UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Study of unstructured finite volume methods for the solution of the Euler equations

Master Thesis

Author:	Iván Padilla Montero
Supervisor:	Roberto Maurice Flores Le Roux
Document:	Report & Appendix
Call:	July 2016

Master's Degree in Aerospace Engineering

Universitat Politècnica de Catalunya

Terrassa - June 17, 2016

Abstract

This work deals with the numerical solution of inviscid compressible flows by means of the Euler equations. It focuses on the description of an unstructured finite volume method for these equations and its numerical application to solve external, two-dimensional steady problems.

On first place, the standard formulation of the Euler equations is presented, reviewing the most important properties that characterize their mathematical behavior. The hyperbolic nature of the system is discussed, emphasizing the fundamental importance of taking into account the propagation of information in the flow field in order to obtain physically meaningful solutions, which also leads to a description of how the boundary conditions should be treated to avoid undesirable behaviors. To complete this presentation, a dimensionless form of the equations is derived, which provides substantial advantages to the numerical solution.

The attention is then focused on the unstructured finite volume formulation, which is based on a central approximation of the fluxes at the volume interfaces. According to the need of properly accounting for the propagation of characteristic variables, the requirement to add artificial dissipation terms to the central discretization is justified. Then, two classical forms of artificial dissipation are defined, namely, the first-order upwind scheme and the Jameson-Schmidt-Turkel high-order model, detailing how to adapt the formulation of the dissipation terms to an unstructured mesh. Eventually, the time integration of the spatially discretized equations is assessed.

With the objective of performing a practical implementation of the theoretical concepts studied, the development of a numerical solver is presented next, briefly describing the program structure and characteristics. After that, five different test cases are solved with the purpose of validating the code, consisting on two transonic flows around a NACA0012 airfoil and three supersonic examples, respectively around a NACA0012 airfoil, a double wedge airfoil and circular cylinder. The results obtained for each case are then analyzed and compared against reference solutions, showing an overall satisfactory performance of the solver developed.

Resumen

Este trabajo trata sobre la solución numérica de flujos compresibles no viscosos mediante las ecuaciones de Euler. La atención se centra en la descripción de un método de volúmenes finitos no estructurado para estas ecuaciones y en su aplicación numérica para la solución de flujos bidimensionales, externos y estacionarios.

En primer lugar se presenta la formulación estándar de las ecuaciones de Euler, revisando las propiedades más importantes que caracterizan su comportamiento matemático. Se describe la naturaleza hiperbólica del sistema, haciendo hincapié en la importancia fundamental de tener en cuenta la propagación de información en el fluido para poder obtener soluciones físicas correctas, lo que lleva directamente a especificar como deben tratarse las condiciones de contorno a fin de evitar posibles comportamientos indeseados. Esta presentación se completa exponiendo la utilización de una forma adimensional de las ecuaciones, lo que aporta ventajas sustanciales a la solución numérica.

A continuación, el documento se centra en la explicación del método de volúmenes finitos no estructurado escogido, que se basa en una aproximación central de los flujos en la entrefase de los volúmenes. De acuerdo con la necesidad de tener en cuenta la propagación de las variables características en el fluido, se procede a justificar el requerimiento de introducir difusión artificial en la discretización producida por el esquema central. Se prosigue entonces con la descripción de dos métodos clásicos de difusión artificial, que son el esquema aguas arriba de primer orden y el modelo de alto orden de Jameson-Schmidt-Turkel, detallando como adaptar la formulación de los términos disipativos a la formulación no estructurada. Finalmente se discute la integración temporal de la discretización espacial de las ecuaciones.

Con el objetivo de realizar una aplicación práctica directa de los conceptos teóricos estudiados, a continuación se presenta el desarrollo de un código de simulación numérica, describiendo en términos generales la estructura del programa y sus características. Con el propósito de validar el código, se pasa entonces a la solución de cinco ejemplos de aplicación, que consisten en dos flujos transónicos alrededor de un perfil NACA0012 y tres casos supersónicos, respectivamente alrededor de un perfil NACA0012, un perfil romboidal y un cilindro circular. Los resultados obtenidos en cada caso se comparan con una solución de referencia, demostrando un comportamiento general satisfactorio del código desarrollado.

Contents

List of figures

List of tables

1 INTRODUCTION

1.1 Motivation

During the last decades, the rapid increase in the computational power available has promoted the development of numerical analysis techniques in many different engineering fields. Due to the technological challenges that characterize the design and construction of aircraft, and the important costs associated with wind tunnel testing, aerospace engineering is one of the areas where numerical methods show a wide range of applications. At present, computational solid mechanics and computational fluid dynamics constitute two fundamental pillars of any serious design process in the aerospace industry. It is necessary, then, for the modern aerospace engineer to develop a solid understanding of the foundations upon which these disciplines build-up.

With the numerical solution of complete flow fields around complex geometries becoming a routine calculation nowadays, there is a strong necessity to develop robust and efficient programs that allow the obtention of accurate results in a short time. However, even with the most recent developments in computing performance, the solution of the full system of Navier-Stokes equations is still an expensive task from the computational point of view, especially if the turbulence effects are to be modelled with high fidelity. This situation has motivated the search for alternate strategies, which focus on simplified flow descriptions that, if applied with care, can capture the relevant physics of the complete model at a lower computational cost.

The best simplified model for the solution of actual aircraft configurations is that given by the Euler equations, which allow the obtention of realistic results in the transonic and low to moderate supersonic regimes when flow separation is not important (Anderson, 2011). These equations, which govern the behavior of inviscid flows, offer a very attractive alternative especially in the case of preliminary design stages. They are able to capture the compressible flow phenomena associated to the presence of discontinuities in the flow field, such as shock waves, which shows to be one of the most important design drivers for any machine flying at or near the supersonic regime.

Although the inviscid flow model is obviously not of universal validity, the importance of its accurate numerical solution also resides in the dominating convective character of the Navier-Stokes equations at high Reynolds numbers, which is the case of the immense majority of flow situations encountered in practice. Since the Euler equations retain the convective properties of the general formulation, almost all of the methods developed for the Euler system are also valid for the Navier-Stokes equations. As a result, another advantage of developing solvers for the Euler equations is that they serve as the base for possible extensions to the complete, viscous model.

1.2 Objectives and scope

With the previous ideas in mind, the objective of this work is the numerical solution of compressible inviscid flows by means of the Euler equations. This has been the subject of extensive research for several years, and many different techniques have been developed and can be found in the literature; see for example (Hirsch, 1990). Here, the aim is to learn the fundamental concepts that lie behind the practical implementation of inviscid flow solvers for the aerospace industry. Accordingly, the scope is focused on the solution of two-dimensional, external, steady inviscid flows, which retain all the important properties of the numerical calculations carried out in practice.

To achieve the proposed goal, a numerical solver has been developed. Due to its natural connection with conservation laws, a finite volume method has been chosen for the discretization of the governing equations, based on an unstructured mesh strategy. Among the different numerical schemes available, a well-established central scheme has been adopted, which relies on a combination of simple artificial dissipation terms and a multistage time integration method to offer a robust solution to most of the flow situations under consideration (Jameson, Schmidt, & Turkel, 1981; Hirsch, 1990).

The use of an unstructured approach has nowadays the enormous advantage of allowing an almost automatic mesh generation on arbitrary geometries. This is an essential aspect of any practical application of computational fluid dynamics to real engineering problems, and although the scope of the numerical solutions treated here is not that big, it is important to follow this philosophy when looking into the future. Furthermore, the possibility to perform local refinements in a certain region without affecting the rest of the domain opens the way to perform flexible mesh adaptation techniques that can optimize the number of grid points for a given level of accuracy.

Five different examples of application have been carried out in order to investigate the performance of the code developed, which cover the transonic and supersonic regimes as well as different body geometries. In all the cases considered, the results obtained are compared against reference solutions.

2 THE EULER EQUATIONS

The Euler equations constitute the standard mathematical model to describe the behavior of inviscid fluids. They are a coupled system of nonlinear conservation laws that state the fundamental physical principles of conservation of mass, momentum and energy for an inviscid fluid.

These equations should simulate real flows in the limit where the transport phenomena of viscosity, mass diffusion and thermal conductivity can be neglected. In this sense, the Euler equations can be directly obtained from the more general set of governing equations, i.e., the Navier-Stokes equations, by removing the corresponding dissipative terms.

From a pure theoretical point of view, there are different but equivalent ways to write the system of Euler equations. For example, it can be formulated either in integral or differential form, and each one can in turn be expressed both in conservation or nonconservation form. However, when focusing on the numerical solution of the equations, we are actually solving an approximate equation that has some inherent limitations, which are due to the discretization and truncation errors of the numerical scheme in use and the available machine precision. These limitations are strongly influenced by which form of the equations is used, and depending on the problem that has to be solved, one form may be preferable over the others.

On the one hand, there is an important difference between the integral and the differential forms of the Euler equations, namely, that the integral form does not require mathematical continuity, whereas the differential form does. In other words, only the integral form of the equations allows for discontinuous solutions. Then, if we want to obtain the numerical solution of a flow where strong gradients such as shock waves are expected, we should use the integral form of the governing equations. For this reason, the integral form can be considered as more fundamental than the differential one.

On the other hand, it is well known, see for instance (Anderson, 1995), that the numerical solution of the Euler equations in nonconservation form may lead to unsatisfactory results for flow fields with shock waves. The shocks may appear in the wrong location and the stability of the solution can be compromised. In contrast, when using the conservation form the solution is generally smooth and stable. The reason behind this behavior is mainly due to the fact that the conservation form uses the conservative variables as the dependent variables, and the jumps in these variables are either zero or small across a shock wave, thus contributing to the numerical quality of the solution.

In order to enhance the stability and accuracy of a numerical solution, the integral conservation form of the equations should be used. Actually, this is the approach taken by the finite volume method, as will be seen in section 3.

2.1 Differential conservation form of the Euler equations

Written in differential conservation form and neglecting body forces, the Euler equations can be expressed in a two-dimensional Eulerian reference frame as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{0} \tag{2.1}$$

where $\mathbf{U}$ is the vector of conservative variables, t represents the temporal dimension and $\mathbf{F}$, $\mathbf{G}$ respectively denote the convective flux vectors in the spatial directions x and y. These vectors are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u h_0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v h_0 \end{bmatrix} \tag{2.2}$$

where ρ, u, v and p are, respectively, the fluid density, x and y velocity components and pressure, which, along with temperature T, constitute the so-called primitive (or nonconservative) fluid variables. The quantities e_0 and h_0 respectively denote the specific total energy and total enthalpy of the fluid, which are defined as

$$e_0 = e + \frac{V^2}{2}, \quad h_0 = e_0 + \frac{p}{\rho} \tag{2.3}$$

with e being the internal energy of the fluid and V its velocity magnitude ($V^2 = u^2 + v^2$).

In order to completely define the system, the fluid constitutive relations have to be specified. Assuming a perfect gas, the associated equation of state can be written as

$$p = \rho R T \tag{2.4}$$

where R is the gas constant. Additionally, assuming that the gas is also calorically perfect, that is, a perfect gas with constant specific heats, the internal energy can be expressed as an algebraic function of temperature only, given explicitly by

$$e = c_v T \tag{2.5}$$

with c_v denoting the specific heat at constant volume. From the definition of a calorically perfect gas, the following relationships also hold

$$R = c_p - c_v, \quad \gamma = \frac{c_p}{c_v} \tag{2.6}$$

in which c_p is the specific heat at constant pressure and γ is the ratio of specific heats.

2.2 Some mathematical properties of the Euler equations

2.2.1 Quasi-linear form

Observing equation (2.2), it can be seen that the flux vectors $\mathbf{F}$ and $\mathbf{G}$ can be expressed as functions of the conservative variables only, that is: $\mathbf{F} = \mathbf{F}(\mathbf{U})$ and $\mathbf{G} = \mathbf{G}(\mathbf{U})$. This fact allows us to write the system in what is known as the quasi-linear form of the Euler equations, namely

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x} + \frac{\partial \mathbf{G}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{0} \tag{2.7}$$

or

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{0} \tag{2.8}$$

In equation (2.8), $\mathbf{A}$ and $\mathbf{B}$ are respectively defined as the Jacobian matrices of the flux vectors $\mathbf{F}$ and $\mathbf{G}$, and represent the derivatives of the flux vectors with respect to the conservative variables. It is important to note that they are not constant but functions of $\mathbf{U}$. Under the previous assumption of a calorically perfect gas, these matrices have explicit analytical expressions, which can be found for example in (Hirsch, 1990).

The flux vectors have the very remarkable property of being homogeneous functions of degree 1 of the vector of conservative variables in the case of a gas satisfying the following relationship

$$\frac{p}{\rho} = f(e) \tag{2.9}$$

which holds under the assumption of a perfect gas presented previously. This is known as the homogeneous property of the Euler equations, and makes it possible to express the flux vectors as

$$\mathbf{F} = \mathbf{AU}\,, \quad \mathbf{G} = \mathbf{BU} \tag{2.10}$$

These relations allow rewriting the system of Euler equations with the flux Jacobians inside the spatial derivatives, giving a mathematically equivalent formulation as that of equations (2.1) and (2.7). Nevertheless, from the numerical point of view, these formulations do not lead to identical discretizations, and once again, one form may be advantageous over the others depending on the numerical scheme to be used.

2.2.2 Eigenvalues of the system

It can be shown that the mathematical nature of the Euler equations is hyperbolic, meaning that the flux Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ have purely real eigenvalues and are diagonalizable, i.e., they have a complete set of linearly independent eigenvectors (Anderson, 1995). The eigenvalues of the flux Jacobians are of fundamental importance in the solution of the Euler equations. On a physical basis, they give the directions and velocities of the propagation of information throughout the flow field, that is, the slopes of the characteristic lines.

Given an arbitrary direction defined by the unit vector $\mathbf{n}$, the associated flux Jacobian, $\mathbf{A}_n$, can be obtained as a linear combination of the Jacobians corresponding to each of the spatial directions of the reference frame

$$\mathbf{A}_n = n_x \mathbf{A} + n_y \mathbf{B} \tag{2.11}$$

where n_x and n_y are the corresponding components of $\mathbf{n}$. Solving the associated eigenvalue problem, the following four eigenvalues are obtained for $\mathbf{A}_n$ (Hirsch, 1990)

$$\begin{gathered} \lambda_1 = \lambda_2 = V_n \\ \lambda_3 = V_n + c \\ \lambda_4 = V_n - c \end{gathered} \tag{2.12}$$

where V_n denotes the fluid velocity component along the direction $\mathbf{n}$, namely $V_n = \mathbf{V} \cdot \mathbf{n}$, and c is the local speed of sound. As will be made clearer in the next section, these eigenvalues physically represent the direction and velocity of wave components that propagate through the flow field. The ones which travel with the same velocity as the fluid, given by λ_1 and λ_2, are entropy and vorticity waves, whereas the other two, which propagate at a sonic speed relative to the fluid, are acoustic waves.

It is very important to notice that if the flow is subsonic, λ_4 will have a different sign from the others, meaning that the perturbations will propagate both upstream as well as downstream. On the contrary, if the flow is supersonic, all the eigenvalues are positive and the information can only travel downstream. This behavior has strong implications in the treatment of boundary conditions, as will be discussed in section 2.3.

2.2.3 One-dimensional linearized characteristic formulation

To further understand the physical meaning behind the mathematical nature of the Euler equations, it is interesting to analyze the linearized characteristic formulation of the system.

For simplicity, let us focus on the one-dimensional case (the present discussion is also valid for multiple dimensions), given by

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} = \mathbf{0} \tag{2.13}$$

Now, the eigenvalues of the flux Jacobian are (Hirsch, 1990)

$$\lambda_1 = u\,, \quad \lambda_2 = u + c\,, \quad \lambda_3 = u - c \tag{2.14}$$

Recalling that the system is hyperbolic, it is possible to write the flux Jacobian as

$$\mathbf{A} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1} \tag{2.15}$$

where $\mathbf{R}$ is a matrix whose columns are the right eigenvectors of $\mathbf{A}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}$, that is

$$\boldsymbol{\Lambda} = \begin{bmatrix} u & 0 & 0 \\ 0 & u+c & 0 \\ 0 & 0 & u-c \end{bmatrix} \tag{2.16}$$

This, in turn, allows expressing equation (2.13) as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1}\frac{\partial \mathbf{U}}{\partial x} = \mathbf{0} \tag{2.17}$$

Equations (2.13) and (2.17) are not linear at all since the Jacobian matrix is a function of $\mathbf{U}$ (this is the reason why such formulations are called quasi-linear). However, considering small perturbations around a reference state, it is acceptable to linearize the behavior of the equation and assume that the Jacobian is constant. Under this simplification, equation (2.17) can be rewritten as follows

$$\frac{\partial}{\partial t}(\mathbf{R}^{-1}\mathbf{U}) + \boldsymbol{\Lambda}\frac{\partial}{\partial x}(\mathbf{R}^{-1}\mathbf{U}) = \mathbf{0} \tag{2.18}$$

where the product

$$\mathbf{R}^{-1}\mathbf{U} = \boldsymbol{\psi} \tag{2.19}$$

is defined as the vector of characteristic variables, and

$$\frac{\partial \boldsymbol{\psi}}{\partial t} + \boldsymbol{\Lambda}\frac{\partial \boldsymbol{\psi}}{\partial x} = \mathbf{0} \tag{2.20}$$

is the linearized characteristic formulation of the one-dimensional Euler equations. As can be observed, since the matrix $\boldsymbol{\Lambda}$ is diagonal, the three characteristic equations in the system are uncoupled. Expanding (2.20), we have

$$\frac{\partial \psi_1}{\partial t} + u\frac{\partial \psi_1}{\partial x} = 0$$

$$\frac{\partial \psi_2}{\partial t} + (u+c)\frac{\partial \psi_2}{\partial x} = 0 \qquad (2.21)$$

$$\frac{\partial \psi_3}{\partial t} + (u-c)\frac{\partial \psi_3}{\partial x} = 0$$

This is a very significant result, which tells us that each of the characteristic variables is governed by a wave equation in which, as introduced before, the wave speed and direction is given by the eigenvalues of flux Jacobian. Moreover, it means that the characteristic variables propagate without any distortion.

In the general case, in which the problem is not linearized, neither u nor c are constant, so the characteristic system is not uncoupled. However, even in this case, it can be shown that the characteristic variables are also governed by a (coupled) system of wave equations. These variables, which remain constant along the characteristic lines, are usually known as the Riemann invariants. Refer to (Hirsch, 1990) for more details.

2.3 Physical boundary conditions for external inviscid flow

A very important aspect of the solution of a given flow field are the boundary conditions. This is the mechanism by which the governing equations can recognize and differentiate one application from another. From a physical point of view, there are two different types of boundary conditions for an external inviscid flow.

On first place, at a nonporous solid wall such as the surface of a body, the flow tangency condition has to be satisfied. This means that the flow velocity vector $\mathbf{V}$ immediately adjacent to the wall has to be tangent to the wall. Defining $\mathbf{n}$ as the unit normal vector at a given point on the surface, this wall boundary condition can be stated as

$$\mathbf{V} \cdot \mathbf{n} = 0 \qquad (2.22)$$

which has to be satisfied for the entire surface. Equation (2.22) simply means that the component of velocity perpendicular to the wall must be zero. This is the only surface boundary condition for an inviscid flow, as the magnitude of the velocity and the surface values of the fluid density, pressure and temperature are obtained as part of the solution.

Secondly, we have the far-field boundary condition, that is, the flow conditions at a large (theoretically infinite) distance away from the body. Far away from the body, as limited by the size of the computational domain, the flow approaches the freestream conditions in all directions. According to the arguments given in the previous section, the propagation of information in the flow field is different depending on whether the flow is locally subsonic or supersonic. Then, in order to properly account for the correct transport of characteristic variables across the far-field boundary, there is a need to differentiate between two cases.

On one side, if the normal freestream velocity component is supersonic, all the characteristics move in the same direction (all positive eigenvalues), meaning that all the flow variables have to be prescribed at the inlet portion the far-field boundary and none of them have to be imposed at the outlet portion. On the other side, for a subsonic normal velocity there is a negative eigenvalue, telling us that one component of the solution moves upstream. In this case, for a two-dimensional flow, only three conditions shall be prescribed at the inlet portion and one at the outlet.

Finally, it is also important to take into account that the far-field boundary condition should be imposed on the characteristic variables only, not on the primitive or conservative ones. If the characteristic variables are not properly prescribed at the boundary, their associated waves may not be able to leave or enter the computational domain in the correct way, causing unwanted reflections which decrease the numerical quality of the solution.

2.4 Nondimensionalization

The use of the equations in nondimensional form is very convenient from the numerical perspective. First, the use of normalized quantities contributes to the numerical quality of the solution by enhancing the conditioning of the system of the equations. Second, the use of dimensionless variables allows us to work with the similarity parameters of the flow, which reduces the number of variables involved in the calculations and yields more general results.

A useful set of dimensionless variables for the Euler equations is

$$\begin{gathered} \tilde{t} = \frac{t c_\infty}{L}\,, \quad \tilde{x} = \frac{x}{L}\,, \quad \tilde{y} = \frac{y}{L}\,, \quad \tilde{u} = \frac{u}{c_\infty}\,, \quad \tilde{v} = \frac{v}{c_\infty} \\ \tilde{p} = \frac{p}{\rho_\infty c_\infty^2}\,, \quad \tilde{e}_0 = \frac{e_0}{c_\infty^2}\,, \quad \tilde{T} = \frac{T}{T_\infty} \end{gathered} \tag{2.23}$$

where L is a characteristic length of the problem under study (for example the chord of an airfoil) and c_∞, ρ_∞ and T_∞ respectively denote the freestream speed of sound, density and temperature of the fluid. For a calorically perfect gas, the speed of sound satisfies the following relationship: $c = \sqrt{\gamma R T}$.

With these variables, the system (2.1) simply transforms in

$$\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{t}} + \frac{\partial \tilde{\mathbf{F}}}{\partial \tilde{x}} + \frac{\partial \tilde{\mathbf{G}}}{\partial \tilde{y}} = \mathbf{0} \tag{2.24}$$

and the respective vectors become

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \tilde{\rho} \\ \tilde{\rho}\tilde{u} \\ \tilde{\rho}\tilde{v} \\ \tilde{\rho}\tilde{e}_0 \end{bmatrix}, \quad \widetilde{\mathbf{F}} = \begin{bmatrix} \tilde{\rho}\tilde{u} \\ \tilde{\rho}\tilde{u}^2 + \tilde{p} \\ \tilde{\rho}\tilde{u}\tilde{v} \\ \tilde{\rho}\tilde{u}\tilde{h}_0 \end{bmatrix}, \quad \widetilde{\mathbf{G}} = \begin{bmatrix} \tilde{\rho}\tilde{v} \\ \tilde{\rho}\tilde{u}\tilde{v} \\ \tilde{\rho}\tilde{v}^2 + \tilde{p} \\ \tilde{\rho}\tilde{v}\tilde{h}_0 \end{bmatrix} \tag{2.25}$$

with

$$\tilde{e}_0 = \tilde{e} + \frac{\tilde{V}^2}{2}, \quad \tilde{h}_0 = \tilde{e}_0 + \frac{\tilde{p}}{\tilde{\rho}} \tag{2.26}$$

Then, the thermodynamic relationships for a calorically perfect gas change as

$$\tilde{p} = \frac{\tilde{\rho}\tilde{T}}{\gamma}, \quad \tilde{e} = \frac{1}{\gamma(\gamma-1)}\tilde{T} \tag{2.27}$$

Similarly, the freestream values of the primitive variables result in the following expressions

$$\tilde{\rho}_\infty = 1, \quad \tilde{u}_\infty = M_\infty^x, \quad \tilde{v}_\infty = M_\infty^y, \quad \tilde{p}_\infty = \frac{1}{\gamma}, \quad \tilde{T}_\infty = 1 \tag{2.28}$$

which conveniently transform the freestream vector of conservative variables in

$$\widetilde{\mathbf{U}}_\infty = \begin{bmatrix} 1 \\ M_\infty^x \\ M_\infty^y \\ \dfrac{1}{\gamma(\gamma-1)} + \dfrac{M_\infty^2}{2} \end{bmatrix} \tag{2.29}$$

where M_∞^x and M_∞^y denote the components of the freestream Mach number along each spatial direction, given by the angle of attack α as

$$M_\infty^x = M_\infty \cos\alpha, \quad M_\infty^y = M_\infty \sin\alpha \tag{2.30}$$

Observing equations (2.28) and (2.29), it is easy to notice that the nondimensional solution of the Euler equations only depends on the freestream Mach number (including the angle of attack), the ratio of specific heats and the shape (but not the size) of the body under consideration. These are precisely the similarity parameters for an inviscid flow (Anderson, 2011), which clearly shows the benefits of working with the equations in nondimensional form.

From now on, for the sake of clarity, the tilde will be omitted in the equations, but it will be implicitly assumed that the dimensionless form of the equations is being used unless otherwise specified.

3 UNSTRUCTURED FINITE VOLUME METHOD FOR THE EULER EQUATIONS

The finite volume method is advantageous for the discretization of conservation laws such as the Euler equations due to its direct connection to the physical flow properties. It is based upon the discretization of the integral form of the conservation equations, as opposed to the finite difference method, which deals with the differential form, and is therefore more general and fundamentally appropriate for the solution of external compressible flows.

Assuming a control volume Ω fixed in space, enclosed by a surface Γ, the integral conservation form of the two-dimensional Euler equations can be written as

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \oint_{\Gamma} \mathbf{F}_n \, d\Gamma = \mathbf{0} \tag{3.1}$$

where $\mathbf{F}_n$ is the flux vector across the boundary of the control volume, given by the unit normal vector $\mathbf{n}$ pointing outwards from $d\Gamma$

$$\mathbf{F}_n = n_x \mathbf{F} + n_y \mathbf{G} \tag{3.2}$$

with n_x and n_y being the components of $\mathbf{n}$. It is important to note that with this formulation, the order of the equations has been decreased by one, which reduces the continuity requirements of the flow variables. This is, as commented before, the reason why this form of the Euler equations allows for mathematical discontinuities in the flow field.

The system of integral conservation laws given by equation (3.1) can be either discretized both in space and time simultaneously, or by a spatial discretization with independent time integration. The later approach converts the system of partial differential equations into a system of ordinary differential equations in time, which are then solved using any suitable time integration method. This offers more flexibility regarding the stability and accuracy properties of the time-marching scheme, and is also the approach taken in this work.

It may be somewhat surprising at first that in order to obtain the numerical solution to a steady flow, the time dependent Euler equations are considered. There is, however, a strong mathematical reason to justify this choice. We have stated previously that the complete system of Euler equations is hyperbolic, no matter the type of inviscid flow being computed. Nevertheless, when the time derivatives are removed from the system, the mathematical behavior of the equations changes, and it can actually be proved that the system of steady Euler equations exhibits a mixed elliptic-hyperbolic nature (Anderson, 1995). Moreover, this mixed behavior is associated with the local flow regime, namely, the equations are elliptic in the subsonic regions and hyperbolic in the supersonic ones. This situation poses enormous numerical difficulties, since any steady technique that is suitable for the solution of the subsonic region is usually not valid for the supersonic counterpart, and vice versa. This is the reason why the unsteady Euler equations are also used for the solution of steady flows. It is the steady-state flow field what we want, and the time-dependent approach is simply a means to that end.

3.1 Spatial discretization

The formulation given by (3.1) expresses that the variation of the conservative flow variables inside the control volume only depends on the net balance of the flux vectors across its boundary. This implies that for an arbitrary division of the domain into smaller subdomains, we can write the integral conservation equations for each subdomain and recover the global conservation law by simply adding up the contribution of each one. This is the basis of the spatial discretization by means of the finite volume method, i.e., the subdivision of the domain Ω into a series of finite control volumes, also known as cells, and the application of the conservation statement to each one of them.

In order to obtain the semi-discrete form of the integral equations, the volume integral in (3.1) is usually replaced by the average value of the vector of conservative variables over the cell, which for a cell with area A is defined as (the finite volume is two-dimensional)

$$\overline{\mathbf{U}} = \frac{1}{A}\int_A \mathbf{U}\, dA \tag{3.3}$$

Regarding the discretization of the flux term, the surface integral can be replaced by the sum over all the bounding faces of the cell, so that the fluxes are assumed constant along each face. This proves to be a second-order approximation, which is the recommended accuracy for the majority of CFD applications. Thus, such an approximation is acceptable for our purpose.

Under the previous considerations, for a given generic cell i with area A^i, the spatial discretization of equation (3.1) may be expressed as

$$A^i \frac{\partial \overline{\mathbf{U}}^i}{\partial t} + \sum_{e=1}^{N_e} \mathbf{F}_n^e\, l^e = \mathbf{0} \tag{3.4}$$

where $\overline{\mathbf{U}}^i$ is the average vector of conservative variables over the cell, N_e is the number of faces of the cell, $\mathbf{F}_n^e$ denotes the numerical flux vector across the cell face e, and l^e is the length of face e. As before, the numerical flux across the face is determined by the local unit outward normal $\mathbf{n}^e$

$$\mathbf{F}_n^e = n_x^e \mathbf{F}^e + n_y^e \mathbf{G}^e \tag{3.5}$$

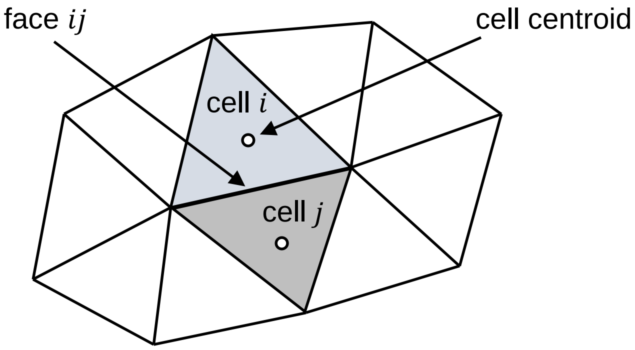

Figure 3.1 Schematic of an unstructured triangular mesh.

where $\mathbf{F}^e$ and $\mathbf{G}^e$ denote the numerical flux vectors along each spatial component evaluated at the face. This is a quite general formulation of the finite volume method applied to the two-dimensional Euler equations.

In practice, when the numerical results obtained from a discretization like the one already presented are to be analyzed, we need to assign the cell-averaged values to a mesh point, for example the center (centroid) of the cell. Taking this into account, it can be shown, see (Lomax, Pulliam, & Zingg, 2001), that the cell-averaged values and the values at the center of the cell only differ by a term of second-order. This means that the volume integral of the original conservation equation can be approximated as the value of the vector of conservative variables at the center of cell, once again with second-order accuracy. As a result, $\overline{\mathbf{U}}^i$ may be substituted by $\mathbf{U}^i$ in the discrete equation, denoting the conservative variables at the centroid of the cell.

At this point, one has to select the type of cell to subdivide the computational domain and choose how to approximate the fluxes at the cell faces. As can be expected, there exist many different options, and a general discussion can be found in (Hirsch, 2007). As announced previously, in this study, an unstructured discretization based on triangular cells has been considered (see Figure 3.1). Triangular cells are the simplest two-dimensional control volumes, and are widely used since they allow a lot of flexibility to discretize the majority of geometries usually encountered (Löhner, 2008). During practical implementation, the calculation of cell areas, centroids, face lengths and face normals is required. The mathematical formulation of such quantities for triangular cells can be found in the Appendix.

The evaluation of fluxes at the cell faces is one of the key aspects of any numerical scheme based on the finite volume method. For the Euler equations, we can distinguish essentially between two families: central and upwind discretization schemes. Both of them have been widely applied to the numerical solution of high-speed inviscid flows with satisfactory results (Hirsch, 1990). From their pure definition, upwind schemes are designed to numerically account for the direction of propagation of information in the flow field, whereas central

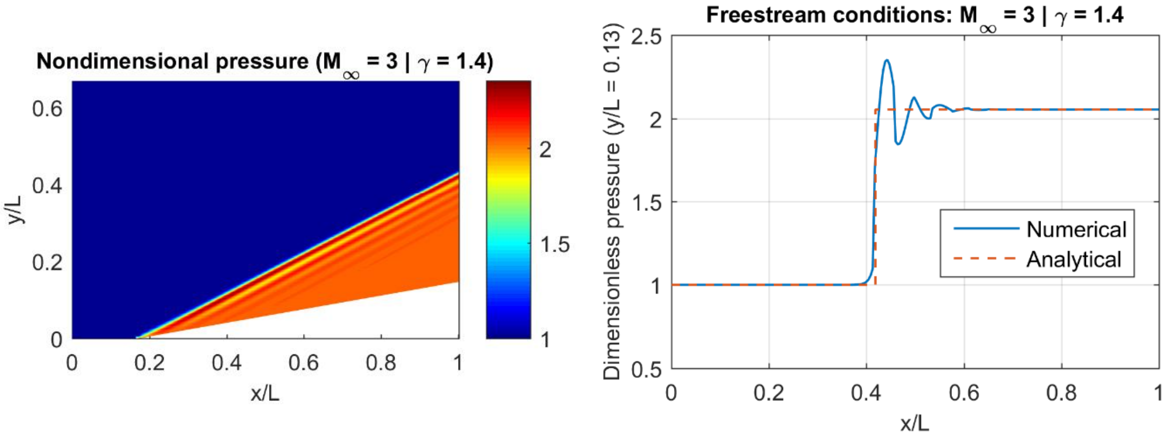

Figure 3.2 Example of strong unphysical oscillations near a shock wave on a 10 degree compression corner. This solution was obtained using a central finite difference scheme with no artificial dissipation.

schemes are based on centered discretizations which can draw numerical information from outside the correct domain of dependence of a given grid point. In this sense, upwind schemes obey more properly the physics of the flow, and can be viewed as a more natural discretization of the Euler equations. However, as will be discussed in the next section, in view of the limitations of pure central discretizations, different corrections have been developed for central schemes, which modify them to mimic the behavior of upwind methods.

Here, a second-order central discretization has been considered.

3.1.1 Central schemes and the need for artificial dissipation

Focusing on a cell-centered approach, which is the most common practice in CFD, and assuming a piecewise constant approximation, that is, a constant value of the fluid variables inside a given cell, the approximation of numerical fluxes at the interface by the second-order central scheme can be formulated for an unstructured mesh as

$$\begin{aligned} \mathbf{F}^{ij} &= \frac{1}{2}\left(\mathbf{F}^{i} + \mathbf{F}^{j}\right) \\ \mathbf{G}^{ij} &= \frac{1}{2}\left(\mathbf{G}^{i} + \mathbf{G}^{j}\right) \end{aligned} \tag{3.6}$$

where $\mathbf{F}^{ij}$ and $\mathbf{G}^{ij}$ are the spatial components of the numerical flux at the face shared by the cells i and j (see Figure 3.1), and $\mathbf{F}^{i}$, $\mathbf{F}^{j}$, $\mathbf{G}^{i}$ and $\mathbf{G}^{j}$ respectively denote the fluxes associated to each one of the cells, namely $\mathbf{F}^{i} = \mathbf{F}(\mathbf{U}^{i})$, $\mathbf{F}^{j} = \mathbf{F}(\mathbf{U}^{j})$, and the same for the y components.

Then, the central discretization approximates the fluxes at a given interface as the simple arithmetic average between the fluxes of the cells sharing that face. This estimation does not take into account the direction of the transport of the characteristic variables in the solution, something which is the source of significant numerical problems, especially in the presence of discontinuities, as described below.

It is known that in processes governed by nonlinear equations such as the Euler and Navier-Stokes systems, there can be a continuous production of high-frequency components in the solution. These are the responsible, for example, of the production of shock waves. In real flows, the production of these high-frequency modes is actually limited by viscosity. However, when dealing with the Euler equations, there is no such limitation. Then, if a nondissipative scheme is used to discretize the equations in the presence of sharp gradients, the numerical errors associated to the discretization usually introduce severe oscillations in the vicinity of discontinuities (see Figure 3.2) and, in some cases, can produce unphysical solutions due to a violation of the entropy condition. As a consequence, the numerical scheme in use must contain some form of numerical dissipation in order to be able to deal with this phenomenon.

Upwind schemes, on the other hand, respect the correct propagation of the flow characteristics and do not present these problems. In fact, they show an intrinsic dissipation. Unfortunately, it can be found that the second-order central approximation to a first derivative is nondissipative, see for instance (Lomax, Pulliam, & Zingg, 2001). This means that if the central scheme presented above is to be used to obtain the numerical solution of a high-speed inviscid flow by means of the Euler equations, some numerical dissipation has to be added to the solution. Otherwise, strong oscillations and incorrect, unphysical solutions may be obtained.

This explicitly added dissipation is usually referred to as artificial dissipation, artificial diffusion or artificial viscosity, and can also be viewed as a means of stabilizing the numerical solution. Different forms of artificial dissipation have been developed for central schemes (Hirsch, 1990), which are usually based upon the properties of upwind schemes. When simplified equations that retain most of the properties of the original system are discretized using upwind-type discretizations, a modified partial differential equation can be obtained in which the inherent dissipative terms that the upwinding procedure generates appear explicitly. These terms can then be used to design an artificial dissipation model such as to correct the behavior of the central schemes.

It can be stated, then, that the artificial dissipation terms introduce an upwind-like correction to the central schemes such as to remove the non-physical effects arising from the central discretization of wave propagation phenomena. In this sense, the terms that are added are not as “artificial” as it may seem, but are strongly connected to the nature of the flow. This is the reason why the use of a central discretization plus dissipation terms can usually be considered equivalent to the use of an upwind technique.

The artificial dissipation models that have been considered in this analysis are described in the following section.

3.2 Artificial dissipation

In order to add artificial dissipation to the finite volume discretization, the central approximation of the interface fluxes given by (3.6) is generally modified as

$$\begin{aligned}\mathbf{F}^{ij} &= \frac{1}{2}(\mathbf{F}^{i} + \mathbf{F}^{j}) - \mathbf{D}_x^{ij} \\ \mathbf{G}^{ij} &= \frac{1}{2}(\mathbf{G}^{i} + \mathbf{G}^{j}) - \mathbf{D}_y^{ij}\end{aligned} \tag{3.7}$$

with $\mathbf{D}_x^{ij}$ and $\mathbf{D}_y^{ij}$ being the artificial dissipation terms associated to the fluxes $\mathbf{F}^{ij}$ and $\mathbf{G}^{ij}$.

For the purpose of properly understanding the nature of the artificial dissipation terms commonly used in practice, it is important to review the basic philosophy of upwind schemes. Recalling the analysis performed in section 2.2.3, and taking into account the homogeneous property of the Euler equations, a characteristic flux vector can be defined as (Flores, Ortega, & Oñate, 2011)

$$\mathbf{f} = \mathbf{\Lambda}\boldsymbol{\psi} \tag{3.8}$$

which allows rewriting the one-dimensional characteristic formulation into

$$\frac{\partial \boldsymbol{\psi}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = \mathbf{0} \tag{3.9}$$

Let us assume for the time being that (3.9) is the system of governing equations that we want to solve using the finite volume method. After applying a spatial discretization equivalent to that given by (3.4), we need to decide how to approximate the characteristic fluxes at the cell interfaces. In this one-dimensional case, the cells become segments and the faces reduce to the midpoint between the nodes defining each segment. We now choose to use an upwind scheme, for simplicity the first-order one. Assuming a segment defined by the nodes i and j, and that the positive direction of the x axis (from left to right) is from i to j, the first-order upwind approximation for each characteristic component k can be written as

$$\begin{aligned} f_k^{ij} &= f_k^{i} \quad \text{if} \quad \lambda_k > 0 \\ f_k^{ij} &= f_k^{j} \quad \text{if} \quad \lambda_k < 0 \end{aligned} \tag{3.10}$$

for $k = 1,2,3$. As can be seen, the previous expression takes into account the direction of propagation of the wave components in the flow, as given by the sign of the corresponding eigenvalues, which are assumed piecewise constant between i and j due to the approximation previously considered. If the wave is propagating from left to right ($\lambda_k > 0$), the value of the interface flux is assumed to be that of node i, that is, upstream of the midpoint ij. On the contrary, if the wave travels from right to left ($\lambda_k < 0$), the assigned flux value is the one of node j, once again upstream of the midpoint. It is clear, then, which is the fundamental idea of upwinding.

Using expression (3.8), equation (3.10) can be transformed in

$$\begin{aligned} f_k^{ij} &= \lambda_k \psi_k^i \quad \text{if} \quad \lambda_k > 0 \\ f_k^{ij} &= \lambda_k \psi_k^j \quad \text{if} \quad \lambda_k < 0 \end{aligned} \tag{3.11}$$

Now, to see the equivalence between the central scheme with artificial dissipation and the upwind scheme, we can also write the upwind approximation in (3.11) as

$$\begin{aligned} f_k^{ij} &= \frac{1}{2}\left(f_k^i + f_k^j\right) - \frac{1}{2}\lambda_k\left(\psi_k^j - \psi_k^i\right) \quad \text{if} \quad \lambda_k > 0 \\ f_k^{ij} &= \frac{1}{2}\left(f_k^i + f_k^j\right) + \frac{1}{2}\lambda_k\left(\psi_k^j - \psi_k^i\right) \quad \text{if} \quad \lambda_k < 0 \end{aligned} \tag{3.12}$$

which is the same as the second-order central approximation plus an additional term which takes into account the direction of propagation of information. Both cases in (3.12) can be combined in a single expression by making use of the absolute value of the eigenvalues (Hirsch, 1990; Lomax, Pulliam, & Zingg, 2001; Lyra & Morgan, 2000)

$$f_k^{ij} = \frac{1}{2}\left(f_k^i + f_k^j\right) - \frac{1}{2}|\lambda_k|\left(\psi_k^j - \psi_k^i\right) \tag{3.13}$$

Then, defining $|\mathbf{\Lambda}|$ as the diagonal matrix which contains the absolute value of the eigenvalues in its diagonal terms, the previous equation can be directly expressed in terms of the vectors of characteristic variables and fluxes, that is

$$\mathbf{f}^{ij} = \frac{1}{2}(\mathbf{f}^i + \mathbf{f}^j) - \frac{1}{2}|\mathbf{\Lambda}|(\boldsymbol{\psi}^j - \boldsymbol{\psi}^i) \tag{3.14}$$

This constitutes the first-order upwind approximation to the interface characteristic fluxes for the formulation given by equation (3.9). Focusing again on the conservative form, the relationship (2.19) can be used to change from characteristic to conservative variables, obtaining

$$\mathbf{F}^{ij} = \frac{1}{2}(\mathbf{F}^i + \mathbf{F}^j) - \frac{1}{2}\mathbf{R}|\mathbf{\Lambda}|\mathbf{R}^{-1}(\mathbf{U}^j - \mathbf{U}^i) \tag{3.15}$$

where the product

$$\mathbf{R}|\mathbf{\Lambda}|\mathbf{R}^{-1} = |\mathbf{A}| \tag{3.16}$$

is defined as the positive flux Jacobian. It is obtained by taking the absolute value of all the eigenvalues of the flux Jacobian $\mathbf{A}$. With this definition, the upwind approximation to the interface flux for the one-dimensional Euler equations becomes

$$\mathbf{F}^{ij} = \frac{1}{2}(\mathbf{F}^{i} + \mathbf{F}^{j}) - \frac{1}{2}|\mathbf{A}|(\mathbf{U}^{j} - \mathbf{U}^{i}) \tag{3.17}$$

This is an important result, since it states that all the information necessary to properly account for the propagation of characteristic quantities in the flow is contained inside the positive flux Jacobian. Then, $|\mathbf{A}|$ plays a key role in any upwind discretization.

The extension to multiple dimensions is straightforward, see for instance (Hirsch, 1990), allowing us to write

$$\mathbf{G}^{ij} = \frac{1}{2}(\mathbf{G}^{i} + \mathbf{G}^{j}) - \frac{1}{2}|\mathbf{B}|(\mathbf{U}^{j} - \mathbf{U}^{i}) \tag{3.18}$$

Now compare equation (3.7) with (3.17) and (3.18). It is clear that choosing

$$\begin{aligned} \mathbf{D}_x^{ij} &= \frac{1}{2}|\mathbf{A}|(\mathbf{U}^{j} - \mathbf{U}^{i}) \\ \mathbf{D}_y^{ij} &= \frac{1}{2}|\mathbf{B}|(\mathbf{U}^{j} - \mathbf{U}^{i}) \end{aligned} \tag{3.19}$$

converts the second-order central scheme into the first-order upwind scheme. Hence, this is a form of artificial dissipation, actually, one of the simplest. However, it is important to bear in mind that in the process, the order of the central scheme is being reduced to first-order, which is usually too low for the desirable accuracy of the solution. In order to maintain second-order accuracy, higher-order artificial dissipation terms have to be used, which, once again, should be based on the foundations of high-order upwind discretizations.

Taking advantage of the definition of the Jacobian along an arbitrary direction, as given by equation (2.11), the components of the dissipation term associated to the face fluxes $\mathbf{F}^{ij}$ and $\mathbf{G}^{ij}$ can be grouped as

$$\mathbf{D}_n^{ij} = n_x^{ij}\mathbf{D}_x^{ij} + n_y^{ij}\mathbf{D}_y^{ij} = \frac{1}{2}|\mathbf{A}_n|(\mathbf{U}^{j} - \mathbf{U}^{i}) \tag{3.20}$$

where n_x^{ij} and n_x^{ij} are the components of the unit vector $\mathbf{n}^{ij}$ normal to the face shared by the cells i and j. Recalling that the normal face flux is then

$$\mathbf{F}_n^{ij} = n_x^{ij}\mathbf{F}^{ij} + n_y^{ij}\mathbf{G}^{ij} \tag{3.21}$$

the two-dimensional first-order upwind approximation can be expressed as follows

$$\mathbf{F}_n^{ij} = \frac{1}{2}\left(\mathbf{F}_n^i + \mathbf{F}_n^j\right) - \frac{1}{2}|\mathbf{A}_n|(\mathbf{U}^j - \mathbf{U}^i) \tag{3.22}$$

In summary, on a general basis, any form of artificial dissipation suitable for our discretization may be written in the following way

$$\mathbf{D}_n^{ij} = \frac{1}{2}|\mathbf{A}_n|\mathbf{d}^{ij} \tag{3.23}$$

in which $\mathbf{d}^{ij}$ represents a difference of conservative variables that dictates the order of the dissipation term.

3.2.1 First-order artificial dissipation

As described in the previous section, a first-order artificial dissipation scheme can be obtained by choosing

$$\mathbf{d}^{ij} = \mathbf{U}^j - \mathbf{U}^i \tag{3.24}$$

which is equivalent to the first-order upwind scheme.

At this point, we have to address how to calculate the positive flux Jacobian $|\mathbf{A}_n|$. In the general case in which the behavior of the equations is not linearized, the Jacobian matrix is not constant. Hence, since the flow states at cells i and j are often slightly different, the obtention of a Jacobian matrix such that

$$\mathbf{F}_n^j - \mathbf{F}_n^i = \mathbf{A}_n(\mathbf{U}^j - \mathbf{U}^i) \tag{3.25}$$

is not trivial. This can be accomplished in an exact manner by the solution of what is known as a Riemann problem (Hirsch, 1990). However, such approach is expensive from the computational point of view, and approximate alternatives are usually considered, such as the use of what are known as approximate Riemann solvers.

In this work, the approximate Riemann solver of Roe has been adopted (Roe, 1981), which is very popular and has a good performance in the majority of situations. Equation (3.25) is then approximated as

$$\mathbf{F}_n^j - \mathbf{F}_n^i \approx \mathbf{A}_n^{Roe}(\mathbf{U}^j - \mathbf{U}^i) \tag{3.26}$$

and the first-order artificial dissipation term becomes

$$\mathbf{D}_n^{ij} = \frac{1}{2}|\mathbf{A}_n^{Roe}|(\mathbf{U}^j - \mathbf{U}^i) \tag{3.27}$$

where $\mathbf{A}_n^{Roe}$ is the flux Jacobian evaluated at the so-called Roe's intermediate (or average) state, which is defined in a specific set of fluid variables. For the two-dimensional case, it is as follows

$$\rho^{ij} = r\rho^i$$
$$u^{ij} = \frac{ru^j + u^i}{r+1}, \quad v^{ij} = \frac{rv^j + v^i}{r+1} \tag{3.28}$$
$$h_0^{ij} = \frac{rh_0^j + h_0^i}{r+1}$$

with the auxiliary parameter

$$r = \sqrt{\frac{\rho^j}{\rho^i}} \tag{3.29}$$

For the details on the derivation and justification of these averages, the reader is referred to (Roe, 1981; Hirsch, 1990). With this intermediate variables, the fluid state at the interface between cells i and j is completely defined.

The positive flux Jacobian can now be calculated using a decomposition similar to that given by (3.16), that is, using the associated set of eigenvalues and eigenvectors, for which there exist explicit analytical expressions. Nevertheless, it is important to notice that this approach involves matrix inversion and multiplication, which are computationally intensive tasks. In view of this limitation, a better alternative is to directly calculate the product of the Jacobian times the difference vector, i.e., $|\mathbf{A}_n^{Roe}|\mathbf{d}^{ij}$, without directly evaluating the matrix. An efficient algorithm for that purpose was developed by (Turkel, 1988), see also (Ortega, 2014; Hu, 2001), which allows a direct calculation of $\mathbf{D}_n^{ij}$. For the two-dimensional case it has the form

$$\mathbf{D}_n^{ij} = \frac{|\lambda_1|}{2}\mathbf{d}^{ij} + \frac{|\lambda_3| - |\lambda_1|}{4} l_1 \begin{bmatrix} \frac{1}{c^{ij}} \\ \frac{u^{ij}}{c^{ij}} + n_x^{ij} \\ \frac{v^{ij}}{c^{ij}} + n_y^{ij} \\ \frac{h_0^{ij}}{c^{ij}} + V_n^{ij} \end{bmatrix} + \frac{|\lambda_4| - |\lambda_1|}{4} l_2 \begin{bmatrix} \frac{1}{c^{ij}} \\ \frac{u^{ij}}{c^{ij}} - n_x^{ij} \\ \frac{v^{ij}}{c^{ij}} - n_y^{ij} \\ \frac{h_0^{ij}}{c^{ij}} - V_n^{ij} \end{bmatrix} \tag{3.30}$$

where u^{ij}, v^{ij} and h_0^{ij} are given by the Roe averages in equation (3.7). The average speed of sound c^{ij} is computed with

$$c^{ij} = \sqrt{(\gamma - 1)\left(h_0^{ij} - k_e^{ij}\right)} \tag{3.31}$$

with k_e^{ij} denoting the specific kinetic energy of the fluid at the intermediate state

$$k_e^{ij} = \frac{1}{2}\left(u^{ij}u^{ij} + v^{ij}v^{ij}\right) \tag{3.32}$$

The eigenvalues are given by (2.12), repeated here for convenience

$$\lambda_1 = \lambda_2 = V_n^{ij}, \quad \lambda_3 = V_n^{ij} + c^{ij}, \quad \lambda_4 = V_n^{ij} - c^{ij} \tag{3.33}$$

with the normal velocity $V_n^{ij} = n_x^{ij}u^{ij} + n_y^{ij}v^{ij}$, and the auxiliary variables

$$\begin{aligned} l_1 &= \frac{\gamma - 1}{c^{ij}}\left[\left(k_e^{ij} - V_n^{ij}\right)d_1^{ij} - \left(u^{ij} - n_x^{ij}\right)d_2^{ij} - \left(v^{ij} - n_y^{ij}\right)d_3^{ij} + d_4^{ij}\right] \\ l_2 &= \frac{\gamma - 1}{c^{ij}}\left[\left(k_e^{ij} + V_n^{ij}\right)d_1^{ij} - \left(u^{ij} + n_x^{ij}\right)d_2^{ij} - \left(v^{ij} + n_y^{ij}\right)d_3^{ij} + d_4^{ij}\right] \end{aligned} \tag{3.34}$$

where d_k^{ij}, with $k = 1$ to 4, denote the components of the difference vector $\mathbf{d}^{ij}$.

One shortcoming of the Roe solver is that it is not able to properly resolve a flow expansion containing a sonic point (Hirsch, 1990), causing it to appear as an expansion shock. This is a nonphysical solution since it does not satisfy the second principle of thermodynamics. A solution to this behavior is to limit the minimum value of the eigenvalues at the sonic point, not allowing the wave speeds to vanish. This diffuses the unphysical expansion shock into an expansion fan, which in turn physically sound. This procedure is usually known as entropy correction, or entropy fix, and the general practice is to set the minimum wave speed as a small fraction of the spectral radius of the flux Jacobian. The spectral radius is defined as the maximum possible value that the eigenvalues can take, that is

$$\rho\left(\mathbf{A}_n^{Roe}\right) = \left|V_n^{ij}\right| + c^{ij} \tag{3.35}$$

and then the entropy correction can be implemented as (Swanson, Radespiel, & Turkel, 1997; Ortega, 2014)

$$\begin{aligned} |\lambda_1| &= \max\left[|\lambda_1|,\, \nu_l \rho\left(\mathbf{A}_n^{Roe}\right)\right] \\ |\lambda_3| &= \max\left[|\lambda_3|,\, \nu_{nl} \rho\left(\mathbf{A}_n^{Roe}\right)\right] \\ |\lambda_4| &= \max\left[|\lambda_4|,\, \nu_{nl} \rho\left(\mathbf{A}_n^{Roe}\right)\right] \end{aligned} \tag{3.36}$$

where ν_l and ν_{nl} are small constants respectively associated to the linear (entropy and vorticity waves) and nonlinear (acoustic waves) eigenvalues. Typical values are $\nu_l = 0.1$ and $\nu_{nl} = 0.2$.

Besides acting as an entropy fix for the Roe solver, the limitation given by (3.6) also plays another important role. In practice, there are different situations in which the eigenvalues can vanish. For example, the acoustic eigenvalues λ_3 and λ_4 tend to zero at sonic points, and λ_1 becomes null at a stagnation point. This can lead to an insufficient numerical diffusion at certain regions, creating difficulties that may cause numerical instability. As a result, the fact of having a lower limit in the eigenvalues is favorable from the numerical point of view, although it has to be taken into account that this is introducing some small error in the solution.

Before finishing this presentation, it is also interesting to describe a simplified artificial dissipation that is commonly used in practical computations. It consists on replacing the positive flux Jacobian by its spectral radius (Swanson, Radespiel, & Turkel, 1997), namely

$$\mathbf{D}_n^{ij} = \frac{1}{2}\rho(\mathbf{A}_n)\mathbf{d}^{ij} \tag{3.37}$$

This form introduces more dissipation into the solution because it scales the difference term by the largest possible wave speed, which should also be evaluated at an intermediate state. Since the scaling factor is a scalar value, i.e., the same for all four components, this form is usually known as scalar artificial dissipation. On the other hand, the case of the complete, matricial form described before is labeled as matrix artificial dissipation. As can be realized, this simplification avoids the use of (3.30), and is therefore more computationally efficient. Note also that choosing $\nu_l = 0.1$ and $\nu_l = 0.1$ reduces the matrix form into the scalar one.

3.2.2 Jameson-Schmidt-Turkel high-order artificial dissipation

It can be shown, see (Lomax, Pulliam, & Zingg, 2001), that when applying a one-sided difference operator to a simple convection model equation such as the one-dimensional wave equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0 \tag{3.38}$$

the upwind discretization that results is equivalent to solving the following modified partial differential equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} - \frac{c\Delta x}{2}\frac{\partial^2 \psi}{\partial x^2} + \frac{c\Delta x^2}{6}\frac{\partial^3 \psi}{\partial x^3} - \frac{c\Delta x^3}{24}\frac{\partial^4 \psi}{\partial x^4} + \cdots = 0 \tag{3.39}$$

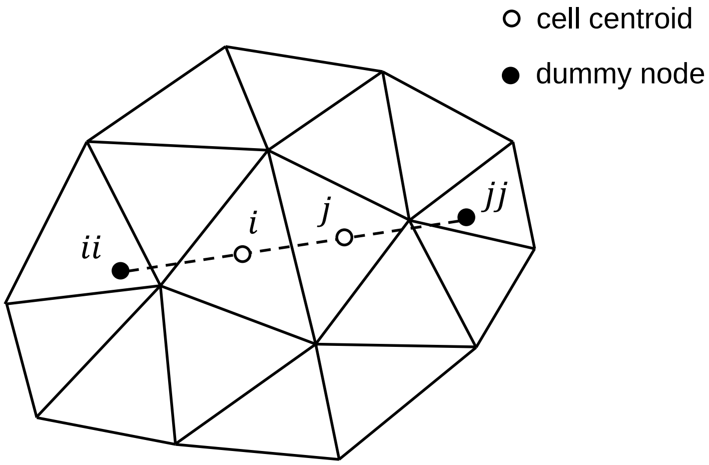

Figure 3.3 Example geometry for reconstruction of the fourth difference stencil. The dummy nodes are located in the line joining the centroids of cells i and j.

in which Δx is the grid spacing used in the discretization. Performing an analysis of the properties of this partial differential equation, it can be found that the even derivative terms control the amplitude of the wave, whereas the odd terms modify its propagation speed (Lomax, Pulliam, & Zingg, 2001). From this result, it is concluded that adding even difference terms to a central discretization of the wave equation is an effective means of achieving a stable solution, i.e., is an appropriate form of artificial dissipation.

When dealing with high-order numerical schemes for the Euler equations, it is also important to take into account Godunov's theorem (Hirsch, 1990), which states that any linear numerical scheme of order higher than one will show oscillations in the vicinity of discontinuities. Then, in order to obtain a satisfactory solution, the high-order scheme must switch to first order in the presence of large gradients.

Taking into account the need of designing higher-order artificial dissipation terms for the Euler equations, a third-order scheme was developed by (Jameson, Schmidt, & Turkel, 1981), in which the difference term $\mathbf{d}^{ij}$ consists on a blend of second and fourth differences. Although the original formulation was designed for structured meshes, it can be adapted to unstructured discretizations, see (Lyra & Morgan, 2000; Lyra & Morgan, 2002; Morgan & Peraire, 1998). With the notation and definitions that are being followed in this work, this higher-order difference term can be expressed as

$$\mathbf{d}^{ij} = \varepsilon_2(\mathbf{U}^j - \mathbf{U}^i) - \varepsilon_4(\mathbf{U}^{jj} - 3\mathbf{U}^j + 3\mathbf{U}^i - \mathbf{U}^{ii}) \quad (3.40)$$

where ε_2 and ε_4 are two nonlinear coefficients defined as

$$\varepsilon_2 = k_2 \max(\sigma_i, \sigma_j)$$
$$\varepsilon_4 = \max(0, k_4 - \varepsilon_2) \tag{3.41}$$

where k_2 and k_4 are constants that depend on the application (usually k_2 is on the order of unity and k_4 about an order of magnitude smaller), and σ_i, σ_j are pressure-based discontinuity sensors given by

$$\sigma_i = \left|\frac{p^j - 2p^i + p^{ii}}{p^j + 2p^i + p^{ii}}\right|$$
$$\sigma_j = \left|\frac{p^{jj} - 2p^j + p^i}{p^{jj} + 2p^j + p^i}\right| \tag{3.42}$$

In equations (3.40) and (3.42), the superscripts ii and jj denote the cells respectively containing two dummy points defined by the following coordinates (see Figure 3.3)

$$\mathbf{x}_p^{ii} = \mathbf{x}_c^i - \mathbf{v}_c^{ij}$$
$$\mathbf{x}_p^{jj} = \mathbf{x}_c^j + \mathbf{v}_c^{ij} \tag{3.43}$$

where $\mathbf{x}_c^i$ and $\mathbf{x}_c^j$ are the coordinates of the centroid of cells i and j (see the Appendix for the calculation of the centroid of a triangular cell) and $\mathbf{v}_c^{ij}$ is the vector connecting both centroids, that is $\mathbf{v}_c^{ij} = \mathbf{x}_c^j - \mathbf{x}_c^i$. This formulation is one of the many different possibilities to extend a fourth difference stencil to an unstructured mesh, a procedure which is usually called reconstruction. For details on the different options that can be used for stencil reconstruction refer to (Lyra & Morgan, 2002; Morgan & Peraire, 1998; Löhner, 2008). For the sake of clarity, the analogy with a conventional structured mesh in this case would be

$$\mathbf{d}^{ij} \leftrightarrow \mathbf{d}^{i+1/2},\ \ \mathbf{U}^{ii} \leftrightarrow \mathbf{U}^{i-1},\ \ \mathbf{U}^{i} \leftrightarrow \mathbf{U}^{i},\ \ \mathbf{U}^{j} \leftrightarrow \mathbf{U}^{i+1},\ \ \mathbf{U}^{jj} \leftrightarrow \mathbf{U}^{i+2} \tag{3.44}$$

and the same for pressure.

The behavior of the difference term given by equation (3.40) is governed by the pressure discontinuity detector in (3.42). When the flow is smooth and pressure variations are small, the value of σ is also small (actually of second order), whereas in the presence of discontinuities such as shock waves, it becomes close to unity. Then, in regions of smooth flow the coefficient ε_2 is of second-order and the dissipation term is of third order. This provides a background dissipation to the solution which helps to enhance its convergence to the steady-state, without affecting the second-order accuracy of the central scheme. On the other hand, near a shock wave the value of ε_2 is of unit order, which switches off the fourth difference by forcing $\varepsilon_4 = 0$ and converts the dissipation term to first-order, making it to be equivalent to the low-order artificial dissipation scheme presented previously. This nonlinear switching behavior serves to comply with the restrictions imposed by Godunov's theorem, thus allowing a sharp capturing of discontinuities with no oscillations.

The reconstruction method chosen here has the advantage of being computationally efficient since the coordinates of the dummy nodes do not change with time for a fixed mesh. Hence, during practical implementation, the procedure to find the cells ii and jj in which each dummy node falls only has to be executed once, namely, at the beginning of the numerical solution. Note also that in the case of boundary cells, the dummy nodes may fall outside the computational domain. In order to handle this situation, the conservative variables for a point located outside the domain are extrapolated from the interior cells following a constant difference extrapolation, that is

$$\begin{aligned} \mathbf{U}^{ii} &= 2\mathbf{U}^{i} - \mathbf{U}^{j} \\ \mathbf{U}^{jj} &= 2\mathbf{U}^{j} - \mathbf{U}^{i} \end{aligned} \tag{3.45}$$

The pressures p^{ii} or p^{jj} to be used in the sensor can then be directly obtained from these extrapolated conservative variables.

The artificial dissipation scheme presented in this section, which can also be implemented in both a scalar or matricial manner, is often also referred to as JST artificial dissipation scheme, or Jameson's artificial dissipation. It has been widely applied to the solution of compressible flows with good results (Hirsch, 1990; Swanson & Turkel, 1990).

3.3 Numerical treatment of boundary conditions

The necessary physical boundary conditions for external inviscid flow were described in section 2.3. In practice, their numerical implementation should be done with care in order to obtain a meaningful numerical solution. The approaches that have been adopted in this study are described next.

3.3.1 Body surface boundary condition

The body surface is assumed to be a nonporous solid wall, so the normal velocity component must be zero at any point in the surface in order to satisfy the flow tangency condition. This boundary condition is introduced naturally in the finite volume method by forcing the flux vector across an face w located at the wall to be

$$\mathbf{F}_n^w = \begin{bmatrix} 0 \\ n_x^w p^w \\ n_y^w p^w \\ 0 \end{bmatrix} \tag{3.46}$$

where n_x^w and n_y^w are the components of the face unit normal vector $\mathbf{n}^w$, pointing into the wall, and p^w is the pressure at the wall. The boundary face w belongs to a single cell i_b located at the boundary, which also has two interior faces shared with other cells of the mesh. Then, it is important to recognize that from the discretization given by equation (3.4), the value of the flow variables in the boundary cell, $\mathbf{U}^{i_b}$, will be a result from the contribution of each of the three faces belonging to the cell, and as a result the boundary condition is only enforced in a weak form. Actually, the fluid at the centroid of the cell i_b does not need to have a tangent velocity to the wall because it is not located at the wall.

For consistency with the piecewise constant approximation, the fluid variables at the wall are assumed to be the same as that of the cell i_b, so that (3.46) becomes

$$\mathbf{F}_n^w = \begin{bmatrix} 0 \\ n_x^w p^{i_b} \\ n_y^w p^{i_b} \\ 0 \end{bmatrix} \tag{3.47}$$

Other methods such as extrapolation or the solution of characteristic compatibility relations can be used in order to estimate the flow at the wall, see (Hirsch, 1990) for a general description. Note also that for the calculation of the flux vector at the boundary face no artificial dissipation is being used. This helps imposing the boundary condition in a more effective way, since the dissipation tends to excessively diffuse the important gradients that are found near the body boundary, reducing the accuracy of the solution.

The wall boundary condition given by (3.47) can also be thought as a central approximation between a “mirror” cell located at the other side of the wall and the boundary cell i_b. This means that this formulation is also valid as a boundary condition for planes of symmetry in symmetric flow fields.

Although this boundary condition is theoretically valid for any inviscid flow, it has to be taken into account that at the beginning of the transient simulation, the impulsive start from the initial conditions can lead to very large gradients in the wall, causing numerical instabilities. This is especially true for the case of high-speed flows using a uniform flow field as initial condition. In order to prevent this from occurring, the wall boundary condition can be imposed in a relaxed way, which can be accomplished by defining the following corrected velocity (Lyra & Morgan, 2002; Flores, Ortega, & Oñate, 2011; Ortega, 2014)

$$\mathbf{V}_{co} = \mathbf{V}^{i_b} - \kappa(\mathbf{V} \cdot \mathbf{n}^w)\mathbf{n}^w \tag{3.48}$$

where κ is a parameter that has a value of zero at the start of the simulation and is progressively ramped up to one after a given number of time steps. Then, the solution is allowed to penetrate the wall at the start but, as time evolves, the normal velocity at the wall goes to zero. With this velocity correction, the body surface boundary flux eventually becomes

$$\mathbf{F}_n^w = n_x^w \begin{bmatrix} \rho^{i_b} u_{co} \\ \rho^{i_b} u_{co}^2 + p^{i_b} \\ \rho^{i_b} u_{co} v_{co} \\ \rho^{i_b} u_{co} h_0^{i_b} \end{bmatrix} + n_y^w \begin{bmatrix} \rho^{i_b} v_{co} \\ \rho^{i_b} u_{co} v_{co} \\ \rho^{i_b} v_{co}^2 + p^{i_b} \\ \rho^{i_b} v_{co} h_0^{i_b} \end{bmatrix} \tag{3.49}$$

where the total specific enthalpy $h_0^{i_b}$ is also computed based on the corrected velocity.

3.3.2 Far-field boundary condition

As explained before, the numerical treatment of the far-field boundary condition has to account for the propagation of wave components across the boundary of the computational domain. Only those components which travel towards the interior of the domain can be prescribed, whereas the ones moving outwards have to be determined from the interior solution. Furthermore, as seen previously, the direction of propagation of one of the acoustic components changes depending on whether the flow is locally subsonic or supersonic.

One way to achieve the correct setup of the far-field boundary condition is to work with characteristic variables, i.e., changing from conservative to characteristic variables, prescribing the necessary components as a function of the local direction of propagation, and transforming back to conservative variables. However, this is not desirable in practice since it is a computationally expensive procedure. Fortunately, the same result can be achieved by means of the positive flux Jacobian, following the principles of upwind schemes discussed before. Denoting by $\mathbf{F}_n^f$ the flux vector across a far-field boundary face, the far-field boundary condition can be weakly enforced as (Lyra & Morgan, 2002; Flores, Ortega, & Oñate, 2011; Ortega, 2014)

$$\mathbf{F}_n^f = \frac{1}{2}\left(\mathbf{F}_n^{i_b} + \mathbf{F}_n^{\infty}\right) - \frac{1}{2}|\mathbf{A}_n^{Roe}|(\mathbf{U}^{\infty} - \mathbf{U}^{i_b}) \tag{3.50}$$

where the superscripts i_b and ∞ respectively denote the flow conditions at the far-field boundary cell and at the freestream. The upwind correction $|\mathbf{A}_n^{Roe}|(\mathbf{U}^{\infty} - \mathbf{U}^{i_b})$ is evaluated at the Roe averages between the states at i_b and ∞ by means of the algorithm described in section 3.2.1, but in this case no limitation is applied on the eigenvalues. This can be easily done by setting $\nu_l = 0$ and $\nu_{nl} = 0$ at the corresponding far-field faces.

To ensure that the condition given by (3.50) works properly, it is important for the far-field boundary to be located enough distance away from the body, so as to have near freestream conditions at the far-field boundary cells.

With this formulation, the correct propagation of information is accounted for in the solution, without the need of directly prescribing any characteristic variables. This should guarantee that no wave reflections are produced at the far-field boundary, thus minimizing undesirable perturbations in the flow field. Moreover, using this method avoids the need of differentiating between the subsonic/supersonic or inlet/outlet portions of the boundary.

3.4 Time integration

Once the spatial discretization presented in (3.4) is complete, a semi-discrete scheme is obtained in the form of a system of nonlinear ordinary partial differential equations. The residual $\mathbf{R}^i$ of a given cell i can now be defined as

$$\mathbf{R}^i = \frac{1}{A^i}\sum_{e=1}^{N_e} \mathbf{F}_n^e \, l^e = -\frac{\partial \mathbf{U}^i}{\partial t} \tag{3.51}$$

which expresses the flux balance over the cell faces per unit area.

There are many different options available for the time integration of a system defined by (3.51). Since our purpose is to obtain steady state solutions, the drivers for choosing a time integration scheme here are robustness and a fast convergence rate. Attending to these considerations, an explicit multistage Runge-Kutta scheme has been selected, given by

$$\begin{gathered} \mathbf{U}_s^i = \mathbf{U}_n^i - \alpha_s \Delta t^i \, \mathbf{R}^i\left(\mathbf{U}_{s-1}^i\right) \\ \mathbf{U}_{n+1}^i = \mathbf{U}_{n_s}^i \end{gathered} \tag{3.52}$$

for $s = 1, \dots, n_s$, with n_s being the number of stages of the scheme. The subscripts n and $n+1$ respectively denote the current and the next time levels, Δt^i stands for the local time step associated to cell i, and α_s are coefficients that depend on the number of stages employed. Note that at each stage, the residual has to be evaluated, as expressed by $\mathbf{R}^i\left(\mathbf{U}_{s-1}^i\right)$. This scheme was originally introduced in the calculation of inviscid flows by (Jameson, Schmidt, & Turkel, 1981) for central schemes with artificial dissipation, and is mainly designed to allow the use of relatively high Courant numbers, in some cases higher than one. The most commonly used option, which is also the one adopted here, corresponds to the choice (Hirsch, 1990; Löhner, 2008; Lyra & Morgan, 2002)

$$\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{1}{3}, \quad \alpha_3 = \frac{1}{2}, \quad \alpha_4 = 1 \tag{3.53}$$

that usually offers a good trade-off between the allowable time step and the computational cost per time iteration (Flores, Ortega, & Oñate, 2011). This method is one of the so-called minimal storage Runge-Kutta schemes, which only require one extra copy of the conservative variables at each time step, that is, $\mathbf{U}_n^i$ and $\mathbf{U}_s^i$ for each cell in the mesh.

In order to minimize the computational cost of the scheme, a common practice is not to update the artificial dissipation terms at each stage of the scheme, but only at specific ones. A usual strategy is to calculate the dissipation terms only at the first stage, although this can cause some instabilities at the beginning of the transient when the flow properties change significantly between stages, especially if a low order dissipation scheme is being used. To increase the robustness of the solution, in this work the artificial dissipation is updated at the first and third stages of the time integration scheme.

In practical implementations, the numerical solution is advanced in time until a specific convergence criterion is satisfied. A suitable criterion for the convergence of inviscid compressible flows is based on the norm of the density residual, see for instance (Hirsch, 2007). When the norm of the residual decreases by a given number of orders of magnitude, the numerical solution is stopped and a steady state is assumed. The L^2 norm, or Euclidean norm, is often considered for that purpose, defined as

$$\left\|\mathbf{R}_\rho\right\| = \sqrt{\sum_{i=1}^{N_c}\left(R_\rho^i\right)^2} \tag{3.54}$$

where R_ρ^i is the density residual of cell i, as given by the first component of the residual vector $\mathbf{R}^i$, and N_c is the number of cells in the mesh.

Another relevant topic related with time integration is the choice of the initial conditions. In the case of high-speed inviscid flows below the hypersonic regime, the way to proceed is almost always to assume the initial flow-field to be the same as the freestream conditions. This creates strong gradients near the walls at the beginning of the time marching, which can in turn produce severe numerical instability. However, as discussed in the previous section, and as will be seen further below, some strategies have been devised in order to control such impulsive start. On the other side, it is interesting to comment that in the case of hypersonic flows, the availability of local surface inclination methods such as the Newtonian theory gives the ability of setting up much more appropriate initial conditions on common geometries. More details can be found in (Anderson, 2006).

3.4.1 Calculation of the time step

Due to the fact that the chosen time integration scheme is explicit, the maximum allowable time step is constrained by the Courant-Friedrichs-Levy (CFL) stability criterion, which for the Euler equations can be formulated as (Hirsch, 1990)

$$\Delta t^i = CFL\,\frac{h^i}{\left(V^i + c^i\right)} \tag{3.55}$$

where CFL is the allowable Courant number, V^i and c^i are the velocity magnitude and the speed of sound at cell i, and h^i is a characteristic size of the cell. A reasonable characteristic cell size is be taken to be the minimum cell height (Flores, Ortega, & Oñate, 2011), which for triangular cells is defined as

$$h^i = \frac{2A^i}{\max(l^e)} \tag{3.56}$$

where l^e is the length of cell side e, with $e = 1, \dots, 3$. Since we are not interested in the transient solution, a local time stepping is chosen in order to accelerate the convergence to steady state, so that each cell progresses at its maximum possible time step.

Attending to the limitation given by the CFL condition in (3.55), it is reasonable to think that an implicit scheme will pay-off in the case of steady state solutions. Actually, it may do. However, its implementation is much more complex than the explicit case, and their flexibility to adapt to changes in the numerical scheme is lower. Besides being simpler to implement, explicit schemes also have the advantage of being straightforward to parallelize (Löhner, 2008). This is very attractive when considering future improvements in the numerical solution, and is one of the main reasons why the explicit option has been adopted here.

3.4.2 A relaxed update procedure to promote the positivity of thermodynamic variables

Due to the impulsive start from freestream conditions, during the first time integration steps negative values of the thermodynamic variables can appear in the flow field, with the subsequent failure of the numerical solution. To prevent the appearance of these local, spurious negative values during the convergence process and increase the robustness of the solution, the density and pressure can be updated using a relaxation process to help them remain positive. One of such relaxed update procedures is given by (Lyra & Morgan, 2002)

$$p_{n+1}^i = p_n^i + \Delta p^i \left[1 + \eta \left(\theta - \frac{\Delta p^i}{p_n^i} \right) \right]^{-1} \tag{3.57}$$

whenever $\Delta p^i / p_n^i \leq \theta$, with the recommended values being $\theta = -0.2$ and $\eta = 2$. The increment Δp^i simply denotes $\Delta p^i = p_{n+1}^i - p_n^i$. The same procedure is then applied for the density ρ. The use of this special update method (and similar ones) is found to play a significant role in the solution of high-speed inviscid flows; see (Yee, Klopfer, & Montagné, 1988).

4 NUMERICAL IMPLEMENTATION AND TEST CASES

4.1 Description of the solver developed

With the objective of putting in practice all the theoretical concepts presented before, a numerical solver has been developed using Fortran. This constitutes a direct implementation of the unstructured finite volume method for the solution of the Euler equations. Fortran has been chosen for its high performance in scientific computing and its simplicity of implementation.

The code is designed to be able to solve two-dimensional steady inviscid flows around closed geometries (external flows), at speeds ranging from subsonic to moderate supersonic regimes, up to about Mach 3. The solution of higher Mach number flows has not been considered due to the complex physical phenomena that takes place when high-temperature effects become important, which would require including real gas effects in the solution (Anderson, 2006).

In order to minimize the memory requirements and reduce the number of calculations needed, a side-based strategy has been taken, see for instance (Löhner, 2008). This approach consists on calculating the residual (3.51) by iterating over faces instead of cells. Any mesh considered has a given number of global faces, which, excluding the boundaries, are shared by two triangular cells each. Similarly, each triangular cell has three local faces. Then, for the numerical scheme chosen, the flux approximation at a global face shared between two cells contributes in the same amount to the residual of the each of the two cells, but with different sign. As a consequence, the use of a side-based solution almost halves the calculations otherwise required in the case of calculating the residual cell by cell.

A diagram of the code structure is shown in Figure 4.1. As can be observed, the program follows a modular approach, which has been adopted to favor the possible implementation of future improvements in the solution. For the generation of unstructured triangular meshes, the pre and postprocessing tool GiD has been used, for which a problem type was created in order to automatically generate the input files for the solver. On the other end, MATLAB has been chosen for the analysis of results and the generation of different types of plots.

The program starts by reading the input data files of the solution. These contain the node coordinates and the cell connectivities of the mesh, as well as the simulation parameters and the freestream flow conditions. Next, the necessary data structures are built, which include the definition of the global faces of the mesh and the calculation of necessary data for the reconstruction of the fourth difference stencils, to be used later in the evaluation of

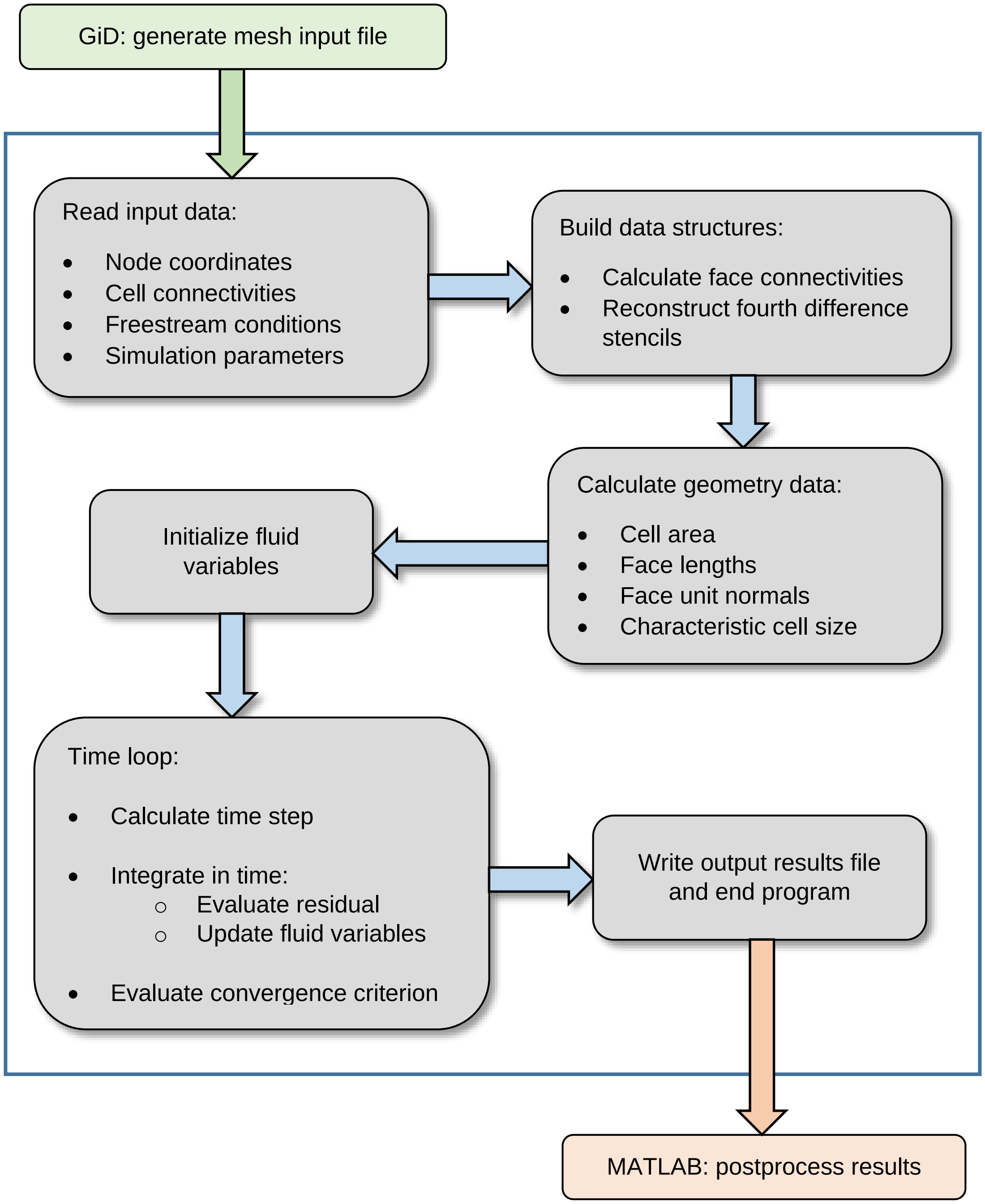

Figure 4.1 Diagram showing the different structural blocks and flow of the developed code.

the JST artificial dissipations terms. Once the face connectivities have been defined, a subroutine that calculates the geometrical data is executed, which produces the lists of cell areas, side lengths and normals, and the characteristic cell sizes required for evaluating the time step. Upon successfully reaching this point, the fluid variables are initialized and the time marching process starts. At each time iteration, the multistage scheme is executed, computing the residual and updating the variables at each stage. Then, the convergence criterion is evaluated. When convergence is achieved, the time loop stops and a function writes an output file with the results prepared to be postprocessed with MATLAB.

Many different solution parameters can be controlled by the user, which mainly include: the stages of the time integration method, the Courant number, the target density residual for the convergence criterion and the different coefficients and form of the artificial dissipation terms.

4.2 Test cases

A set of five different test cases have been solved in order to validate the numerical results obtained with the developed solver. The first three of them are based on the transonic and supersonic flow around a NACA0012 airfoil, which correspond to a series of rigorous airfoil benchmark tests performed by (Pulliam & Barton, 1985). The fourth case deals with the supersonic flow past a double wedge airfoil, which is an interesting benchmark problem since it has an analytical solution, and the fifth and last case tackles the difficult problem of the inviscid supersonic flow past a circular cylinder. A summary of the different cases, specifying the flow configuration for each one, can be found in Table 4.1.

Case ID	Geometry	M_∞	α (degrees)
1	NACA0012	0.85	1
2	NACA0012	0.95	0
3	NACA0012	1.2	7
4	Double wedge airfoil	2	0
5	Circular cylinder	3	-

Table 4.1 Summary of the different test cases considered.

The reference length L for these cases is taken to be the chord of the body, which is defined as the linear distance between the leading and trailing edges. For convenience, a unit value of the chord is adopted in all five cases, so $L = 1$.

There are some other parameters that are also common to all the examples. On one side, the ratio of specific heats is always chosen to be the standard value $\gamma = 1.4$. On the other side, the convergence criterion is always set to be a reduction of 8 orders of magnitude in the density residual with respect to its initial value. Moreover, as commented before, a four-stage integration scheme is selected in all the tests, with the coefficients given by (3.53), in which the artificial dissipation terms are only calculated at the first and third stages. Note that thanks to the use of the nondimensional equations, for a fixed value of γ only the freestream Mach number and the angle of attack are needed to completely define the flow in each case.

Regarding the body surface boundary condition (refer to section 3.3.1), the parameter κ is implemented so as to increase with the number of time iterations i_t following

$$\kappa = 1 - 0.9^{i_t} \tag{4.1}$$

which can be assumed to reach a value of one at about 100 iterations.

4.2.1 Calculation of nondimensional quantities for the analysis of results

In order to properly validate the numerical results obtained, there are different important nondimensional quantities that should be verified. The analysis of nondimensional results is more general and offers a simpler and unambiguous framework to perform comparisons against reference results. In this study, the quantities considered are the Mach number, the pressure coefficient, the nondimensional change in entropy and the aerodynamic force and moment coefficients.

Recalling the nondimensional variables defined in section 2.4, the Mach number can be calculated from the results of the simulation as

$$M = \sqrt{\frac{\tilde{u}^2 + \tilde{v}^2}{\tilde{T}}} \tag{4.2}$$

Similarly, the following expression can be derived for the pressure coefficient

$$C_p = \frac{2\left(\tilde{p} - \frac{1}{\gamma}\right)}{M_\infty^2} \tag{4.3}$$

In a similar fashion, the change in entropy for a calorically perfect gas can be given by

$$\Delta s = s - s_\infty = c_p \ln \tilde{T} - R \ln(\gamma \tilde{p}) \tag{4.4}$$

which can be made dimensionless using for example the specific heat at constant pressure, that is

$$\Delta \tilde{s} = \frac{\Delta s}{c_p} = \ln \tilde{T} - \frac{\gamma - 1}{\gamma} \ln(\gamma \tilde{p}) \tag{4.5}$$

Regarding the calculation of aerodynamic force coefficients, the lift and drag coefficients for a two-dimensional body are defined as (Anderson, 2011)

$$\begin{aligned} c_l &= c_n \cos\alpha - c_a \sin\alpha \\ c_d &= c_n \sin\alpha + c_a \cos\alpha \end{aligned} \tag{4.6}$$

where c_n and c_a respectively denote the force coefficients in directions perpendicular and parallel to the body chord. Assuming a unit chord, as explained in the previous section, these coefficients are given for an inviscid flow by

$$\begin{aligned} c_n &= \int_0^1 (C_{p,l} - C_{p,u}) d\tilde{x} \\ c_a &= \int_0^1 \left(C_{p,u} \frac{d\tilde{y}_u}{d\tilde{x}} - C_{p,l} \frac{d\tilde{y}_l}{d\tilde{x}} \right) d\tilde{x} \end{aligned} \tag{4.7}$$

where the subscripts u and l refer to the upper and lower surfaces of the body. In a similar manner, the moment about the leading edge becomes

$$c_m^{LE} = \int_0^1 (C_{p,u} - C_{p,l}) \tilde{x} d\tilde{x} + \int_0^1 \left(C_{p,u} \frac{d\tilde{y}_u}{d\tilde{x}} \tilde{y}_u - C_{p,l} \frac{d\tilde{y}_l}{d\tilde{x}} \tilde{y}_l \right) d\tilde{x} \tag{4.8}$$

In some references, the moment about the quarter-chord point is considered. This can be directly obtained with

$$c_m^{1/4} = c_m^{LE} + \frac{c_n}{4} \tag{4.9}$$

It is interesting to note that since we are dealing with an inviscid flow, the only surface force responsible for the aerodynamic forces and moments is pressure. Moreover, recalling what is known as d’Alembert’s paradox, see for instance (Anderson, 2011), drag is identically zero for a two-dimensional closed body in an inviscid flow with no discontinuities. As a result, any deviation from zero in the computed drag coefficient can only come from two sources: the presence of discontinuities such as shock waves, leading to what is known as wave drag, or numerical errors.

Finally, it is also important to note that the piecewise constant, cell-centered approach adopted gives a constant value of the flow variables in each cell. However, at the time of representing the results, it is more convenient to use the solution values at the mesh nodes, that is, at the cell vertices. In order to obtain the flow variables at the nodes, a simple procedure has been considered. For a given node, the nodal values of the desired quantities are computed as the arithmetic average between respective values at the cells shared by the node. This introduces a small additional error in the solution.

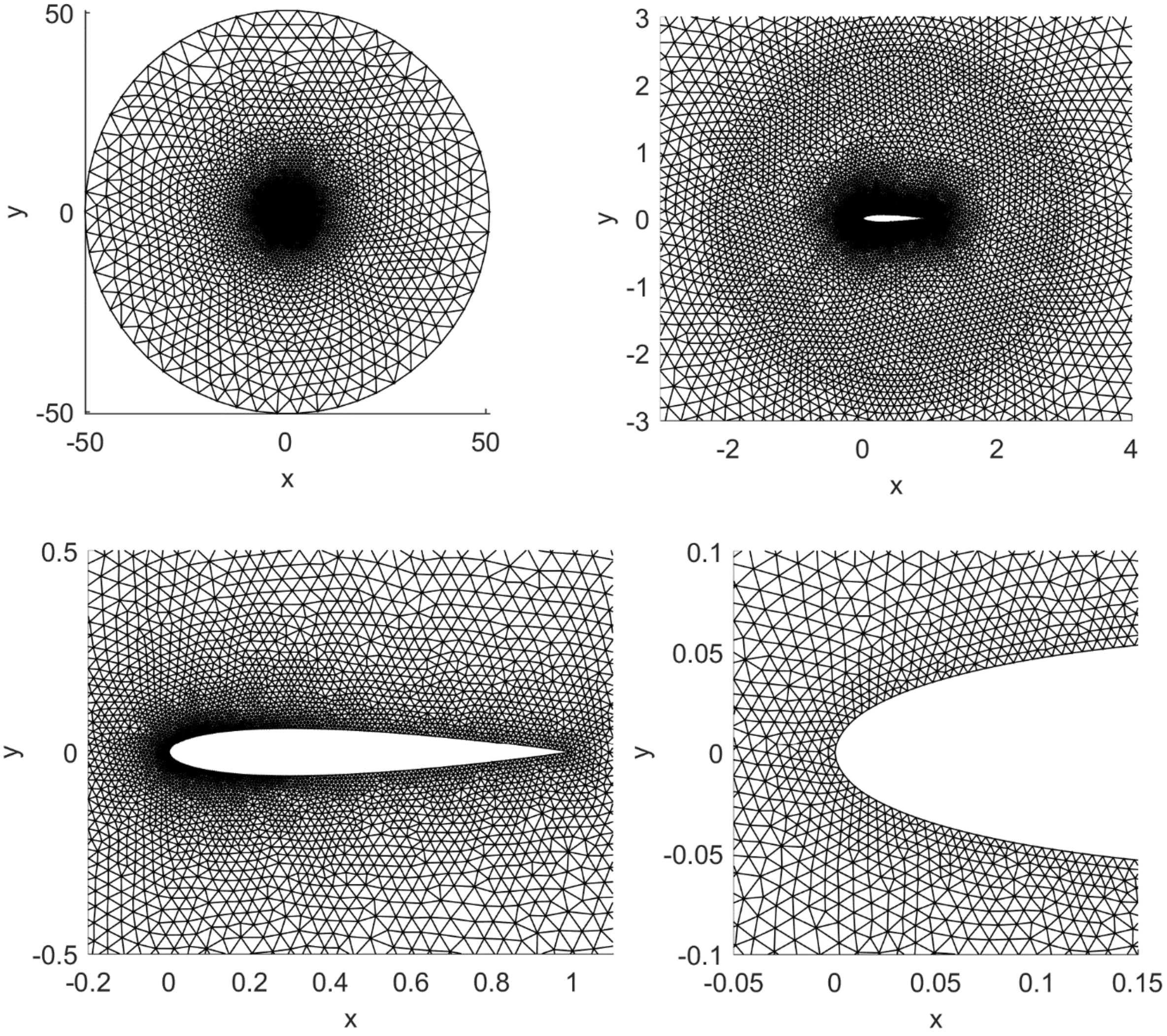

Figure 4.2 Different views of the mesh used for the NACA0012 test cases.

4.2.2 Transonic flow around a NACA0012 airfoil: case 1

The first case focuses on the solution of the complete flow field around a NACA0012 airfoil at $M_\infty = 0.85$ and $\alpha = 1º$. This is one of the test cases that were carried out by (Pulliam & Barton, 1985). In this reference, the solutions were obtained using the scalar variant of the JST artificial dissipation scheme (as described in 3.2.2) in a structured mesh. In order for the comparison of results to be as close as possible, the same configuration has been adopted here, so the solver was setup to use the scalar JST artificial dissipation terms. The values of the remaining relevant solution parameters that have been set for this problem are: $CFL = 1$, $k_2 = 1$ and $k_4 = 1/16$, along with a regular update of the variables.

The distance of the far-field boundary shows to have a strong impact in the solution. For radial distances below 50 chords between the body and the far-field, significant numerical errors can be encountered in the results, especially affecting the aerodynamic force and moment coefficients (Pulliam & Barton, 1985). In their reference calculations, they chose a radial distance of 48 chords and additionally applied a far-field vortex correction. In this work, a distance of 50 chords has been selected, without any further corrections.

The unstructured mesh that has been used for the three NACA0012 cases is shown in Figure 4.2. It consists of 11098 nodes and 21878 triangular cells. As can be observed, the cell size is progressively decreased towards the airfoil surface, which also contains some refinement around the leading edge. The cell size at the airfoil surface is 0.011 (0.004 at the leading edge), which leads to around 250 surface cells. Since mesh adaptation is not used, this constitutes an appropriate number of cells to capture the desired flow details with good resolution. See (Esfahani, 2013) for a similar example.

The results obtained for this case are presented in Figure 4.3 and the top of Figure 4.6, which respectively show the pressure coefficient and Mach number distributions, the convergence history and the generation of entropy in the solution. As can be observed, both the upper and lower shock waves are sharply captured with very small oscillations, demonstrating the good performance of the artificial dissipation scheme in use. Note also the general smoothness of the flow field, which is indicative of the satisfactory quality that can be achieved regardless of the use of a simple piecewise constant approximation. Overall, the results obtained show a very good agreement with the ones found in the reference calculation, with almost identical surface distributions of the pressure coefficient and the Mach number. The calculated aerodynamic force and moment coefficients are given in Table 4.2, which also show small differences with the reference values.

	c_l	c_d	$c_m^{1/4}$
Own results	0.3697	0.0596	-0.1285
Reference	0.3938	0.0604	-0.1378

Table 4.2 Comparison of the aerodynamic force and moment coefficients for case 1.

Although the Euler equations neglect the dissipative mechanisms of the Navier-Stokes model, the fact that they allow for discontinuities in the flow field means that there can be different regions where the entropy of the fluid is different. In each region, however, the entropy must remain constant along the streamlines. This is the case, for example, of a flow field with shock waves, such as the ones considered here. This behavior can be seen in the entropy change plots displayed in the right of Figure 4.6. A part from indicating the presence of shock waves, the entropy generated in the solution is a good indicator of the regions where more artificial dissipation is being introduced in the solution, and provides a direct measure of the zones where a finer mesh should be used in order to improve the quality of the solution.

4.2.3 Transonic flow around a NACA0012 airfoil: case 2

The next case considered also deals with the transonic flow around a NACA0012 airfoil, but at $M_\infty = 0.95$ and $\alpha = 0º$. As announced before, this is also another of the benchmark cases found in (Pulliam & Barton, 1985). The mesh and the simulation parameters used are the same as that of the previous example.

The results for this case are summarized in Figure 4.4 and the middle of Figure 4.6. This case is characterized by a flow structure known as a fishtail shock, due to the triangular shock pattern appearing behind the trailing edge of the airfoil. Since the freestream Mach number is very close to one, the flow over almost all the airfoil surface is supersonic, resulting in two oblique shocks very close to the trailing edge. The flow after these shocks is still slightly supersonic, and a weak normal shock is formed far downstream in order to decelerate it back to the freestream velocity. The computed strength and location of the weak normal shock is in excellent agreement with the reference results, as well as the rest of the flow-field and the computed aerodynamic coefficients, summarized in Table 4.1. Note also the faster convergence rate of this case with respect to the previous one, which is due to the presence of larger supersonic regions in the flow field.

	c_l	c_d	$c_m^{1/4}$
Own results	0.0003	0.1111	-0.0002
Reference	0.0000	0.1103	-0.0000

Table 4.3 Summary of the aerodynamic force and moment coefficients for case 2.

During numerical tests, it has been found that the far-field boundary can greatly influence the position of the weak normal shock if the outer boundary is not far enough. For instance, using a mesh with an outer boundary located at only 6 chords away from the airfoil, the normal shock appears located at only about 2 chords downstream from the trailing edge, when the correct position is around 4 chords. This constitutes, once again, another reason to stress out the fundamental importance of using large enough computational domains if meaningful results are to be obtained.

4.2.4 Supersonic flow around a NACA0012 airfoil

The last application example focusing on the NACA0012 geometry considers a supersonic freestream at $M_\infty = 1.2$ and $\alpha = 7º$. Once again, the same mesh and parameters are used, since this is also the approach taken in the reference solution.

For this case, the resulting quantities are displayed in Figure 4.5, the bottom of Figure 4.6 and Table 4.4. It can be seen how a weak detached shock wave appears at a considerable distance upstream of the airfoil's leading edge, which is properly resolved by the numerical scheme without significant oscillations. This is due to the low supersonic Mach number considered, which is close to unity. In fact, as the freestream velocity is increased, the bow shock rapidly becomes stronger and moves towards the airfoil surface. Due to the larger angle of attack used in this case, it can be appreciated how the stagnation point moves to the lower surface. On the other side, an oblique shock is also captured at the trailing edge, which slows down the flow back to the freestream state. Once again, a good agreement is achieved with the reference solution. Note the very small differences in the resulting aerodynamic coefficients.

	c_l	c_d	$c_m^{1/4}$
Own results	0.5203	0.1560	-0.1091
Reference	0.5232	0.1554	-0.1105

Table 4.4 Comparison of lift, drag and moment coefficients for case 3.

In a similar fashion as before, the supersonic nature of the flow enhances the convergence rate of the solution. On other hand, it is interesting to notice the streamlined pattern in the entropy color map, which for this case is clearer than in the previous ones thanks to the detached shock. At the near normal portion of the shock, which is found in front of the stagnation point, the entropy increase for the incoming flow is larger than further away, where the shock is more oblique to the freestream.

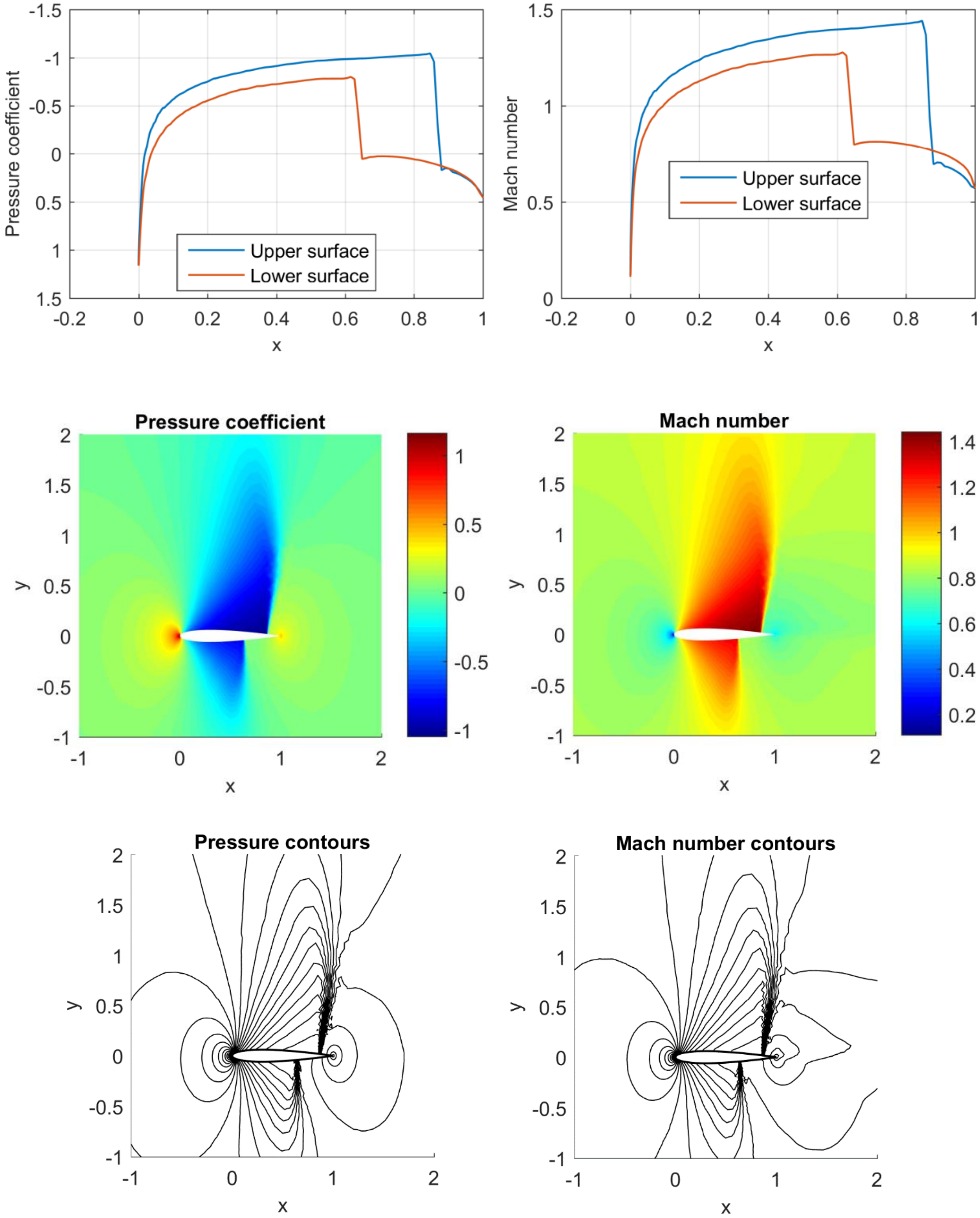

Figure 4.3 Pressure coefficient and Mach number results for case 1.

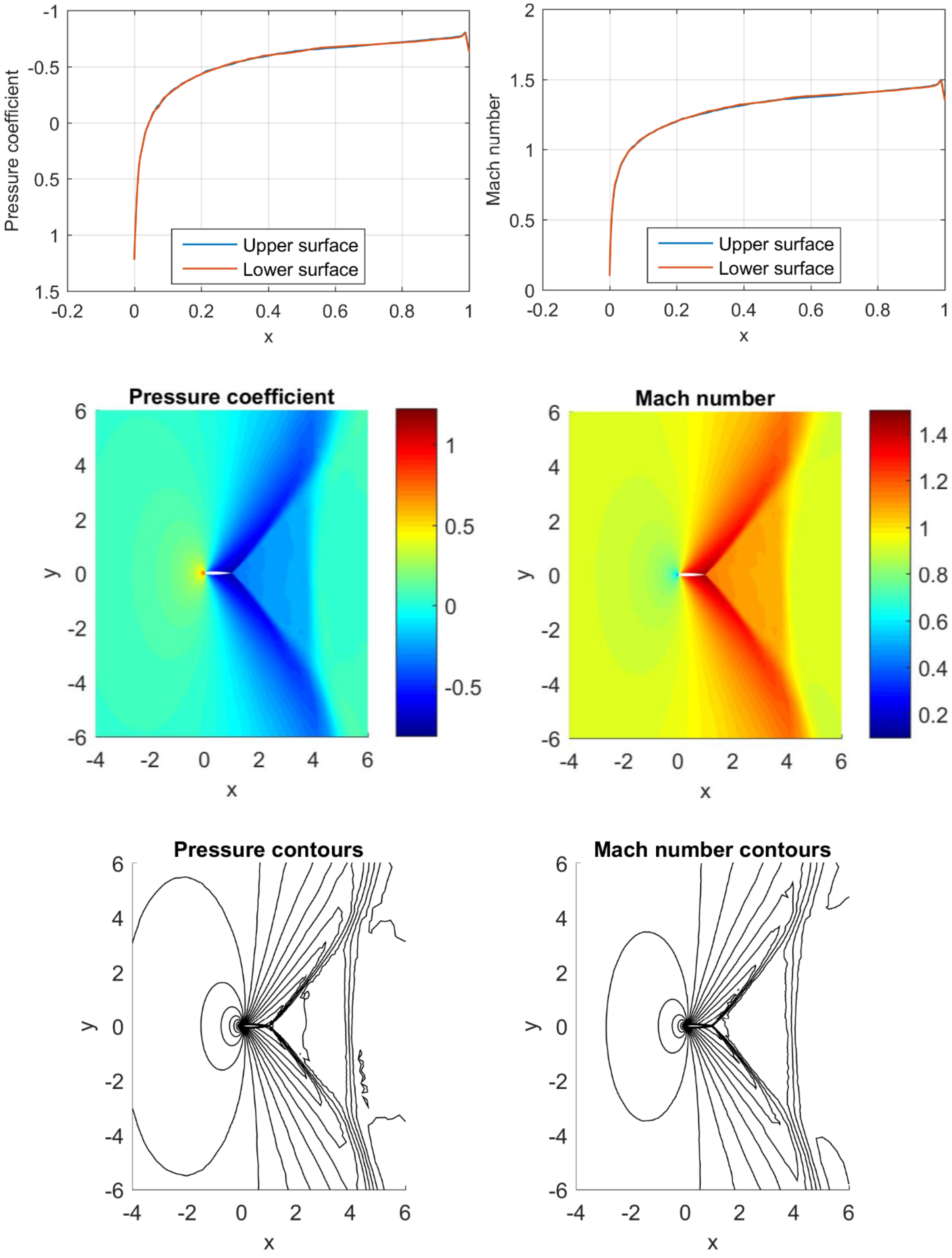

Figure 4.4 Pressure coefficient and Mach number results for case 2.

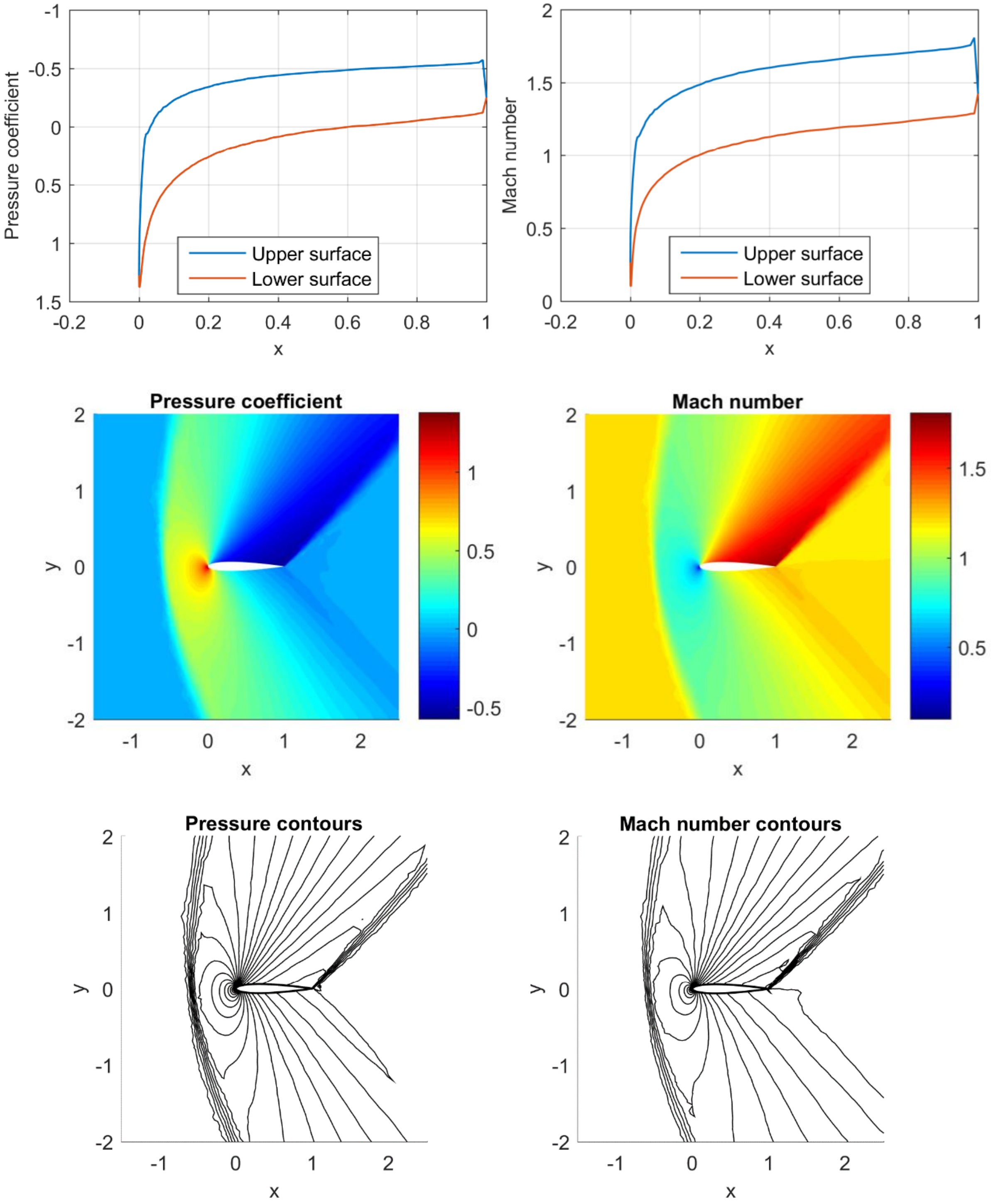

Figure 4.5 Pressure coefficient and Mach number results for case 3.

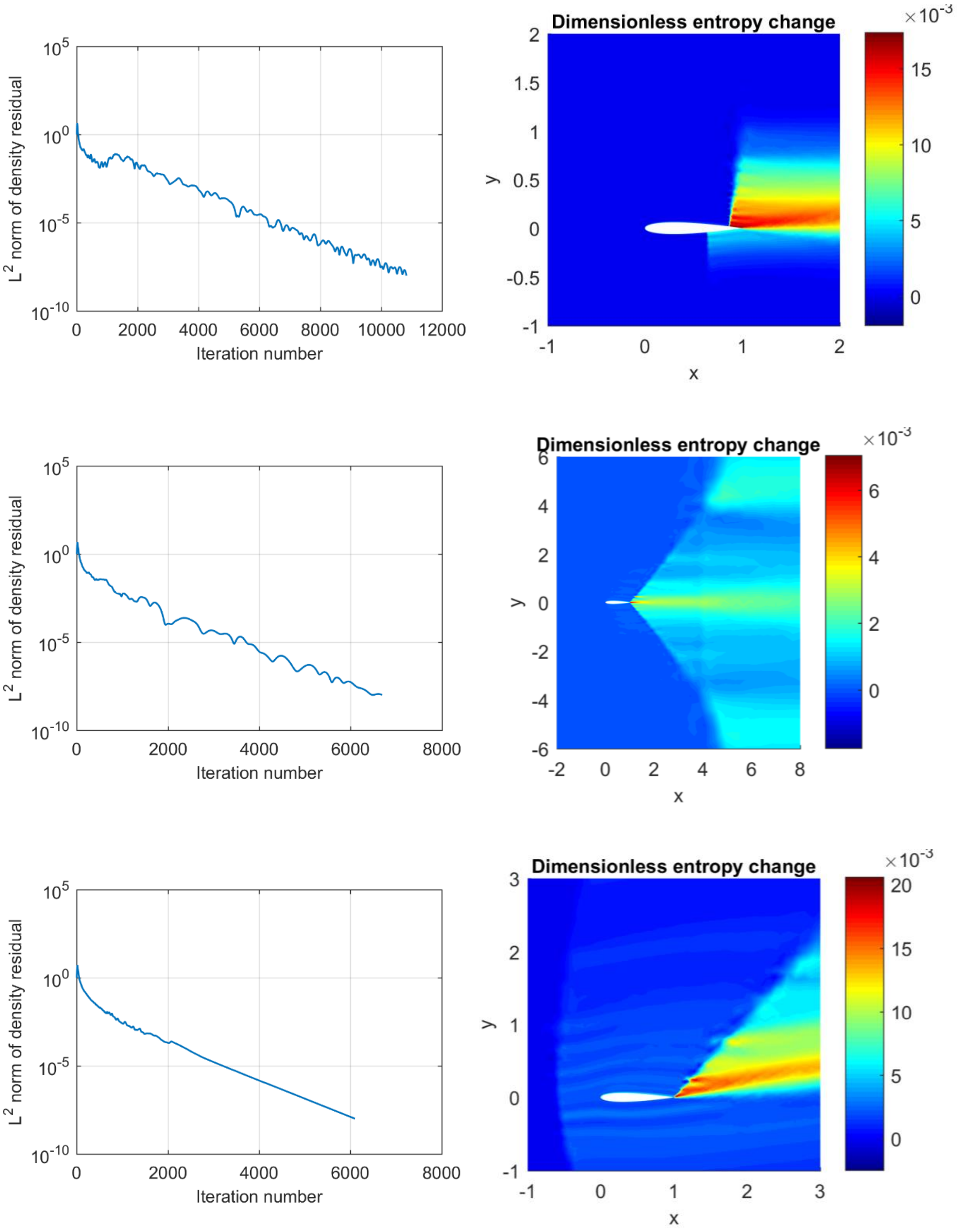

Figure 4.6 Convergence history and entropy generation for cases 1 (top), 2 (middle) and 3 (bottom).

4.2.5 Supersonic flow past a double wedge airfoil

This fourth test case aims to the solution of the supersonic flow past a double wedge airfoil, also known as a diamond-wedge airfoil. The wedge angle is 15 degrees, which forces the flow to turn a value of 30 degrees at the expansion points located at half of the chord, and the freestream Mach number and angle of attack are respectively $M_\infty = 2$ and $\alpha = 0º$. This case is attractive as a benchmark problem because it has an analytical solution, and contains both shock and expansion waves. The analytical solution is given by the Rankine-Hugoniot relationships and the Prandtl-Meyer function, see for instance (Anderson, 2011) for the specific details.

The mesh considered for this problem is sketched in Figure 4.9, which contains 23778 cells and 12076 nodes. Only the cells near the body are shown because the rest of the domain is equal the one for the NACA0012 airfoil discussed before. The outer boundary is once again 50 chords away from the airfoil, and some refinement is applied in the leading edge and the expansion points. This time, the matrix form of the JST artificial dissipation terms have been used, which improve the accuracy of the solution. The parameters chosen are $CFL = 0.6$, $k_2 = 1$, $k_4 = 1/16$, $\nu_l = 0.05$ and $\nu_{nl} = 0.1$.

A summary of the numerical results is provided in Figure 4.7 and Figure 4.8. As can be observed, the surface quantities show a good agreement with the analytical solution. The pressure coefficient is perfectly predicted, but the Mach number after the expansion wave presents a small deviation. It has been found in numerical experiments that at the expansion point, the numerical scheme in use tends to introduce an excessive amount of artificial dissipation at the surface, affecting only the surface Mach number predicted by the solution (the values computed far from the surface are correct). Actually, the use of a scalar artificial dissipation for this problem leads to significant errors in the Mach number behind the expansion wave. On the other hand, some small oscillations are also present at the shock and expansion waves. Although the JST scheme mimics a first-order upwind scheme in the vicinity of discontinuities, it does not lead to the same exact formulation as the first-order artificial dissipation scheme given in section 3.2.1, and use of a more elaborated pressure sensor may lead to better results. It has been checked that with the use of the first order scheme, these oscillations disappear.

The aerodynamic force coefficients also have been calculated for this problem, and can be found in Table 4.5. The values are close to the analytical solution, demonstrating that the overall quality of the solution is satisfactory.

	c_l	c_d
Numerical	0.0003	0.1704
Analytical	0	0.1715

Table 4.5 Comparison of aerodynamic force coefficients for case 4.

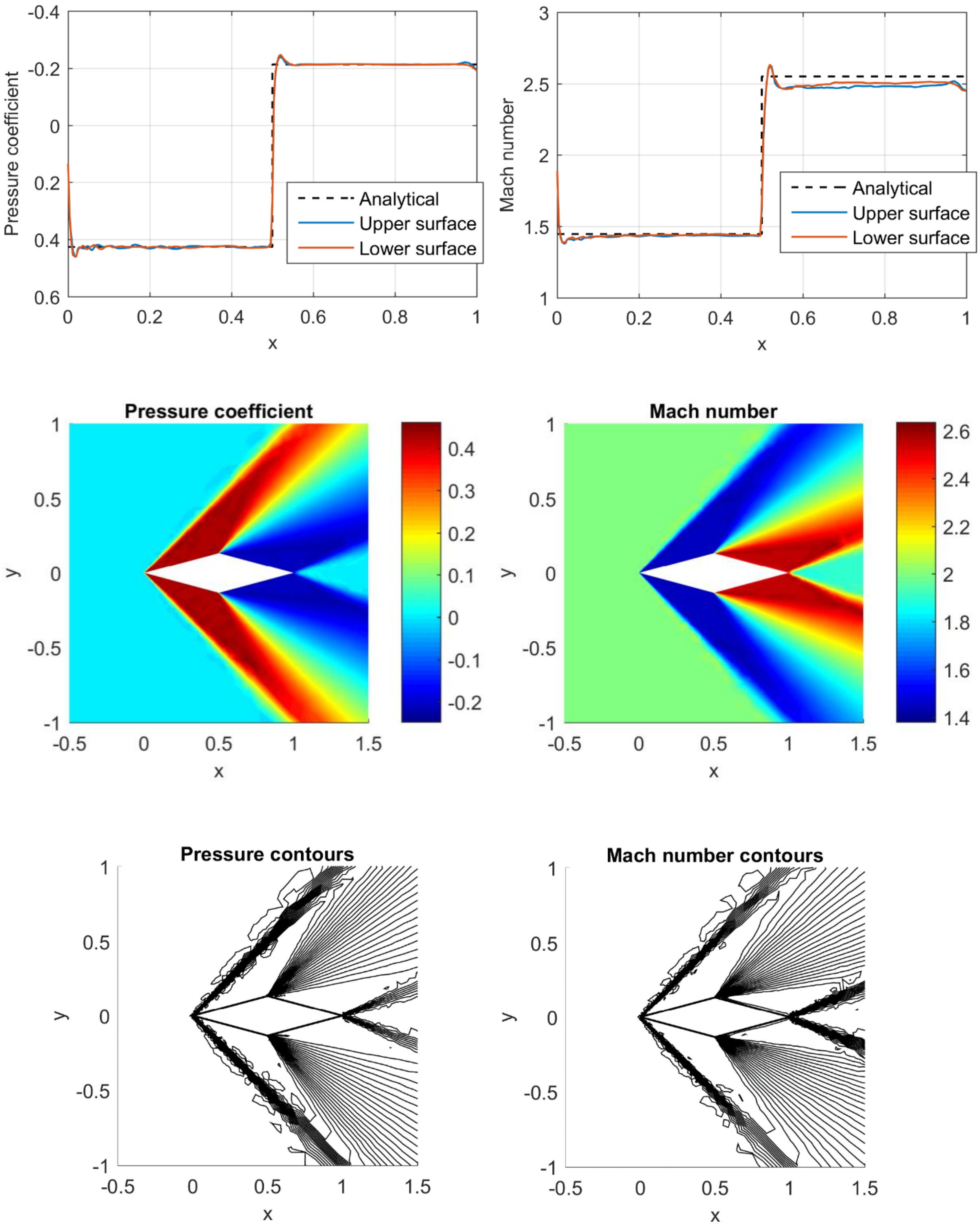

Figure 4.7 Pressure coefficient and Mach number results for case 4.

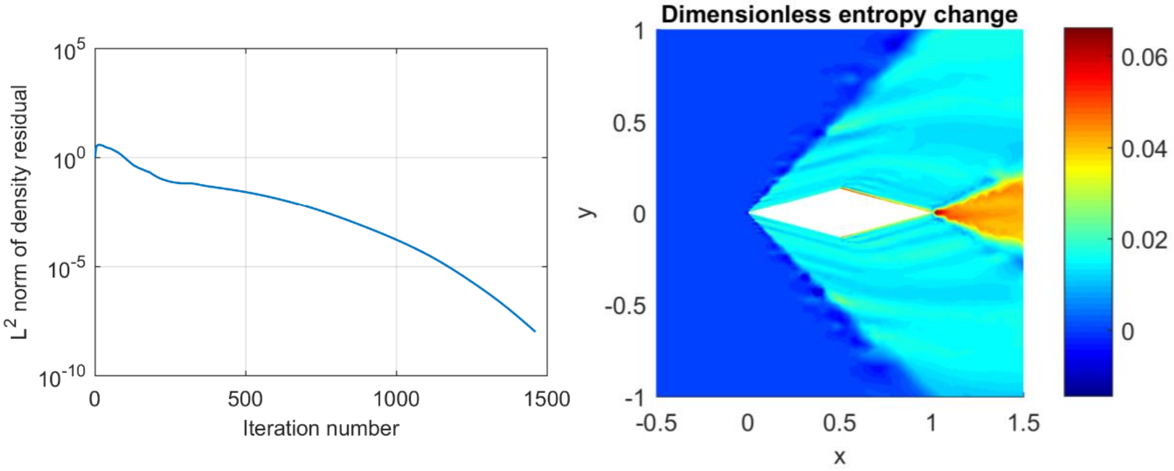

Figure 4.8 Convergence history and entropy change for case 4.

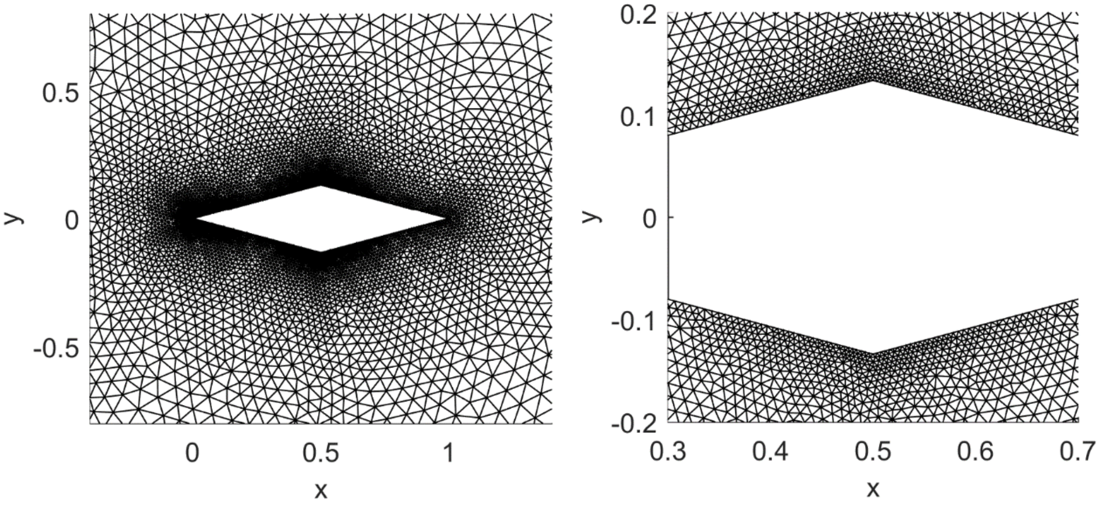

Figure 4.9 Details of the mesh used for the numerical solution of case 4.

4.2.6 Supersonic flow past a circular cylinder

The last test case that has been attempted is the solution of the complete supersonic flow field around a circular cylinder. The configuration of this case is based on a solution that was performed by (Lyra & Morgan, 2002), with a freestream Mach number of $M_\infty = 3$. This problem is challenging in terms of stability behavior due to the presence of a strong bow shock close to the cylinder surface and a complex, rarefaction zone at back. The strong perturbations created by a very blunt object such as the circular cylinder, combined with the relatively high Mach number considered creates very strong gradients in the solution which are usually difficult to manage. As a result, this constitutes a firm benchmark to test the robustness of the numerical solution. It is important, however, to be aware that the solution of this case through the Euler equations is by no means realistic. In the real case, the flow field presents a large detachment region at the back of the cylinder, so the consideration of viscous phenomena is essential for reproducing the correct physical behavior.

As usual, a mesh with a radius of 50 chords has been constructed for this case, as represented in Figure 4.12. The cell size is on the same order as the previous cases, resulting in 9999 nodes and 19779 cells. The reference solution is based on a finite element high-order upwind scheme, which is quite different from the approach considered in this analysis. Different numerical trials have been performed with the code developed. However, due to the difficulty of this case, only the solution using the matrix low-order artificial dissipation scheme has been obtained. The application of the JST dissipation terms has always resulted in bad convergence behavior and computational instability.

The solution obtained here is presented in Figure 4.10 and Figure 4.11. It was computed with $CFL = 0.5,\ \nu_l = 0.2,\ \nu_{nl} = 0.2$ and the use of the special update procedure described in section 3.4.2. This, along with the relaxed imposition of the body boundary condition, have been found to be of significant importance in order to be able to advance the solution in time at the first steps. Observing the results, it can be seen that despite the use of a low-order scheme, the important flow field features are captured with an acceptable resolution. Actually, although the solution is over-diffusive due to the low-order upwinding, the obtained pressure coefficient distribution agrees very well with the values found in the reference calculation. Both the strength and location of the detached shock are correct, as well as the predicted pressure values at the back of the cylinder. Regarding the Mach number, the jump across the bow shock is well captured. However, in a similar way as in the previous case, the large expansion that takes place at the second half of the cylinder is not accurately resolved. The strong shock wave that is encountered at the rear part is in the correct location, but the surface Mach number just before the shock is about 4.5, whereas in the reference solution the value reached is only 4. In their studies, (Yee, Klopfer, & Montagné, 1988) and (Lyra & Morgan, 2002) suggest that the use a more elaborated entropy fix than that given by equation (3.36) is required when the freestream Mach number starts to become relatively high. This may be one of the possible explanations why an excessive amount of artificial dissipation is added to the solution in the strong expansion region.

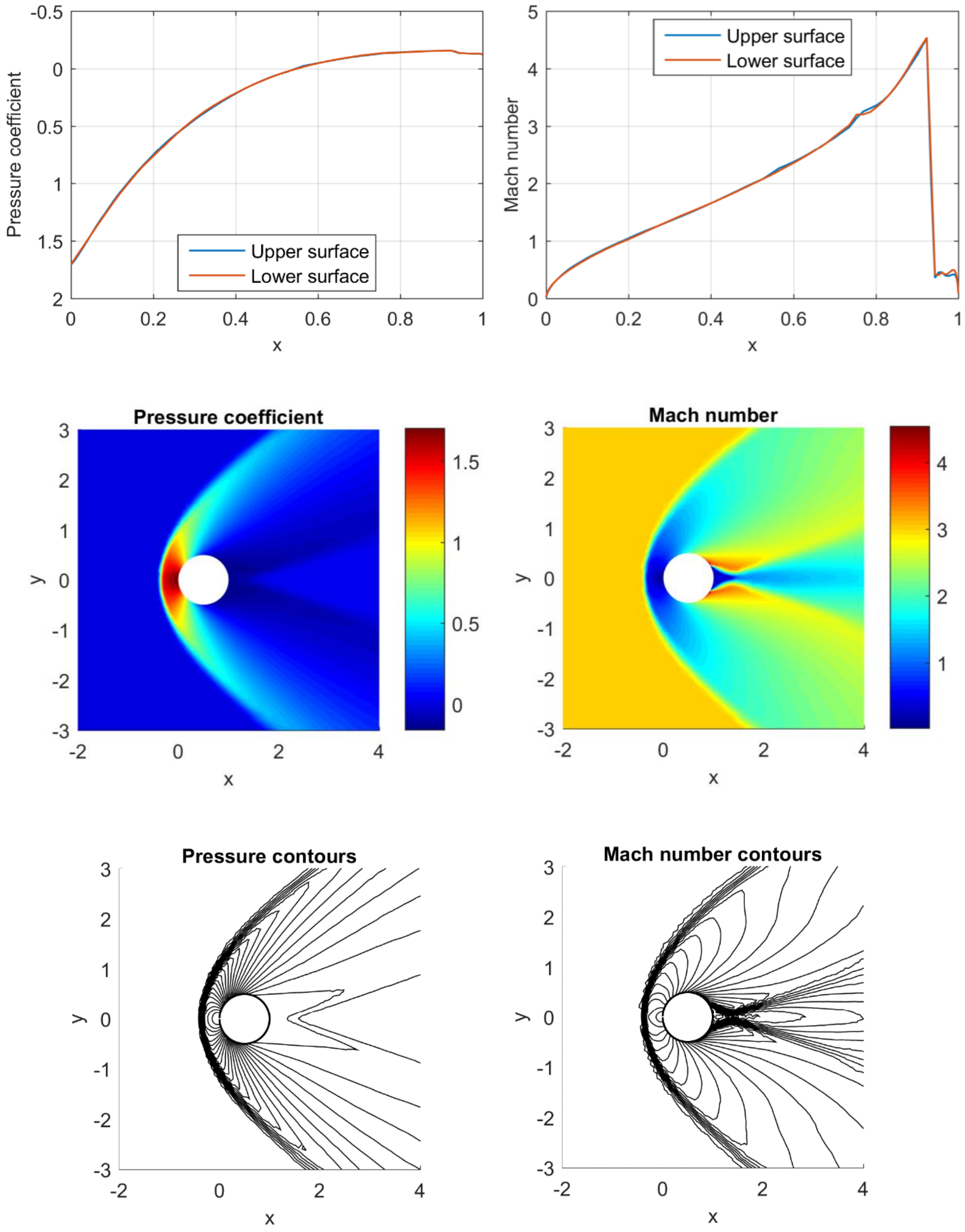

Figure 4.10 Pressure coefficient and Mach number results for case 5.

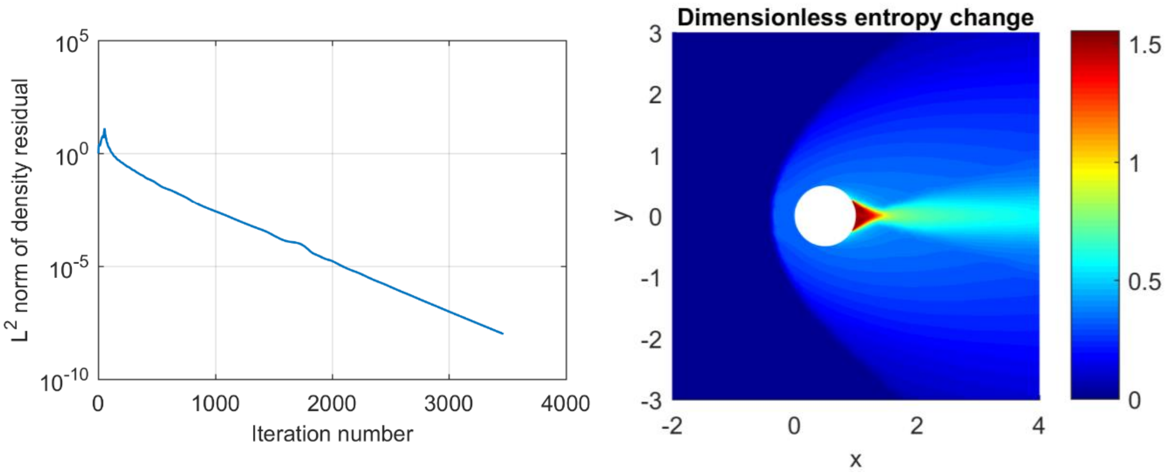

Figure 4.11 Convergence history and entropy change for case 5.

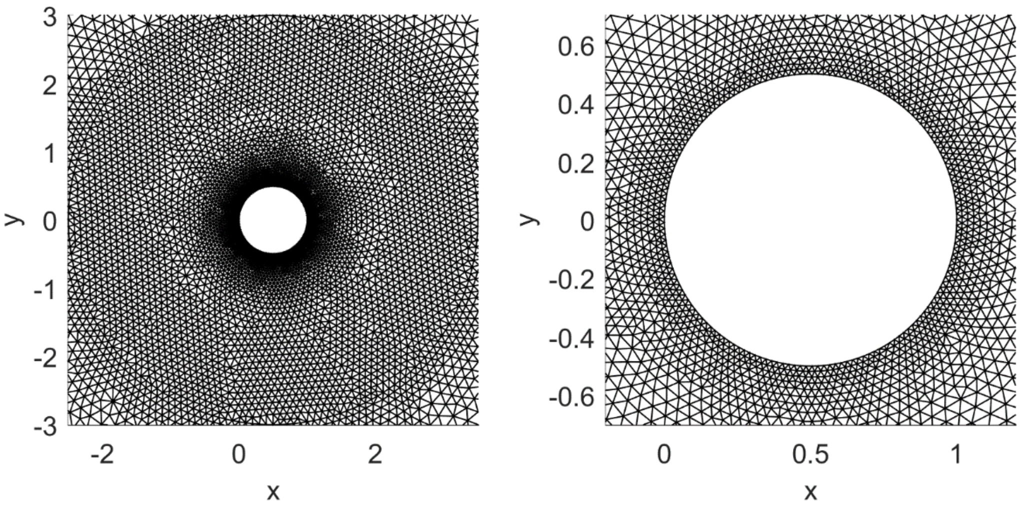

Figure 4.12 Detail views of the mesh for case 5.

5 CONCLUSIONS AND FUTURE WORK

In summary, the main objective of this work has been achieved upon the obtention of satisfactory results with the numerical solver developed.

Due to the hyperbolic mathematical nature of the Euler equations, the fact of taking into account the propagation of information in the fluid is found to be of paramount importance in order to obtain meaningful numerical results. The wave components that travel through the flow field have been described and analyzed, emphasizing the relevance of carefully imposing the boundary conditions and justifying the need of using artificial dissipation when central numerical schemes are considered.

The unstructured finite volume discretization described has been successfully applied to all the test cases considered, demonstrating that the decisions taken regarding the piecewise constant approximation of values inside the cells, and the implementation of boundary conditions are perfectly valid for the solution of inviscid compressible flows. Besides, the use of the Jameson-Schmidt-Turkel artificial dissipation terms has provided satisfactory results in most of the test cases considered, which also confirms the feasibility of the stencil reconstruction technique adopted and the choices made regarding the calculation of the positive flux Jacobian.

Focusing on the results obtained, it has been found that the overall numerical solution has a very good performance in the three NACA0012 airfoil examples calculated, showing an excellent agreement with the reference results. A good solution has also been achieved for the supersonic flow past a double wedge airfoil, matching the analytical results with very reasonable accuracy, although some small deviation is found in the Mach number behind the expansion waves. Finally, for the complete supersonic flow around a circular cylinder, only the solution with a low-order artificial dissipation scheme could be obtained. The numerical difficulties associated to the large perturbations created by the cylinder and the presence of a strong bow shock and rarefaction zones make this problem quite challenging. In order to obtain acceptable results for this case, the use of a more robust numerical scheme should be considered, such as a high-order upwind method. Significant discrepancies are also observed in this case with respect to the Mach number predicted in the large expansion region at the second half of the cylinder. This, similarly to the expansion in the double wedge airfoil case, appears to be caused by the introduction of excessive artificial dissipation in the surface. A possible explanation for this behavior may be the need for a better entropy fix, and is an aspect that should be further investigated.

Other improvements of the numerical solver can be considered for future implementation, which mainly include mesh adaptation, parallelization, the extension to three dimensional problems and the introduction of a thermally perfect gas behavior, to allow modeling the variation of specific heats with temperature in the case of higher Mach number flows.

Bibliography

Anderson, J. D. (1995). *Computational fluid dynamics: the basics with applications.* New York: McGraw-Hill.

Anderson, J. D. (2006). *Hypersonic and high-temperature gas dynamics* (2nd ed.). Reston: AIAA.

Anderson, J. D. (2011). *Fundamentals of aerodynamics* (5th ed.). New York: McGraw-Hill.

Esfahani, M. K. (2013). *A contribution to the finite element analysis of high-speed compressible flows and aerodynamic shape optimization.* PhD thesis, Universitat Politécnica de Catalunya.

Flores, R., Ortega, E., & Oñate, E. (2011). PUMI: un código explícito no estructurado para resolver las ecuaciones de Euler. *Rev. Int. Mét. Num. Cálc. Dis. Ing., 27*, 129-145.

Hirsch, C. (1990). *Numerical computation of internal and external flows. Volume 2: Computational methods for inviscid and viscous flows.* Chichester: John Wiley & Sons.

Hirsch, C. (2007). *Numerical computation of internal and external flows. Volume 1: Fundamentals of computational fluid dynamics* (2nd ed.). Oxford: Butterworth-Heinemann.

Hu, G. (2001). *The development and applications of a numerical method for compressible vorticity confinement in vortex-dominant flows.* PhD thesis, Virginia Polytechnic Institute and State University.

Jameson, A., Schmidt, W., & Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA 14th Fluid and Plasma Dynamic Conference.* Palo Alto, California, June 23-25.

Löhner, R. (2008). *Applied computational fluid dynamics techniques: an introduction based on finite element methods* (2nd ed.). Chichester: John Wiley & Sons.

Lomax, H., Pulliam, T. H., & Zingg, D. W. (2001). *Fundamentals of computational fluid dynamics.* Springer.

Lyra, P., & Morgan, K. (2000). A review and comparative study of upwind biased schemes for compressible flow computation. Part I: 1-D first-order schemes. *Arch. Comput. Meth. Engng., 7*(1), 19-55.

Lyra, P., & Morgan, K. (2000). A review and comparative study of upwind biased schemes for compressible flow computation. Part II: 1-D higher-order schemes. *Arch. Comput. Meth. Engng., 7*(3), 333-377.

Lyra, P., & Morgan, K. (2002). A review and comparative study of upwind biased schemes for compressible flow computation. Part III: Multidimensional extension on unstructured grids. *Arch. Comput. Meth. Engng., 9*(3), 207-256.

Morgan, K., & Peraire, J. (1998). Unstructured grid finite-element methods for fluid mechanics. *Reports on Progress in Physics, 61*, 569-638.

Ortega, E. (2014). *Development and applications of the finite point method to compressible aerodynamics problems.* PhD thesis, Universitat Politècnica de Catalunya.

Pulliam, T. H., & Barton, J. T. (1985). Euler computations of AGARD working group 07 airfoil test cases. *AIAA 23rd Aerospace Sciences Meeting.* Reno, Nevada, January 14-17.

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics, 43*(2), 357-372.

Swanson, R. C., & Turkel, E. (1990). On central-difference and upwind schemes. *ICASE report 90-44*.

Swanson, R. C., Radespiel, R., & Turkel, E. (1997). Comparison of several dissipation algorithms for central difference schemes. *ICASE report 97-40*.

Turkel, E. (1988). Improving the accuracy of central difference schemes. *ICASE report 88-53*.

Yee, H., Klopfer, G., & Montagné, J.-L. (1988). High-resolution shock-capturing schemes for inviscid and viscous hypersonic flows. *Technical Memorandum 100087, NASA Ames Research Center*.

APPENDIX: Practical calculation of geometrical quantities for triangular cells

Given an arbitrary triangle in two-dimensional space defined by vertices A, B and C (see Figure A.1), with respective coordinates (x_A, y_A), (x_B, y_B) and (x_C, y_C), the following formulas can be used for an easy computation of geometrical quantities for triangular cells.

Centroid coordinates

The centroid coordinates (x_c, y_c) of the cell are given by

$$x_c = \frac{1}{3}(x_A + x_B + x_C)\,, \quad y_c = \frac{1}{3}(y_A + y_B + y_C)$$

Cell area

The cell area A can be directly calculated by taking the absolute value of the determinant of the following matrix

$$\frac{1}{2}\begin{pmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{pmatrix}$$

so

$$A = \frac{1}{2}|x_A y_B + y_A x_C + x_B y_C - y_B x_C - y_A x_B - x_A y_C|$$

Side length

Having the coordinates of the vertices, the calculation of side lengths is straightforward (see Figure A.1)

$$l^a = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$l^b = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

$$l^c = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}$$

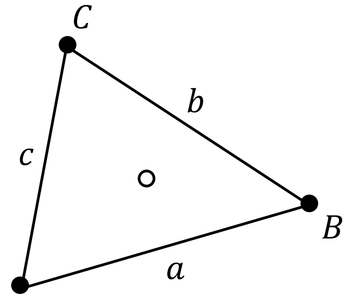

Figure A.1 Arbitrary triangular cell.

Side unit normal

The calculation of side normals is also readily achieved having the coordinates of the vertices. For instance for side a, a vector along the side is

$$\mathbf{v}^a = \left(v_x^a, v_y^a\right) = (x_B - x_A, y_B - y_A)$$

and a normal vector to the side is then

$$\mathbf{n}^a = (v_y^a, -v_x^a)$$

Since the direction of the normal is important, it has to be checked. We are usually interested in the outward normal to the triangle. To verify the normal direction, calculate the vector along the other side of the triangle that is shared by the same vertex from which $\mathbf{v}^a$ emanates, in this case $\mathbf{v}^c$, defined as

$$\mathbf{v}^c = (x_C - x_A, y_C - y_A)$$

Now, take the dot product $\mathbf{n}^a \cdot \mathbf{v}^c$. It shows that if the result is positive, then the normal is pointing into the triangle, so change its sign. Otherwise it is already pointing outwards.

Finally, the unit normal vector is obtained dividing $\mathbf{n}^a$ by l^a.

The same procedure applies for the normal of sides b and c.