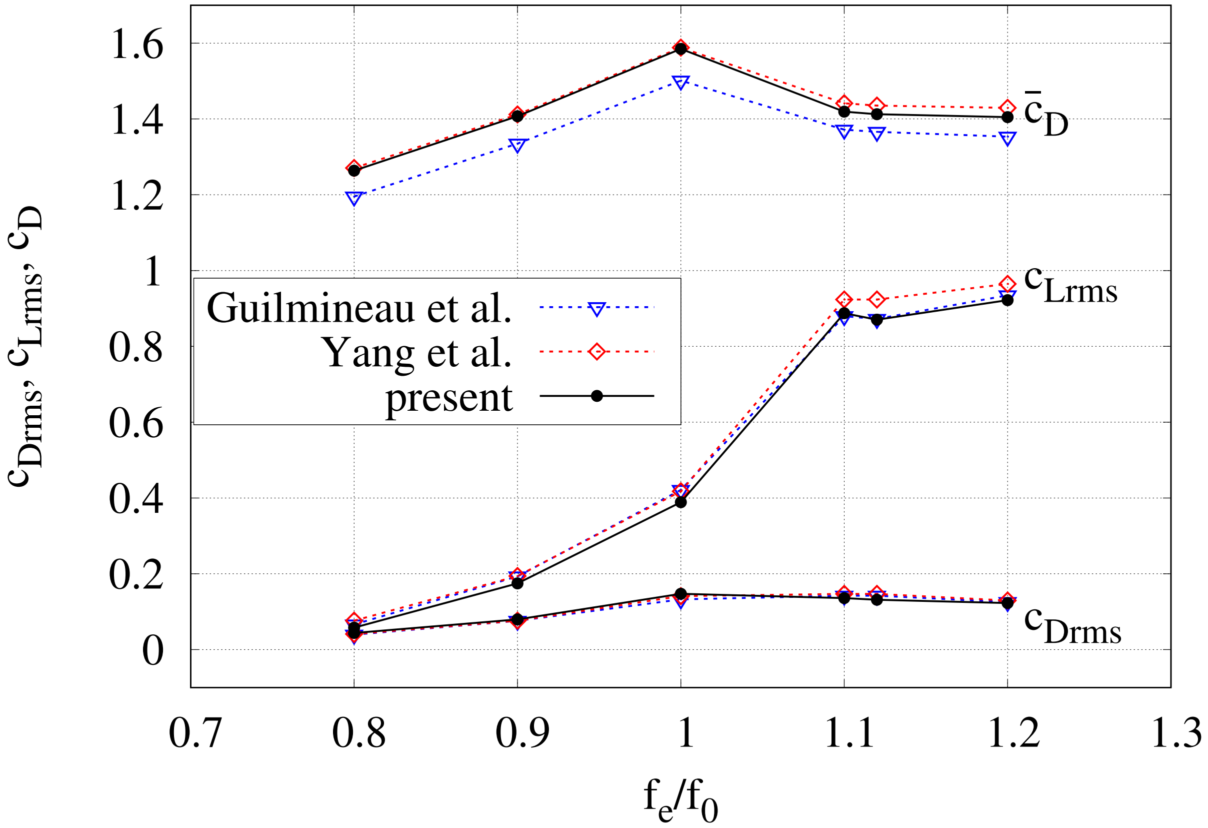
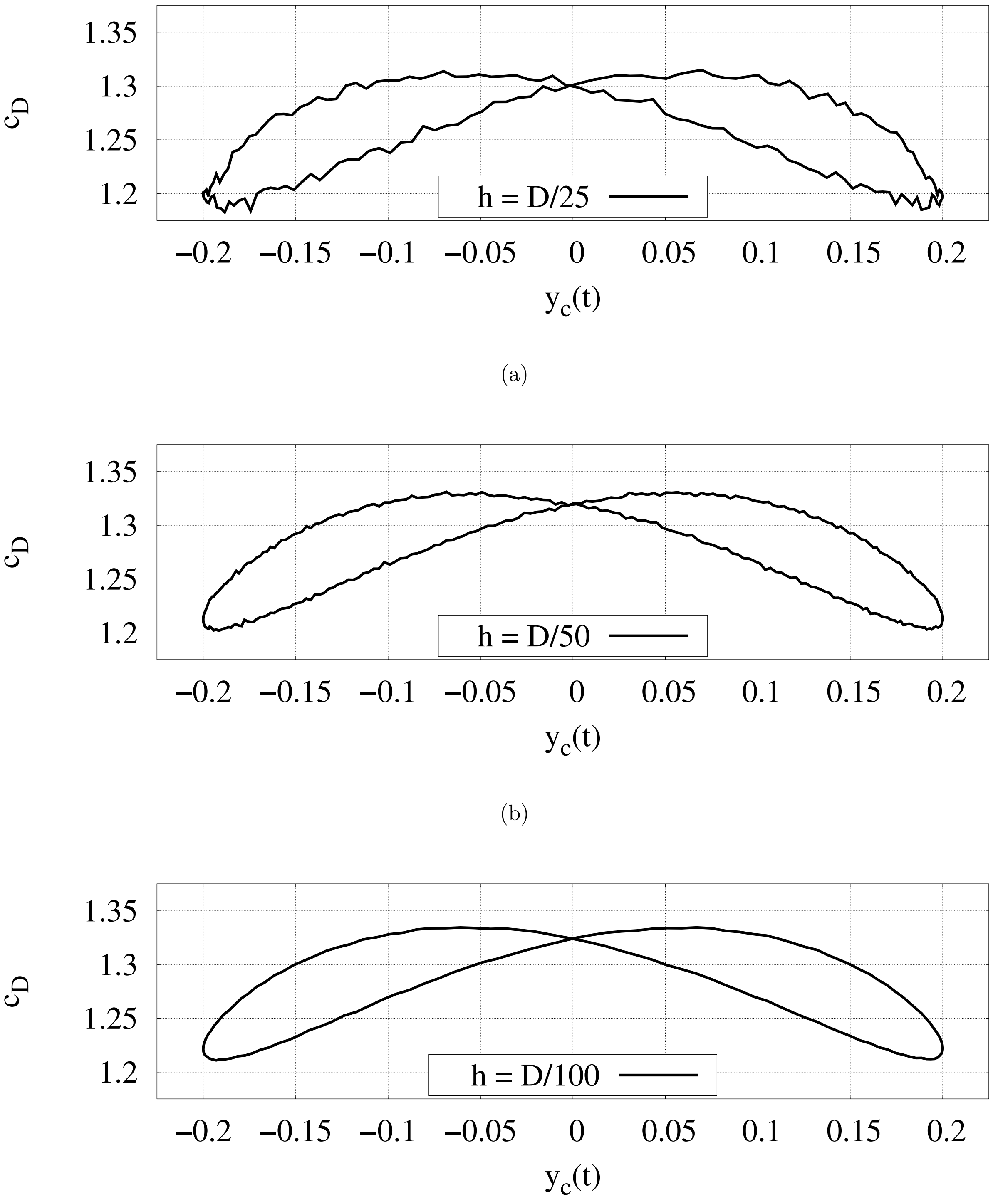


Figure 8.19: Variation of the force coefficients statistics for a transversely oscillating cylinder at $Re = 185$ and $M = 0.25$ respect of f_e/f_0. In the figure our computation (black circles) is compared with Guilmineau and Queutey (2002) (blue dashed triangles) and Yang and Balaras (2006) (red dashed squares).

To highlight the low level of noise introduced by the present method, in Figure 8.20 the drag coefficient for the case at $f_e/f_0 = 0.8$ has been plotted as a function of the location of the cylinder. Magnifying the drag history over a single period, we can see that even without introducing any additional dissipation the signal is very clean. In Figure 8.21 the time history of the lift and drag coefficient is reported for all the reduced frequency spectrum, while in Figure 8.22 the z-vorticity contours are shown.

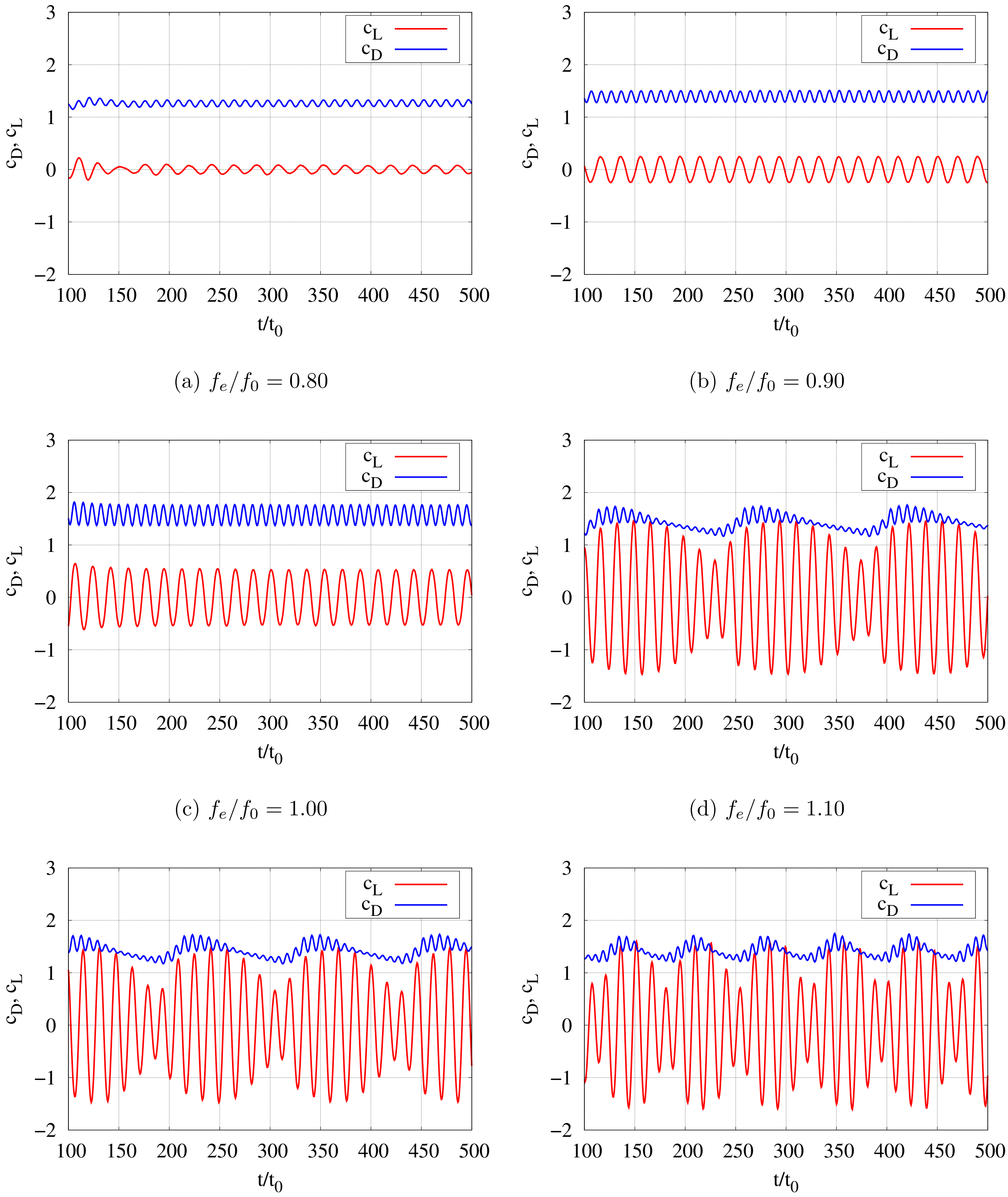


(a)

(b)

(c)

Figure 8.20: Resolution comparison at different grid spacing. Here the drag coefficient for the case at $f_e/f_0 = 0.8$ is plot as a function of the cylinder centre location over a single period. Three results, with an increasing level of resolution $h = \{D/25, D/50, D/100\}$ are reported respectively in panel 8.20a, 8.20b and 8.20c.



(a) $f_e/f_0 = 0.80$

(b) $f_e/f_0 = 0.90$

(c) $f_e/f_0 = 1.00$

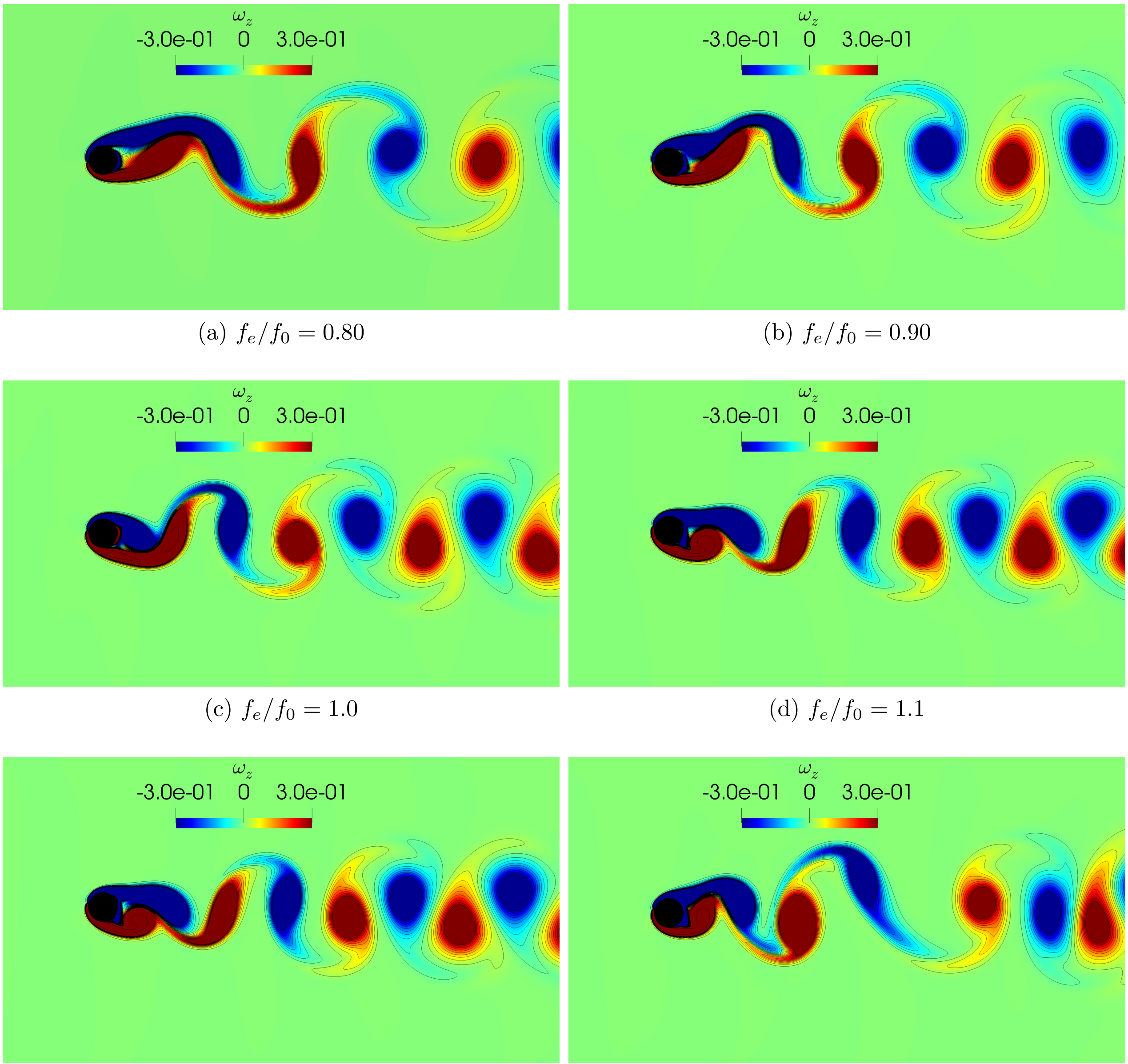
(d) $f_e/f_0 = 1.10$

(e) $f_e/f_0 = 1.12$

(f) $f_e/f_0 = 1.20$

Figure 8.21: Time history of the lift and drag coefficient past a moving cylinder in a cross flow configuration with $Re = 185$, $M = 0.25$ and a $0.8 \leq f_e/f_0 \leq 1.2$. Here the results are associated to the mesh with $\Delta x = \Delta y = D/50$.

According to the results of our simulations and in particular looking at Figure 8.19 some comments are demanding to discuss the complex dynamics related to a moving cylinder in a transitional Reynolds regime. Looking at the case at $f_e/f_0 < 1$ we can easily conclude that force coefficients statistics depend linearly to this ratio. On the other hand, increasing the wavenumber, the trend is not confirmed, and something happens in the flow domains. Looking closer to the Figure 8.22 we can observe that, up to $f_e/f_0 > 1$ the interaction between the body motion and the vortex shedding becomes strongly non-linear, resulting in a vortex an opposite circulation detached from the lower side of the cylinder. This fact causes the force exchange to suddenly invert its behaviour, showing a typical knot effect in the system dynamics. The phenomenon is clearly highlighted also in Figure 8.21. Here the force coefficient time history, up to $f_e/f_0 = 1$, exhibits a secondary frequency, higher than the fundamental one, resulting in a secondary vortex shedding. Above the good agreements with the results available in the literature, from this analysis, we can conclude that our methodology is properly predictive in a wide range of body motions and also the strong non-linear effects related to fluid-structure interactions are properly caught and resolved.



(a) $f_e/f_0 = 0.80$

(b) $f_e/f_0 = 0.90$

(c) $f_e/f_0 = 1.0$

(d) $f_e/f_0 = 1.1$

(e) $f_e/f_0 = 1.12$

(f) $f_e/f_0 = 1.2$

Figure 8.22: Non-dimensional vorticity contours (ω_z) of a cross flow oscillating cylinder at $M = 0.25$ and $Re = 185$. Here the results are presented in function of the oscillation frequency $0.8 \leq f_e/f_0 \leq 1.2$.

8.2.2 Moving cylinder against a steady shock wave

Here we show the performance of the present method concerning moving objects in compressible conditions. The same case described in the previous §8.1.5 is repeated in a difference reference frame. Here we will focus on the frame jointed to the shock in which the cylinder is moving from left to right. Due to the Galilean invariance, the physics in the two frame is identical once the translational speed is accounted for. However, it should be remarked that the intrinsic non-linearity of WENO scheme does not guarantee a perfect Galilean invariance transformation.

A similar set up as previous paragraph was employed. A Cartesian grid featuring $N_x \times N_y = 700 \times 400$ grid points was employed while the domain sized $L_x \times L_y = [14D; 8D]$. The Reynolds and the Mach numbers were set equal to 2000 and 2.81 and followed the same convections in §8.1.5.

In Figure 8.23, the drag coefficient history is plotted in the two references. As we can notice, a perfect agreement is found with the small differences barely noticeable. In Figure 8.24 two instantaneous fields in terms of density contours are shown comparing the steady with the moving cylinder concluding that our method is properly predictive also in case of moving objects in strong compressible conditions with shock waves.

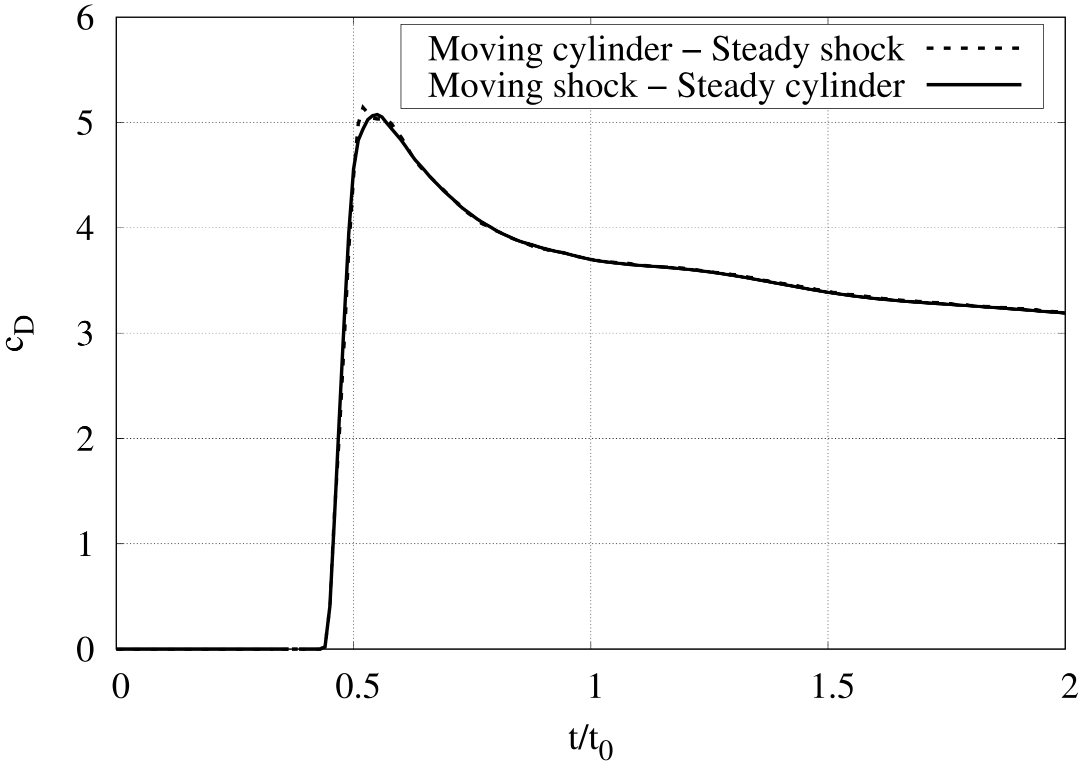
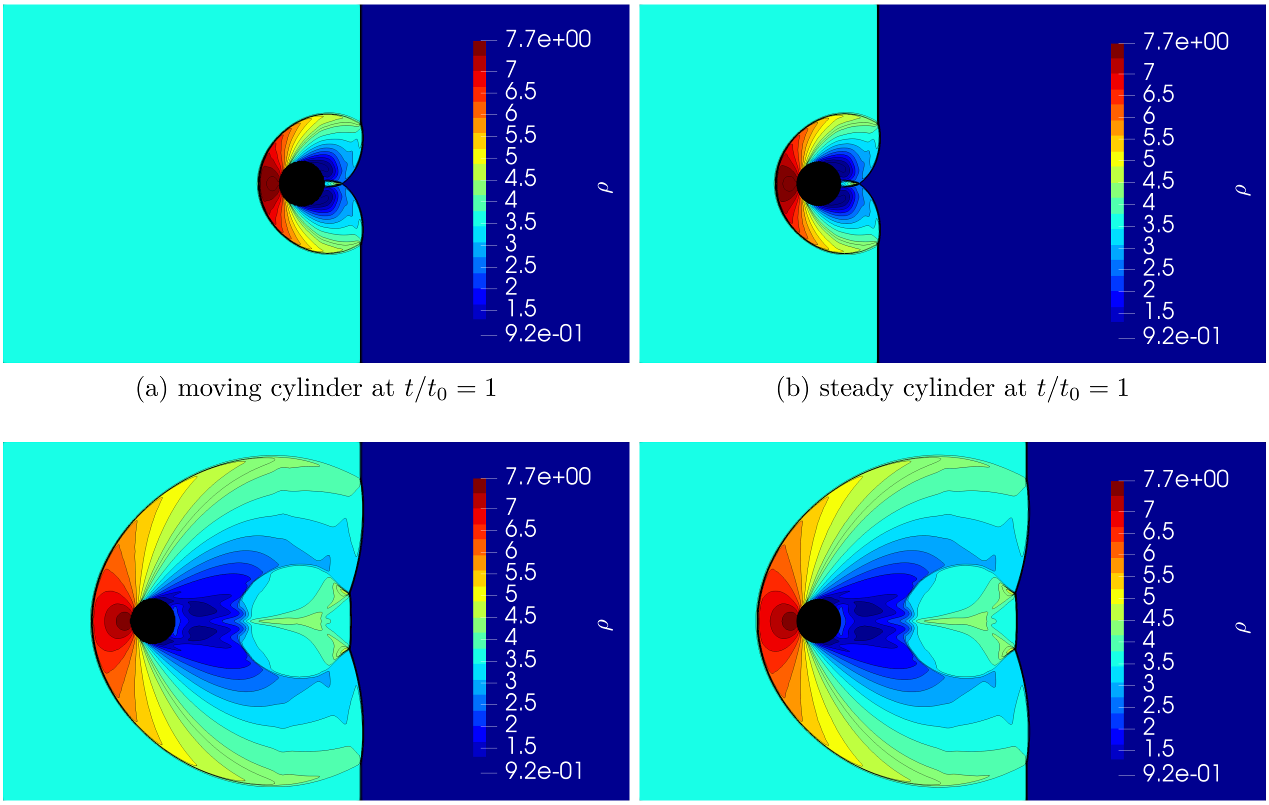


Figure 8.23: Comparison between the time history of the drag coefficient in two different reference frames. The dashed line reports the results for a moving cylinder against a steady shock while the solid line shows the results for a moving shock against a steady cylinder. The cylinder (the shock) is travelling at $Mach = 2.81$, while the Reynolds number is set equals to 2000.



(a) moving cylinder at $t/t_0 = 1$

(b) steady cylinder at $t/t_0 = 1$

(c) moving cylinder at $t/t_0 = 2$

(d) steady cylinder at $t/t_0 = 2$

Figure 8.24: Non-dimensional density field in two homologous instants of a moving cylinder against a steady shock (left panel) and of a moving shock against a steady cylinder (right panel).

8.2.3 Supersonic flow past a confined moving square

Finally, to conclude with a robustness benchmark for the entire methodology we select the supersonic flow past a moving confined square as a good and delicate benchmark of a compressible non-smooth moving object. The presented results provide reasonable qualitative behaviour since we did not find previous experimental data or numerical simulations to compare with. The simulation consisted in a $M_\infty = 3$ and $Re = 3000$ flow over a single harmonic oscillating square along the crossflow direction. Slip-wall conditions were enforced at the top and at the bottom side of the domain while a supersonic inflow condition was enforced at the left side of the domain. The outlet bound was treated characteristically. The movement was analytical superimposed as $y(t) = A\sin(2\pi f_e t)$ and here $A = 0.2D$ is the amplitude associated to the harmonic oscillation, while f_e is its effective oscillation frequency. The latter was set equal to $0.5f_0$ where $f_0 = St_0 u_\infty = 0.147u_\infty$ represents a physical consistent value whose order of magnitude is the same of the natural shading frequency of a square in similar reference viscosity conditions (i.e. $\mu_\infty \sim M_\infty/Re_\infty$). The simulation was carried out over a uniform Cartesian mesh featuring $N_x \times N_y = [800 \times 200]$ grid points for a domain of $L_x \times L_y = [32D; 8D]$. Four reference lengths were reserved upstream the squared centre and the problem was run till a statistically steady convergence of the flow. The CFL number was set equal to 0.5. For this setting, a mean drag coefficient equal to $\bar{c}_D = 1.62$ has been recovered. The instantaneous temperature fields in five different time levels are reported in Figure 8.25. As we notice, the frontal shock is dynamically stretched following the square centre square.

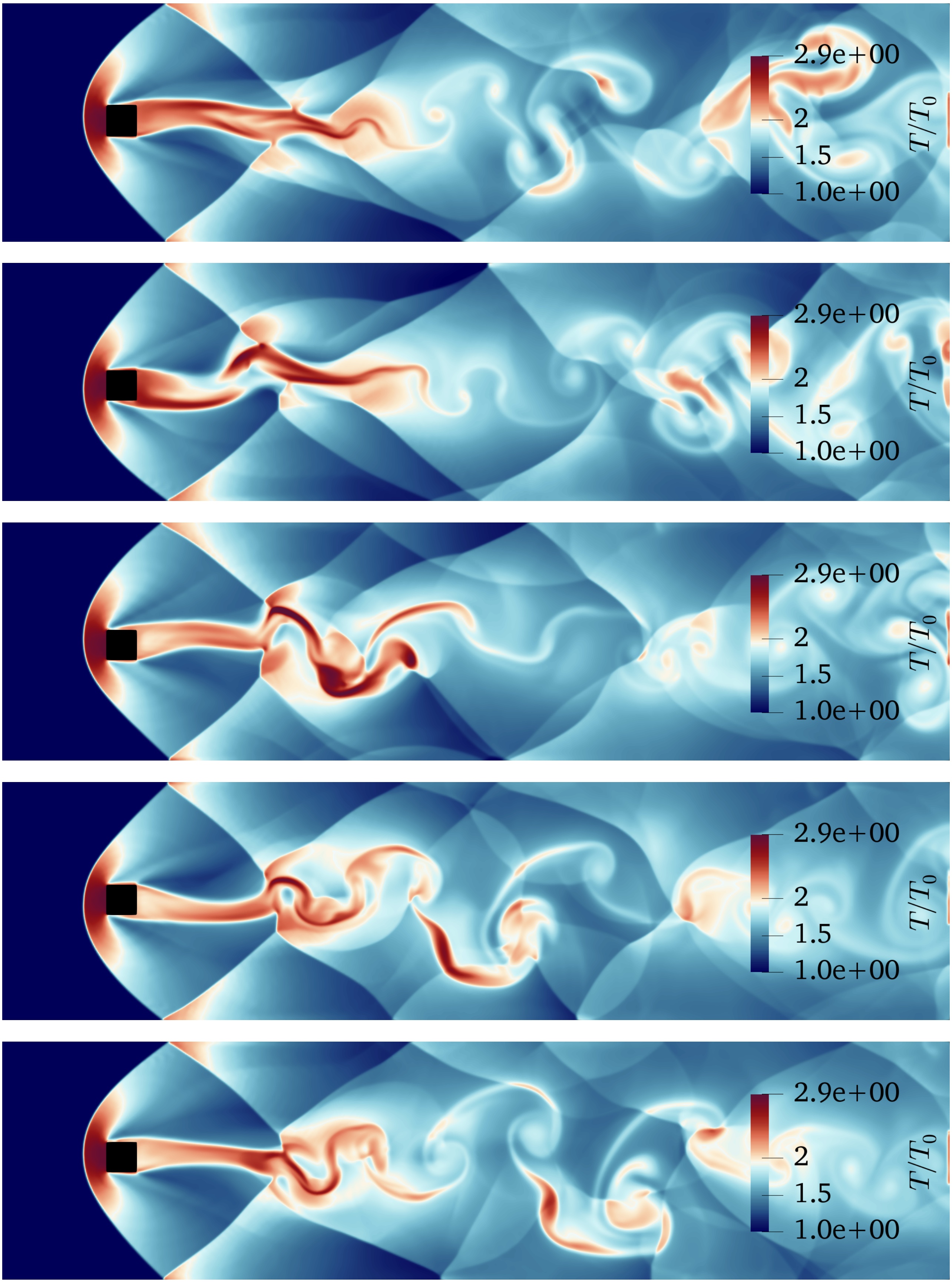


Figure 8.25: Instantaneous temperature fields for a confined moving square at $M_\infty = 3.0$ and $Re = 3000$.

8.3 Summary

The present chapter described the numerical benchmarks in the validation of the GPFM applied to objects aerodynamics in a wide range of both Reynolds and Mach numbers. In particular, we proved that using the fully-split-convective formulation of Kennedy-Grüber and Pirozzoli [91] in combination with IBM and a WENO procedure for shock-capturing, represents a suitable numerical strategy able to evolve moving objects in compressible flows stably. The methodology avoids any artificial viscosity injection even in the case of moving boundaries; thus, it appears a promising strategy for accurate direct and large-eddy simulations of compressible turbulent flows with moving boundaries. The capability of the method to reproduce a broad set of flow conditions, from subsonic laminar flows (e.g. flow over a confined and free cylinder at low Mach numbers), to transitional flows (e.g., vortex shedding downstream a cylinder) up to high-speed shock-wave interactions has been quantitatively proved both in steady and moving cases reproducing well-documented datasets and showing very good agreements for the entire validation campaign. In particular, the low-Mach computations and the strong compressible simulations of a moving cylinder a have highlighted the vanishing level of noise introduced by the present approach.

9
Conclusions & Future perspectives

Sapere aude! Have courage to use your own understanding!

Immanuel Kant

Contents

9.1 Conclusions

Turbulent compressible flows represent an exciting and challenging topic for many practical and scientific applications, and today few models can accurately describe the mechanics inherent of such systems. Undoubtedly, the Direct Numerical Simulation (DNS) and the Large-Eddy Simulation (LES) of the Navier-Stokes system of equations represent the most suitable approaches to deeply understand the physics of the fluid motion and today the availability of a large variety of accurate and efficient methods, moreover, the dramatical increase of the computational power, make numerical scientists able to address the problem.

The objective of the present three-year work was to develop and validate a high-resolution numerical model suitable for the simulation of high-speed flows in the accurate frame of DNS/LES. The work consisted of the PhD activity of the author, who started to develop the solver from scratch. Thanks to the support of his two thesis advisors and to the whole Industrial Engineering Department of the University of Padova the development process arrived at the point described in the present dissertation, consisting in a satisfactory ending point.

To describe in detail the workflow, the present dissertation was organised as a numerical toolkit. In particular, the most difficult issues, the alternatives, and the actuated coding choices were highlighted to build a reference manual for future developments. Thus, starting from the mathematical model describing the flow mechanics inherent an unsteady compressible flow, we moved to the numerical methods suitable for the discretisation of the Navier-Stokes system of equations. Many emphases were reserved for the treatment of convective and viscous terms. The latter represents the core of the numerical discretisation and lot of coding effort was reserved to this portion of the work, both in term of development time and optimisation processes. Every chapter of the work was organised in such a way that a theoretical description of the problem advances the practical implementation and the numerical validation, resulting (we hope!) in clear and usable documentation.

A sufficient detailed description was also reserved for the parallelisation strategy. The latter, even more today, represents a demanding target for three-dimensional turbulent flows simulations. Thus, the whole solver was tested over a fully turbulent test case, at a moderate Reynolds number, and consisting in a channel flow in slightly-compressible conditions. The entire validation campaign has shown excellent agreements in respect of all the most accurate and well-documented tests and benchmarks available in the literature, both analytical, numerical and experimental, results that make us able to conclude that our numerical tool is properly predictive in a wide range of problems and applications.

A huge part of the work consisted in coupling the solver with a sharp-interface Immersed Boundary Method whose peculiarity was represented by the possibility of treating moving objects in both subsonic and supersonic cases. This part of the work was divided into two chapters. The first dealt with the theory of the method and pointed out a detailed description of the numerical algorithms implemented in the present work. The second concerned the results obtained in term of the aerodynamics of steady and moving objects in a wide range of both Reynolds and Mach numbers.

9.2 Future perspectives

As far as future perspectives, we hope that this work can not be considered arrived at its end and a lot of further improvements and further analysis are feasible. Here a list of possible developments, with an increasing level of coding effort, will be presented.

1. At first sight, the solver can be immediately employed in the analysis of fundamental fluid mechanics topics concerning the turbulent behaviour of compressible flows. For instance, the dynamics of homogeneous turbulent systems consists in challenging research area for contemporary physics (see, e.g., [127, 128, 129]) and represents a feasible target with actual resources.

2. Going to more applicative problems, the dynamics associated to the interaction between strong-compressible and viscous phenomena and the shock-wave boundary layer interaction problem represent challenging and essential tasks for the modern propulsion system and flight devices (see, e.g. [95, 94, 86]). The problem can be straightforwardly faced with the actual solver hoping to provide a significant role in this research field.

3. Improving the LES accuracy dealing with more sophisticated models seams to be a challenging task for future computations and a straightforward goal for the actual state of the solver. In particular, the *wall-modelled* Large-Eddy Simulations (WMLES) [64] represents a promising strategy able to increase the LES accuracy in the case of wall-bounded flows. The task is strongly motivated also by the turbulence community who fixed a thematic challenge on this topic in 2021 (https://wmles.umd.edu).

4. The inclusion of hybrid models for turbulence, like DES/DDES/IDDES, represent quite an easy job and requires to add (at least) one more differential equation in the set the

evolutionary variables. The latter techniques represent a suitable description for thin-bodies aerodynamics with superior accuracy in respect of a standard RANS model (see, e.g., Bernardini et al. (2016)) and consists in a feasible improvement in a decent range of time;

5. Finally, even if Cartesian meshes represent a very efficient and straightforward solution for high-resolution numerical schemes, this choice represents also a stiff technology dealing with very complex geometries. To face this target, two possible improvements can be seen:

   (a) the first consists in dealing with generalised-curvilinear grids similarly to what we already have done in case of non-uniform Cartesian meshes, thus employed a fully curvilinear discretisation of the Navier-Stokes system of equations (see, e.g., Pirozzoli (2011b));

   (b) the second consists of dealing with Adaptive Mesh Refinement (AMR) solutions, which represent a very efficient and flexible technology for high-order Cartesian numerical methods. Various AMR strategies have been proposed in the last two decades, and the most efficient solutions are fully open-source and available online (see, e.g., SAMRAI, AMREX). However, AMR represents a tough coding problem, especially because of the incomplete and unclear related documentation.

Acknowledgements

Indeed the section dedicated to the acknowledgments is the most complex and challenging to write and surely a thesis work that lasted three years can not be divided in term of help, the support, and knowledge between many people. So, hoping to fulfil this delicate task, now I am going to start with the the last page.

First of all I want to dedicate my most sincere thanks to Prof. Ernesto Benini and Prof. Francesco Picano. The first, to have granted me the opportunity to continue my studies with the Ph.D. and to have allowed the development of such a particular project. The second, for his constant support, both technical and moral, and the trust reserved within the countless choices taken during these years.

A special gratitude also goes to my colleagues. Thanks to Andrea D., Andrea M., Federico and Luca for patience and for the long days spent in the *warmth* of the COMETES office.

A special thanks must also go to my colleagues across the Alps, who welcome me in their research group. Thanks to Prof. Arnaud Mura who welcomed me and with from I received excellent advices and suggestions. Thanks also to all his collaborators and students: Aimad, Marcos, Luis, and Song. I am sure we are only at the beginning of our carries and we will get in touch again.

Thanks to my family members whose support in these years has always been full of affection and tenderness. Thanks, therefore, to my uncles Giorgio, Renato and Roberto, whose concern for my goals is always among their first thoughts. Thanks to my grandmother Clara, whose bold age of 92 years seems not to have touched her vitality. So thanks to my grandparents Anna and Alessandro that even though they are no longer here with me, I am sure they are protecting my way.

And finally, among all those who support me and love me my last thanks goes to my girlfriend, Lorenza, and my beloved parents. Lorenza. Thanks for the countless experiences we had together and for all the best memories. At each memories my soul is filled with joy. And finally, mom and dad. Thank you. Without you, I would certainly not be the person I am, and I would not have achieved all this goals.

Sincerly,
Francesco De Vanna

Venezia, 27th September 2019

A

Unsteady Rankine-Hugoniot conditions

It is well known that the Rankine-Hugoniot (RH) conditions consist of a compatible relation for two-fluid regions divided by shock wave. These relations exploit the conservation principles and the equation of state for an ideal gas and, for and steady shock, they read

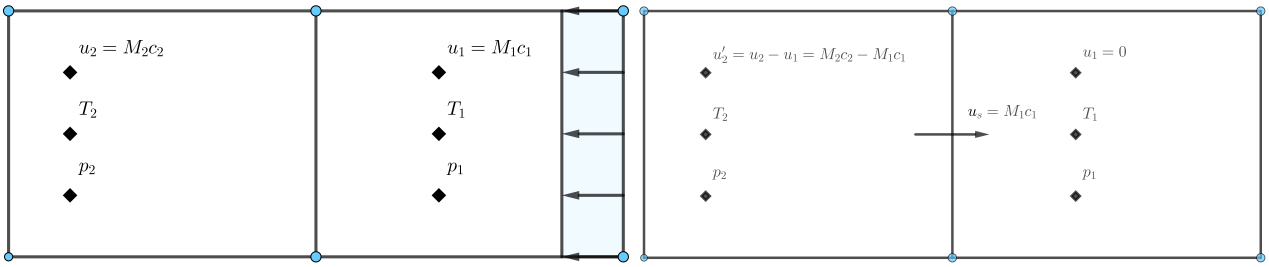
$$\frac{M_2}{M_1} = \frac{1}{M_1}\sqrt{\frac{1+\frac{\gamma-1}{2}M_1^2}{\gamma M_1^2 - \frac{\gamma-1}{2}}} \tag{1.1a}$$

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)M_1^2}{2+(\gamma-1)M_1^2} \tag{1.1b}$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1}\left(M_1^2 - 1\right) \tag{1.1c}$$

$$\frac{T_2}{T_1} = \left[\frac{2\gamma}{\gamma+1}\left(M_1^2 - 1\right)\right]\frac{2+(\gamma-1)M_1^2}{(\gamma+1)M_1^2} \tag{1.1d}$$

Here $(\cdot)_1$ are the inflowing flow variables while $(\cdot)_2$ are the *post-shocked* condition, thus M_1 and M_2 are respectively the inflowing and the post-shocked Mach numbers. In Figure 1.1a a sketch of the two flow region divided by a steady shock is reported. Knowing the inflow conditions, the RH system of equations provides a compatible relation for the flow quantities in the post-shocked region. Here we want to extend the RH conditions for a moving shock. The problem appears quite often in compressible fluid dynamics, especially as initial (analytical) flow condition.



(a) Steady shock at $M = M_1$

(b) Moving shock at $M = M_1$

Figure 1.1: Sketch of the transformation process employed in order to enforced an unsteady-shock initial condition.

Dealing with the system in Figure 1.1a a simple reference frame transformation solves the problem. In fact, adding a translational speed $u_\tau = -u_1$ to both the two flow regions the post-shock condition starts moving with a speed equal to $u_2' = u_2 - u_1$ while the inflow region rests. Consequently the shock wave moves with $u_s = M_1 c_1$, where M_1 represents the original inflow Mach number. Figure 1.1b sketches the problem.

Bibliography

[1] Acker, F., Borges, R. B., and Costa, B. (2016). An improved WENO-Z scheme. *Journal of Computational Physics*, 313:726–753.

[2] Anderson, D. (2010). Fundamentals of Aerodynamics. *AIAA Journal.*

[3] Anderson, D. and Eberhardt, S. (2001). *Understanding flight.*

[4] Balsara, D. S., Garain, S., and Shu, C. W. (2016). An efficient class of WENO schemes with adaptive order. *Journal of Computational Physics*, 326:780–804.

[5] Balsara, D. S. and Shu, C. W. (2000). Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, 160(2):405–452.

[6] Batchelor, G. (2000). *An introduction to fluid dynamics.* Cambridge university press.

[7] Bernardini, M., Modesti, D., and Pirozzoli, S. (2016). On the suitability of the immersed boundary method for the simulation of high-Reynolds-number separated turbulent flows. *Computers & Fluids*, 130:84–93.

[8] Bernardini, M., Pirozzoli, S., and Orlandi, P. (2014). Velocity statistics in turbulent channel flow up to Re $\tau = 4000$. pages 171–191.

[9] Biollo, R. and Benini, E. (2013). Recent advances in transonic axial compressor aerodynamics. *Progress in Aerospace Sciences*, 56:1–18.

[10] Blanchard, R. C. and Desai, P. N. (2012). Mars Phoenix Entry, Descent, and Landing Trajestory and Atmosphere Reconstruction. *Journal of Spacecraft and Rockets.*

[11] Blasius, H. (1908). Grenzschichten in Flussigkeiten mit kleiner Reibung. *Zeitschrift für angewandte Mathematik und Physik.*

[12] Boukharfane, R., Eugênio Ribeiro, F. H., Bouali, Z., and Mura, A. (2018). A combined ghost-point-forcing / direct-forcing immersed boundary method (IBM) for compressible flow simulations. *Computers and Fluids*, 162:91–112.

[13] Breugem, W. P. (2012). A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *Journal of Computational Physics*, 231(13):4469–4498.

[14] Bridges, J. and Wernet, M. (2004). Measurements of the Aeroacoustic Sound Source in Hot Jets. *Instrumentation.*

[15] Bryson, A. E. and Gross, R. W. F. (1961). Diffraction of strong shocks by cones, cylinders, and spheres. *Journal of Fluid Mechanics*, 10(1):1–16.

[16] Castro, M., Costa, B., and Sun, W. (2011). High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 230(5):1766–1792.

[17] Chang, S. M. and Chang, K. S. (2000). On the shock-vortex interaction in Schardin's problem. *Shock Waves*, 10(5):333–343.

[18] Chao, J., Haselbacher, A., and Balachandar, S. (2009). A massively parallel multi-block hybrid compact-WENO scheme for compressible flows. *Journal of Computational Physics*, 228(19):7473–7491.

[19] Chaudhuri, A., Hadjadj, A., and Chinnayya, A. (2011). On the use of immersed boundary methods for shock/obstacle interactions. *Journal of Computational Physics*, 230(5):1731–1748.

[20] Coppola, G., Capuano, F., Pirozzoli, S., and de Luca, L. (2019). Numerically stable formulations of convective terms for turbulent compressible flows. *Journal of Computational Physics*, 382:86–104.

[21] Costa, B. and Don, W. S. (2007). High order Hybrid central-WENO finite difference scheme for conservation laws. *Journal of Computational and Applied Mathematics*, 204(2 SPEC. ISS.):209–218.

[22] Courant, R., Isaacson, E., and Rees, M. (1952). On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics.*

[23] Culbert B. Laney (1998). *Computational Gasdynamics.* Cambridge University Press.

[24] Dan, B. and John, S. H. A. N. D. (1991). On turbulent spots in plane Poiseuille flow. 228.

[25] De Palma, P., de Tullio, M. D., Pascazio, G., and Napolitano, M. (2006). An immersed-boundary method for compressible viscous flows. *Computers and Fluids*, 35(7):693–702.

[26] de Tullio, M. D., De Palma, P., Iaccarino, G., Pascazio, G., and Napolitano, M. (2007). An immersed boundary method for compressible flows using local grid refinement. *Journal of Computational Physics*, 225(2):2098–2117.

[27] Deane, A. (2006). *Parallel Computational Fluid Dynamics : Theory and Applications.*

[28] Dennis, S. C. R. and Chang, G.-Z. (1970). Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(3):471–489.

[29] Dessoky, A., Bangga, G., Lutz, T., and Krämer, E. (2019). Aerodynamic and aeroacoustic performance assessment of H-rotor darrieus VAWT equipped with wind-lens technology. *Energy.*

[30] Ducros, F. (1999). Large-eddy simulation of the shock/turbulence interaction. *Journal of Computational Physics*, 152:517—-549.

[31] Ehsan Khalili, M., Larsson, M., and Müller, B. (2018). Immersed boundary method for viscous compressible flows around moving bodies. *Computers and Fluids*, 170:77–92.

[32] El-Hamalawi, A. (2003). Mesh Generation - Application to Finite Elements. *Engineering Construction and Architectural Management.*

[33] Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *Journal of Computational Physics*, 161(1):35–60.

[34] Ferri, F., Karatekin, Ö., Lewis, S. R., Forget, F., Aboudan, A., Colombatti, G., Bettanini, C., Debei, S., Van Hove, B., Dehant, V., Harri, A. M., Leese, M., Mäkinen, T., Millour, E., Muller-Wodarg, I., Ori, G. G., Pacifici, A., Paris, S., Patel, M., Schoenenberger, M., Herath, J., Siili, T., Spiga, A., Tokano, T., Towner, M., Withers, P., Asmar, S., and Plettemeier, D. (2019). ExoMars Atmospheric Mars Entry and Landing Investigations and Analysis (AMELIA). *Space Science Reviews*, 215(1):1–21.

[35] Ferziger, J. H. and Peric, M. (2002). *Computational Methods for Fluid Dynamics.*

[36] García Rosa, N., Dufour, G., Barènes, R., and Lavergne, G. (2015). Experimental Analysis of the Global Performance and the Flow Through a High-Bypass Turbofan in Windmilling Conditions. *Journal of Turbomachinery*, 137(5):051001.

[37] Garnier, E., Adams, N., and Sagaut, P. (2009). *Large eddy simulation for compressible flows.* Springer Science \& Business Media.

[38] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A*, 3(7):1760–1765.

[39] Ghias, R., Mittal, R., and Lund, T. (2013). A Non-Body Conformal Grid Method for Simulation of Compressible Flows with Complex Immersed Boundaries. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*, number January, pages 1–10. American Institute of Aeronautics and Astronautics.

[40] Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics [English title page]. *Matematicheskii Sbornik.*

[41] Goldstein, M. (1984). Aeroacoustics of Turbulent Shear Flows. *Annual Review of Fluid Mechanics.*

[42] Golias, N. A. and Dutton, R. W. (1997). Delaunay triangulation and 3D adaptive mesh generation. *Finite Elements in Analysis and Design.*

[43] Gottlieb, S. and Shu, C.-W. (1998). Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society*, 67(221):73–85.

[44] Green, D. and Unruh, W. G. (2006). The failure of the Tacoma Bridge: A physical model. *American Journal of Physics.*

[45] Gritskevich, M. S., Garbaruk, A. V., Schütze, J., and Menter, F. R. (2012). Development of DDES and IDDES formulations for the k-ω shear stress transport model. *Flow, Turbulence and Combustion.*

[46] Gropp, W., Gropp, W. D., Lusk, A. D. F. E. E., Lusk, E., and Skjellum, A. (2000). *Using MPI: Portable parallel programming with the message-passing interface*, volume 40.

[47] Gropp, W., Lusk, E., Doss, N., and Skjellum, A. (1996). A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing.*

[48] Gu, W., Chyu, C., and Rockwell, D. (1994). Timing of vortex formation from an oscillating cylinder. *Physics of Fluids*, 6(11):3677–3682.

[49] Guilmineau, E. and Queutey, P. (2002). A numerical simulation of vortex shedding from an oscillating circular cylinder. *Journal of Fluids and Structures*, 16(6):773–794.

[50] Güner, H., Thomas, D., Dimitriadis, G., and Terrapon, V. E. (2019). Unsteady aerodynamic modeling methodology based on dynamic mode interpolation for transonic flutter calculations. *Journal of Fluids and Structures.*

[51] Guo, Y. and Shi, Y. (2018). Seventh order compact-WENO scheme for hyperbolic conservation laws. *Computers and Fluids*, 176:193–209.

[52] Haller, G. (2005). An objective definition of a vortex. *Journal of Fluid Mechanics*, 525:1–26.

[53] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R. (1987). Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics.*

[54] Hatcher, A. (2002). *Algebriac topology.*

[55] Hendrickson, T., Kartha, A., and Candler, G. V. (2018). An Improved Ducros Sensor for the Simulation of Compressible Flows with Shocks. pages 1–13.

[56] Hirsch, C. (2007). *Numerical computation of internal and external flows, volume 1: Fundamentals of numerical discretization*, volume 1.

[57] Hoeger, M., Baier, R.-D., Fischer, S., and Neudorfer, J. (2011). High Turning Compressor Tandem Cascade for High Subsonic Flows - Part 2: Numerical and Experimental Investigations. *Proceedings of 47th Joint Propulsion Conference*, (AIAA-2011-5601):1–14.

[58] Iaccarino, G. and Verzicco, R. (2003). Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews.*

[59] Jiang, G. S. and Shu, C. W. (1996). Efficient implementation of weighted WENO schemes. *Journal of Computational Physics*, 126(126):202–228.

[60] Kuhn, G. D. (1986). The limiting behaviour of turbulence near a wall. *Journal of Fluid Mechanics.*

[61] Kundu, Pijush K., Cohen, Ira M., Dowling, D. R. (2014). *Fluidic Mechanics.*

[62] Kurzke, J. (2009). Fundamental Differences Between Conventional and Geared Turbofans. *ASME TURBO EXPO 2009: Power for Land, Sea and Air*, pages 145–153.

[63] Kyprianidis, K. G., Rolt, A. M., and Grönstedt, T. (2013). Multidisciplinary Analysis of a Geared Fan Intercooled Core Aero-Engine. *Journal of Engineering for Gas Turbines and Power*, 136(January):1–11.

[64] Larsson, J., Kawai, S., Bodart, J., and Bermejo-Moreno, I. (2016). Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews.*

[65] LeVeque, R. (1998). Finite difference methods for differential equations. *Draft version for use in AMath*, pages 1998–2005.

[66] LeVeque, R. (2002). Finite Volume Methods for Hyperbolic Problems. *Cambridge University Press*, 54:258.

[67] Li, H. and Ben-Dor, G. (1995). A shock dynamics theory based analytical solution of double Mach reflections. *Shock Waves*.

[68] Linnick, M. and Fasel, H. (2012). A High-Order Immersed Boundary Method for Unsteady Incompressible Flow Calculations. Number January, pages 1–17.

[69] Lodato, G. (2011). *Tridimensional Boundary Conditions for Direct and Large-Eddy Simulation of Turbulent Flows . Sub-Grid Scale Modeling for Near-Wall Region Turbulence.* PhD thesis.

[70] Lodato, G., Domingo, P., and Vervisch, L. (2008). Three-dimensional boundary conditions for direct and large-eddy simulation of compressible viscous flows. *Journal of Computational Physics*.

[71] Luchini, P. (2017). Universality of the Turbulent Velocity Profile. *Physical Review Letters*, 118(22):1–8.

[72] Luo, K., Zhuang, Z., Fan, J., and Haugen, N. E. L. (2015). A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions. *International Journal of Heat and Mass Transfer*, 92:708–717.

[73] Mattingly, J. D. (2006). *Elements of Propulsion: Gas Turbines and Rockets*.

[74] Merlin, C., Domingo, P., and Vervisch, L. (2013). Immersed boundaries in large eddy simulation of compressible flows. *Flow, Turbulence and Combustion*, 90(1):29–68.

[75] Millikan, C. (1938). A critical discussion of turbulent flows in channels and circular tubes. In *Proceedings of the Fifth International Congress for Applied Mechanics, held at Harvard University and the Massachusetts Institute of Technology, September 12-16*.

[76] Mittal, R. and Iaccarino, G. (2005a). Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261.

[77] Mittal, R. and Iaccarino, G. (2005b). Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261.

[78] Modesti, D. and Pirozzoli, S. (2016). Reynolds and Mach number effects in compressible turbulent channel flow. *International Journal of Heat and Fluid Flow*, 59:33–49.

[79] Modesti, D. and Pirozzoli, S. (2019). Direct numerical simulation of supersonic pipe flow at moderate Reynolds number. *International Journal of Heat and Fluid Flow*.

[80] Moller, T. and Trumbore, B. (2005). *Fast, Minimum Storage Ray/Triangle Intersection*.

[81] Niazi Ardekani, M., Abouali, O., Picano, F., and Brandt, L. (2018). Heat transfer in laminar Couette flow laden with rigid spherical particles. *Journal of Fluid Mechanics*, 834:308–334.

[82] Nicoud, F. and Ducros, F. (1999). Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor. *Flow, Turbulence and Combustion*, 624:183–200.

[83] Oerlemans, S. and Schepers, J. G. (2009). Prediction of Wind Turbine Noise and Validation against Experiment. *International Journal of Aeroacoustics*.

[84] Orlandi, P. (2012). *Fluid flow phenomena: a numerical toolkit*, volume 55. Springer Science & Business Media.

[85] O'Rourke, J. (1998). *Computational Geometry in C.*, volume 64. Cambridge University Press.

[86] Pasquariello, V., Hickel, S., and Adams, N. A. (2017). Unsteady effects of strong shock-wave/boundary-layer interaction at high Reynolds number. *Journal of Fluid Mechanics*, 823:617–657.

[87] Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271.

[88] Picano, F., Breugem, W. P., and Brandt, L. (2015). Turbulent channel flow of dense suspensions of neutrally buoyant spheres. *Journal of Fluid Mechanics*, 764:463–487.

[89] Piquet, A., Roussel, O., and Hadjadj, A. (2016). A comparative study of Brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows. *Computers and Fluids*, 136:272–284.

[90] Pirozzoli, S. (2002). Conservative hybrid compact-WENO schemes for shock-turbulence interaction. *Journal of Computational Physics*, 178(1):81–117.

[91] Pirozzoli, S. (2010). Generalized conservative approximations of split convective derivative operators. *Journal of Computational Physics*, 229(19):7180–7190.

[92] Pirozzoli, S. (2011a). Numerical Methods for High-Speed Flows. *Annual Review of Fluid Mechanics*, 43(1):163–194.

[93] Pirozzoli, S. (2011b). Stabilized non-dissipative approximations of Euler equations in generalized curvilinear coordinates. *Journal of Computational Physics*.

[94] Pirozzoli, S. and Bernardini, M. (2011). Direct numerical simulation database for impinging shock wave/turbulent boundary-layer interaction. *AIAA Journal*, 49(6):1307–1312.

[95] Pirozzoli, S., Bernardini, M., and Grasso, F. (2010). Direct numerical simulation of transonic shock/boundary layer interaction under conditions of incipient separation. *Journal of Fluid Mechanics*, 657:361–393.

[96] Pirozzoli, S. and Colonius, T. (2013). Generalized characteristic relaxation boundary conditions for unsteady compressible flow simulations. *Journal of Computational Physics*, 248:109–126.

[97] Poinsot, T. (2004). Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 99(2):352.

[98] Pope, S. B. (2001). Turbulent Flows. *Measurement Science and Technology.*

[99] Prandtl, L. (1904). Verhandlungen des dritten internationalen Mathematiker-Kongresses. *Heidelberg, Leipeizig*, pages 484–491.

[100] Qiu, Y. L., Shu, C., Wu, J., Sun, Y., Yang, L. M., and Guo, T. Q. (2016). A boundary condition-enforced immersed boundary method for compressible viscous flows. *Computers and Fluids*, 136:104–113.

[101] Qu, Y., Shi, R., and Batra, R. C. (2018). An immersed boundary formulation for simulating high-speed compressible viscous flows with moving solids. *Journal of Computational Physics*, 354:672–691.

[102] Quarteroni, A. (2014). *Numerical Models for Differential Problems.*

[103] Quarteroni, A. and Valli, A. (2008). *Numerical Approximation of Partial Differential Equations.*, volume 23. Springer.

[104] Rajani, B. N., Kandasamy, A., and Majumdar, S. (2009). Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3):1228–1247.

[105] Ren, W., Shu, C., and Yang, W. (2013). An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions. *International Journal of Heat and Mass Transfer*, 64:694–705.

[106] Rieth, M., Proch, F., Stein, O. T., Pettit, M. W. A., and Kempf, A. M. (2014). Comparison of the Sigma and Smagorinsky LES models for grid generated turbulence and a channel flow. *Computers and Fluids*, 99:172–181.

[107] Rohde, A. (2001). Eigenvalues and eigenvectors of the Euler equations in general geometries. *AIAA Paper 2001-2609*, (1):1–6.

[108] Sahin, M. and Owens, R. G. (2004). A numerical investigation of wall effects up to high blockage ratios on two-dimensional flow past a confined circular cylinder. *Physics of Fluids*, 16(5):1305–1320.

[109] Salvadore, F., Bernardini, M., and Botti, M. (2013). GPU accelerated flow solver for direct numerical simulation of turbulent flows. *Journal of Computational Physics*, 235:129–142.

[110] Schneiders, L., Hartmann, D., Meinke, M., and Schröder, W. (2013a). An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics*, 235:786–809.

[111] Schneiders, L., Hartmann, D., Meinke, M., and Schröder, W. (2013b). An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics*, 235:786–809.

[112] Seo, J. H. and Mittal, R. (2011). A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *Journal of Computational Physics.*

[113] Shu, C.-W. (1999). High Order ENO and WENO Schemes for Computational Fluid Dynamics. pages 439–582.

[114] Spalart, P. R. (2009). Detached-Eddy Simulation. *Annual Review of Fluid Mechanics.*

[115] Talamelli, A., Bellani, G., and Rossetti, A. (2014). The “long pipe” in CICLoPE: A design for detailed turbulence measurements. In *Springer Proceedings in Physics*.

[116] Toda, H. B., Cabrit, O., Balarac, G., Bose, S., Lee, J., Choi, H., and Nicoud, F. (2010). A subgrid-scale model based on singular values for LES in complex geometries. *Center for Turbulence Research, Proceedings of the Summer Program*, pages 193–202.

[117] Toro, E. F. (2009). *Riemann solvers and numerical methods for fluid dynamics: A practical introduction*. Springer.

[118] Touber, E. and Sandham, N. D. (2008). Oblique Shock Impinging on a Turbulent Boundary Layer : Low-Frequency Mechanisms Mach number. *Fluid Dynamics*, (June):4170–4170.

[119] Touber, E. and Sandham, N. D. (2009). Large-eddy simulation of low-frequency unsteadiness in a turbulent shock-induced separation bubble. *Theoretical and Computational Fluid Dynamics*, 23(2):79–107.

[120] Tritton, D. J. (2006). Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 6(04):547.

[121] Uhlmann, M. (2005). An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476.

[122] van der Hoef, M., van Sint Annaland, M., Deen, N., and Kuipers, J. (2008). Numerical Simulation of Dense Gas-Solid Fluidized Beds: A Multiscale Modeling Strategy. *Annual Review of Fluid Mechanics*.

[123] Von Karman, T. (1937). The Fundamentals of the Statistical Theory of Turbulence. *Journal of the Aeronautical Sciences*, 4:131–138.

[124] Vreman, A. W. and Kuerten, J. G. (2014). Comparison of direct numerical simulation databases of turbulent channel flow at reτ = 180. *Physics of Fluids*, 26(1).

[125] Vreman, B., Geurts, B., and Kuerten, H. (1995). Subgrid-modelling in LES of compressible flow. *Applied Scientific Research*, 54(3):191–203.

[126] Wang, B.-s., Li, P., Gao, Z., and Sun, W. (2018a). An improved fifth order alternative WENO-Z finite difference scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 374:469–477.

[127] Wang, J., Gotoh, T., and Watanabe, T. (2017). Spectra and statistics in compressible isotropic turbulence. *Physical Review Fluids*, 2(1):1–26.

[128] Wang, J., Wan, M., Chen, S., and Chen, S. (2018b). Kinetic energy transfer in compressible isotropic turbulence. *Journal of Fluid Mechanics*, 841:581–613.

[129] Wang, J., Wan, M., Chen, S., Xie, C., Wang, L. P., and Chen, S. (2019). Cascades of temperature and entropy fluctuations in compressible turbulence. *Journal of Fluid Mechanics*, 867:195–215.

[130] Wang, Y., Shu, C., and Yang, L. M. (2016). Boundary condition-enforced immersed boundary-lattice Boltzmann flux solver for thermal flows with Neumann boundary conditions. *Journal of Computational Physics*, 306:237–252.

[131] Wodicka, G. R., Stevens, K. N., Golub, H. L., and Shannon, D. C. (1990). Spectral Characteristics of Sound Transmission in the Human Respiratory System. *IEEE Transactions on Biomedical Engineering.*

[132] Woodward, P. and Colella, P. (1984). The numerical simulation of two-dimensional fluid flow with strong shocks.

[133] Yang, J. and Balaras, E. (2006). An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1):12–40.

[134] Yang, X., Zhang, X., Li, Z., and He, G. W. (2009). A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836.

[135] Yoo, C. S., Wang, Y., Trouvé, A., and Im, H. G. (2007). Characteristic boundary conditions for direct simulations of turbulent counterflow flames. 7830.

[136] Yuen, A. C., Yeoh, G. H., Timchenko, V., Cheung, S. C., and Chen, T. (2016). Study of three LES subgrid-scale turbulence models for predictions of heat and mass transfer in large-scale compartment fires. *Numerical Heat Transfer; Part A: Applications*, 69(11):1223–1241.