SAPIENZA
UNIVERSITÀ DI ROMA

Shock-fitting techniques on 2D/3D unstructured and structured grids: algorithmic developments and advanced applications.

La Sapienza University of Rome
PhD course in Space and Aeronautical Engineering (XXXV cycle)

Alessia Assonitis
ID number 1613481

Advisor
Prof. Renato Paciorri

Co-Advisors
Prof. Aldo Bonfiglioli
Prof. Carl Ollivier-Gooch
Prof. Sergio Pirozzoli

Reviewers
Prof. Matthew J. Zahr
Dr. Mario Ricchiuto

Board of Examiners
Prof. Sergio De Rosa
Prof. Pierluigi Di Lizia
Prof. Andrea Ferrero

Academic Year 2022/2023

Thesis defended on January 23th 2023

Shock-fitting techniques on 2D/3D unstructured and structured grids: algorithmic developments and advanced applications.
PhD thesis. Sapienza University of Rome

© 2022 Alessia Assonitis. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: alessia.assonitis@uniroma1.it

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Prof. Renato Paciorri for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.
I am also deeply grateful to Prof. Aldo Bonfiglioli for his constant support, guidance and assistance throughout this project.
Special thanks also to Prof. Carl Ollivier-Gooch and Prof. Sergio Pirozzoli for their invaluable advices and technical support on my study.
Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

Over the last decades the simulations of compressible flows featuring shocks have been one of the major drivers for developing new computational algorithms and tools able to compute also complex flow configurations. Nowadays, Computational fluid dynamics (CFD) solvers are mainly based on shock capturing methods, which rely on the integral form of the governing equations and can compute all type of flows, including those with shocks, using the same discretization at all grid points. Consequently, these methods can be implemented with ease and provide physically meaningful solutions also for complex flow configurations, features particularly attractive for CFD community. Although shock capturing methods have been the subject of development and innovations for more than 40 years, they are plagued by several numerical problems due to the shocks capture process, such as discontinuities finite-width, numerical instabilities and reduction of accuracy order in the shock downstream region, which are still unsolved and probably will never find a solution. For this reason, there is a renewed interest in shock-fitting techniques: in particular, these methods explicitly identify the discontinuities within the flow field and compute them by enforcing the Rankine-Hugoniot jump relations. Because of this modelling, shocks are represented by zero thickness discontinuities, so that significant advantages can be gained in terms of solution quality and accuracy improvements. Furthermore, this class of methods is immune to the numerical problems linked to shock capture process. Following this research line, the presented Thesis proposes new developments and advanced applications of shock-fitting techniques, which prove that these methods are an effective option regarding shock capturing ones in simulating flows with shocks, able to provide also a better understanding of all the phenomena linked to shock waves.

Contents

Introduction

Computational fluid dynamics (CFD) is a branch of fluid mechanics that employs numerical simulation to predict flows behaviour and solve problems regarding fluids motion. Specifically, CFD study for high speed flows in aerospace engineering was initiated in the early 1960s and nowadays aerospace industry is one of the most prevalent user of CFD tools: as a matter of fact, these solvers are of paramount importance for designing components and evaluating aerodynamic performance of flight vehicles and re-entry capsules, such as the drag, lift, noise and thermal loads. Flow simulation is a tough task, especially when dealing with high speed flows featuring shocks: in this case, numerical computations demand particular attention to asses their reliability. As an example, let us consider shock/boundary layer interactions, which are commonplace in almost every supersonic viscous flow and especially severe for space vehicles during re-entry manoeuvrers. These interactions are often the cause of the presence of extended separation regions and localized heating, thus becoming a critical factor in determining the design of vehicles or propulsion systems since they could lead to unexpected components failures: such was the experience on October 1967 of NASA's X-15-2 research airplane, when flying at Mach 6.7 [1, 2]. Therefore, it is evident that a correct numerical modelling of the shock waves is of fundamental importance in order to obtain reliable CFD solutions, especially when considering complex flow configurations.
Nowadays, CFD commercial solvers are generally based on the so-called shock capturing methods, which rely on the integral form of the governing equations and are capable of computing all type of flows, including those with shocks, using the same algorithm at all grid points/cells. As a consequence, shock capturing schemes can be implemented with ease, a feature particularly attractive for CFD community. Actually, the widespread use of this class of solvers hides some unsolved problems [3], especially when the computation of shocked flows is involved. First of all, captured shocks are characterized by a numerical thickness, which generally spans two or three computational cells: the width of a captured shock is therefore orders of magnitude larger than the true physical shock-width. Not only this introduces an error in the shock position, which is of the order of the local mesh spacing [4], but also the internal shock-structure is grossly mis-represented [5]: as a matter of fact, cells crossed by captured shocks are characterized by intermediate states between upstream and downstream ones, that are clearly unphysical. These states are supposed to be the origin of numerical artefacts such as the carbuncle phenomenon [5, 3], which is particularly severe when considering hypersonic flows around blunt bodies [6], or the production of spurious perturbations. Generally, all these numerical problems affect the flow field downstream of the shock, thus making the computed CFD

solution unreliable. Even though many studies focused their attention on factors that influence these phenomena and tried to find remedies, still no definitive cure is available. Furthermore, among the shock capturing methods shortcomings, it is worth mentioning the reduction of accuracy to first order of the shock capturing scheme, regardless the design accuracy order. Specifically, the spatial order of accuracy of the discretisation scheme must be reduced to first order in the neighbourhood of the discontinuity in order to avoid the appearance of unphysical oscillations of the flow variables: this is typically achieved by the use of so called limiter functions [7], non-linear devices that, when sensing the presence of a discontinuity, locally reduce the design order of the discretisation to first order. However, the capturing scheme accuracy order deteriorates to first order not only in the immediate neighbourhood of the captured discontinuity, where limiters are switched on, but in the entire region downstream of the captured discontinuity, as reported in Ref. [4, 8].
In the last decades, many attempts have been made to overcome shock capturing drawbacks: some researchers [9, 10] focused on developing multi-dimensional upwind shock capturing schemes, capable of making the computed solution not sensitive to the orientation of the cell faces relative to the shock front, since the mis-alignment between mesh faces and shock front generally causes artificial spurious waves to arise and to propagate in the shock downstream region, worsening the solution quality. Nevertheless, even this class of capturing schemes is not entirely exempt from the problems which plagued the traditional shock capturing methods. A different way to proceed that can help in improving the quality of shock capturing calculations consists in coupling the flow solver with an a posteriori mesh adaptation algorithm, as in Ref [11]. This option could be advantageous for two main reasons: not only mesh adaptation allows to bring the captured shock width closer to its physical value thanks to the cells clustering in the vicinity of the discontinuity, but it also allows to improve the solution quality by aligning the faces of the grid cells with the shock front, which aims to reduce the arise of spurious waves. However, these solutions only alleviate the shock capturing problems by reducing their amplitude and do not remove them completely.
Because of the aforementioned shock capturing troubles, it appears sensible to investigate a class of computational methods which are alternative to shock capturing, the so-called shock fitting techniques: specifically, shock fitting methods are able to overcome all the capturing drawbacks previously described, despite nowadays they are less popular and used with respect to shock capturing ones. Historically, shock fitting was proposed by Emmons [12] and further developed and made popular starting from 1960s in the structured grid context by Moretti and co-workers [13]. Modelling shock-waves via shock fitting consists in locating and tracking the motion of the discontinuities which are treated as boundaries between regions where a smooth solution to the governing equations exists. The appropriate jump relations (algebraic equations connecting the states on either side of the discontinuity and its local speed) are used to calculate the space–time evolution of the discontinuity, whereas the flow in the smooth regions can be approximated using any available gas-dynamic solver. Beside Moretti, other researchers applied this technique to 2D and 3D flows in the 1970s and 1980s and in the early 1990s (see, for instance, Refs. [14, 15, 16]), following two different approaches: the boundary shock fitting and the floating shock fitting. In the first approach [17, 18], shocks are made to coincide with boundaries of

the multi-block structured grids which compose the computational domain, so that the enforcement of the jump relations amounts to prescribe a new set of boundary conditions, but it poses strong limitations for modelling shocks interactions. The floating shock-fitting version [14, 19] was developed to deal with more complex flow configurations, including shock/shock and shock/boundary layer [20] interactions, since in this approach, discontinuities can freely move over a background structured mesh and the shock-fronts are described by their intersections with the grid-lines: anyway, these methods are in general very complicated to implement because they require massive changes within the gas-dynamic solver. In the last century, both the aforementioned approaches used in association with structured grid solvers based on the quasi-linear form of the equations, allowed to accurately compute flows characterised by strong discontinuities using the modest computational resources available at that time and to provide high quality numerical solutions. Indeed, in this class of methods shock waves are represented as mathematical discontinuities featuring null thickness, specifically poly-lines or surfaces, resp. in 2D and 3D space. As a consequence, by explicitly identifying and computing discontinuities within the flow field, the carbuncle phenomenon and the generation of spurious waves are prevented [6] and the design accuracy order is recovered downstream the shocks [21]. Nevertheless, the price to pay is an increasing coding complexity, especially for dealing with complicated shock patterns. However, in the past years the development of shock fitting techniques was interrupted mainly for a reason: the tight link between shock fitting algorithms and grid topology, that prevented the coding of general proposed solvers able to compute all kinds of flows, including those featuring complex shock configurations. Indeed, it must be noticed that, at the very beginning of CFD, numerical computations were performed on structured grids: therefore, it is clear that these shock fitting methods had faced the rigid topology of these grids, which does not lend itself to model complex shock configurations. Moreover, shock fitting methods were complex to code and required important modifications of the computational kernel to handle the presence of a shock within the domain: for this reason, they were progressively abandoned since 1990s.

A turning point in the development of shock fitting methods was marked in 2009, when Paciorri and Bonfiglioli proposed a new shock fitting algorithm based on unstructured grids [22], which allows to relieve most of the algorithmic difficulties that have contributed to the dismissal of the shock fitting technique in favour of the simpler shock-capturing methods: specifically, this novel technique breaks free from the limitations imposed by structured grids framework, and it is able to compute several 2D complex flow configurations by exploiting unstructured meshes flexibility, also when involving shocks interactions [23] as well as 2D unsteady flows [24]. The technique described in Ref [22] was further developed and extended to simulate 3D compressible flows with shocks [25], even if in this case some limitations in shocks interactions modelling still existed. Indeed, in 3D space interacting shocks are represented by several surfaces, anchored to a line (the interaction line) which can move, therefore causing surfaces deformations. Consequently, the generation of shocks surfaces grids and their insertion within the background volume mesh, as described in Ref. [25], are not trivial tasks and require tailored softwares and libraries to be coupled to shock fitting algorithms to identify and handle the shock fronts intersections.

Thanks to these new developments concerning unstructured shock-fitting techniques, several research teams worldwide have applied shock-fitting in conjunction with different discretization techniques including Finite Volume and Finite Element methods. In particular, the modular approach adopted in the shock-fitting code designed by Paciorri and Bonfiglioli [22] allows to plug in different gas-dynamic solvers with a limited coding effort: for instance, a C++ version of the shock-fitting algorithm proposed by these authors has been coupled to the public-domain CFD code COOLFluiD [26, 27]. Moreover, in the past years, the shock-fitting technique in Ref. [23] was coupled also with another gas-dynamic solver, namely *NEO*, developed by Ricchiuto [28, 29], with the aim of improving the capability of the shock-fitting algorithm to deal with unsteady flows [30]. More recent developments of the technique described in Ref. [23] borrow ideas from Shifted Boundary Methods [31], which have led to the so-called "Extrapolated Discontinuity Tracking" method. Specifically, by using this technique [32, 33] the fitted (or tracked) discontinuity carves a mesh-less hole in the computational domain over which it is floating and data transfer between the discontinuity and the boundaries of the hole relies on extrapolation via truncated Taylor series expansions. It allows to apply the shock fitting technique without imposing any constraint on the topology of the mesh and on the data structure of the underlying flow solver and without requiring complex mesh operations other than flagging the cells crossed by the discontinuity. This approach was recently extended also to the structured grids context and gave rise to the *Structured Extrapolated shock-fitting*, that will be described in this Thesis.
At the same time, shock-fitting techniques for unstructured grids alternative to the one proposed in Ref. [22] were developed. The group headed by Jun Liu at Dalian University of Technology, China, developed the Mixed Capturing and Fitting Solver by combining a shock-fitting algorithm with an existing shock-capturing, cell-centered Finite Volume solver using triangular meshes [34, 35, 36, 37]. Shock-fitting ideas made their way also through the Finite Element community. We refer to the Streamline Upwind Petrov-Galerkin technique of [38] and the Discontinuous Galerkin Finite Element methods independently developed by two different research teams: [39, 40, 41, 42] and [43, 44, 45].
However, even if the introduction of unstructured shock fitting techniques allowed to overcome the troubles associated to the structured grid context, this class of methods can not still be considered general as shock capturing one to compute complex shock patterns: besides the aforementioned limitations in simulating 3D shocks interactions, there are still problems for handling changes in the shock topology that may occur in unsteady flows, such as the disappearance of weakening fronts and the appearance of new fronts at boundaries or the collisions of other fronts as summarized by Glimm and co-workers [46].
In this Thesis some new developments of shock-fitting techniques will be presented. Specifically, two parallel paths have been followed: the former covers new advancements of shock-fitting methods in the unstructured grid context, by describing algorithmic improvements of the shock fitting technique proposed in Ref. [23] for computing 2D shocked flows and the recent developments of the shock fitting solver for 3D flows described in Ref. [25]. Specifically, in this case will be emphasised the implementation within the shock fitting algorithm of software tools based on the GRUMMP library, developed some years ago by Zaide and Ollivier-Gooch [47], capa-

ble of not only inserting surfaces describing the shocks fronts within the background volume mesh, but also of handling these multiple interacting shocks surfaces during their space/time evolution. The development of these tools represents therefore a significant forward step for simulating also 3D shocks interactions, that historically were considered a stumbling block for shock-fitting developments in 3D space.
At the same time, this Thesis proposes novel shock fitting techniques to be implemented in conjunction with structured solvers. Because of the aforementioned limitations due to the structured meshes topology, in the past years the development of classical shock fitting techniques on structured grids was progressively abandoned. Nevertheless, the aforementioned merging of the shock-fitting techniques with the shifted boundary methods [31] has broken the strong link between computational grid and fitted shocks and recently it has allowed the development of a new shock-fitting technique for 2D/3D flows that will be presented in this Thesis. As a matter of fact, the development of high order structured solvers based on shock fitting could be an important achievement, since these solvers are the ones that most could take advantage of shock fitting techniques: indeed, these methods allow to provide high-order of accuracy solutions in the overall flow-field with low numerical error, even using coarse grids, thus reducing the computational cost. This is of paramount importance especially for the simulation of turbulent flows through DNS and LES techniques or in the acoustic direct simulation, where gas-dynamic solvers for structured grids are more computationally efficient and accurate than unstructured ones: therefore, these simulations could really gain advantages by shock fitting, since it would avoid the shortcomings due to the shock capture process.
Moreover, it is worth to underline that shock fitting methods allow not only to improve solution quality and accuracy, but also to achieve a better understanding of the physics behind complex shock patterns and interactions. For this reason, in this Thesis some interesting applications fields of the proposed techniques will be also provided, such as the investigation of different shock/boundary layer and shock/shock interactions, to give proof of the understanding capabilities offered by the shock-fitting techniques. As an example, the transonic flow past a NACA0012, where shock pattern exhibits a fishtail configuration will be presented. By modelling the interaction points via shock-fitting, for the first time it was possible to verify that the conditions of the Von Neumann paradox [48] occur in this shock configuration and to compute explicitly a solution based on the Guderley's four-waves model [49].
In view of the advantages stated above, a reappraisal of shock fitting methods is timely and the development of a new class of high accuracy solvers for both structured and unstructured grids based on these techniques certainly could represent a step forward for CFD.

Objectives and Thesis outline

This study focuses on the developments and on the advanced applications of shock-fitting techniques in order to make them an effective option for simulating compressible flows with shocks with respect to the most popular shock capturing methods. With this purpose, the presented Thesis illustrates and discusses the recent advancements of shock-fitting techniques both in structured and unstructured grids

frameworks, which have been carried on during the PhD program.
Specifically, this Thesis is divided into three Parts. Generalities are provided in Part I, which is composed of three Chapters: Chapter 1 presents an examination of the compressible flows governing equations. Afterwards, numerical models for compressible flows are presented in Chapter 2, where special attention is provided for introducing shock capturing and shock fitting approaches. Specifically, the former methods are described by referring to two main categories: finite volume (FV) and fluctuation splitting (FS) schemes, that will find application in the following chapters. Moreover, a brief history about the shock-fitting techniques is described in this Chapter, highlighting also those factors which jeopardized in the past decades the development of this approach. Since the development of shock fitting technique has always been strictly linked to the grids used for computing compressible flows, Chapter 3 gives an overview on the grids used in CFD and mesh generation methods considered in this Thesis.
Part II presents the different shock fitting techniques developed during the PhD program and is based on four Chapters. Unstructured shock-fitting techniques are presented in Chapters 4 and 5, respectively dealing with compressible flows with shocks in 2D and 3D space. This additional split is required since in the former case fitted discontinuities are represented by poly-lines in the computational domain, whereas shocks are modelled using zero thickness, triangulated surfaces for 3D applications. In each Chapter, will be also included some examples of numerical simulations performed using both shock capturing and shock fitting methods, in order to compare the two different modelling approaches regarding solution quality, accuracy and grid convergence properties.
A similar organization is proposed to present the new developments in the structured grids framework. Chapter 6 examines a new shock-fitting technique for 2D structured grids, able to overcome the limitations of the classical shock-fitting schemes in the structured context. This novel method is extended also for 3D computations in Chapter 7, where for the first time shock fitting is applied in full fitted mode for simulating a 3D shock/shock interaction.
Applications of shock fitting techniques are presented in Part III. Specifically, in Chapter 8 shock/boundary layer interactions computed for the first time by using the shock fitting technique for unstructured grids described in Chapter 4 are presented: in particular three different test-cases regarding laminar and turbulent flows are analyzed and discussed. Chap. 9 provides instead an example of how shock-fitting can be also considered as a tool to investigate complex shock patterns and shock interactions topology, referring specifically to a transonic flow past a NACA0012 featuring a fishtail configuration, where conditions for the Von Neumann paradox occur.
Finally, in Chap. 10 conclusions and future perspective are summarized.

Part I

Generalities

Chapter 1

Mathematical models

Contents

1.1 Governing equations

In this Chapter the mathematical models used in this Thesis will be described. A particular relevance will be given to the conservation laws and the jump relations, which describe the thermodynamic and kinetic variables behaviour across a discontinuity. For details, please refer to Ref. [50]

1.1.1 Conservations laws and Navier-Stokes equations

Conservation of mass Let consider a fixed volume $\mathcal{V}$ with boundary $\mathcal{A}$ and outward normal $\mathbf{n}$, as shown in Fig. 1.1. The volume contains fluid that at time t and position $\mathbf{x}$ has density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{u}(\mathbf{x}, t)$: the integral form of the conservation of mass states that the rate of change of the mass of fluid in $\mathcal{V}$ equals the mass flux through $\mathcal{A}$. In particular, the mass of fluid per unit time leaving $\mathcal{V}$ through an element of surface area dA is $\rho\mathbf{u} \cdot dA$, whereas the mass of fluid in an element of volume dV is ρdV. Therefore, the integral form of the mass conservation law is Eq.1.1:

$$\frac{d}{dt}\int_{\mathcal{V}} \rho dV = -\int_{\mathcal{A}} \rho\mathbf{u} \cdot \mathbf{n}\ dA. \tag{1.1}$$

Where the negative sign is due to the outward normal, and hence the integral measures the outward flow of fluid.

Conservation of momentum Similarly, the integral form of the equation of momentum conservation asserts that the rate of change of the momentum of the

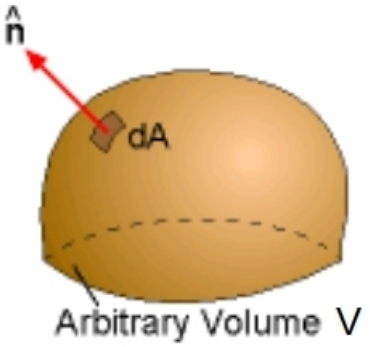


Figure 1.1. Arbitrary control volume $\mathcal{V}$

fluid in $\mathcal{V}$, plus the momentum flux on $\mathcal{A}$, equals the total body force $\mathbf{f}$ exerted on the volume $\mathcal{V}$, plus the total surface force $\mathbf{T}$ exerted on the surface $\mathcal{A}$

$$\frac{d}{dt}\int_{\mathcal{V}} \rho \mathbf{u} dV + \int_{\mathcal{A}} \rho \mathbf{u}(\mathbf{u}\cdot\mathbf{n})\, dA = \int_{\mathcal{V}} \rho \mathbf{f} dV + \int_{\mathcal{A}} \mathbf{T} dA. \tag{1.2}$$

where the dependency of $\mathbf{T}$ on $\mathbf{n}$ is expressed by means of the stress tensor $\mathbb{T}$, so that: $\mathbf{T} = \mathbb{T}\cdot\mathbf{n}$. It is worth to notice, that $\mathbb{T}$ takes into account both the internal stresses due to the fluid pressure p in addition to the stress due to viscous forces, modelled by the viscous stress tensor τ.

$$\mathbb{T} = -p\mathbb{I} + \tau \tag{1.3}$$

By adopting this notation, we obtain the integral form of the momentum conservation law:

$$\frac{d}{dt}\int_{\mathcal{V}} \rho \mathbf{u} dV + \int_{\mathcal{A}} \rho \mathbf{u}(\mathbf{u}\cdot\mathbf{n})\, dA = \int_{\mathcal{V}} \rho \mathbf{f} dV - \int_{\mathcal{A}} p\cdot\mathbf{n} dA + \int_{\mathcal{A}} \tau\cdot\mathbf{n} dA. \tag{1.4}$$

Conservation of energy The equation of energy conservation asserts that the rate of change of total energy E in $\mathcal{V}$, plus energy flux on $\mathcal{A}$, equals the work of the body force in $\mathcal{V}$, plus work done by the surface forces over $\mathcal{A}$, plus the heat flux on $\mathcal{A}$. In particular, the specific total energy E is the sum of internal energy e and the kinetic energy $\frac{1}{2}u^2$, thus $E = e + \frac{1}{2}u^2$. The integral form of the conservation of energy is:

$$\frac{d}{dt}\int_{\mathcal{V}} \rho E dV + \int_{\mathcal{A}} \rho E(\mathbf{u}\cdot\mathbf{n})\, dA = \int_{\mathcal{V}} \rho \mathbf{u}\cdot\mathbf{f} dV - \int_{\mathcal{A}} \mathbf{u}\cdot(-p\mathbb{I}+\tau)\mathbf{n} dA - \int_{\mathcal{A}} \mathbf{q} dA. \tag{1.5}$$

The set of integral conservation equations Eqs. 1.1, 1.4, 1.5 represents the Navier-Stokes (NS) equations, which describe the motion of viscous fluids from a macroscopic point of view. It is worth to notice that, when dealing with viscous turbulent flows, as referred in Sec.8, one approach to account for the effects of turbulence is to time-average the unsteady Navier-Stokes (NS) equations on a time-scale larger than the turbulence time-scale. Such averaging results in a new set of equations, the so

called Favre-averaged NS (FANS) equations, which differs from the NS equations for the presence of the Reynolds' stress tensor, representing the effects of turbulence on the averaged flow field. The appearance of this tensor yields a closure problem, which is solved by adopting an algebraic or differential turbulence model. Making use of the Boussinesq's approximation and selecting a differential model, results in the Favre-averaged equations becoming formally identical to the NS equations except for two differences: *i*) the effective viscosity in the momentum and energy equations, and the thermal conductivity in the energy equations is the sum of laminar and turbulent contributions, and *ii*) the addition of one or more turbulence-related differential equations in the system of conservations laws. In the present work, the Spalart-Allmaras [51] (S-A) one-equation model for the turbulent viscosity is considered. Referring to the control volume $\mathcal{V}$, fixed in space and bounded by the control surface A with inward normal $\mathbf{n}$ as shown in Fig. 1.1, the integral form of the mass, momentum and energy conservation laws and the turbulence transport equation take the following form:

$$\int_V \frac{\partial \mathbf{U}}{\partial t}\, dV = \oint_A \mathbf{F}\cdot\mathbf{n}\, dA - \oint_A \mathbf{G}\cdot\mathbf{n}\, dA + \int_V \mathbf{S}\, dV \tag{1.6}$$

where $\mathbf{U}$ is the vector of conserved variables and $\mathbf{F}$ and $\mathbf{G}$ represent the inviscid and viscous fluxes, respectively:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho E \\ \rho\mathbf{u} \\ \tilde{\nu} \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}H \\ \rho(\mathbf{u}\otimes\mathbf{u}) + p\mathbb{I} \\ \tilde{\nu}\mathbf{u} \end{pmatrix}, \quad \mathbf{G} = \frac{1}{\mathrm{Re}} \begin{pmatrix} 0 \\ \mathbf{u}\cdot\tau + q \\ \tau \\ \frac{1}{\sigma}\left[(\nu+\tilde{\nu})\,\nabla\tilde{\nu}\right] \end{pmatrix} \tag{1.7a}$$

and ν is the kinematic viscosity. The source term $\mathbf{S}$ in Eq. (1.6) has a non-zero entry only in the last row corresponding to the turbulence transport equation proposed in Ref. [51]:

$$\mathbf{S}^t = \left(0, 0, \mathbf{0}^t, c_{b1}\left[1-f_{t2}\right]\tilde{S}\tilde{\nu} + \frac{1}{\sigma\,\mathrm{Re}}\left[c_{b2}\left(\nabla\tilde{\nu}\right)^2\right] - \frac{1}{\mathrm{Re}}\left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left[\frac{\tilde{\nu}}{d}\right]^2\right). \tag{1.7b}$$

Where the following nomenclature proposed by the authors in Ref. [51] has been used:

c_{b1}, c_{b2}, c_{w1} empirical constants in the turbulence model;

f_{v2}, f_{t2}, f_w empirical functions in the turbulence model;

κ turbulent kinetic energy;

σ turbulent Prandtl number;

$\tilde{S}$ production term ;

$\mathbf{d}$ distance from the wall.

Specifically, the stress tensor, $\mathbb{T}$, and heat flux vector, q, that appear in the viscous flux $\mathbf{G}$ include both the laminar and turbulent contributions. In particular, the appearance of the Reynolds number:

$$\mathrm{Re} = \frac{|\mathbf{u}_{\mathrm{ref}}|\, L}{\nu\,(T_{\mathrm{ref}})}$$

in Eqs. (1.7) is a consequence of making all variables dimensionless using a suitable choice of reference variables. The turbulent variable $\tilde{\nu}$ that appears in Eqs. (1.7) is a scalar quantity related to the kinematic turbulent eddy viscosity ν_t via a damping function [51]. As stated before, Eqs. (1.7) are the FANS equations, whereas the NS equations are retrieved by removing the source term $\mathbf{S}$, the differential equation associated with the turbulence variable, and setting the effective viscosity and thermal conductivity to their laminar values. Finally, the Euler equations are instead recovered by additionally removing the flux vector $\mathbf{G}$, so that:

$$\int_V \frac{\partial \mathbf{U}}{\partial t}\, dV = \oint_{\partial A} \mathbf{F}\cdot\mathbf{n}\, dA \tag{1.8}$$

The differential form of Eq. 1.8 is obtained by applying the divergence theorem:.

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla\cdot\mathbf{F} = 0 \tag{1.9}$$

1.1.2 Rankine-Hugoniot relations

When a solution is characterized by discontinuities within the domain (*weak solution*), the differential form in Eq. 1.9 can be applied to the discontinuity-free sub-domains delimited by the discontinuities. These regions are connected by the Rankine Hugoniot relations, which describe the conservation of energy, momentum and mass across a discontinuity: [52, 53, 54]

$$w_n\, [\![\mathbf{U}]\!] = [\![\mathbf{F}]\!]\cdot\mathbf{n} \tag{1.10}$$

where $[\![\cdot]\!]$ denotes the jump of a quantity, with $\mathbf{n}$ the normal vector to the discontinuity, and with w_n the speed of the discontinuity along the direction $\mathbf{n}$. The Rankine-Hugoniot relations can be expressed also by introducing the velocity components along the discontinuity tangential and normal directions, resp. $\mathbf{u_t}$ and $u_n = \mathbf{u}\cdot\mathbf{n}$.

$$[\rho u_n] = w_n[\rho] \tag{1.11}$$
$$[p + \rho {u_n}^2] = w_n[\rho u_n] \tag{1.12}$$
$$[\rho u_n \mathbf{u_t}] = w_n[\rho \mathbf{u_t}] \tag{1.13}$$
$$[E u_n + p u_n] = w_n[E] \tag{1.14}$$

In particular, these relations model two different types of discontinuities:

- slip-lines (or slip-streams) with respect to which the (relative) normal flow speed is zero and as a consequence, $u_n - w_n = 0$. Moreover, in this case one finds trivially that there is no jump in the pressure $[p] = 0$ and along the normal direction $[u_n] = 0$.

- shock waves for which the flow goes into the discontinuity. By assuming $u_n - w_n \neq 0$, Eqs. 1.11- 1.14 are expressed by:

$$[\rho u_n] = w_n[\rho] \tag{1.15}$$
$$[p + \rho {u_n}^2] = w_n[\rho u_n] \tag{1.16}$$
$$[\mathbf{u_t}] = 0 \tag{1.17}$$
$$[E u_n + p u_n] = w_n[E] \tag{1.18}$$

It is worth to underline that this treatment is exact if the flow is governed by the Euler equations: when the effects of viscosity are accounted for, the transition between the pre- and post-shock states takes place over a finite, albeit very thin, region of space. However, in laminar high-speed flows, whenever the Reynolds number is sufficiently high, "the shock thickness can be neglected and a sharp discontinuity, satisfying the Rankine-Hugoniot conditions, can be assumed in a flow otherwise satisfying the Navier-Stokes equations"[55]. The first two test-cases examined in Sect. 8.2 fall into this regime and, therefore, the R-H jump conditions (1.10) will be used across all shock-waves, without trying to resolve their internal structure. Finally, when dealing with turbulent flows by solving the FANS equations, as in Sect. 8.2.3, it would be necessary to take into account that: "the classical jump conditions hold for the instantaneous flow and are no longer exact for the mean flow" [56]. Indeed, as shown in [56], the jump relations that are derived from the FANS equations, except for mass conservation, not only involve the jumps of the mean flow quantities, as in Eq. (1.10), but also those of the turbulent fluxes, and these latter require modelling. Although it may be possible to model at least some of these extra-terms even using a 1-equation turbulence model, this is left for future work and the simplified approach used here consists in neglecting the jumps of the turbulent fluxes: indeed, as far as the turbulent variable used in the S-A model is concerned, this Thesis adopt the simplified approach suggested by the model developers [57], which consists in carrying the turbulent viscosity unchanged through the shock. Therefore Eq. (1.10) is used also in the turbulent case by adding following jump relation for the eddy viscosity, which amounts to set the shock-downstream value of the turbulent kinematic viscosity equal to the corresponding shock-upstream value:

$$[\tilde{\nu}] = 0 \tag{1.19}$$

Chapter 2

Numerical models

High speed flows are often characterised by the presence of shock waves, which play an important role that affects the overall flow behaviour: we could mention numerous examples testifying the importance of shock waves and their effects. Among these are: the flow around space vehicles re-entering the atmosphere at hypersonic speeds, the buffeting phenomenon induced by oscillating shock waves on transonic wings and turbo-machinery blades, shock-induced noise and shock-induced boundary-layer separation that occur inside highly over-expanded nozzles. It is therefore not surprising that computing shocks correctly has always been one of the most important issues in the numerical simulation of compressible flows. More in general, this task can be achieved using mainly two different approaches: shock-fitting (S-F) and shock capturing (S-C). The former was developed inside gas-dynamic solvers based on structured grids starting from the 1960s: it consists in identifying and tracking the motion of the discontinuities, which are treated as internal boundaries of zero thickness which bound regions of the flow-field where a continuous solution of the governing governing partial differential equations (PDEs) exists. Until the early 1990s, S-F techniques, used in conjunction with numerical schemes based on the quasi-linear equations, allowed to accurately compute flows with strong discontinuities using the modest computational resources available at that time(see, for instance, Refs. [58, 14, 15, 16]. On the other hand, they required a significant coding effort especially when dealing with shock-shock interactions. Furthermore, in the 1990s the strong growth in computing resources promote the development of modern S-C schemes, which are built upon the integral conservation law form of the governing equations. Historically, S-C methods made their appearance in 1950s [59, 60]: in particular, in 1959 Godunov [61] proposed a monotone scheme based on the assumption that the flow variables within each computing cell are constant and on the solution of the Riemann problem as applied to flux calculations. In later years, this scheme was further developed and gained particularly popularity in the Soviet Union [62], whereas it became widely known in Western countries with the appearance of its high-order extensions, as the one proposed by Van Leer in Ref. [63]. The following years saw the active development of high order shock-capturing methods based on Godunov approach [64, 65], which can be implemented with ease and are able to provide physically meaningful solutions also for complex flow configurations, features particularly appreciated by the CFD community. For

this reason the interest in S-F techniques progressively diminished, since they were considered too complex and not general enough as S-C schemes. Although S-C methods have been the subject of development and innovations for more than 30 years, they are plagued by several numerical problems, especially around shocks. Nevertheless, captured shocks always have a thickness of two or more mesh cells which worsen the solution quality; it is important to underline that this thickness is a product of the numerical method and has nothing to do with the physical thickness of the shock wave. In order to bring the numerical thickness of the captured wave closer to its physical size, mesh refinement in the shock normal direction is needed, which inevitably causes an increase in computational cost. Also, the solution states computed in the cells inside this region are unphysical, and this is believed to be the origin [5] of some numerical artefacts such as the carbuncle phenomenon [6], which is particularly severe when considering hypersonic flows around blunt bodies [6], or the production of spurious perturbations. Moreover, numerical computations obtained by means of S-C schemes are generally characterized by problems concerning the stability, the accuracy and the quality of the captured solutions [3]. The efforts of many researchers [9, 10] have been focussed on counteracting the flaws and shortcomings of the S-C methods, which however have not been completely overcome and probably will never find a definitive cure. S-F techniques, on the contrary, does not suffer from these numerical problems due to the enforcement of the R-H relations to compute discontinuities: for this reason, the developments of S-F techniques could open up to several possibilities for CFD simulations regarding aerospace applications.

2.1 Shock capturing methods

As previously stated, S-C methods are based on the integral form of the governing equations reported in Chap. 1: the computation of all flow types, including those with shocks, is performed using the same discretization of the conservation law form of the governing equations at all grid cells. This implies coding simplicity, since a single numerical scheme is used and the same set of operations is repeated within all control volumes of the mesh, also when dealing with complex flow configurations. For this reason, nowadays S-C schemes enjoy great popularity among the CFD community. In the proposed Thesis, mainly two classes of S-C schemes will be considered and here briefly described: the Finite Volume (FV) and the Fluctuation Splitting (FS) schemes.

2.1.1 Finite Volume schemes

In the finite volume method, the discretized unknowns are referred to the average value in each cell. In order to illustrate the main features of the Finite Volume method, we can consider a 1D system described by the Euler equations (Eq. 1.9). We can now discretize the spatial domain into N computing cells $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ of size $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$, with i = 1,....,N. We also define a control volume as $\mathcal{V} = I_j x[t^n, t^{n+1}]$, as shown in Fig. 2.1. By integrating Eq. 1.9 over the space-time

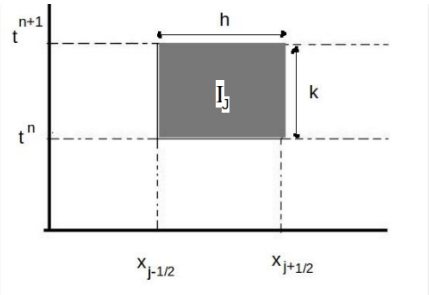


Figure 2.1. Finite volume methods: control volume example for a 1D spatial domain

rectangular domain I_j, we obtain:

$$\int_{t^n}^{t^{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} dx dt = 0 \tag{2.1}$$

That can be rewritten as:

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} [U(x, t^{n+1}) - U(x, t^n)] dx + \int_{t^n}^{t^{n+1}} [F(U_{(x_{j+\frac{1}{2}}, t)}) - F(U_{(x_{j-\frac{1}{2}}, t)})] dt = 0 \tag{2.2}$$

Eq. 2.2 represents the integral form of the Euler equations. At this point we define the average integral values of U as:

$$\bar{U}_j^n = \frac{1}{h} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(x, t^n) dx \tag{2.3}$$

In a similar way, the time average value of the numerical flux is given by Eq. 2.4, where the time step k is defined by $k = t^{n+1} - t^n$

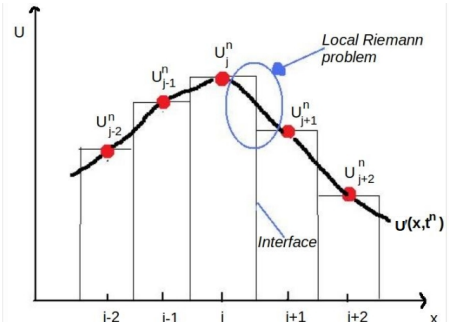
$$\bar{F}_{j\pm\frac{1}{2}}^n = \frac{1}{k} \int_{t^n}^{t^{n+1}} F(U_{(x_{j\pm\frac{1}{2}}, t)}) dt \tag{2.4}$$

According to these new definitions, Eq. 2.2 becomes:

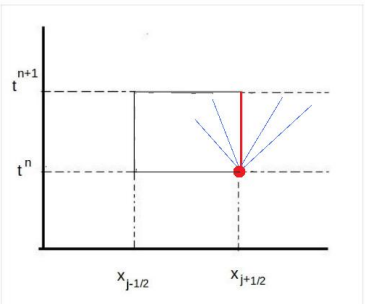
$$\bar{U}_j^{n+1} = \bar{U}_j^n - \frac{k}{h} (\bar{F}_{j+\frac{1}{2}}^n - \bar{F}_{j-\frac{1}{2}}^n) \tag{2.5}$$

It is important to notice that no approximations have been made so far, indeed Eq. 2.5 is not a numerical scheme. At this point, different numerical algorithms can then be devised from Eq. 2.5 according to the method used to calculate the fluxes $F_{j+\frac{1}{2}}^n$ and $F_{j-\frac{1}{2}}^n$ at each interface. Anyway, the numerical flux evaluation in Eq. 2.4 involves the computation of the integral of $F_{j\pm\frac{1}{2}}$ over the interfaces $x_{j\pm\frac{1}{2}}$, so that the knowledge of the solution $U_{(x_{j\pm\frac{1}{2}}, t)}$ is also required for each $t \in [t^n, t^{n+1}]$. Godunov methods compute these fluxes by solving Riemann problems at the interfaces: indeed, at the interface between adjacent cells the state variables $\bar{U}_j$ manifest a jump as highlighted in Fig. 2.2a, thus generating a sequence of local Riemann problems. In particular, left ($\bar{U}_L$) and right ($\bar{U}_R$) states for each Riemann problem can be

determined by defining the solution at the interface in each cell: the procedure that allows to approximate the solution at the cell interfaces by knowing the solution cell average value is referred as *reconstruction*. As an example, Godunov first order method, which was introduce in 1959 [61], is based on a piecewise constant reconstruction, so that the approximate solution at time t^n is constant in each cell, as shown in Fig. 2.2a.



(a) Finite volume methods: piecewise reconstruction in Godunov first order method



(b) Finite volume methods: Riemann problem solution at interface $x_{j+\frac{1}{2}}$ and time t^n

Figure 2.2. Finite Volume methods: Godunov scheme

Moreover, since the solution to the Riemann problem is self-similar (i.e. it does not depend on x and t separately but only on $\frac{x-x_{j\pm\frac{1}{2}}}{t-t^n}$), the solution for U at $x_{j\pm\frac{1}{2}}$ is time independent, so that:

$$\bar{U}_{(x_{j\pm\frac{1}{2}},t)} = U_{(x_{j\pm\frac{1}{2}},t^n)}$$

It is no longer true whether the waves originating from the adjacent interfaces reaches the interface. Waves interactions within cell I_j is therefore avoided by imposing the following limitation on the time step k:

$$k \leq \frac{h}{\lambda_{max}} \tag{2.6}$$

where λ_{max} is the maximum wave speed: specifically, Eq. 2.6 represents the so called Courant, Friedrichs, Lewy condition, also known as *CFL condition*.
As a consequence of the Riemann self similarity property, the integral in Eq. 2.4 is drastically simplified:

$$\bar{F}^n_{j\pm\frac{1}{2}} = \bar{F}(U_(x_{j\pm\frac{1}{2}}, t^n)$$

For this reason, the evaluation of the numerical flux at the interfaces requires only the knowledge of the solution at the interfaces. This task can be achieved by using two different methods for solving each local Riemann problem: the former regards the use of the exact Riemann solver, such as the one preposed by Gottlieb and Groth [66], that however requires computationally expensive iterative processes when the Euler governing equations are considered. An alternative is the approximated Riemann solver, as proposed by Roe [67], where the governing equations are locally

linearized and then the associated Riemann problem is solved exactly. Once the boundary fluxes are evaluated, it is possible to advance the solution to the next time step according to Eq. 2.5.
As stated before, the aforementioned Godunov scheme, based on piece wise constant reconstruction, is only first order accurate. The extension to higher order of accuracy is achieved by replacing piecewise constant with piecewise polynomial reconstructions: as an example, Van Leer [63] obtained second order accuracy by using monotonic piecewise linear slopes for cell reconstruction, whereas the the piecewise parabolic method (PPM) of Colella and Woodward [68] provides third-order accuracy. More in general, the major steps of a generic Finite Volume method can be summarized as follows:

1. Reconstruction process for evaluating the solution at cell interfaces
2. Solution of the local Riemann problem at each cell interfaces
3. March the solution in time from time t to time $t + kt$

It is worth to provide some details regarding how higher order of accuracy Godunov schemes are developed. As previously stated, for instance second order of accuracy is achieved when using a piecewise linear reconstruction, such as:

$$U_{(x,t^n)} = \bar{U}_j^n + s_j^n(x - x_j) \text{ where } x \in [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \tag{2.7}$$

where s_j^n represents the slope of the piecewise linear reconstruction in I_j. Since by using the reconstruction process in Eq. 2.7, the solution at the interface is time dependent, the Riemann problem left and right states are no more uniform and therefore in this case a *generalized* Riemann problem must be solved. Multiple choices are available for defining s_j^n, so that the formula used to compute this parameter characterizes different numerical schemes: in particular, the choice $s_j^n = 0$ provides the previously described Godunov first order scheme. Possible alternatives can be based on:

forward differences formula: $s_j^n = \frac{\bar{U}_{j+1}^n - \bar{U}_j^n}{h}$

backward differences formula: $s_j^n = \frac{\bar{U}_j^n - \bar{U}_{j-1}^n}{h}$

centred differences formula: $s_j^n = \frac{\bar{U}_{j+1}^n - \bar{U}_{j-1}^n}{2h}$

However, whether the solution features a discontinuity, the aforementioned expressions can lead to oscillations around the discontinuity. For this reasons, in general s_j^n is defined by using functions, namely slope limiter functions [7], able to prevent spurious oscillations at discontinuities by avoiding cells crossed by shocks for evaluating s_j^n. One of the simplest choices is given by the use of the minmod function as in Eq. 2.8, which is able to switch between a forward and a backward formula in order to employ the slope having a smaller magnitude:

$$s_j^n = \frac{1}{h} minmod(\bar{U}_{j+1}^n - \bar{U}_j^n, \bar{U}_j^n - \bar{U}_{j-1}^n) \tag{2.8}$$

where:

$$minmod(x,y) = \begin{cases} x & \text{if } |x| \leq |y| \text{ for } xy \geq 0 \\ y & \text{if } |x| > |y| \text{ for } xy > 0 \\ 0 & \text{for } xy < 0 \end{cases} \tag{2.9}$$

Specifically, the choice based on the minmod slope limiter has been implemented within the *AFFS* gas-dynamic solver [69, 70] for 2D/3D flows on structured grids, an in-house shock capturing code that will be widely applied for computing different high speed shocked flows configurations in Chapter 6 and 7.
Anyway, whether the two slopes in Eq. 2.8 have opposite signs, as occur while crossing discontinuities or sharp gradients, s_j^n is set equal to zero, so that the order of the spatial accuracy is locally reduced to first order. In order to overcome this drawback, different alternatives for constructing uniformly high-order accurate shock-capturing schemes have been developed. As an example, the family of essentially non oscillatory (ENO) schemes [64] employs polynomials with adaptive stencils in order to achieve high-order reconstruction: specifically, since different stencils are available for reconstructing the solution at the interfaces, ENO schemes select that one which avoids as much as possible interpolation across discontinuities, thus minimizing spurious oscillations [71]. Specifically, the selection criterion is to compare the local smoothness of the reconstruction polynomials, measured by divided differences: as a consequence, ENO schemes rely on several logical operations, which in general imply an higher computational cost. An alternative is provided by the class of the weighted essentially-non-oscillatory (WENO) schemes, which were first introduced by Liu [65] and further generalized and improved by Jiang and Shu [72]. The basic idea of WENO approximation is the following: instead of using only one of the candidate stencils to form the reconstruction, as using the ENO approach, one uses a convex combination of all of them to form a large stencil [73]. Specifically, the available stencils are combined using weights selected to achieve maximum formal order of accuracy in smooth regions, and assign (nearly) zero weight to reconstructions on stencils crossed by discontinuities, as described in Ref. [71]. WENO schemes are in general characterized by high accuracy and robustness, even if the large number floating point operations required for the evaluation of the smoothness measurements makes WENO schemes computationally expensive.

Finite volume methods previously described can be extended with ease to higher dimensionality. As an example, let us consider a 2D case. For the sake of simplicity, we can consider a Cartesian computational grid characterized by uniform spacing Δx and Δy, as proposed in Ref. [74]. Elementary computational cells of rectangular shape centred on $x_i = i\Delta x$ and $y_j = j\Delta y$ are referred as $C_{i,j} = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}})$, an example of which is provided by Fig. 2.3.

Therefore, in this case the integral of the Euler equations (Eq. 1.9) over the domain $C_{i,j} \times (t^n, t^{n+1})$ is given by Eq. 2.10, where the numerical flux F components

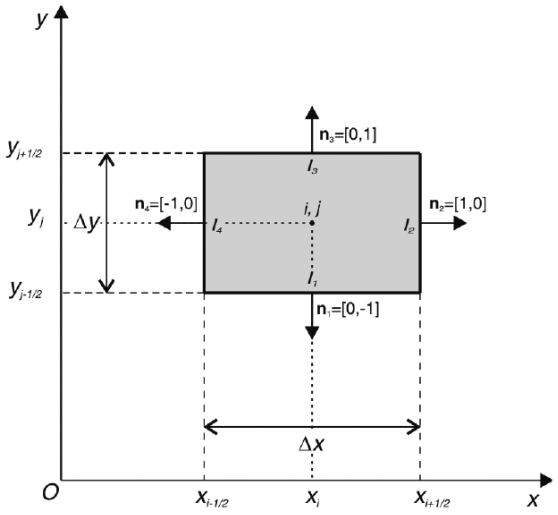


Figure 2.3. Cartesian control volume $C_{i,j}$

f and g along the Cartesian axis have been used, so that $\nabla F = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}$:

$$\int_{C_{i,j}} U(x,y,t^{n+1}) - U(x,y,t^n)dxdy = \int_{t^n}^{t^{n+1}} (\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(U_{(x_{i-\frac{1}{2}},y,t)}) - f(U_{(x_{i+\frac{1}{2}},y,t)})\ dy + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(U_{(x,y_{j-\frac{1}{2}},t)}) - g(U_{(x,y_{j+\frac{1}{2}},t)})\ dx)\ dt \tag{2.10}$$

Let us introduce the cell average values of $U_{(x,y,t)}$, where $A_{C_{i,j}} = \Delta x \Delta y$ is the area of the elementary cell $C_{i,j}$. In particular, at time level t^n, it amounts to say:

$$\bar{U}^n_{i,j} = \frac{1}{A_{C_{i,j}}} \int_{C_{i,j}} U(x,y,t^n)dxdy \tag{2.11}$$

And the time average values for the numerical flux components, which are defined as follows:

$$\bar{F}^n_{i\pm\frac{1}{2},j} = \frac{1}{k\Delta y} \int_{t^n}^{t^{n+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(U_{(x_{i\pm\frac{1}{2}},y,t)})dt$$

$$\bar{G}^n_{i,j\pm\frac{1}{2}} = \frac{1}{k\Delta x} \int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(U_{(x,y_{j\pm\frac{1}{2}},t)})dt$$

By doing so, the fully discrete scheme for the updating of the average vector of conserved variables at the new time level t^{n+1} becomes:

$$\bar{U}^{n+1}_{i,j} = \bar{U}^n_{i,j} - \frac{k}{\Delta x}(\bar{F}^n_{i+\frac{1}{2},j} - \bar{F}^n_{i-\frac{1}{2},j}) - \frac{k}{\Delta y}(\bar{G}^n_{i,j+\frac{1}{2}} - \bar{G}^n_{i,j-\frac{1}{2}}) \tag{2.12}$$

As stated for the 1D case, in order to advance the solution in time, the computation of the integrals of $\bar{F}^n_{i\pm\frac{1}{2},j}$ and $\bar{G}^n_{i,j\pm\frac{1}{2}}$ is required. Also in this case, it can be

achieved by solving the Riemann problems occurring at each corner of cell $C_{i,j}$, as described in Ref. [74, 75].

2.1.2 Fluctuation Splitting schemes

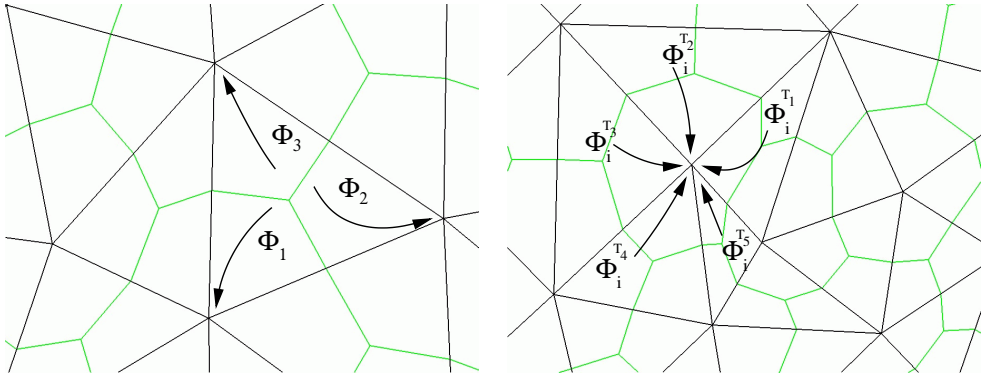
A different class of numerical methods considers Residual Distribution (RD) or Fluctuation Splitting (FS) schemes [76, 77]: in its most classical formulation, the RD approach provides discrete approximations of the compressible Euler equations on simplicial grids, starting from values of the dependent variables stored at the vertices of the mesh. The second order variant of the methods exploits a classical continuous piece-wise linear finite element interpolation of the unknowns. The main idea behind these methods is summarized in Fig. 2.4: discrete equations for the steady state values of the unknowns are assembled over each element. The steady solution can be obtained as the limit of the (pseudo-)time iteration:

$$|C_i|\frac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\Delta t}+\sum_{T\ni i}\Phi_i^T=\text{B.C.s} \tag{2.13}$$

with Δt a (pseudo-)time step, and $|C_i|$ usually taken as the area of the median dual obtained joining the gravity centres of the cells surrounding i to the mid-points of the edges running into the node (the polygonal-shaped boundaries in Fig. 2.4). The nodal fluctuations (or nodal residuals) verify in each triangle, T the consistency constraint:

$$\sum_{j\in T}\Phi_j^T=\Phi^T:=\int_T\nabla\cdot F(\mathbf{U}^h)\,\mathrm{d}V \tag{2.14}$$

In Eq. (2.13) the right hand side represents the boundary condition terms, which are left out of the discussion. Several design criteria exist for (2.13)-(2.14), impacting both the practical evaluation of the element fluctuation (or residual) Φ^T, as well as the definition of the nodal residuals (2.13). Among the most important we mention:



(a) The flux balance (or cell residual) of triangle T is scattered among its vertices.

(b) Grid-point i gathers the fractions of cell residuals from the surrounding triangles.

Figure 2.4. Residual distribution concept.

Conservation The consistency condition (2.14) implies conservation if the evaluation of the element residual is such that the equality:

$$\Phi^T = \oint_{\partial T} F\left(\mathbf{U}^h\right) \cdot \mathbf{n}\, \mathrm{d}S \tag{2.15}$$

holds for some continuous approximation of the normal flux. The condition above can be shown to be essential for solutions of scheme (2.13)-(2.14) to converge to weak solutions [78, 79]. In practice there exist two approaches to fulfill this condition. The first is to introduce a local conservative linearization of the equations. For the Euler equations with perfect gas equation of state, this can be achieved by means of a multi-dimensional extension of Roe's linearization [67, 80]. This is the approach used in `EulFS` [81], which also uses the conservative linearization to evaluate all quantities necessary for the splitting. In alternative, one can compute the element residual by approximating directly the contour integral on ∂T by some quadrature formula, and with some assumption on the (continuous) polynomial interpolation used to perform the flux evaluation in quadrature points [82, 83]. This is the formulation implemented in `NEO` [84, 85], which makes use of a set of physical variables (pressure, density, velocity) for both the interpolation, and the averaging of the flux Jacobians where necessary.

Accuracy The consistency/truncation error condition originally introduced in [86] (see also [77, 76]) provides a necessary condition for second order of accuracy. Schemes verifying this condition at steady state can be generally cast in a FEM like form:

$$\Phi_i^T = \int_\Omega \boldsymbol{\omega}_i \nabla \cdot F\Big(\mathbf{U}^h\Big)\, \mathrm{d}V \qquad\qquad \sum_{j \in T} \boldsymbol{\omega}_j|_T = \boldsymbol{I} \tag{2.16}$$

where $\boldsymbol{\omega}_i$ is a bounded test function, and the second relation equivalent to the consistency condition (2.14).

Monotonicity The theory of positive coefficient schemes has been used systematically to study these schemes in the scalar case, proving a discrete maximum principle equivalent to the preservation in time of the initial bounds on the discrete solution (cf. [77] and references therein). Formal matrix generalizations of this condition have been considered in several works [87, 88]. Linear monotone schemes are only first order accurate, and several different nonlinear approaches exist to combine second (or higher) order of accuracy with a monotonicity preserving property.

Upwinding and multidimensional upwinding A bias of the residual distribution in the direction of propagation of the information is often present. This notion is clear in one space dimension [67], and a geometrical generalization to the multi-dimensional case can be provided for scalar problems on linear finite elements [89, 90, 91], see also [77]. For multidimensional systems, this notion becomes less clear, unless one focuses on steady two-dimensional supersonic flows for which exact decompositions in scalar waves exist [81, 92].

The aforementioned FS approach lies behind the gas-dynamic solvers `EulFS` and `NEO`, which will be widely used in conjunction with the unstructured S-F technique proposed in Sec. 4 and that are briefly described below:

EulFS

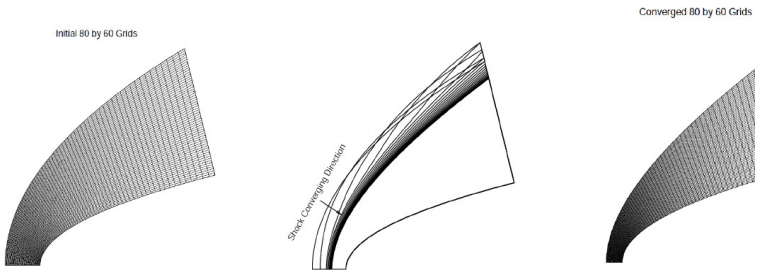
The EULFS code is an in-house, unstructured CFD solver that has been developed over the last 20 years by Aldo Bonfiglioli (see [81] for a detailed description). This tool is able to work in both 2D and 3D space and stores the solution at the vertices of triangles, in 2D, and tetrahedra, in 3D. In both cases, the solution is assumed to vary linearly and continuously in space. The inviscid cell fluctuation ϕ^e is evaluated over each triangular/tetrahedral element e by means of a conservative linearization [93, 67] based on the parameter vector $Z = (\sqrt{\rho}, \sqrt{\rho}H, \sqrt{\rho}u, \sqrt{\rho}v)^T$ and scattered to the element vertices using signals ϕ_i^e. Within a cell e, the signals have to sum up to the net flux for conservation, $\sum_{i\in e} \phi_i^e = \phi^e$. The different Fluctuation Splitting schemes proposed in the literature differ by the way cell residuals are split into signals. The schemes that may used are several and based on the different features that they present. Starting from a monotonicity preserving but first-order-accurate scheme, named N scheme, and a second-order accurate, which may lead to unphysical oscillations in the neighbourhood of a captured discontinuity, called LDA scheme; EULFS is also provided with a non-linear scheme, which captures discontinuities monotically and preserves second order of accuracy in smooth regions, that blends the linear N and LDA schemes in such a way that the former is activated only where discontinuities occur. The blend is based on a smoothness sensor that makes the new scheme non-linear.

NEO

The NEO code has been developed by Mario Ricchiuto [28] and has been mainly used to study time-dependent problems. It is based on a different formulation aimed at designing explicit Runge-Kutta residual distribution schemes exhaustively described in [28]. This explicit approach is based on three main ingredients: first recast the RD discretization as a stabilized Galerkin scheme, then use a shifted time discretization in the stabilization operator, and lastly apply high order mass lumping on the Galerkin component of the discretization. In particular, this approach turned out to be very useful in simulating unsteady flows by coupling NEO with the presented shock-fitting technique. Some of the results obtained from this work have been published in the chapter [24], whose contributions showed very promising results in the development of the unsteady shock-fitting version. The computations shown in the chapter show the possibility of using the aforementioned linear first-(monotone) and second-order N and LDA schemes, which are based on a multidimensional upwind distribution of the cell residual, their non-linear blend (B scheme), and two non-upwind methods. In particular, the explicit predictor-corrector formulation of the second order linear Streamline-Upwind (SU) method proposed in [28] and the nonlinear blended central (Bc) discretization obtained when blending the SU method with a limited Lax-Friedrich's distribution.

2.2 Shock-fitting techniques

S-F techniques, as conceived by Gino Moretti and his collaborators starting in the 1960s, have been originally developed within the structured-grid framework with two different variants blossoming over the years: "boundary" [13] and "floating" [14, 19] S-F. In the first approach, the shock is made to coincide with one of the boundaries of the computational domain so that the enforcement of the R-H jump relations amounts to prescribe boundary conditions. When the shock moves, it causes a deformation of the entire block grid, which can be remeshed by modifying the nodal distribution along the coordinate line in the shock normal direction, as highlighted in Fig. 2.5. It



(a) Initial grid. (b) The shock and the grid move. (c) Final grid.

Figure 2.5. Boundary S-F: grid deformation during shock motion.

is evident that the boundary S-F technique is quite simple to implement in existing gas-dynamic solvers, since it only requires the coding of a boundary condition that accounts for the presence of the fitted shock and no other change is necessary inside the computational kernel of the solver. For this reason, the application of boundary S-F techniques was particularly appreciated to compute high speed flows around blunt bodies [17, 18, 94]. Over the years, some researchers applied boundary S-F techniques also to different flow configurations [95, 96]. However, boundary S-F has clearly strong topological limitations inherent in the use of structured multi-block grids, especially when shocks interactions occur within the flow-field: moreover, it is not possible to model those discontinuities which arise when the solution advances in time. The floating S-F version [14, 19] was developed to deal with more complex flow configurations, including shock/shock and shock/boundary layer [20] interactions. In this approach, discontinuities can freely move over a background structured mesh and the shock-fronts are described by their intersections with the grid-lines, as illustrated in Fig. 2.6. Specific procedures are required to move shock-points along the grid according to computed shock velocity, detect the formation of new shocks [97, 98] and the disappearance of shocks as they weaken. Even though the floating S-F approach allows to overcome most of the limitations incurred by the boundary S-F, it is very complex to code as it requires to modify the computational stencil in the neighbourhood of the floating discontinuities to avoid taking differences across the discontinuity, see e.g.[99]. Front-tracking (F-T) methods [100, 101, 102], as conceived by James Glimm and collaborators, have some similarities with floating S-F. The F-T approach of Klingenberg et al. [103] involved solving the flow equations

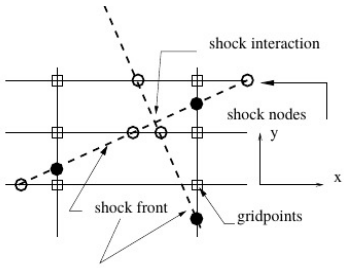


Figure 2.6. Floating S-F: shock fronts and interaction point description.

on both sides of the discontinuity never using the data on the front, and use ghost states at intersections to obtain finite difference expressions to evolve the closest points. This approach has many common points with classical well known cut cell approaches, and introduces small cells which may degrade the accuracy and stability (CFL condition) of the method (cf. e.g. [104] and references therein). Similar approaches had also been followed by other researchers [105, 106, 107]. All these techniques require quite extensive local geometrical operations to side step the small cut cell issue. In 2009 Paciorri and Bonfiglioli [22] introduced a new, unstructured-grid S-F technique that takes advantage of the geometrical flexibility offered by unstructured meshes and combines features of both the boundary and floating S-F algorithms developed in the structured-grid setting. This new unstructured-grid S-F technique, which allows to overcome at least some of the difficulties incurred by the traditional S-F/F-T methods, has been further improved in recent years by making it capable of dealing with multiple interacting discontinuities [23], unsteady 2-D flows [30, 24] and shock/boundary-layer interactions [108], as will be widely discuss in Sec. 8. Most of the aforementioned developments have also been included in an open-source platform [109] that will be presented in Chapter 4, whereas the extension of this technique to 3D flows will be subject of Chapter 5. In recent years the spreading of the unstructured S-F methods sparked a renewed interest towards S-F/F-T techniques, which have been applied by several research teams worldwide in conjunction with different discretization techniques including Finite Volume (FV) and Finite Element (FE) methods. The group headed by Jun Liu at Dalian University of Technology, China, developed the Mixed Capturing and Fitting Solver (MCFS) by combining a shock-fitting algorithm with an existing shock-capturing, cell-centered FV solver using triangular meshes [34, 35, 36, 37]. S-F/F-T ideas made their way also through the FE community, as proposed by the Streamline Upwind Petrov-Galerkin (SUPG) technique of [38] and the Discontinuous Galerkin (DG) FE methods independently developed by two different research teams: [39, 40, 41, 42] and [43, 44, 45]. All three aforementioned techniques simultaneously solve for the location of the grid-points, in addition to the flow variables, to constrain certain edges of the tessellation to be aligned with the discontinuities. The use of shape-functions that are continuous across the element interfaces, which is the case with SUPG, or discontinuous, such as in DG, has implications on how discontinuities are fitted. In the SUPG-FE algorithm of [38] the discontinuities are internal boundaries of zero thickness: by doing so, a finite jump in the dependent variables takes place

while crossing the discontinuity. On the other hand, numerical methods that employ a data representation which is discontinuous across the cell interfaces, which is the case with DG-FEM [110, 111, 112], but also with cell-centred FV methods, allow to fit discontinuities as a collection of edges of the mesh, without introducing internal boundaries. Regardless of the chosen approach, in all the aforementioned unstructured-grid S-F/F-T techniques the mesh has to be modified, either locally or globally and in a (pseudo) time-dependent fashion, to follow the motion of the discontinuities. This may not always be an easy task from the meshing view-point, especially when dealing with complicated shock interactions and/or 3D flows, in which case it may also be computationally costly. For these reasons, Ciallella et al. have recently developed the extrapolated DIscontinuity Tracking (eDIT) technique, which combines the unstructured S-F technique developed by [22] with the shifted boundary method by [31]. In eDIT [32, 33] the fitted (or tracked) discontinuity carves a mesh-less hole in the computational domain over which it is floating and data transfer between the discontinuity and the boundaries of the hole relies on extrapolation via truncated Taylor series expansions. This approach allows to retain high-order convergence properties without imposing any constraint on the topology of the mesh, and on the data structure of the underlying flow solver and without requiring complex mesh operations other than flagging the cells crossed by the discontinuity. This technique was extended also to structured grids and gave rise to the *Structured Extrapolated S-F (SESF)*, that will be described in detail in Chapters 6 and 7, and was recently applied to compute 2D/3D high speed flows with shocks [113, 114]: as a matter of fact, one of the main advantages of this new technique is the capability to compute shock waves and the adjacent flow areas without re-meshing around the shock, breaking the strong link between grid topology and shock position that prevented the development of S-F techniques in the structured grid framework. Indeed, for boundary S-F, the fitted shocks must always be part of the computational grid, limiting the application of S-F only to simple flows and structured solvers. Moreover, in the floating SF schemes is required a change of stencil in the gas-dynamic solver to avoid that cells crossed by the shock waves are involved in the computation. On the other hand, it should be noticed that for implementing SESF almost no extra coding is needed in the gas-dynamic solver: indeed, if the solver is able to treat cell blanking, it can be considered as a black-box by the S-F algorithm, which only takes care of giving an input solution and the blanked elements that must be excluded from the computation. This algorithmic flexibility greatly simplifies the coding effort that characterized the floating S-F, and allows to handle flows with complex shock patterns. Therefore, the structured extrapolated S-F technique can be considered as an important achievement since structured solvers are the ones that most can take advantage of S-F techniques. Indeed, these solvers are still very much in use today for simulations of turbulent flows (through DNS and LES) and acoustics since they are more computationally efficient and accurate than the unstructured ones: therefore, these simulations could really benefit from a more accurate and efficient shock modelling free of all those problems associated to the capturing process.

Thank to these recent developments, some of which will be presented in the following Chapters, S-F techniques are bidding to be an effective option regarding the S-C methods in simulating flows with shocks.

Chapter 3

CFD grids and mesh generation

Contents

A preliminary step for performing CFD numerical simulations is the discretization of the domain of interest: in particular, we define a mesh as a subdivision of the geometric domain into smaller simple shapes, such as quadrilaterals and triangles in 2D space or hexahedra and tetrahedra for 3D applications. Anyway, when complex geometries are considered, domain decomposition process is not trivial, and several challenges must be faced to achieve this goal: as noticed by Baker in Ref. [115], due to these implications, mesh generation became progressively a discipline in its own, borrowing ideas from several fields, as mathematics and computer science. Indeed, over the last 50 years many studies have been led regarding the development of new methods for generating grids [116, 117, 118, 119] as well as the effects of the computational mesh on CFD solutions quality and accuracy [115, 120, 121] and nowadays it is still a very active area of research and development. This Chapter aims to provide a summary regarding the types of grids commonly used for CFD computations, paying also particular attention on the most popular mesh generation techniques. At the very least, this Chapter provides a theoretical framework that allows to evaluate how mesh refinement is related to CFD solution error, in order to further investigate the relationship between solution accuracy and grid properties.

3.1 CFD meshes and grid generation methods

A basic classification for the grids used for CFD applications is based on the topology of the elements that fill the computational domain and the mesh data organization.

According to this sorting, mainly two categories can be defined: *structured* and *unstructured* meshes. The former are composed by cells which are well ordered, and a simple scheme (e.g., I,J,K indices) can be used to label elements and identify neighbouring cells. An alternative is represented by unstructured grids, where the elements structure is arbitrary and therefore cells connectivity must be defined and stored. In the following subsections these two classes of grids are widely discussed, specifically referring to their advantages and disadvantages, and an overview about the grids generation methods commonly use is also provided.

3.1.1 Structured meshes

Structured grids are characterized by elements (quadrilaterals for a surface grid, hexahedra for a volume grid) are arranged in an IxJ (or $IxJxK$ in 3D space) array where I, J, and K refer to the grid's dimensions, as shown in Fig. 3.1. Because of their structure, it was possible to easily identify the neighbouring cells adjacent to each element by increasing or diminishing the indices values: consequently, structured CFD solvers are characterized by less memory usage, since they have no need to store the neighbour connectivity information. This feature was well suited to the modest computing resources available at dawn of CFD, therefore originally numerical simulations were performed on structured grids.

Since then, several algorithms have been developed for generating structured grids,

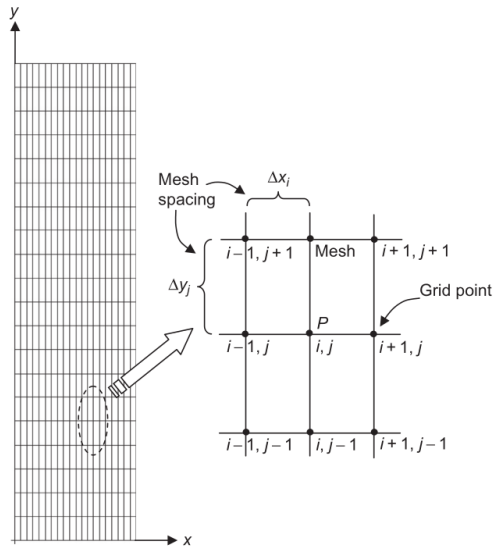


Figure 3.1. Example of structured grid on a 2D rectangular domain. Re-printed from Ref. [122], with permission

even if they can be cast mainly into two categories: algebraic methods and differential equations techniques, based on numerically solving PDEs [117]. Algebraic methods [123, 124] are particularly appreciated for the ease of implementation: this class of methods is based on the use of mathematical interpolation functions to interpolate between some known grid points (usually on the boundaries, but they can also belong to a specific line which one desires that the grid lines pass), in order to generate the grid in between these points. The interpolation method or

technique depend on the algebraic method that is taken into account, but the idea behind is more or less the same. By doing so, users have explicit control of the physical grid shape and spacing, and the implementation of these methods is in general simple [123]. Anyway, the major drawback of algebraic grid generation is the determination of the interpolation functions which control a grid.
Like algebraic methods, differential equation methods are also used to generate meshes by solving PDEs, that can be cast into three types: elliptic, parabolic, and hyperbolic. Elliptic schemes became popular in the 1970s and 1980s [125, 126] and are based on either the Laplace equations or the Poisson equations, obtained by adding functions that control the spacing of the coordinate lines. By doing so, elliptic methods generate very smooth grids, while requiring as input prescribed boundary points: anyway, the choice of the control functions may become not trivial when dealing with complex boundaries of the computational domain and moreover, the solution of the elliptic system can consume a large amount of computational time. A different class of methods is given by the hyperbolic grid generation schemes [127, 118], generally applied to problems with open domains: in this case, grids are generated by marching from an initial curve or surface in the normal direction, respectively for 2D and 3D space, as illustrated in Fig. 3.2. Additional grid nodes are placed by numerically solving hyperbolic governing equations, that are derived from orthogonality relations and cell area/volume constraints and using both a marching step size and final distance set the user. When exact control of all the mesh boundaries is not needed, less work is required using hyperbolic methods since only one boundary has to be prescribed instead of four, as for elliptic methods: this advantage is particularly appreciated when 3D grids are considered, because only one surface is required as input. Finally, parabolic schemes proposed in Ref. [128] try

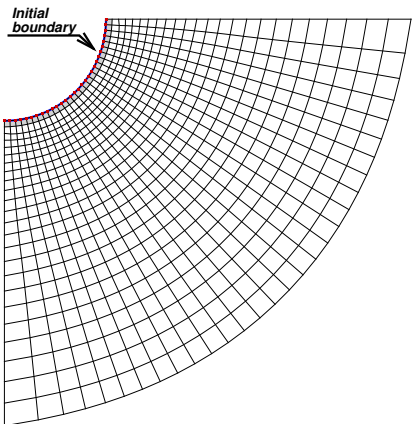


Figure 3.2. Example of 2D hyperbolic structured grid.

to combine the smoothing properties of elliptic methods with the hyperbolic schemes efficiency: they are similar to the hyperbolic ones, since internal grid points are placed by advancing from the initial curve or surface, but they allow outer boundary grid points distribution to be prescribed.

3.1.2 Multiblock grids and overset methods

Regardless the mesh generation method used, it must be noticed that the generation of structured grids is often difficult or impossible when concerning complex geometries. However, this drawback can be alleviate by using a multi-block strategy: it consists in breaking up the domain into several smaller blocks, where separate meshes are generated using the aforementioned methods and then assembled and connected, as illustrated in Fig. 3.3. Specifically, the different grids could be connected by

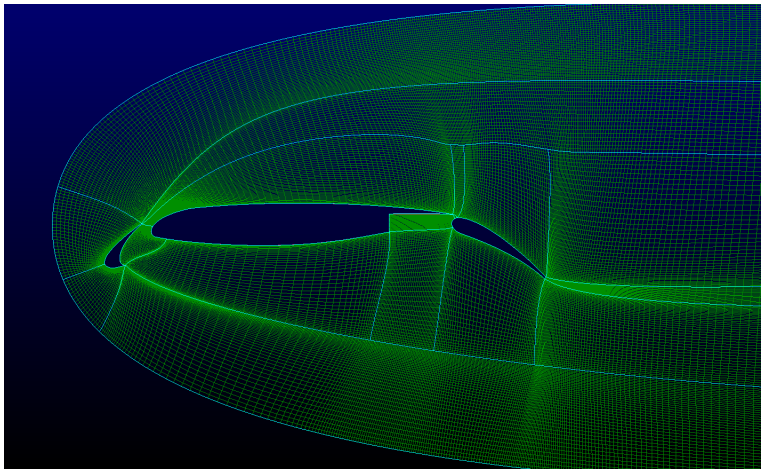


Figure 3.3. Multi-block 2D structured grid around an airfoil.

requiring continuity among the mesh lines across the block interfaces (*composite multiblock*) or not, as done for *patched multiblock* [115]. Both the strategies are proposed in Fig. 3.4. Overset methods can be considered an extreme variant of the latter multiblock strategy, since in this case a complex geometry is decomposed into different geometrically simple overlapping grids, which exchange information via interpolation processes: specifically, multiblock structured grids with overlapping blocks are also referred in literature as Chimera grids. The advantages of employing such grids are that complex domains are treated with ease, due to the lack of any constraint at the block boundaries, which greatly simplifies the individual blocks generation. Furthermore, overset grids methodology can also be used to simulate the unsteady flows around moving rigid bodies in relative motion [129], such as a store releasing from a wing [130, 131], the space launchers stage separation [132] or an airfoil-flap configuration, which is described in Fig. 3.5. This flexibility has a price to pay: as stated before, it is required to transfer information between overlapping meshes, so that interpolation formulae must be determined to provide an accurate data transfer.

Anyway, blocks definition within the computational domain is not an automated process, which becomes not trivial when complex geometries are taken into account: blocks position and their discretizations at the interfaces must be manually defined even if many grid generation softwares have been developed for simplifying this task, so that the final multiblock decomposition is strictly dependent on the user capability and experience in grids generation field.

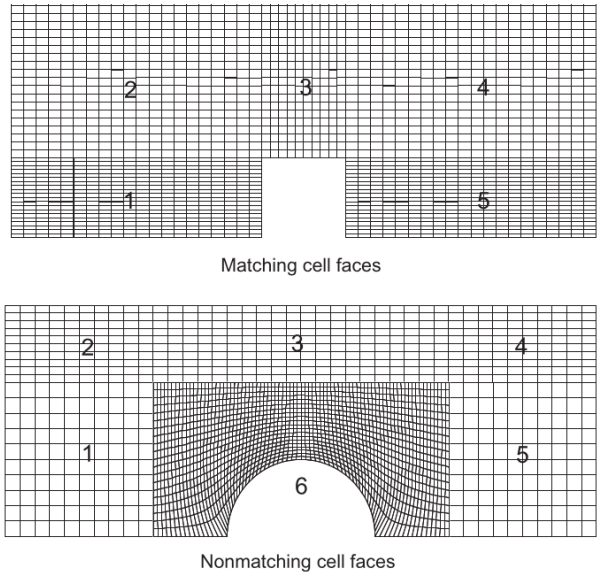


Figure 3.4. Multiblock 2D structured mesh with matching and nonmatching cell faces.

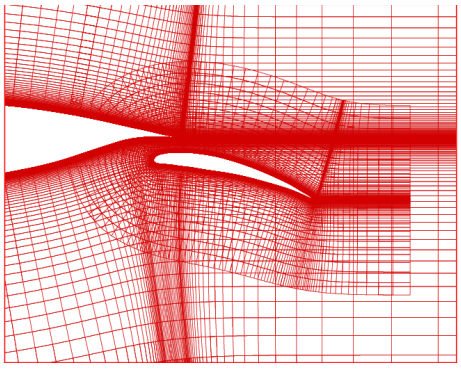


Figure 3.5. Example of overset grids around an airfoil with flap.

3.1.3 Unstructured meshes

Last decades were marked by the widespread of unstructured grids, so that nowadays the majority of the CFD commercial solvers are based on these meshes, aided also by the progressive increase of computational resources. Indeed, even if unstructured grids require higher memory usage with respect to structured grids to store connectivity information, the use of this kind of grids offers several advantages: just to name a few, the ability of handling arbitrary complex geometries as well as the possibility of locally adapting the grid to follow the flow features. The most typical shape of an unstructured element is a triangle in 2D or a tetrahedron in 3D space. Nonetheless, any other shape including quadrilateral or hexahedral cells is also possible. Several algorithms can be taken into account for generating unstructured grids using triangles/tetrahedra. As an example, in the advancing-front techniques, some of which are proposed in Ref. [133, 134], an unstructured mesh is generated by adding individual elements one at a time to an existing front of generated elements. Generation of a 2D grid starts with a discretization of the geometry boundaries as a

set of edges, which form the initial front that must be advanced out into the field until the entire domain has been covered with elements. In order to clarify how these algorithm works, let us consider the 2D domain shown Fig. 3.6a, where the front is depicted using a dashed line. As described in Ref. [119], a particular edge of this front is selected, and a new triangle is formed with this edge as its base by joining the two ends of the current edge either to an existing point on the front or to newly created point, namely points P and Q in Fig. 3.6 b. Specifically, new vertices are placed according different criteria, such as the quality of the resulting element and the desired mesh spacing imposed by the user: most important, it must be ensure that the new triangle created using this node do not cross other edges of the triangulation. Then, the current edge is removed from the front, since it is now obscured by the new triangle. Similarly, the remaining two edges of the new triangle are either assigned to the front or removed from the front, depending on their visibility, as illustrated in Fig. 3.6.

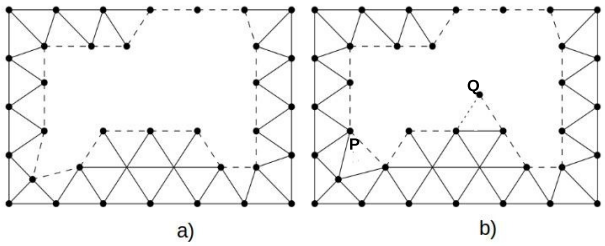


Figure 3.6. Advancing front methods: initial front (a) and the new front obtained by connecting vertices P and Q (b). Re-printed from Ref. [119], with permission

Advancing front methods can be also applied to 3D grid generation: in this case, a surface grid is first constructed by creating a 2D triangular mesh on the surface boundaries of the domain. This triangular mesh forms the initial front, which is then advanced into the flow field by placing new points ahead of the front and forming tetrahedral cells. The required intersection checking now involves triangular front faces rather than edges as in the 2D space. Advancing front methods generally provide smooth high quality grids, but difficulties may arise in regions where multiple fronts must be merged, so that the procedure is not trivial.
Delaunay methods are an alternative to advance front techniques, particularly appreciated for their algorithmic simplicity: a 2D Delaunay triangulation (DT) for given a set of discrete points is a triangulation such that no point falls within the circumcircle of any triangle of the tessellation, as shown in Fig. 3.7. When 3D grids are considered, Delaunay criterion still applies by considering the circumsphere (circumscribing sphere) associated with a tetrahedron instead of circumcircles. In general, several algorithms have been developed in order to compute Delaunay triangulations. In general, Delaunay algorithm starts with an initial set of points, as the one illustrated in Fig. 3.8a: triangles are created by imposing the empty circumcircles condition, which generate elements as the one in Fig. 3.8b, so that the final tessellation is similar to the one in Fig. 3.8d.
There exists a Delaunay triangulation for any set of points in 2D and in particular, it is always unique as long as no four nodes in the point set are co-circular. Anyway, it must be notice that in 3D space it is not always possible to construct a Delaunay

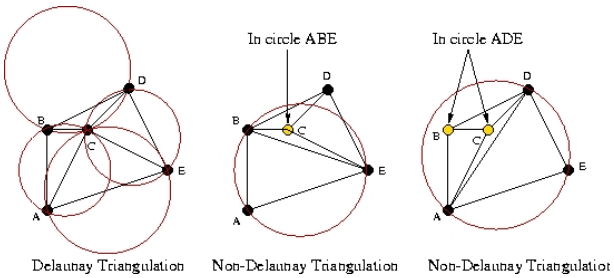


Figure 3.7. Empty circumcircle property for Delaunay triangulations

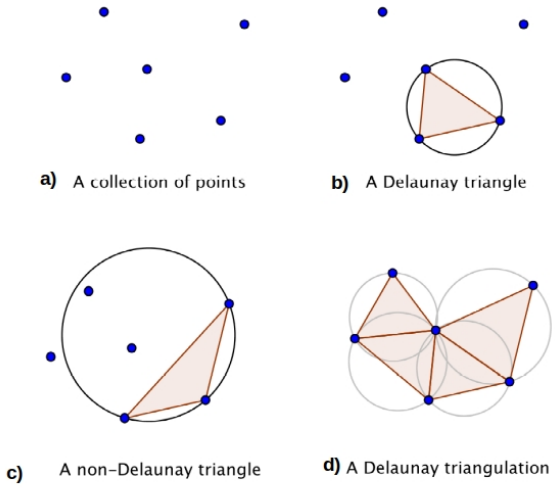


Figure 3.8. Example of Delaunay triangulation in 2D space.

tetrahelization for a given set points, so that the addition of further vertices, referred as Steiner points, is required for generating volume meshes which conform to the Delaunay criterion. The main disadvantages of Delaunay techniques is related to inability to guarantee boundary integrity if not convex domain are involved, as reported in Ref. [119]: in order to overcome this drawback, Chew [135] proposed a Constrained Delaunay Triangulation (CDT), that relaxes the "empty-circle" condition in order to preserve the boundaries. In particular, CDT constraints are mainly two: some pre-specified edges must be included in the triangulation, thus respecting the given boundary discretization, and the final tessellation must be as close as possible to a Delaunay triangulation. An example of how this algorithm work is provided in Fig. 3.9: red edges in Fig. 3.9a represent boundary edges that must be constrained to be a part of the final triangulation, illustrated in Fig. 3.9b. It is worth to notice that even if domain boundaries are maintained, some of the generated triangles may not verify the empty circumcircle property. This requirement is satisfied by adding further vertices to the initial point set, as proposed by the Conforming Constrained Delaunay triangulation (CCDT) algorithms [136]: in this case, a constrained Delaunay tessellation is performed as previously described, moreover edges are split into smaller edges by inserting additional points until the Delaunay empty circumcircles condition is verified in the overall domain. Alterna-

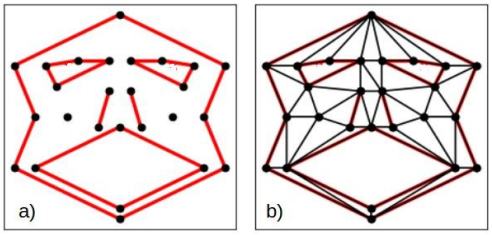


Figure 3.9. Example of Constrained Delaunay triangulation for a 2D domain. Re-printed from Ref. [137], with permission

tively to Delaunay methods, quadtree/octree techinques generate an unstructured mesh through a recursive subdivision of the domain down to a prescribed resolution. The vertices of the resulting quadtree or octree structure are used as grid points, and the tree quadrants or octants are divided up into triangular or tetrahedral elements, respectively when considering 2D or 3D space. It must be underlined that the vertices of the quadtree/octree cells intersecting the boundary edges/surfaces must somehow be displaced or wrapped in order to coincide with the domain boundaries. These methods are relatively simple and inexpensive and are capable of producing good quality mesh in interior regions of the domain. Anyway, one drawback of the method is that it has a tendency of generating an irregular cell distribution near boundaries, as reported in Ref. [122].

3.1.4 Adaptive grids

An adaptive mesh is a grid network that automatically or dynamically clusters the grid points in regions where steep gradients exist in the computed solution flow field. The idea behind adaptive methods is to use an error estimate to assess the quality of a numerical solution and systematically modify the mesh to improve the solution quality by clustering grid points in regions where large gradients occur. Historically, adaptive techniques were developed in the structured grids framework [138], but nowadays are also applied to unstructured meshes, since here cells and grid points do not follow any kind of global structure, so that it is also possible to add or remove mesh cells and points with ease [139]. Specifically, adaptive gridding techniques can be generally subdivided into two main classes: adaptive mesh redistribution and adaptive mesh refinement. The former methods continuously reposition a fixed number of points so that they improve the resolution in particular locations of the flow domain featuring large gradients. However, with a fixed number of points, a better local resolution is achieved at the expense of depreciating the resolution in other regions, where the grid becomes coarser: in many cases the depreciation is quite minor because the other regions often have an overabundance of points that would not significantly alter the accuracy of the simulation. Problems linked to use a fixed number of points then only occurs when there are not enough grid nodes to well resolve both the local phenomena and the other regions. For this reason, sometimes is preferred to adopt a different way to proceed, such as the one used by the adaptive mesh refinement (AMR) techniques [140, 138]: in these methods, further nodes are added to regions where higher accuracy of the solution

is desired. Nodes can also be removed from locations where the solution is smooth and therefore requires less grid resolution. Anyway, due to node addition/removal, the grid connectivity changes from one adaptation cycle to another, so that data interpolation is required to transfer the solution among the consecutive grids. These methods are particularly attractive for their flexibility, especially when applied on unstructured grids, as in Ref. [139]. In particular, AMR techniques modify the grid nodes number according two main approaches: grid subdivision and grid remeshing. When using grid subdivision, new nodes are added to the edges of the cells identified for refinement: as a result, these elements are divided into smaller cells, so that the data transfer between the "old" and the "new" grid is also simplified. This method is particularly known to be efficient and fast, but several subdivision can also cause grid distortion, especially when tetrahedral grids are taken into account [139]. An alternative approach is proposed by the adaptive grid refinement methods, which feature high flexibility and generate good quality grids during the adaptation process: nevertheless, grid generation in these methods is in general computational expensive and the cost of solution interpolations between subsequent meshes is higher with respect to the grid subdivision approach.
As state before, grid adaptation methods are based on the knowledge of the CFD solution as it evolves in time, in order to identify the regions which require better resolution and dynamically modify the computational mesh. Specifically, adaptive gridding techniques are used in conjunction with an error estimate, which determines refinement required, so that regions with higher error will end up accumulating more mesh cells, as described in Ref. [141]. Indeed, grid adaptation is an iterative process: as the solution marches in time, the numerical error is estimated and the grid points are moved to adapt to the evolution of the large flow gradients, which are in general linked to the presence of discontinuities such as shocks, until steady state is reached. Adaptive grids are considered particularly attractive by the S-C methods [142, 143], which have been presented in Chap. 2.1, especially when high speed flows featuring shocks are computed. Indeed, particular attention was posed in the previous Chapter to describe the S-C numerical problems linked to the shock capture process, such as the discontinuity finite thickness, the spurious disturbances and the accuracy order degradation in the shock downstream regions [144, 4]. A way to alleviate these drawbacks is the use of S-C methods in conjunction with an a posteriori mesh adaption technique [11], since by reducing the size of cells in the vicinity of the discontinuity, the shock width is reduced and brought closer to its physical value. Anyway, not only the cluster of grid nodes near the shock is necessary, but also the alignment of the mesh elements with the shock wave is important, otherwise the captured shock results to be smeared over some mesh intervals and thus the S-C scheme accuracy order deteriorate to first order, regardless of the design accuracy of the discretization used. In order to provide an example of the benefits that can be obtained by coupling S-C methods and adaptive grids methods, let us consider Fig. 3.10 and Fig. 3.11, which show the numerical computation of a transonic flow past a NACA0012 ($M_\infty = 0.95$) using the EulFS in-house code introduced in Chap. 2.1 on two different unstructured grids, where the one in Fig. 3.11 is generated using an AMR technique. Specifically, this test case is characterized by a complex shock pattern, referred as fishtail configuration (please refer to Chap. 9 for details). The availability of both the solutions allow to compare the Mach flow-fields computed

using the same S-C solver on the two meshes: the one in Fig. 3.11 is obtained using an AMR technique, so that it is locally refined and, most importantly, aligned to the shocks, as highlighted by Fig. 3.12, whereas the grid in Fig. 3.10 is generated with a priori refinement around the airfoil, so that no dynamically adaption has been performed during the computation. By evaluating the computed Mach flow-fields, it can be seen that simulation performed on the adapted grid exhibits a higher quality with respect to Fig. 3.10, also using a third of the grid nodes and cells.

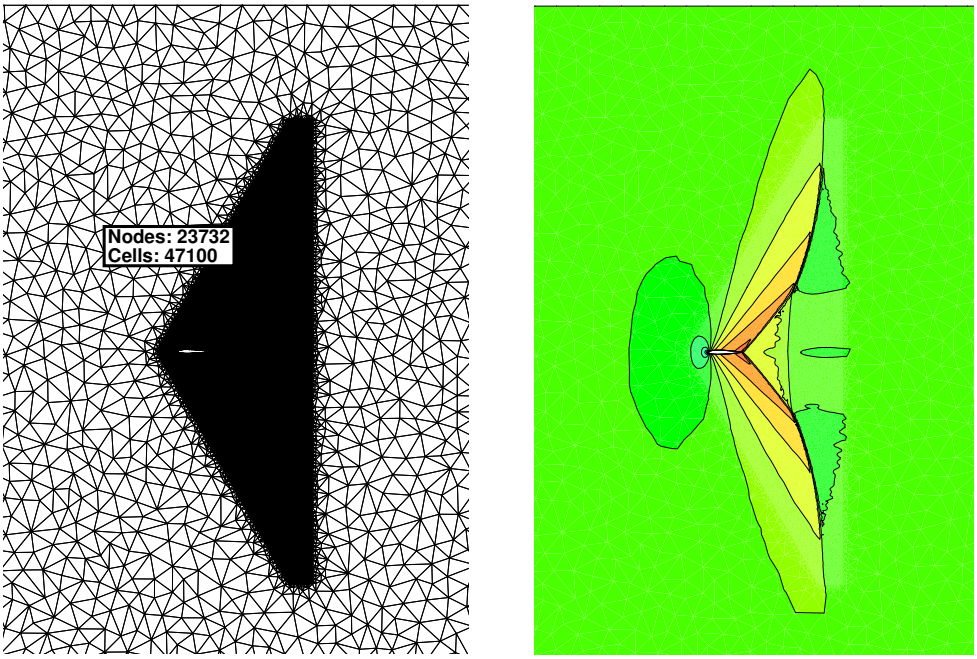


Figure 3.10. Transonic flow past a NACA0012: computational grid (left) and Mach flow-field computed using a S-C solver (right)

3.1.5 Not body-fitted grids: immersed boundary methods analysis

The necessity to compute flows past complicated shapes is one of the main reasons to prefer unstructured grids with respect to structured meshes. Nevertheless, the increasing complexity in the geometry may deteriorate the grid quality also when unstructured grids are considered, so that numerical solutions computed on these meshes can be affected by high numerical errors. Because of this drawback, in the past years different approaches were developed both regarding grid generation and CFD algorithms in order to simplify the computation of these flows: among the different solutions that have been proposed by the CFD community, the class of immersed boundary methods (IBMs) deserves particular attention. These techniques were first introduced by Peskin to simulate the blood flow in the heart [145]: the main innovation of this study was that the numerical computation was carried out on a Cartesian grid, which did not conform to the geometry of the heart, and as a consequence, a novel procedure was formulated for imposing the effect of the heart immersed boundary (IB) on the flow. Specifically, these methods mark a turning

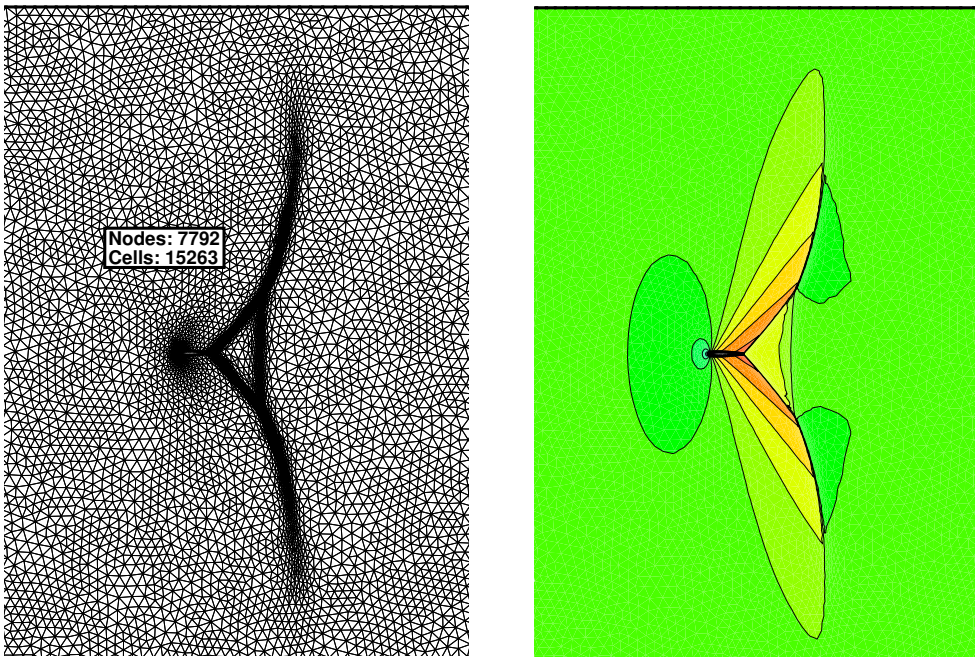


Figure 3.11. Transonic flow past a NACA0012: computational grid generated by AMR (left) and Mach flow-field computed using a S-C solver (right)

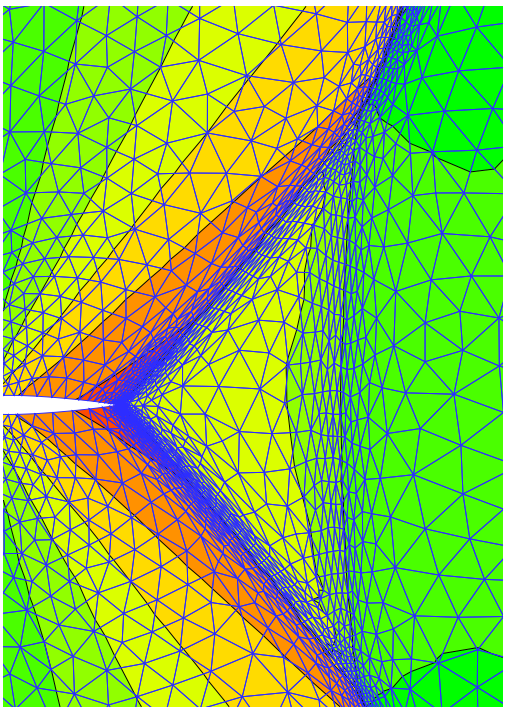


Figure 3.12. Transonic flow past a NACA0012: detail of the computational grid generated by AMR around the interaction region

point for computational techniques: indeed, in the former techniques, first the body geometry is specified and then a structured/unstructured mesh, which include the body boundaries, is generated. Immersed boundary methods use instead a fixed grid, in general a Cartesian mesh, where body boundaries are not constrained to be a part of the mesh, as illustrated in Fig. 3.13, so that the solid boundary cuts some computational cells. Moreover, due to the fact that the grid does not conform to the body, the boundary conditions imposition is achieved using some modifications of the governing equations near to the boundary [141, 146]: specifically, the way these modifications are implemented is the key factor in developing the different immersed boundary algorithms. Let us provide a brief overview of the main IB approaches,

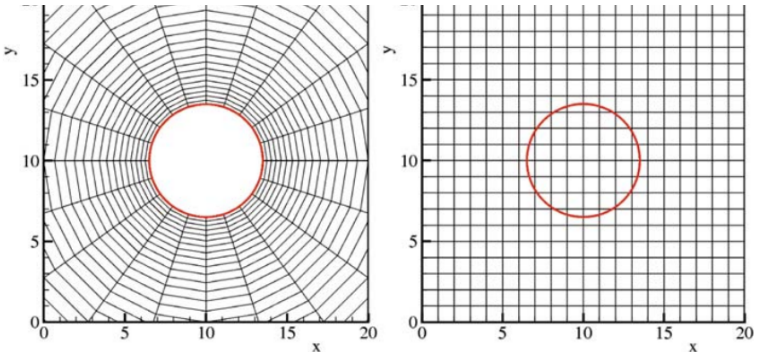


Figure 3.13. Difference between a body-fitted (left) non body fitted (right) 2D grids for blunt body, boundaries of which are depicted in red.

underlining also the advantages and drawbacks of these promising methods. As described in Ref. [146], in order to reproduce the effect of the body boundaries source terms, also referred as forcing functions, can be added in the governing equations: many methods have been designed depending on the forcing function expression. However, mainly two alternatives exist, namely the *"continuous"* and the *"discrete"* forcing approach [146]. The former is based on including the forcing function into the continuous governing equations, which are then discretized and applied to the entire domain: for this reason, it can be considered independent of the spatial discretization. In the latter, the forcing is introduced after the equations are discretized on the grid, without regard to the IB: in particular, the discretization is adjusted to account for the boundary presence only in the cells near of the IB. As a consequence, the discrete forcing procedure is strictly linked to the discretization approach so that it is not as practical as the continuous forcing approach, but it enables a sharp representation of the IB, which is especially desirable for high Reynolds number flows [146].
As previously stated, one of the advantages of using an immersed boundary method is that grid generation is much easier, because the grid must not conform to the body boundaries. Also, it implies that immersed boundary method can handle moving bodies, since their boundaries are able to move on a background fixed mesh as the solution advances in time. Because of these interesting advantages, the popularity of immersed boundary methods is increasing especially during the last decades. Indeed, these techniques has been widely used to simulate different kinds of problems, such as fluid/structure interactions fluid [147, 148] as well as compressible flows past moving solid boundaries [149, 150]. Anyway, these methods are not free

from drawbacks: the main disadvantage of IBMs is that imposing of the boundary conditions is not straightforward compared to the traditional methods. In addition, it is worth to underline that the alignment between the grid lines and the body surface in body-fitted grids allows better control of the grid resolution in the vicinity of the body, which enhances the solution quality.

3.2 Mesh refinement analysis to assess CFD solution accuracy

CFD has been one of the major drivers for developing mesh generation algorithms to discretize the computational domain. As a consequence, a need arose to investigate the close link between the accuracy of the CFD solution computed on a given mesh and the grid properties, with particular interest on the influence of mesh refinement on the behaviour of the solution numerical error. For this purpose, grid convergence analysis are the most common and reliable techniques for the quantification of numerical uncertainty and play an important role also for CFD codes validation, as reported in Ref. [144, 151, 115]: specifically, since in this Thesis it is of relevant interest to evaluate the solution accuracy provided by the recent developed S-F techniques with respect to the one obtained using S-C methods, it is worth to describe in detail how these analysis will be carried out in the following Chapters. According to Ref. [115, 144], a grid convergence study is based on the availability of solutions computed on at least two meshes, that can be generally obtained by coarsening the finest one or by subsequently refining the finest grid. As an example, let us consider the 2D computational domain Ω in Fig. 3.14, which was discretized by using three nested meshes $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ containing N_1, N_2, N_3 nodes respectively, such that $N_3 > N_2 > N_1$, which have been obtained using the former approach. By doing so, the grid nodes of the coarsest mesh also belong to the finest grid, as highlighted by Fig. 3.14. Specifically, the availability of the solution on different

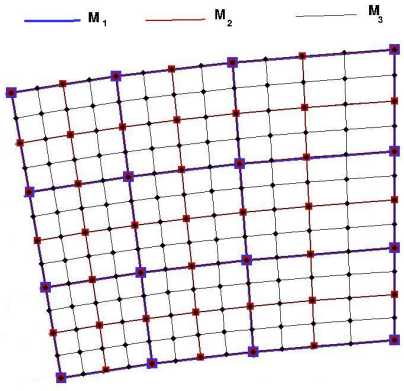


Figure 3.14. Example three nested structured grid levels on a 2D domain

grid levels is useful to perform a posteriori analysis that allows to verify whether the effective order of convergence of the numerical method is equal to the design

(formal) order n, which thus must be known a priori. More in general, whether there is a good agreement of the observed order of accuracy $\tilde{n}$ with the formal order of the numerical method, grids are saying to be in the asymptotic range of convergence. When the solution is computed on three nested grids, as those in Fig. 3.14, the observed order of accuracy $\tilde{n}$ can be calculated as follows [152, 144]:

$$\tilde{n} = \frac{\log R^{-1}}{\log r} \tag{3.1}$$

where:

$$R = \frac{u_2 - u_3}{u_1 - u_2} \tag{3.2}$$

is the so called convergence monitor. In particular, u_i in Eq. 3.2 denotes the flow variable used for evaluating the convergence order computed on the i-th grid level, where the subscript $i = 1, 2, 3$ refers to the coarse, intermediate and fine grid levels, respectively. Moreover, r is the grid refinement ratio, which needs to be constant in order to apply Eq. 3.1: for the computational domain in Fig. 3.14, r is taken equal to $r = 2$. The analysis can be further expanded by computing the Richardson Extrapolated (RE) solution $\tilde{u}_{RE}$ and the Grid Convergence Index (GCI) [144, 152, 153], which represents a bandwidth error of the numerical solution. In particular, the GCI is defined as the difference between the numerical u and the extrapolated solution $\tilde{u}_{RE}$ computed assuming $n = \tilde{n}$, multiplied for a safety factor F_s. For instance, referring to the *i-th* grid level:

$$\mathrm{GCI}_i = F_s \left| u_i - \tilde{u}_{RE} \right| \tag{3.3}$$

where the Richardson extrapolated value $\tilde{u}_{RE}$ is provided by:

$$\tilde{u}_{RE} = u_3 - \frac{u_2 - u_3}{r^{\tilde{n}} - 1}. \tag{3.4}$$

The factor of safety is recommended to be F_s=3 for comparisons of two grids and F_s=1.25 for comparisons over three or more grids [144, 154].
Furthermore, whenever an exact solution to the set of governing PDEs is available, grid convergence can be also evaluated using two grid levels. In this case, we define the local discretization error ϵ_i on the i-th grid as the difference between the solution u_i and the exact solution u_{exact}:

$$\epsilon_i(\mathbf{x}) = u_i(\mathbf{x}) - u_{exact}(\mathbf{x}) \tag{3.5}$$

with subscript $i = 1, 2$. Consequently, the observed order of accuracy can be computed as in Eq. 3.1, where the convergence monitor R is given by [21]:

$$R = \frac{u_2 - u_0}{u_1 - u_0} = \frac{\epsilon_2}{\epsilon_1} \tag{3.6}$$

As previously stated, the aforementioned analysis requires the knowledge of the exact solution, which is not always available. When the exact solution is not known, the discretization error of a numerical solution as well as the convergence properties of a numerical scheme can be evaluated using the Generalized Richardson Extrapolation

(RE) techniques. In particular, it amounts to compute an approximation of $u_{exact}(\mathbf{x})$ by using the design order of convergence n, which is assumed to be known a priori, as follows:

$$\tilde{u}_{exact} = u_2 - \frac{u_1 - u_2}{r^n - 1} \tag{3.7}$$

Eq. 3.7 can be used to provide an estimate of the exact solution in those situations in which an exact solution of the governing PDEs is not known, as described in Ref. [21]. It must be noticed that this approximation is only available pointwise within the N_1 nodes of the coarser grid level $\mathcal{M}_1$, where it allows to evaluate also the L_q norm of the discretization error according to:

$$L_q(\epsilon_1) = (\frac{\sum_{j=1}^{N} |u_1 - \tilde{u}_{exact}|^q}{N_1})^{\frac{1}{q}} \tag{3.8}$$

Part II

Recent developments of Shock Fitting techniques

Chapter 4

UnDiFi-2D

Contents

In Chap. 2 an overview of the numerical models commonly used for computing compressible flows featuring shock waves was presented. In particular, attention was drawn on S-C methods and S-F techniques by placing emphasis on the drawbacks and advantages of both the approaches.
Despite CFD computations are nowadays performed using predominantly S-C solvers, in last decades the interest in S-F methods has been renewed, since in 2009 Paciorri and Bonfiglioli [22] developed an unstructured S-F technique, by taking advantage of the gradual shift that has taken place in the CFD community from structured towards unstructured grids. Indeed, this novel method inherits features of Moretti's techniques for structured solvers while exploiting the geometrical flexibility offered by the use of unstructured triangular meshes. The algorithm proposed in Ref. [22] was capable of simulating steady, 2D flows featuring only one fitted shock-wave: multiple shocks could be handled using a hybrid approach, whereby only one shock was fitted, whereas all other discontinuities and their mutual interactions were captured. Later developments [23] made the algorithm capable of fitting also contact discontinuities, shock/shock and shock/wall interactions. An order of accuracy analysis of the steady, 2D scheme has been conducted in [21] highlighting one of the major strength points of S-F: indeed, by using S-F techniques it is possible to preserve the design order of the spatial discretization scheme within the entire shock downstream region, whereas S-C schemes drops to first order. In recent years, the unstructured S-F algorithm of Paciorri and Bonfiglioli was further developed to compute also time-accurate simulations of 2D unsteady flows, as reported in [30, 24]. These encouraging results gave rise to the idea of combining these contributions from different research teams

in order to release a S-F solver for 2D unstructured grids, which is publicly accessible to further promote collaborations and developments. This new open-source CFD code, referred hereafter as *UnDiFi-2D*, is presented in this Chapter, where will be also provided a detailed description the algorithmic steps of the code and few results focusing on the test-cases that better represent all the implemented features.

4.1 *UnDiFi-2D*: a general description

UnDiFi-2D is an open source (free software) *Un*structured-grid, *Di*scontinuity *Fi*tting code [109]. The aim of *UnDiFi-2D* is to model gas-dynamic discontinuities in 2D flows as if they were true discontinuities of null thickness that bound regions of the flow-field where a smooth solution to the governing PDEs exists. *UnDiFi-2D* therefore needs to be coupled with an unstructured CFD solver that is used to discretize the PDEs within the smooth regions of the flow-field: specifically, two different codes, `EulFS` and `NEO`, described in Sec. 2.1.2, have been included in the current distribution. Specifically, the present and future versions of the code `UnFiDi-2D` can be downloaded in the open-source repository made available at `https://github.com/UnDiFi/UnDiFi-2D`. As shown in Fig. 4.1, the main directory `UnFiDi-2D` contains the following sub-directories:

1. `bin`: where all the executables are installed;
2. `lib`: where various libraries and their source codes are stored;
3. `doc`: which contains the documentation;
4. `source`: where the source files of the *UnDiFi-2D* code are stored;
5. `source_utils`: contains
   - the source files of various I/O format converters;
   - the `Triangle` [155, 156] mesh-generator;
6. `tests`: contains various test-cases, some of which are described in Sect.4.3;
7. `tools`: contains the source code of the `f77split` and `f90split` programs [157];
8. `EulFS.3.7`: where the source files of the `EulFS` [81, 158] gas-dynamic solver are stored.
9. `NEO`: where the source files of the `NEO` solver [159, 28, 29] are stored;

In addition to these sub-directories, the main directory contains the script (`compile_all.sh`) which compiles all the software packages. Full description on how to download, compile and run the code can be found at the documentation page `https://github.com/UnDiFi/UnDiFi-2D`.

The directory tree highlights the fact that the software is made up of three key components:

1. the shock-fitting module *UnDiFi-2D* which handles the motion of the discontinuities and their interactions (if any), but also drives the other two components, i.e.

2. the gas-dynamic solver, either `EulFS` or `NEO`, which is used to discretize the governing PDEs in smooth regions of the flow-field;

3. the meshing software `Triangle`, which is used to locally re-mesh while the discontinuities move throughout the computational domain.

Communication among the driver *UnDiFi-2D*, the gas-dynamic solver and the meshing software is handled using format converters (to be found in the `source_utils` folder) that rely on disk I/O. This programming approach is certainly not the best from the standpoint of computational efficiency, one of the reasons being that one has to switch among the different data-structures used by the three different modules. However, this approach is very convenient, since it allows us to use off-the-shelf gas-dynamic solvers and mesh generation tools that are treated as black boxes and can be replaced by similar ones only by changing the format converters, with a modest coding effort.

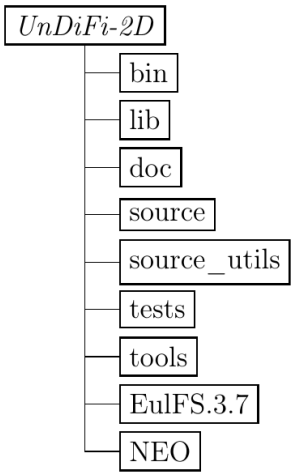


Figure 4.1. *UnDiFi-2D* project directory tree.

4.2 *UnDiFi-2D*: algorithmic features

In this Section, a brief description of the data storage and key algorithmic ingredients of the S-F algorithm is presented. Regardless of whether steady or time-accurate simulations are performed, the approach is inherently time-dependent, because both the solution and the grid change with time, due to the displacement of the fitted discontinuities. When a steady solution exists, the shock speed will asymptotically vanish and the tessellation of the flow domain will not any longer change. As far as data storage is concerned, the dependent variables and grid velocity vector are available within all grid-points of a 2D triangulation that covers the entire computational domain; this is what we call the background mesh. In addition to the

background mesh, the fitted discontinuities (either shocks or slip-lines) are discretized using a collection of grid-points (the shock-points) which are mutually joined to form a connected series of line segments (the shock-edges), as described in Fig. 4.2a; shock-points and shock-edges make up what we call the shock-mesh. In contrast to the grid-points of the background mesh, where a single set of dependent variables is stored, the shock-points are duplicated items that share the same geometrical location, but store two different sets of dependent variables, corresponding to the two sides of the discontinuity. This is schematically shown in Fig. 4.2d. In particular, shock-edges, which connect the shock-points on both sides of the discontinuity (see Fig. 4.2d where the width of the discontinuity has been increased to improve readability) also overlap, so that each fitted discontinuity behaves like a double-sided internal boundary of zero thickness. It is important to notice that the spatial location of the fitted discontinuities is independent of the location of the grid-points that make up the background grid, as shown in Fig. 4.2a. The sequence of operations that leads from the available mesh and solution at time t to an updated mesh and solution at time $t + \Delta t$ can be split into the seven steps that will be described hereafter.

Step 1: cell removal around the shock front

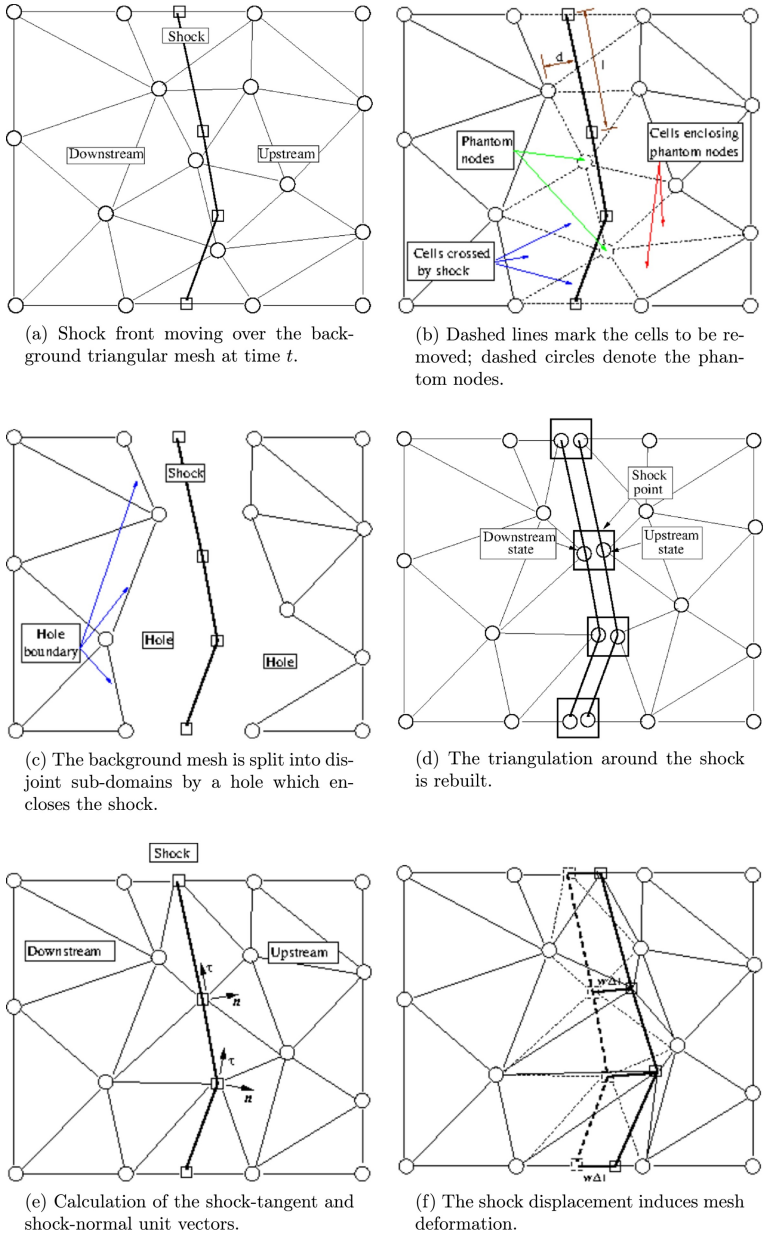
The first step consists in removing those triangles of the background mesh which are crossed by the discontinuity, and also the triangular cells that have at least one of their vertices that are too close to the shock-front. By doing so, a mesh-less hole, which contains the discontinuity, is carved within the background mesh, as shown in Figs. 4.2a, 4.2b and 4.2c. We shall hereafter call phantom-points those grid-points of the background mesh that have been removed in this first step.

Step 2: local re-meshing around the shock front

Following the cell removal step, the hole dug by the fitted front is then re-meshed using a Constrained Delaunay Tessellation (CDT): the edges that make up the fitted discontinuity and the boundary of the hole are both constrained to be part of the final tessellation. The computational mesh, see Fig. 4.2d, that is used to advance the solution in time from t to $t + \Delta t$ is made up of the two different grids that have been generated in the previous and the current step: the background mesh with the hole and the CDT that fills the hole.

Step 3: calculation of the tangent and normal unit vectors to the shock front

In order to apply the R-H jump relations, the tangent, $\boldsymbol{\tau}$, and normal, $\mathbf{n}$, unit vectors have to be calculated within each shock-point. The computation of $\boldsymbol{\tau}_i$, where the subscript i refers to the numbering of shock-points, relies on finite difference formulae which involve the coordinates of shock-point i and those of its neighbouring shock-points. By reference to Fig. 4.3, $\mathbf{r}_i$ denotes the position of shock-point i at time level t. Shock-points $i-1$ and $i+1$ are located on the two sides of shock-point i and their position $\mathbf{r}_{i-1}$ and $\mathbf{r}_{i+1}$ at time level t will be used to compute the tangent and normal unit vectors in shock-point i. Depending on the local flow regime, it may



(a) Shock front moving over the background triangular mesh at time t.

(b) Dashed lines mark the cells to be removed; dashed circles denote the phantom nodes.

(c) The background mesh is split into disjoint sub-domains by a hole which encloses the shock.

(d) The triangulation around the shock is rebuilt.

(e) Calculation of the shock-tangent and shock-normal unit vectors.

(f) The shock displacement induces mesh deformation.

Figure 4.2. Sequence of steps needed to generate the computational mesh at time t

be necessary to use upwind-biased formulae to avoid the appearance of geometrical instabilities along the fitted discontinuities. The criterion used to assess which of the shock-points neighboring i falls in its range of influence is described in [22]. In this respect, three different situations may arise:

1. both shock-points $i-1$ and $i+1$ are in the range of influence of shock-point i;
2. shock-point $i+1$ is outside the range of influence of shock-point i;
3. shock-point $i-1$ is outside the range of influence of shock-point i;

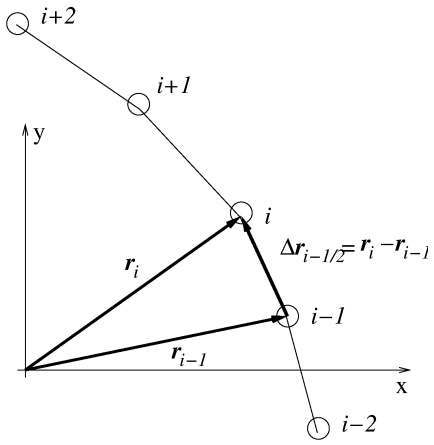


Figure 4.3. Details of the shock-front geometry used in the tangent vector calculation at shock-point i.

When case 1 applies, the computation of $\boldsymbol{\tau}_i$ must involve the shock-points on both sides; therefore:

$$\boldsymbol{\tau}_i = \frac{l_{i+\frac{1}{2}}}{l_{i-\frac{1}{2}}\left(l_{i-\frac{1}{2}} + l_{i+\frac{1}{2}}\right)} \boldsymbol{\Delta r}_{i-\frac{1}{2}} + \frac{l_{i-\frac{1}{2}}}{l_{i+\frac{1}{2}}\left(l_{i-\frac{1}{2}} + l_{i+\frac{1}{2}}\right)} \boldsymbol{\Delta r}_{i+\frac{1}{2}} \tag{4.1a}$$

where $l_{i-\frac{1}{2}} = |\boldsymbol{\Delta r}_{i-\frac{1}{2}}|$.
When case 2 applies, shock-point $i+1$ must not be used in the computation of the tangent vector $\boldsymbol{\tau}$, and the upwind-biased formula (4.1b), which involves shock-point $i-2$, instead of $i+1$, is used:

$$\boldsymbol{\tau}_i = \frac{l_{i-\frac{1}{2}} + l_{i-\frac{3}{2}}}{l_{i-\frac{1}{2}}\, l_{i-\frac{3}{2}}} \boldsymbol{\Delta r}_{i-\frac{1}{2}} - \frac{l_{i-\frac{1}{2}}}{l_{i-\frac{3}{2}}\left(l_{i-\frac{1}{2}} + l_{i-\frac{3}{2}}\right)} \left(\boldsymbol{\Delta r}_{i-\frac{1}{2}} + \boldsymbol{\Delta r}_{i-\frac{3}{2}}\right) \tag{4.1b}$$

Both approximations (4.1a) and (4.1b)can be shown to be second-order-accurate even if the shock-points are un-evenly spaced (in terms of the curvilinear abscissa) along the fitted shock-front. Further details are given in Appendix A 11.1. The third case is specular to the second one, but the corresponding formula involves shock-points i, $i+1$ and $i+2$.

The grid-convergence properties of the Finite Difference (FD) formulae (4.1) have been analyzed in [108]. Finally, the normal unit vector $\mathbf{n}_i$, which is perpendicular to $\boldsymbol{\tau}_i$, is chosen such that it points towards the shock-upstream region, i.e. $\mathbf{u} \cdot \mathbf{n} < 0$.

Step 4: solution update using a shock-capturing code

Using the computational grid in input, a single time-step calculation is performed using one of the unstructured, vertex-centered, shock-capturing solvers described in Sect.2.1.2, which returns updated nodal values at time $t + \Delta t$ within all grid-points of the computational mesh. Since the discontinuities are seen by the S-C code as internal boundaries (of zero thickness) moving with the velocity of the discontinuity, there is no need to modify the spatial discretization scheme already implemented in

the PDEs solver to account for the presence of the fitted discontinuities. In practice, the S-C solver is used as a black-box: it receives in input the computational grid, the nodal values of the solution and grid velocity at time t and returns the updated solution at time $t + \Delta t$.
The solution returned by the S-C solver at time $t + \Delta t$ is however missing some boundary conditions. If we consider the upstream state of a shock wave, the update provided by the shock-capturing solver at time level $t + \Delta t$ is entirely correct. Indeed, within the supersonic, upstream (low-pressure) region, all waves (acoustic, entropy and vorticity) propagate towards the shock so that no boundary condition is required on this internal boundary. The situation is different within the subsonic region attached to the downstream (high-pressure) side of the shock. Here the entropy, vorticity and forward moving acoustic wave propagate away from the discontinuity. Therefore, the provisional values computed by the shock-capturing code within the grid-points located on the downstream side of the shock are wrong, since three (in 2D space) boundary conditions, corresponding to the aforementioned waves, are missing on the shock-downstream internal boundary. Since the downstream region is subsonic in the shock-normal direction, however, the backward moving acoustic wave carries the following signal:

$$R_d{}^{t+\Delta t} = \tilde{a}_d^{t+\Delta t} + \frac{\gamma - 1}{2} \tilde{\mathbf{u}}_d^{t+\Delta t} \cdot \mathbf{n} \tag{4.2}$$

from the shock-downstream region towards the downstream side of the shock. Due to the upwind nature of the spatial discretization, one can therefore assume that the Riemann variable defined by Eq. 4.2 is correctly computed by the S-C code. In Eq. 4.2 the quantities $\tilde{a}_d^{t+\Delta t}$ and $\tilde{\mathbf{u}}_d^{t+\Delta t}$ are the values of the sound speed and flow velocity of the downstream state of the shock nodes computed by the S-C solver. These flow variables have been marked with a "tilde" to underline the fact that these are the provisional (incorrect) values computed at time $t + \Delta t$ by the S-C code before enforcing the jump relation across the discontinuity in the following step.

Step 5: enforcement of the R-H relations

As explained in details elsewhere [22], the shock-downstream values of the dependent variables within the shock-points need to be corrected by enforcing the R-H relations across each pair of shock-points. This amounts to solve using Newton's root-finding algorithm the following system of five (in the 2D space) non-linear algebraic equations within each pair of shock-points, which also supplies the local velocity of the discontinuity, w_s, along its normal.

$$\rho_d^{t+\Delta t} \left[(u_n)_d^{t+\Delta t} - w_s^{t+\Delta t} \right] = \rho_u^{t+\Delta t} \left[(u_n)_u^{t+\Delta t} - w_s^{t+\Delta t} \right] \tag{4.3a}$$

$$\rho_d^{t+\Delta t} ((u_n)_d^{t+\Delta t} - w_s^{t+\Delta t})^2 + p_d^{t+\Delta t} = \rho_u^{t+\Delta t} ((u_n)_u^{t+\Delta t} - w_s^{t+\Delta t})^2 + p_u^{t+\Delta t} \tag{4.3b}$$

$$\frac{\gamma}{\gamma - 1} \frac{p_d^{t+\Delta t}}{\rho_d^{t+\Delta t}} + \frac{1}{2} ((u_n)_d^{t+\Delta t} - w_s^{t+\Delta t})^2 = \frac{\gamma}{\gamma - 1} \frac{p_u^{t+\Delta t}}{\rho_u^{t+\Delta t}} + \frac{1}{2} ((u_n)_u^{t+\Delta t} - w_s^{t+\Delta t})^2 \tag{4.3c}$$

$$(u_\tau)_d^{t+\Delta t} = (u_\tau)_u^{t+\Delta t} \tag{4.3d}$$

$$a_d^{t+\Delta t} + \frac{\gamma - 1}{2} (u_n)_d^{t+\Delta t} = R_d^{t+\Delta t} \tag{4.3e}$$

Equations (4.3a)-(4.3d) are the R-H jump relations and Eq. (4.3e) accounts for the characteristic variable that is conveyed towards the shock from the shock-downstream region (see the discussion already made by reference to Eq. (4.2)). Observe that in

writing Eqs. (4.3) the fluid velocity has been written using its components in the $(\mathbf{n}, \boldsymbol{\tau})$ reference frame defined in step 2. The five unknowns in Eqs. (4.3) are the four shock-downstream values of the primitive variables $(\rho_d, u_{n_d}, u_{\tau_d}, p_d)$ and the component of the shock-speed, w_s, in the shock-normal direction. This will be used in step 6 to move the shock-front. The same approach is used when dealing with contact discontinuities, except that the jump relations must be modified accordingly, see [23] for details.

Step 6: shock displacement

The enforcement of the jump relations provides the speed, w_s, at which each pair of grid-points located on the discontinuity move along its local normal unit vector, n. The position of the discontinuity at time $t + \Delta t$ is computed in a Lagrangian manner by displacing all its grid-points, as shown in Fig. 4.2f where the dashed and solid lines represent the discontinuity at time t, resp. $t + \Delta t$. When simulating steady flows, this can be accomplished using the following first-order-accurate (in time) integration formula :

$$\mathbf{r}_i\left(t + \Delta t\right) = \mathbf{r}_i\left(t\right) + w_{s_i}^t \mathbf{n}_i^t \, \Delta t \tag{4.4}$$

where the (pseudo)-time step Δt is taken small enough to ensure that at time $t + \Delta t$ the shock-front still lies within the hole that has been carved in the mesh in Step 1. The low temporal accuracy of Eq. 4.4 does not affect the spatial accuracy of the steady state solution which only depends on the spatial accuracy of the gas-dynamic solver and that of the tangent and normal unit vectors in Eqs. (4.1).
On the contrary, when dealing with unsteady flows, as in Sec. 4.3.3, the temporal accuracy of the shock motion has to be the same as that of the spatial discretization, i.e. second order accurate in our case. This can be accomplished using a predictor-corrector type temporal integration scheme [30, 24], or a Runge-Kutta multi-step scheme. In the former case, the predictor step estimates the position of the discontinuity at time level $t + \frac{\Delta t}{2}$ using the explicit Euler scheme:

$$\mathbf{r}_i\left(t + \frac{\Delta t}{2}\right) = \mathbf{r}_i\left(t\right) + w_{s_i}^t \mathbf{n}_i^t \, \frac{\Delta t}{2} \tag{4.5}$$

The speed of the discontinuity $w_{s_i}^{t+\frac{\Delta t}{2}}$ and the normal unit vector $\mathbf{n}_i^{t+\frac{\Delta t}{2}}$ at time level $t + \frac{\Delta t}{2}$ are then computed using the intermediate position of the discontinuity $\mathbf{r}_i\left(t + \frac{\Delta t}{2}\right)$ and, finally, the position of each shock-point is updated at time level $t + \Delta t$ in the corrector step:

$$\mathbf{r}_i\left(t + \Delta t\right) = \mathbf{r}_i\left(t\right) + w_{s_i}^{t+\frac{\Delta t}{2}} \mathbf{n}_i^{t+\frac{\Delta t}{2}} \, \Delta t \tag{4.6}$$

One further observations is concerning the discontinuity displacement step. Fig. 4.2f shows that even when the background mesh is fixed in space, the triangular cells that have one of their edges on the discontinuity are constrained to move with it, thus these elements are deformed. This implies that the S-C solver used in Step 4 must be capable of handling moving meshes, i.e. it must be capable of solving the governing PDEs written using an Arbitrary Eulerian Lagrangian (ALE) formulation.

Step 7: interpolation of the phantom points

Upon completion of step 6, nodal values of the dependent variables at time-level $t + \Delta t$ are available within all grid-points of the computational mesh. To be able to compute the subsequent time-step, nodal values of the dependent variables are required within all grid-points of the background mesh, including the so-called phantom-points. The phantom-points are the grid-points of the background mesh that have been temporarily removed in step 1 because of their proximity to the fitted shock-front. Therefore, phantom-points do not belong to the computational mesh and their nodal values have not been updated to time level $t + \Delta t$ in step 4. Due to the shock-displacement, however, the current phantom-points may re-appear at the subsequent time step and it is therefore necessary that meaningful values of the dependent variables are available also within the phantom-points. This is accomplished by first locating the phantom-points inside the current computational mesh and then using linear interpolation.

Upon completion of step 7, the computational mesh can be discarded, and the solution can be advanced over the next time-interval by repeating steps 1 through 7.

4.3 Numerical results

4.3.1 Shock formation due to the coalescence of compression waves

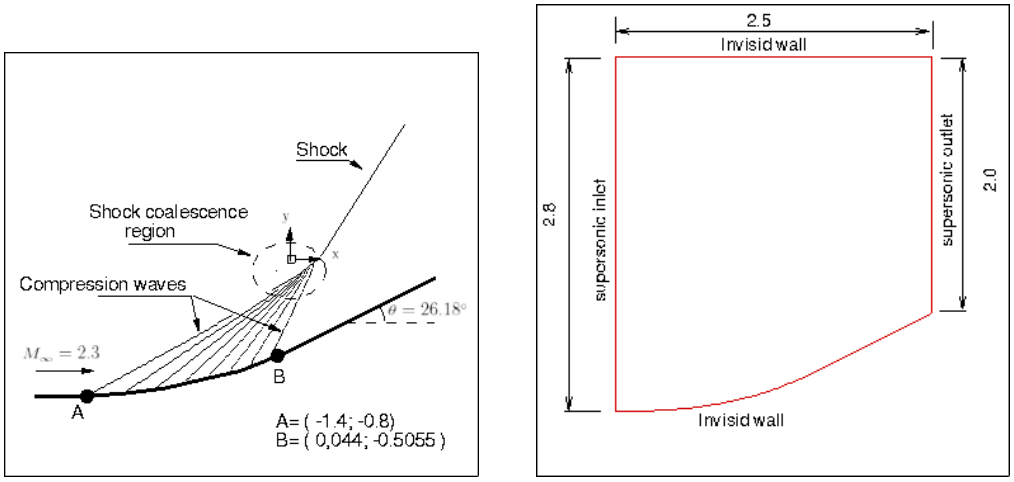
Figure 4.4a shows a supersonic stream ($M_\infty = 2.3$) being deflected by a convex wall. The wall is straight up to point A and beyond point B, whereas the shape of the wall smoothly changes between A and B according to the following cubic polynomial law:

$$y = 0.039923\,x^3 + 0.25144\,x^2 + 0.46928\,x - 0.52683 \qquad -1.4 \leq x \leq 0.044 \quad (4.7)$$

As sketched in Fig. 4.4a, the Mach waves that originate in A and B bound a simple-wave compression region made of straight characteristic lines that merge into a shock-wave at some distance from the wall. Figure 4.4b shows the computational domain and the boundary conditions. The background mesh, which is made of 25531 triangles and 12965 grid-points, has been generated using the `delaundo` mesh generator by specifying a uniform distribution of the boundary nodes with spacing $h = 0.025$.

Figure 4.5 compares the Mach iso-contour lines computed with the shock-capturing and shock-fitting approaches using the `eulfs` CFD solver.

Inspection of Fig.4.5b, which refers to the shock-fitting calculation, clearly reveals that the fitted shock-mesh (shown using a white solid line) starts well ahead of the point where the characteristics merge into the shock. This is because, in Moretti's words [160]: "Premature fitting of the shock in the region where compression waves tend to coalesce is not harmful at all, provided that the shock behaves as one of the characteristic surfaces coalescing into a finite discontinuity". Modeling shock-formation as described by Moretti requires an ad-hoc calculation of the normal to the shock within the end-point of the shock-mesh, so that the present test-case checks that this functionality works correctly.



(a) Flow sketch **(b)** Computational domain

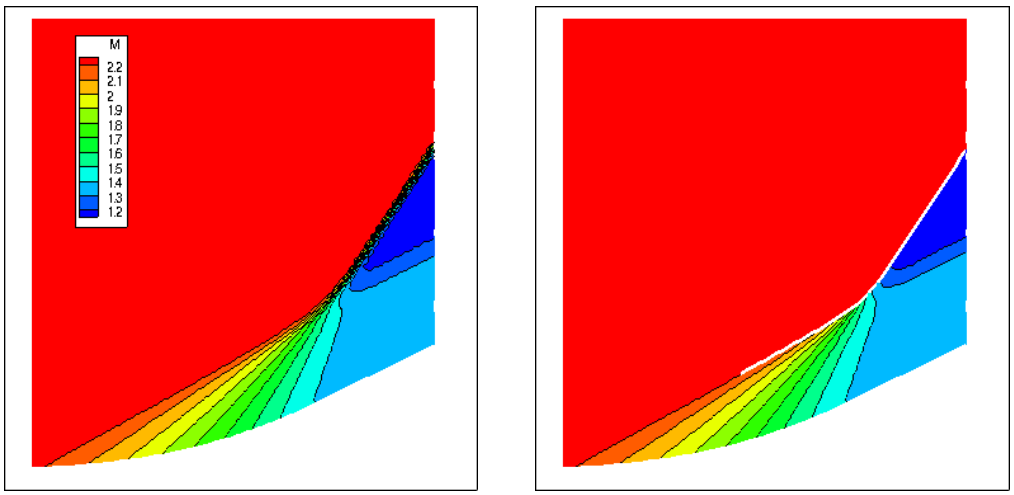
Figure 4.4. Shock formation due to the coalescence of compression waves.

4.3.2 Steady Mach reflection

Whenever the flow deflection imposed by a solid surface on an impinging weak oblique shock is larger than the maximum allowable deflection for the downstream Mach number, a so-called Mach reflection takes place. This is schematically shown in Fig. 4.6: a uniform, supersonic ($M_\infty = 2$) stream of air undergoes a $\Theta = 14°$ deflection through the incident shock, I1. Regular reflection of I1 is however impossible for the chosen pair of M_∞, Θ parameters, so that a steady triple-point (TP) arises which joins I1, the reflected shock (R1), the Mach stem (MS) and the slip-stream (SS).

This test-case has been split into two different simulations: the `MachReflection-1` directory contains the hybrid simulation, whereas a fully-fitted simulation can be run in the `MachReflection-2` directory. In the former case the incident shock and the slip-stream are captured, whereas the reflected shock and the Mach stem are fitted as a single shock. In the latter case all discontinuities, as well as the triple-point, are fitted. From the viewpoint of code-checking the two simulations are very different. The hybrid simulation only demonstrates the capability of the algorithm to run in hybrid mode, but tests the same functionalities already checked in the circular cylinder simulation described in Ref. [109], with the only addition of the interaction between a normal shock (the MS) and a flat wall. On the contrary, the fully fitted simulation tests the capability of the `UnDiFi-2D` code to fit the triple point.

The computational domain used for both simulations has been marked in green in Fig. 4.6. Inside this area a background grid of almost equilateral triangles has been generated using the `delaundo` [161] mesh generator by specifying a uniform distribution of grid-points along the domain boundaries with spacing $h = 0.0167\,L$, see Fig. 4.6. Table 1 reports the number of triangles and grid-points of the background mesh along with those of the computational meshes at steady-state for both the hybrid and the fully-fitted simulations; the corresponding number of shock-points is also shown.



(a) Shock-capturing. **(b)** Shock-fitting.

Figure 4.5. Shock formation due to the coalescence of compression waves: Mach iso-contour lines.

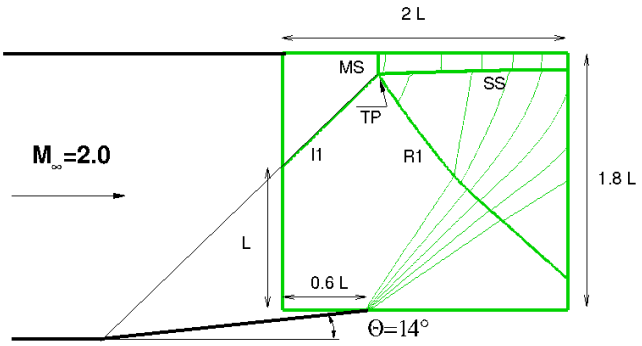


Figure 4.6. Steady Mach reflection: flow configuration.

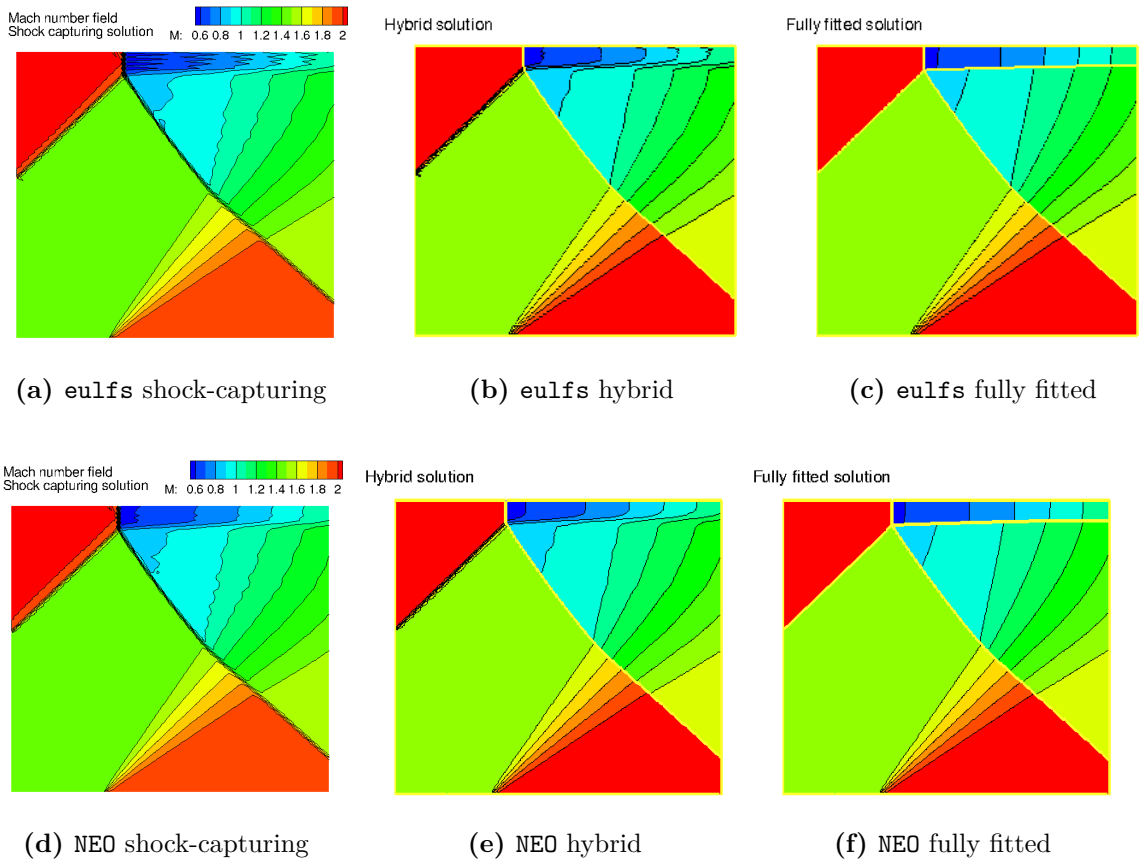
Table 1. Grid-points and triangles of the background and computational meshes.

Background Mesh		Computational meshes at steady-state					
		Hybrid simulation			Fully fitted simulation		
Cells	Nodes	Cells	Nodes	Shock-points	Cells	Nodes	Shock-points
29214	14833	29292	17633	142	29365	19633	294

Shock-capturing, hybrid and fully-fitted calculations obtained using the two different Residual Distribution codes, `NEO` and `eulfs`, are compared in Fig. 4.7.

Even though the two solvers use very similar numerical recipes, as explained in Sect. 2.1.2, it can be seen that the two shock-capturing solutions, Figs. 4.7a and 4.7d, exhibit non-negligible differences, in particular downstream of the Mach stem. These differences are significantly reduced in the hybrid simulations, Figs. 4.7b and 4.7e, and have almost disappeared in the fully fitted ones, Figs. 4.7c and 4.7f.

Further analyses about these numerical solutions can be found in [22, 23], whereas



(a) `eulfs` shock-capturing **(b)** `eulfs` hybrid **(c)** `eulfs` fully fitted

(d) `NEO` shock-capturing **(e)** `NEO` hybrid **(f)** `NEO` fully fitted

Figure 4.7. Mach reflection: Mach number iso-contour lines computed using three different shock-modeling options and the two different shock-capturing codes.

the algorithmic details concerning the treatment of the triple point are reported in [162].

4.3.3 Shock-vortex interaction

This last test-case, which is used to verify that *UnDiFi-2D* works correctly also when dealing with unsteady flows, features the interaction between a moving vortex and a standing shock.

Figure 4.8 shows the computational domain along with the boundary and initial conditions.

The flow-field is initialized by adding the perturbation velocity field induced by the vortex to the uniform flow past a steady normal shock. As shown in Fig. 4.8, at the initial time $t = 0$, the vortex is located 0.2 unit lengths L ahead of the standing shock. Using a cylindrical reference frame attached to the vortex core, the perturbation velocity field reads:

$$\widetilde{u}_\theta = -\epsilon|\boldsymbol{u}_\infty|\tau \exp^{\alpha\left(1-\tau^2\right)} \tag{4.8a}$$

$$\widetilde{u}_r = 0 \tag{4.8b}$$

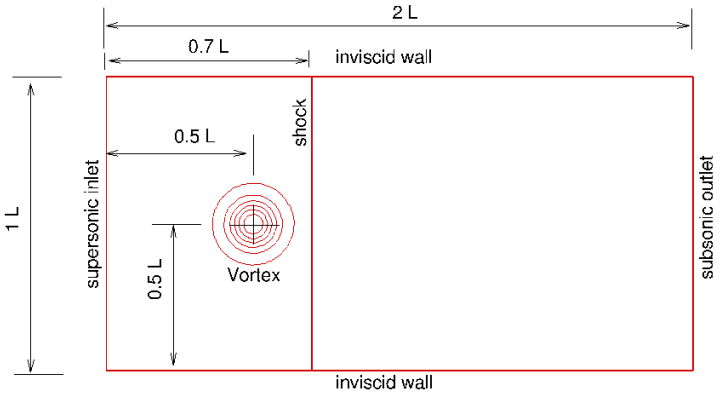


Figure 4.8. Shock–vortex interaction: computational domain and initial condition.

where the dimensionless radial distance from the vortex core, $\tau = \frac{r}{r_c}$, has been used in Eq. (4.8a), with $r_c = 0.05\,L$. The two dimensionless parameters α and ϵ, which respectively control the width and magnitude of the velocity perturbation, are mutually related via the shock and vortex Mach numbers:

$$\epsilon = \frac{M_v}{M_s} \frac{\sqrt{2\alpha}}{\exp^{(\alpha - \frac{1}{2})}} \tag{4.9}$$

where:

$$M_s = \frac{|\boldsymbol{u}_\infty|}{a_\infty} \qquad M_v = \frac{\max |u_\theta|}{a_\infty} \tag{4.10}$$

For the chosen pair of shock and vortex Mach numbers: $M_s = 2$, $M_v = 0.2$, which gives rise to a weak shock-vortex interaction, according to the nomenclature of [163], the vortex strength $\epsilon \approx 8.6\,10^{-2}$ follows from Eq. (4.9), having set $\alpha = 0.204$.

The Delaunay triangulation of the computational domain was generated using `Triangle`; it features 217569 grid-points and 433664 triangles and a mesh spacing along the boundaries equal to $h/L = 0.00375$. The shock-capturing and shock-fitting simulations were performed using the `NEO` solver. A qualitative comparison between the two shock-modeling options is given in Fig. 4.9, where total enthalpy iso-contour lines at three subsequent time instants are shown: shock-capturing on the top row and shock-fitting on the bottom row. In particular, besides the oscillations related to the approximation of the shock, it can be clearly seen that the contours downstream of the discontinuity are much less smooth in the captured solution.

The fitted computations, on the other hand, show very nice and smooth contours. Further details about this test-case and further simulations dealing with shock-vortex interactions can be found in [30, 24].

4.4 Summary

In this Chapter the state-of-the-art of a 10-year-long development of a S-F technique for unstructured meshes was presented: specifically, an overview was provided on

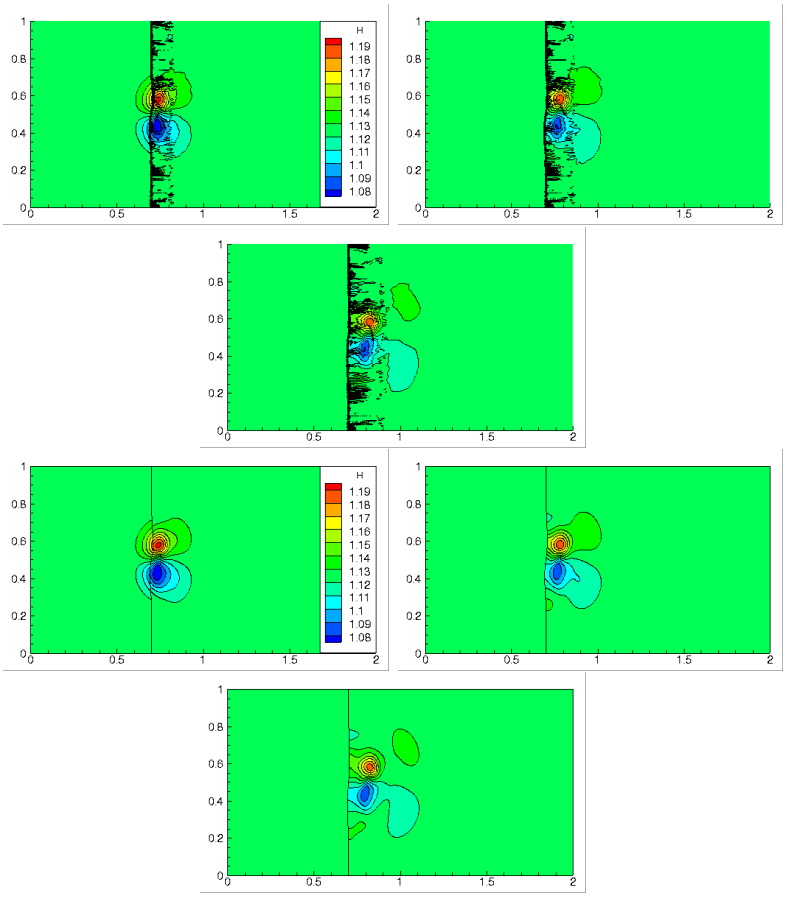


Figure 4.9. Shock-vortex interaction: total enthalpy contours at times $t = 0.3$ (left), $t = 0.4$ (centre) and $t = 0.5$ (right) using the LDA scheme; shock-capturing in the upper row of frames and shock-fitting in the lower row.

the key algorithmic features of the technique, which has been implemented in an open-source code, available in a dedicated repository. A selection of ready-to-run test-cases demonstrates the features and current capabilities of the code, which is able to deal with compressible flows featuring either isolated or mutually interacting discontinuities. The superior quality of fitted shock-waves over captured ones has also been emphasized.

However, some issues require particular attention, as detailed hereafter. In the presented test-cases the effects of viscosity have been neglected. When dealing with viscous flow, shocks cease to be discontinuities in a mathematical sense, and have a finite thickness. At high Reynolds number, however, the width of a shock is comparable to the microscopic length-scales, hence orders of magnitude smaller than any macroscopic length scale, such as Kolmogorov's, for instance. Under this circumstances, it is still appropriate to ignore the inner shock-structure and consider shock-waves as having zero thickness. The capability of dealing with shock/boundary-layer interactions will be addressed in Chapter 8.

Furthermore, when dealing with complex shock-shock and/or shock-boundary interactions in steady flows, a preliminary S-C calculation, followed by an automatic

shock-detection and shock-pattern identification strategy [164], is capable of supplying a reasonably good initial guess of the shock position to the S-F algorithm. When dealing with unsteady flows, however, things get much harder and the currently unsolved issues that remain to be addressed are well summarized in a 1986 paper by Glimm and co-workers [46]:

1. Treating changes of the topology of regions bounded by fronts from simply connected to multiply connected regions.
2. Treating the disappearance of weakening fronts and the appearance of new fronts at boundaries or at collisions of other fronts

As far as the first issue is concerned, changes in the shock topology may occur in unsteady flows for several reasons. For example, new discontinuities may appear when a shock wave moving along a wall encounters an abrupt change in the slope of the wall or when two discontinuities begin to interact. Regarding the second issue, beside the formation of a shock due to the steepening of compression waves, as examined in Sect. 4.3.1, an existing shock may progressively weaken and finally disappear. In order to be able to manage all these topological changes it will be necessary to develop new algorithmic tools capable of detecting the occurrence of a change in the shock-topology and modify accordingly the fitted discontinuities and their mutual interactions. However, the development of a general purpose tool will not be trivial and will probably require the use of advanced, multi-disciplinary techniques such as those used in [164]. The effort which will be necessary to complete the development of the unstructured S-F technique and bring it to full maturity, is not modest and probably requires the merging of different skills and expertise. In order to be successful, it will be necessary to broaden the audience of developers and technical expertise involved. This is the reason behind the launch of the present project. Indeed, *UnDiFi-2D* has clearly some important limitations, as stated before: nevertheless, the software platform presented in this Chapter represents a solid starting point for developments to overcome these limitations in a collaborative framework.

Chapter 5

UnDiFi-3D

Contents

Recently hypersonic flows computations around re-entry vehicles have experienced a growing interest among the CFD community due to the difficulties for reproducing the re-entry conditions in ground-based experiments. The major challenge for numerical simulations is a more reliable prediction of the wall heating, which can be obtained with a more accurate modelling of shock waves. As stated before in Chap. 2, this task can be achieved using two different approaches: shock-capturing (S-C) and shock-fitting (S-F). The S-F techniques were originally developed in the structured grids framework. The reliance on these types of grids made the development of a S-F solver able to solve flows with shock interactions very difficult and complex [165]. Over the last decade, however, some researchers [22] have been developing a novel S-F technique for unstructured grids which allows to relieve much of the algorithmic complexity that plagued the traditional structured S-F techniques. In later years, this technique was further developed in order to be applied for the first time on 3D unstructured grids [25] even if limited to the computation of inviscid flows. In this preliminary version of the 3D S-F solver, the mesh generation and shock surface handling relied on general propose codes (Tetgen [166] and Yams [167]). Nevertheless, since these mesh generation codes were not specifically designed for unstructured S-F, this S-F solver for 3D flows had strong limitations. Some years ago, Zaide and Ollivier-Gooch [47] developed software tools based on the GRUMMP library for inserting lines/surfaces into existing unstructured 2D/3D meshes: their software represented a good starting point for the development of a mesh generator and handling kernel for the 3D unstructured S-F technique able i) to join together multiple shock surfaces at reflection/interaction lines, ii) to re-mesh shock surfaces and iii) to move shock surfaces through the volume mesh. This Chapter describes recent development of the version of the 3D unstructured S-F based on the GRUMMP

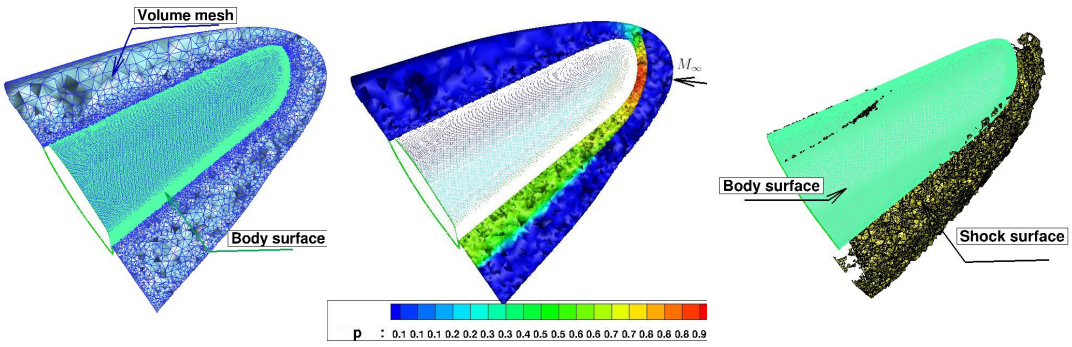
library and provides also some details about the shock surface extraction process from a S-C solution, which is a preliminary action of every S-F computation. Moreover, different test-cases will be also presented in Sec.5.2.

5.1 *UnDiFi-3D*: algorithmic features

The main steps of the S-F algorithm for 3D compressible flows are briefly described below. The S-F technique models shock waves in 3D space as zero-thickness, double-sided, triangulated surface. An initial surface that approximates the position of the shock wave has to be given as input to start the S-F simulation, as well as the background grid and the computed S-C solution, which is used as an initial condition by the proposed S-F algorithm, providing also the upstream states for each point on the shock mesh. Specifically, a simple approach to extract the shock surface starting from an S-C solution are provided in step 0.

Step 0: Shock surface extraction and insertion in the background volume mesh

Let us consider a hypersonic flow past a generic blunt body. Figure 5.1a highlights a portion of the volume mesh that surrounds the blunt body, while Fig. 5.1b displays the normalized pressure flow field computed using a S-C solver on the mesh shown in Fig. 5.1a. A first attempt shock surface, close to the actual shock position, could be extracted by the S-C solution by means of a specific detection technique, such as the one described in Ref. [164] for 2D applications. However, a shock surface can be also obtained using the shock feature of the commercial software TECPLOT [168]. Figure 5.1c shows the result provided by this software tool.

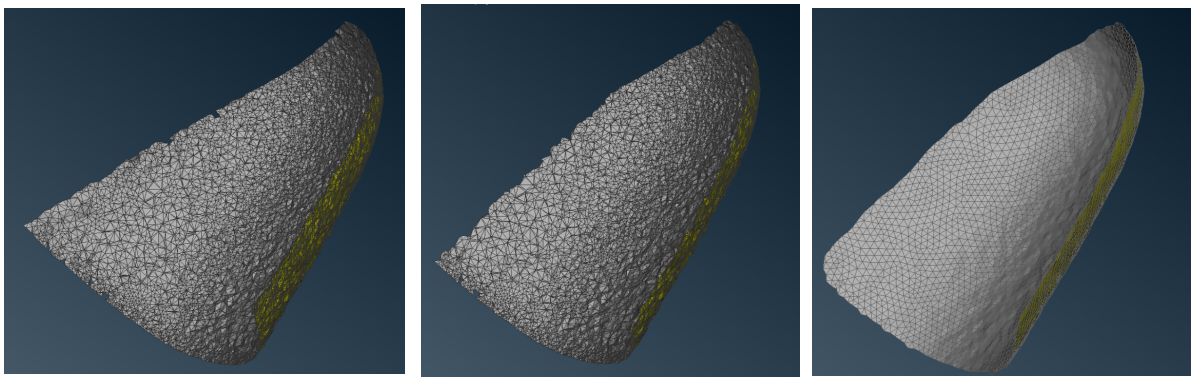


(a) S-C computational mesh **(b)** Normalized pressure field computed by the S-C solver **(c)** Extracted shock surface from S-C solution

Figure 5.1. Example of shock surface extraction process

Indeed, as you can observe in Fig. 5.1c, the surface extracted using TECPLOT is very irregular and, moreover, it is characterized by many spurious points. For this reason, this surface must be processed before considering it as an input for the

S-F solver. Figure 5.2 shows the steps of the post extraction process that can be achieved using different commercial and open source softwares, for instance MeshLab and ANSA. The first step is the removal of all the artefacts (see fig. 5.2a). Then the shock surface is cut in order to remove its side regions where the shock is very weak (see Fig. 5.2b). The resulting shock surface can still be very irregular or its triangulation provides low mesh quality. Using specific tools for mesh smoothing and mesh reconstruction, the shock surface is smoothed and re-meshed to obtain a smooth shock surface composed of triangles whose sizes ranging within an assigned range, as highlighted by Fig. 5.2c. The final result of these alterations, shown in Fig. 5.2c, is ready to be provided as input to the S-F procedure and to be inserted in the background grid, as shown by the nosetip detail in Fig. 5.4. This surface represents the initial position of the shock wave in the computing process: subsequently this surface is deformed and moved by the S-F technique until the correct position is reached.



(a) Artifacts and spurious points removal from the original surface.

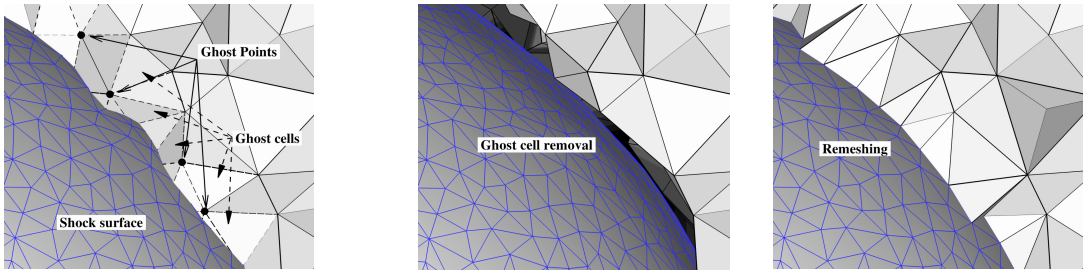
(b) side region removal

(c) Surface meshing and smoothing

Figure 5.2. Example of shock surface meshing process

At this point, this surface, which represents the initial position of the shock wave, must be inserted within the background volume mesh, in order to generate the computational grid used by the gas-dynamic solver for advancing the solution in time. The insertion of the shock surface into the background grid [47] is based on two stages: the former re-discretizes the shock surface according to the length scale of the existing volume mesh. Specifically, the input triangulated shock surfaces are converted to interpolated spline surfaces using CGM [169]: the shock surface final discretization is obtained by sampling each smooth surface [170], starting with the perimeter of the shock surface, then sampling the interior of the surface to ensure topological correctness, geometric fidelity, and mesh quality. Once that the new surface triangulation is generated, it is inserted within the volume mesh. First of all, a cavity is carved in the volume mesh around the surface, by removing those elements which have at least a vertex too close to the shock surface: in particular, all vertices in the volume mesh that are closer than half of the local length scale to any points on the re-discretized surface are removed. This condition avoids the creation of short edges in the computational grid: anyway, an additional criterion is used for

removing the vertices of the background mesh, in order to ensure that all the triangles of the discretized surface can be recovered after insertion: therefore, any volume mesh points that fall inside a sphere whose equator is the circumcircle of a surface triangle is also removed. The deleted vertices will be hereafter called "phantom points", whereas the removed elements of the background grid are defined "ghost cells". This procedure is shown in Fig. 5.3a. Then, these elements are removed from the volume mesh (see Fig. 5.3b). Finally, the surface points are inserted, the surface triangles are recovered and the shock surface is duplicated to produce upstream and downstream shock points (Fig. 5.3c): in particular, this is achieved by replacing each shock point by two superimposed mesh points, respectively belonging to the downstream and upstream region, so that the downstream and upstream states are correspondingly assigned. By doing so, the so called computational volume mesh is generated, which is therefore cut into non-communicating parts by the doubled shock surface: it must be noted that the aforementioned procedure modifies the background volume mesh only around the inserted surface, so that the final grid is nearly identical to the background tetrahedral mesh. At this time, the iterative S-F procedure for computing the solution gets start until the steady state is reached.



(a) Identification of the phantom nodes and ghost cells of the background mesh

(b) Elements tagged as ghost cells are removed

(c) Generation of the modified mesh

Figure 5.3. Inserting a shock surface in the mesh

Step 1: Computation of the tangent and normal unit vectors

The normal unit vectors **n** on the shock surface at each shock point are required to compute the jump relations, as described in step 3. The computation of these vectors at a generic shock point is carried out by averaging the face normals of the shock faces that share that given shock point and that fall inside the range of influence of the shock point itself, as in Ref. [25]. Indeed, when the shock-downstream region is subsonic, all shock points that surround a given shock point belong to its range of influence, since the speed of sound is always larger that the flow speed. On the other hand, in a supersonic region, only a subset of the shock points that surround the test point belong to its range of influence. In order to illustrate how which criterion is used fo establishing which of the shock faces that surround a given shock point fall into its range of influence, the shock point J depicted in Fig. 5.5 is considered, and two points connected to J such as A and B: moreover, let Δt be a testing time increment, which is fixed smaller than the physical time step. At time $t + \Delta t$ the

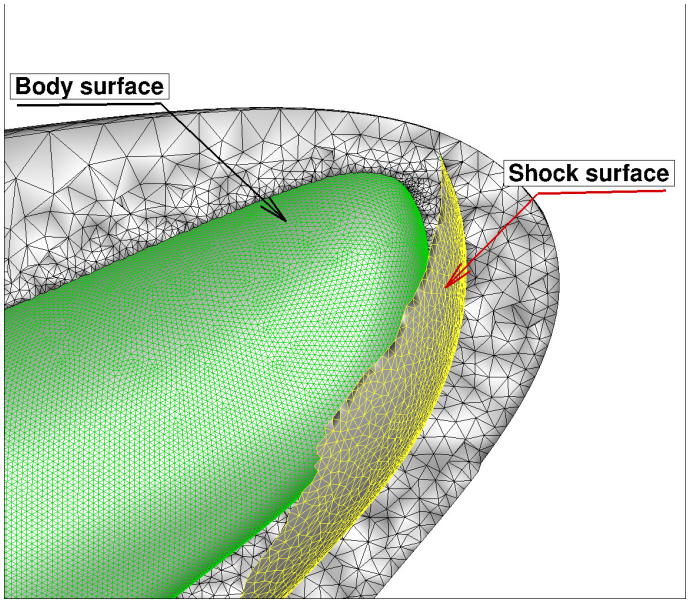


Figure 5.4. Example of the merging between the shock surface and the background mesh

fluid particles that at time t were at A and B have moved to points A'' and B'', according to the values of their flow velocity u_A and u_B: in particular, a generic shock point belongs to the dependence domain of point J if a sound signal emitted from it at time t will eventually reach point J. This is equivalent to require that the distance between the acoustic front emitted from that point at time $t + \Delta t$ and the shock point J must be smaller than the distance separating the two shock points: for this reason, referring to Fig. 5.5, point A falls in the range of influence of point J, but B does not.

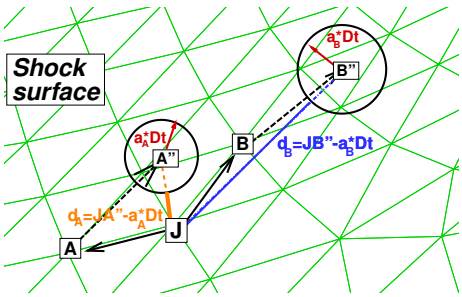


Figure 5.5. Shock-point normal vector computation: definition of the dependence domain.

Step 2: Solution update using a shock-capturing code

Using the computational grid as input, a single time-step calculation is performed using an unstructured, vertex-centered, S-C solver, obtaining nodal values updated at time $t + \Delta t$ at all grid-points of the computational mesh generated in step 0, before the iterative S-F procedure. The gas-dynamic solver used in the present implementation is EulFS [81], the in-house code based on the Fluctuation Splitting

approach previously described in Sec. 2.1.2. The solution returned by the S-C solver at time $t + \Delta t$ is however missing some boundary conditions, as previously discussed regarding step 4 of the *UnDiFi-2D* algorithm in Sec.4.2. These missing pieces of information will be therefore determined in the following step.

Step 3: Shock nodes calculation

As explained in detail elsewhere [25], the shock-downstream values of the dependent variables within the shock-points need to be corrected by enforcing the jump relations across the upstream and downstream states of each pair of shock-points. This amounts to solving the following system of six non-linear algebraic equations (in the 3D space) at each pair of shock-points:

$$\rho_d{}^{t+\Delta t}(u_{dn} - w) = \rho_u{}^{t+\Delta t}(u_{un} - w) \tag{5.1}$$

$$\rho_d{}^{t+\Delta t}(u_{dn} - w)^2 + p_d{}^{t+\Delta t} = \rho_u{}^{t+\Delta t}(u_{un} - w)^2 + p_u{}^{t+\Delta t} \tag{5.2}$$

$$\frac{\gamma}{\gamma - 1}\frac{p_d{}^{t+\Delta t}}{\rho_d{}^{t+\Delta t}} + \frac{(u_{dn} - w)^2}{2} = \frac{\gamma}{\gamma - 1}\frac{p_u{}^{t+\Delta t}}{\rho_u{}^{t+\Delta t}} + \frac{(u_{un} - w)^2}{2} \tag{5.3}$$

$$\mathbf{u}_d{}^{t+\Delta t} - u_{dn}\mathbf{n} = \mathbf{u}_u{}^{t+\Delta t} - u_{un}\mathbf{n} \tag{5.4}$$

$$\sqrt{\gamma \cdot \frac{p_d{}^{t+\Delta t}}{\rho_d{}^{t+\Delta t}}} + \frac{\gamma - 1}{2}u_{dn} = R_d{}^{t+\Delta t} \tag{5.5}$$

Eqs. 5.1- 5.4 are the R-H jump relations. The last equation is referred to the Riemann variable associated with the acoustic wave which moves upstream towards the shock, which was already discussed in Eq. (4.2)) in the previous Chapter. The aforementioned system is therefore solved using a Newton–Raphson algorithm and supplies also the local shock speed w.

Step 4: Shock displacement and solution interpolation at the phantom nodes

The position of the shock at time-level $t + \Delta t$ is computed by moving all shock points according to the shock speed ($w^{t+\Delta t}$) obtained in the previous step. If Δt is kept sufficiently small, the shock surface will overtake only phantom nodes but none of the active points of the computational mesh. All phantom vertices are checked to determine if they are far enough from the shock that they should be re-inserted into the active mesh. Those that will be re-activated must have their solution state updated. This is done by identifying the volume cell they fall within. The solution at all active vertices in the mesh has already been updated to $t + \Delta t$, and this solution is linearly interpolated to the phantom vertices. These phantoms are then re-inserted into the mesh, taking care to preserve triangles on the shock. Non-phantom vertices near the shock are checked to see if they are too close to the shock. Those that verify this condition are removed from the mesh topology and tagged as phantoms: thus the computational grid is locally remeshed around the shock front, to take into account of the newly add/removed phantom points. After these updates to the computational mesh and solution, the solution at new time step $t + 2\Delta t$ can be computed repeating the computing procedure from the step 1.

5.2 Numerical results

In the following subsections the results obtained in the simulation of high-speed flows past two different blunt bodies using both S-C and S-F solver will be analysed and discussed in order to highlight the differences between these two shock-modelling options. Specifically, the former test-case regards the hypersonic flow past an hemisphere and allows to perform a quantitative assessment of the order of convergence of the two techniques: an hypersonic viscous flow past a sphere has been also computed by considering the Blottner flow configuration [171] in the second test case, in order to provide also comparison between the experimental data and the results obtained using both the shock modelling approaches.

5.2.1 Hypersonic flow past an hemisphere

The hypersonic flow ($M_\infty = 10$) past an hemisphere has been numerically computed using the computational domain shown in Fig. 5.6: the hemisphere has been carved within an orthogonal parallelepiped, half of which is shown in Fig. 5.6; R is radius of the hemisphere and the triangulated surface of the bow-shock is shown in blue. Specifically, this test case has been computed using both S-F and S-C, where the S-C solution has been obtained using EulFS code described in Sec. 2.1.2. Moreover S-C and S-F calculations have been performed on nearly identical tetrahedral grids, since the modified S-F mesh differs from the background tessellation only in the neighbourhood of the fitted shock-surface (compare Figs. 5.7a and 5.7b) where it has been locally modified by the GRUMMP library to give room to the triangulated shock surface. In order to evaluate the grid-convergence properties of the two different shock-modeling options, calculations have been performed using two nested grid levels, the finest one obtained by splitting each cell of the coarser mesh into eight tetrahedra.

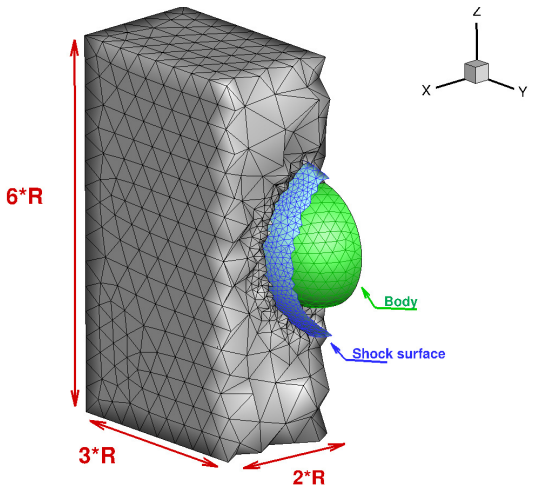
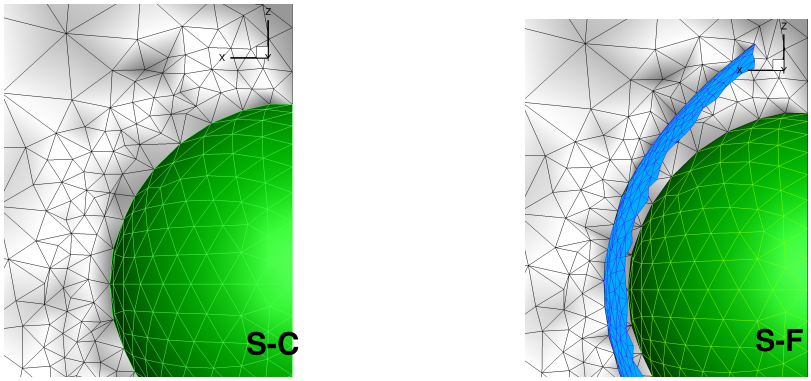


Figure 5.6. Computational domain

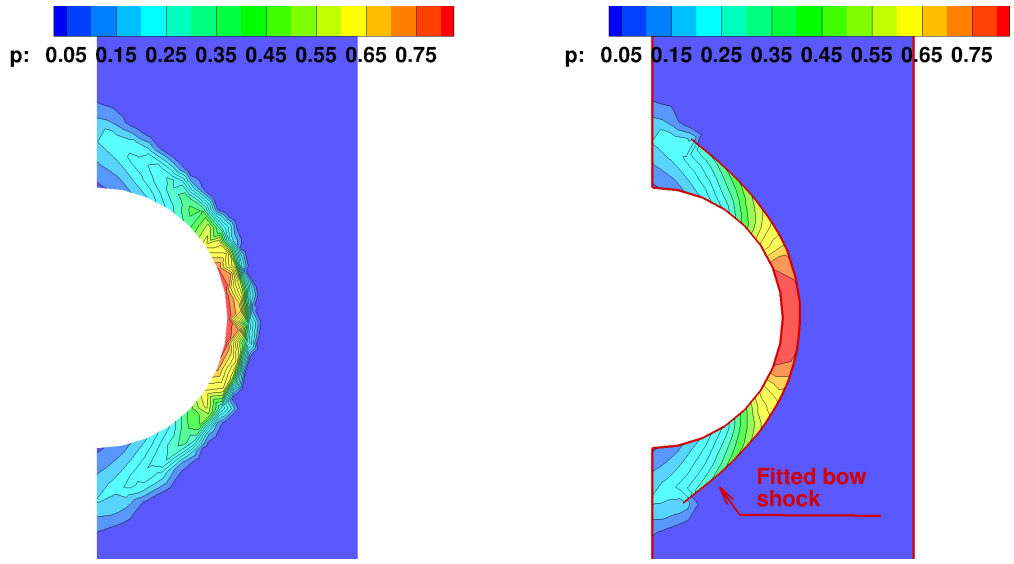
The superior performance of the S-F technique is evident in Fig. 5.8, which compares the dimensionless pressure field within the XZ plane computed using both



(a) S-C mesh side view **(b)** S-F mesh side view

Figure 5.7. Hypersonic flow past a hemisphere: grid detail (plane XZ view)

S-C and S-F on the coarsest grid. The S-C calculation (see Fig. 5.8a) features a shock-thickness which is comparable to the shock stand-off distance and is characterized by the presence of spurious disturbances within the entire shock layer, due to the mis-alignment between the shock-surface and the triangular faces of the grid. On the contrary, the S-F calculation, Fig. 5.8b, features a discontinuity of zero thickness and a smooth pressure-field downstream of the bow shock.

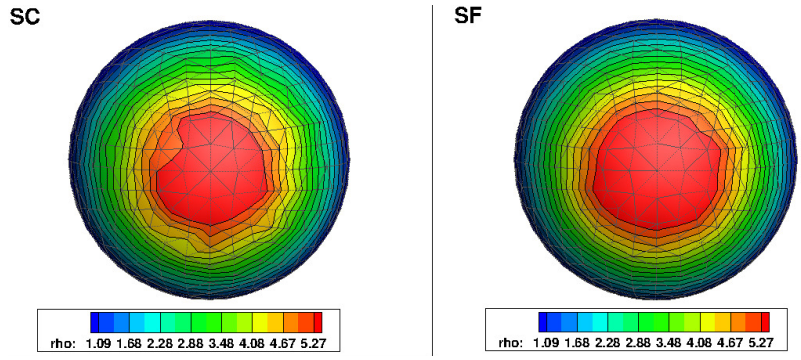


(a) S-C computation **(b)** S-F computation

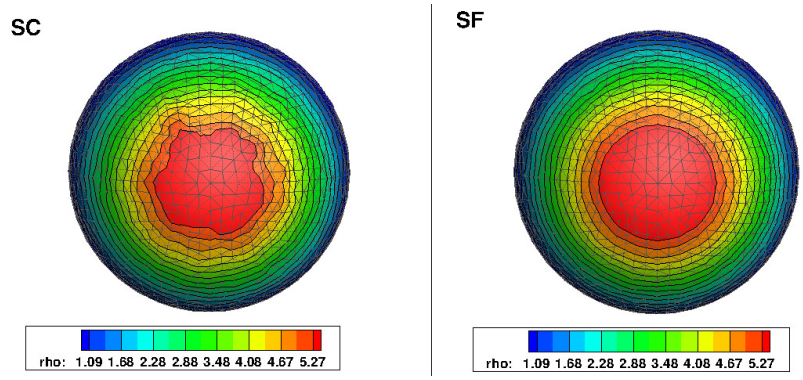
Figure 5.8. Hypersonic flow past a hemisphere: dimensionless pressure field in the XZ plane.)

The S-F improvements are further confirmed by the comparison of the dimensionless density distributions over the body surface, which is shown in Figs. 5.9a and 5.9b for the coarse and fine grid levels. Indeed, in the S-C calculations shown in the two frames on the left of Fig. 5.9, the density iso-contour lines are characterized

by severe oscillations that pollute the solution, in particular close to the stagnation point. In contrast, the S-F calculation reveals a reasonably good circumferential symmetry (particularly on the finer mesh) despite the use of a fully unstructured tetrahedralization.



(a) Coarse grid level



(b) Fine grid level

Figure 5.9. Hypersonic flow past a hemisphere: dimensionless density field over the body surface.

The two nested meshes also allow to assess the observed order of accuracy, $\tilde{n}$, and discretization error, ϵ_h to estimate the difference between the numerical solution u_h computed on a mesh of spacing h and the exact solution, u_0: in particular, the exact solution u_0 is the exact stagnation point value of either pressure or density, which can be analytically determined because the bow shock is a normal shock at the point where it is crossed by the stagnation streamline. Specifically, the observed order of accuracy has been calculated according to Eq. 3.1, where the grid refinement ratio is equal to $r = 2$ for nested meshes, whereas the convergence monitor R is computed following Eq. 3.6.
Table 2, which shows the numerically and analytically computed values of pressure and density at stagnation point clearly reveals that S-F provides values much closer to the exact value than S-C.

The results of the grid-convergence analysis are reported in Tab. 3: on both grid levels the discretization error incurred by S-F is one order of magnitude smaller than that of S-C. In particular, this observation holds true for both pressure and density. Furthermore, the comparison between the pressure and density computed

Table 2. Inviscid flow past a hemisphere: stagnation point data.

	Shock capturing	Shock fitting	Analytical
Quantity	fine grid	fine grid	
P/P_∞	125.04	128.93	129.21
ρ/ρ_∞	5.80	6.13	6.15

values highligths that S-F solutions are characterized by a convergence trend which is better that the S-C one. By evaluating the data in Table. 3,is clear that S-F exhibits an observed order closer to the designed order $\tilde{n} = 2$ with respect to S-C, especially when considering the density analysis.

Table 3. Inviscid flow past an hemisphere: convergence analysis.

	Pressure analysis				Density analysis			
	ϵ_{2h}	ϵ_h	R^{-1}	$\tilde{n}$	ϵ_{2h}	ϵ_h	R^{-1}	$\tilde{n}$
S-C	11.76	4.16	2.82	1.49	0.698	0.357	1.96	0.97
S-F	1.07	0.29	3.73	1.90	0.06	0.02	3.00	1.59

5.2.2 Hypersonic laminar flow past a sphere

To evaluate the quality of the numerical solutions that the S-F technique is able to offer, also the well known benchmark proposed by Blottner [171] was calculated and the numerical solutions analyzed. The hypersonic laminar flow ($M_\infty = 5$, Re=$1.88 \cdot 10^6$) past a sphere has been numerically computed in a tetrahedral computational domain. Some mesh statistics are given in Tab. 4; Figure 5.10 shows the surface of the sphere in red, and the triangulated surface of the bow shock wave (composed of 2615 nodes and 5175 elements) in blue.

The superior performance of the S-F technique is revealed by Fig. 5.11, which compares the Mach flow-field in the symmetry XZ plane computed using both S-C and S-F on nearly identical tetrahedral grids. Indeed, it is worth to remind that the S-C mesh coincides in general with the background mesh used in the S-F calculation since the computational S-F grid differs from the background tessellation only in the neighbourhood of the fitted shock-surface, as previously discussed. Also in this case the S-C solution (see Fig. 5.11a) features a shock-thickness which is comparable to the shock stand-off distance and is characterized by the presence of spurious disturbances within the entire shock layer. On the contrary, the S-F solution (Fig. 5.11b) features a discontinuity of zero thickness and provides smoother Mach iso-contours distribution downstream of the bow shock.

The improved solution quality offered by S-F is further confirmed by the dimensionless pressure distribution over the body surface, which is shown in Fig. 5.12. Indeed, in the S-C calculations shown in Fig. 5.12a, the pressure iso-contour lines are characterized by severe oscillations that pollute the solution, especially near

Table 4. Computational domain features

$N°$ nodes	$N°$ elements	First cell height
93309	542160	$4.9 \cdot 10^{-6}$

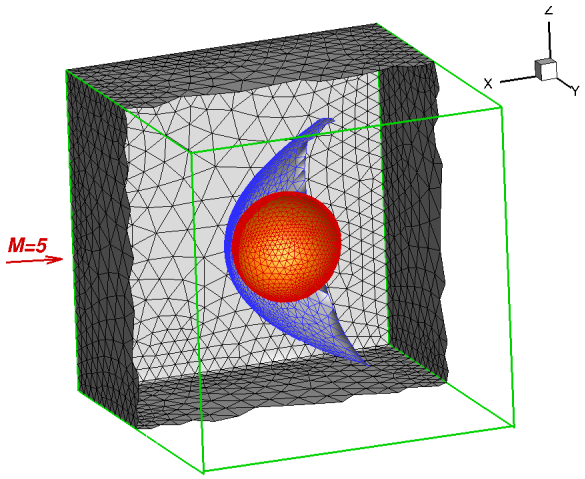
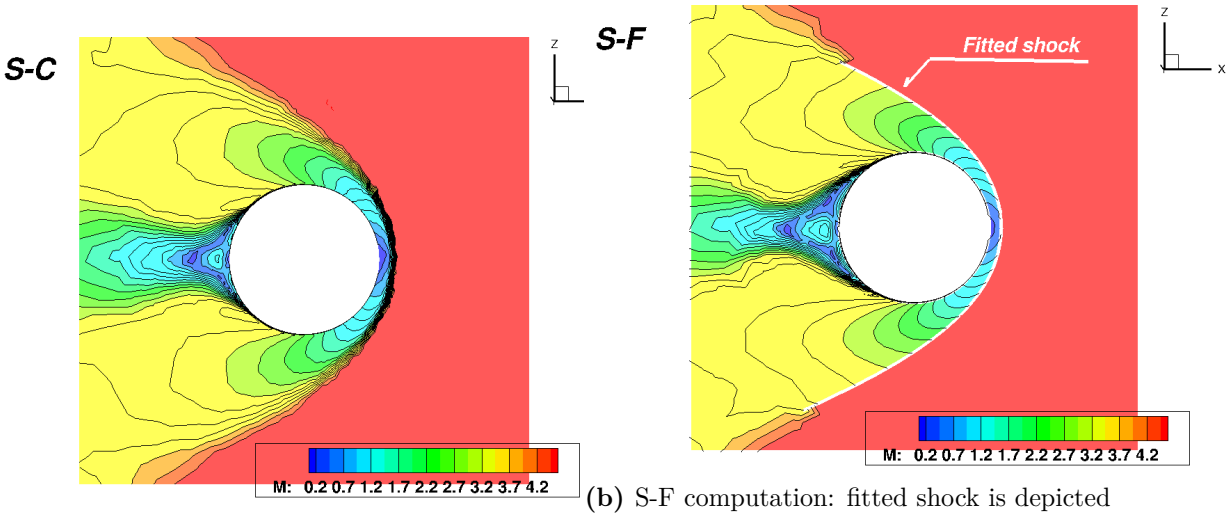


Figure 5.10. Hypersonic flow past a sphere (M_∞=5, Re_∞=$1.88 \cdot 10^6$): computational domain

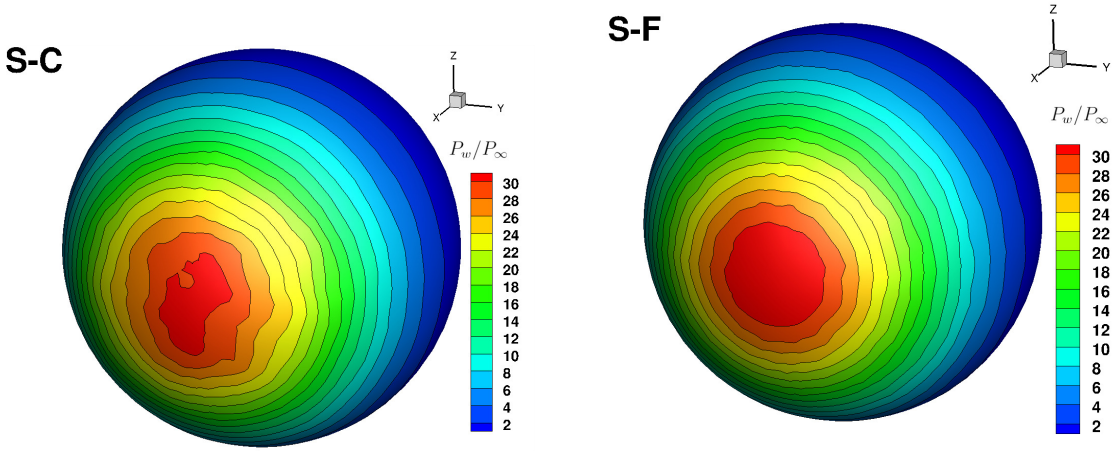


(a) S-C computation

(b) S-F computation: fitted shock is depicted using a white solid line.

Figure 5.11. Hypersonic flow past a sphere (M=5): Mach flow-field.

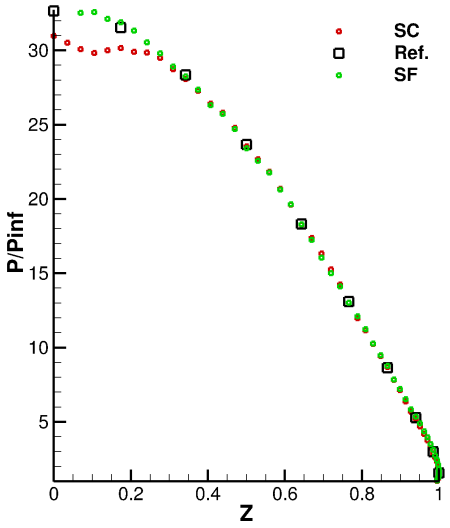
the stagnation point. In contrast, the S-F calculation reveals iso-contours without oscillations and with reasonably good circumferential symmetry (see Fig. 5.12b). A further comparison between the two shock modeling approaches can be carried out comparing these results with the data available in Ref. [171]. The surface pressure distribution computed by the S-F technique is confronted in fig. 5.13a with the Blottner data and with the one computed using the S-C solver (EulFS), which is the same solver used for the smooth regions calculation by the S-F technique (see step 3). Both the solutions are in good agreement with the Blottner pressure data except in the stagnation region, where the S-F technique continues to provide a good agreement, whereas the S-C technique shows results significantly different from the



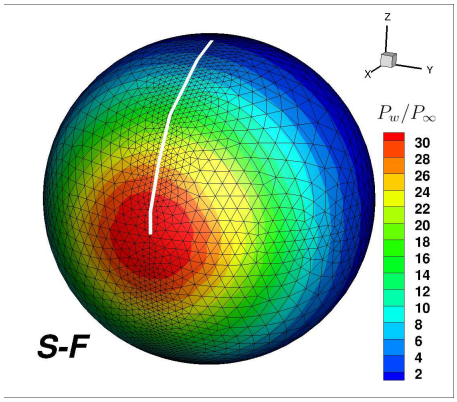
(a) S-C computation. **(b)** S-F computation

Figure 5.12. Hypersonic flow past a sphere (M=5): normalized wall pressure distribution

reference ones.



(a) Surface pressure distribution along the reference line shown in fig. 5.13b.

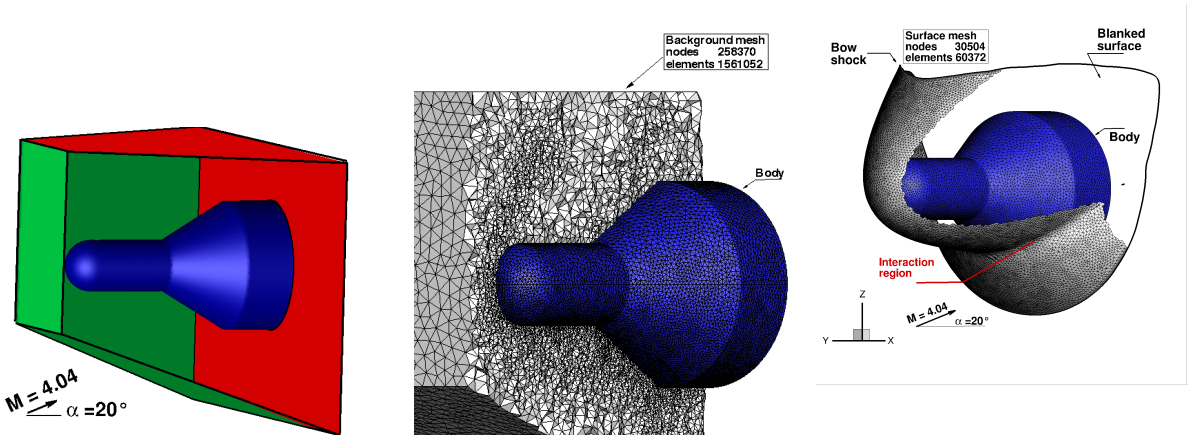


(b) Surface pressure distribution and reference line (marked in white)

Figure 5.13. Hypersonic flow past a sphere (M=5): comparison between pressure reference data [171] and numerical simulations

5.3 Future Work: an example of shock-shock interaction.

Results obtained in the previous section and in Ref. [172] using the unstructured S-F approach, encourage a further development of the proposed technique. Up to date, the GRUMMP library routines implemented in the S-F code do not handle multiple shocks. For this reason, high speed flows characterized by shock-shock interaction can be calculated by the present 3D S-F technique only in a hybrid mode, where the strongest shock is fitted, while the remaining discontinuities and their interactions are captured. A new version of the code require to be developed that can handle multiple shock surfaces and manage shock interactions. To test the new features that must be included in new version of the code, a new specific test case is considered. The test case considers a supersonic flow past a blunt-nosed body, featured by the presence of interacting discontinuities. The body geometry consists of a cylinder with a hemispherical nose, a conical flare making an angle of 30° w.r.t. the body axis and a cylindrical tail, as described by Fig. 5.14a, whereas Fig. 5.14b shows a portion of the volume mesh used for performing the numerical computation using both S-F and S-C solvers. The free-stream Mach number is $M_\infty = 4.04$ and the angle of attack $\alpha = 20°$, as reported in Fig. 5.14c. This test case was experimentally studied in Ref. [173]. As stated before, the flow-field is characterized not only by a bow shock, but also by an embedded shock which originates at the cylinder-cone junction. These two shocks interact giving rise to a type VI shock-shock interaction, as classified by Edney [174]. As explained before, the embedded shock surface is not



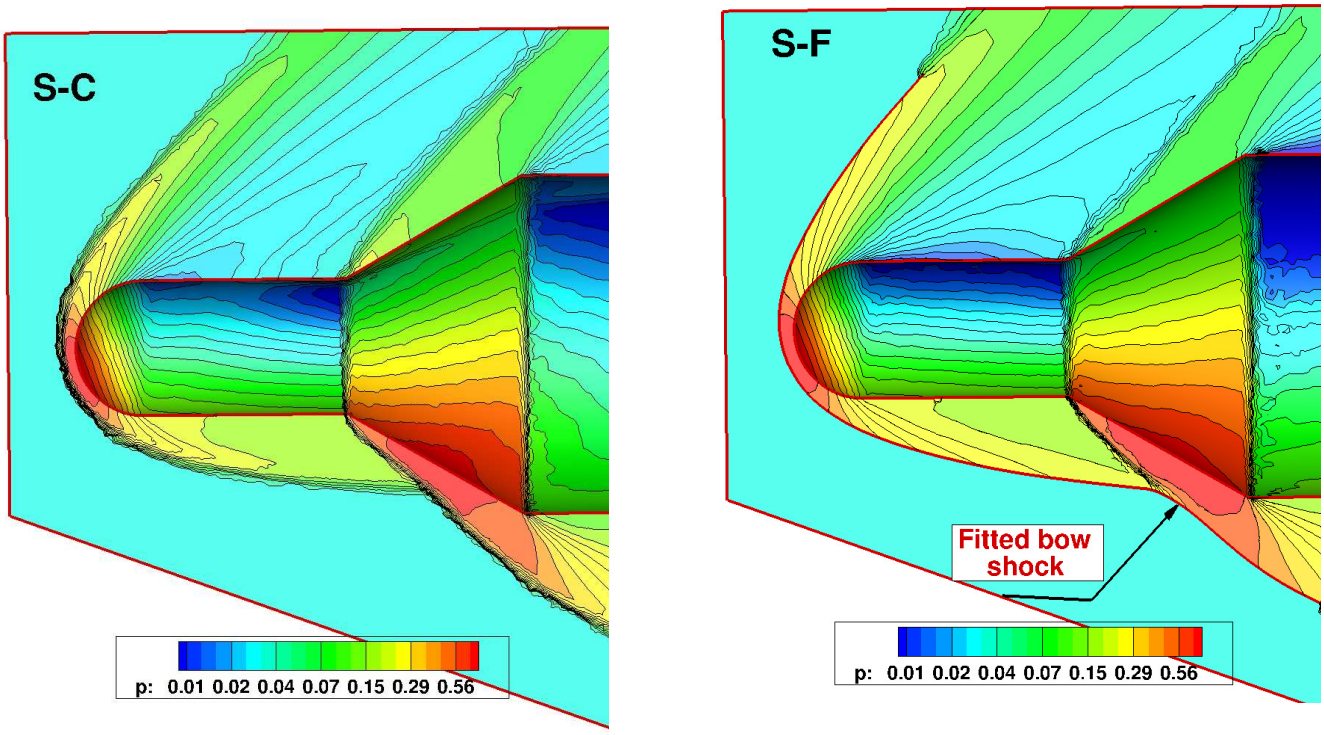
(a) Computational domain **(b)** Background mesh **(c)** Bow shock surface

Figure 5.14. Hypersonic flow past a flare: computational domain and mesh details

visible in Fig. 5.14c, because the S-F technique has been used in an hybrid manner, whereby only the bow shock has been fitted, whereas the embedded one has been captured.

A qualitative comparison between the S-C and the S-F solutions is available in Figs. 5.15, which show the dimensionless pressure field computed using the two shock-modeling options respectively on the symmetry plane XZ, close to the hemispherical

nose and in the region where the shock-shock interaction takes place. By comparing the left and right frames of these figures, it is clear that significant differences exist between the pressure field computed by S-C and S-F downstream of the bow shock, both in the symmetry plane and over the body surface. Indeed, S-C solutions are characterized by finite shock thickness, which spans a few cells, the size of which is comparable to the shock stand-off distance. By modelling shocks using the S-F technique, the shock thickness is brought near to its physical size, that is an order of magnitude greater than the gas mean free path, therefore it allows to obtain high quality solutions on a nearly identical grid to the one used by S-C solver. In particular, Fig. 5.15 highlights the future capabilities of the new code: it shows a zoom of the region where the shock-shock interaction takes place in the hybrid simulation. Due to the numerical thickness of the embedded shock that is captured by the gas-dynamic solver the interaction point is smeared in a region. This region has no physical meaning and is only a numerical artifact. It is evident that the use of a fully S-F technique where all the shocks and interactions are fitted may further improve the solution quality with respect to the hybrid solution if the embedded shock and the shock-shock interaction were also fitted, rather than being captured as in Fig. 5.15. This requires an improvement of the capabilities in the mesh generation and handling of the S-F solver currently under development using the GRUMMP library.



(a) S-C solution **(b)** S-F solution

Figure 5.15. Hypersonic flow past a flare: computed pressure field in the XZ plane

5.4 Summary

The diffusion of S-F techniques in the CFD world is severely limited by the great difficulty of developing sufficiently general S-F solvers able to handle 3D flows with interacting shock waves. Past experience has shown that these difficulties are insurmountable if the S-F technique is implemented in a gas-dynamic solver for structured grids. However, the development of S-F techniques for unstructured grids has shown that most of these difficulties can be removed by having suitable software procedures capable of inserting line/surface grids within a background grid and subsequently moving these lines/surfaces inside the mesh. For some years now, a collaboration has been underway between three different universities for the development of a 3D S-F solver for unstructured grids that uses the GRUMMP library for the mesh generation and handling [175, 172]. This collaboration produced a first S-F solver capable of fitting single shock surfaces whose qualitative and quantitative results are shown in this Chapter. Efforts must be still concentrated on the development of an improved version of the unstructured S-F solver capable of dealing with 3D flows with interacting shock waves since it would represent a significant step in the development of S-F techniques for unstructured grids and could crack the barrier that has hitherto precluded the wide use of S-F methods in the world of CFD.

Chapter 6

SESF for 2D high speed flows

Contents

One of the aims of this Thesis was the development of a S-F technique able to overcome the limits of the previously designed S-F approaches for structured solvers: the boundary S-F and the floating S-F (see Sec. 2.2 for details). A good starting point for achieving this task was the extrapolated DIscontinuity Tracking (eDIT) technique developed by Ciallella et al. [32] for unstructured grids, which combines the unstructured-grid S-F technique developed in Ref. [22] with the shifted boundary method in Ref. [31]. In eDIT [32, 33] the fitted (or tracked) discontinuity carves a mesh-less hole in the computational domain over which it is floating and data transfer between the discontinuity and the boundaries of the hole is carried out through interpolation/extrapolation via truncated Taylor series expansions. This approach allows to retain high-order convergence properties without imposing any constraint on the topology of the mesh, and on the data structure of the underlying flow solver and without requiring complex mesh operations other than blanking the cells crossed by the discontinuity. The eDIT's features lend themselves well also to the structured-grid framework. Using this feature, it is proposed a new S-F approach called *Structured Extrapolated S-F* (*SESF*), that will be described and discussed in this Chapter. One of the key features of SESF is its capability to keep track of the discontinuities and compute the surrounding smooth-flow areas without re-meshing around the discontinuity; by doing so, *SESF* breaks the tight link between grid topology and shock topology which troubled for many years the development of S-F methods for structured grids. Moreover, a very limited amount of extra coding is required to couple SESF with virtually any existing structured-grid CFD code as long as the latter is capable of handling cell blanking, which amounts to disable the computation within a given set of cells. CFD codes dealing with

overset meshes typically posses this capability. Taking advantage of cell-blanking, the CFD code can be seen as a black-box by the *SESF* algorithm, which only takes care of supplying an input solution and the list of blanked cells. This algorithmic flexibility greatly simplifies the coding effort compared to the traditional floating S-F technique, while retaining the capability to handle flows featuring complex shock patterns. Therefore, the *SESF* technique can be considered as an important achievement because structured-grid solvers are the ones that can take the greatest advantage of S-F modeling. Indeed, this class of solvers is still very much in use today for simulating turbulent and aero-acoustic flows (via DNS or LES) because they are computationally more efficient and accurate than the unstructured-grid ones. Therefore, these simulations could really benefit from a more accurate and efficient shock-modeling, free of all those problems associated with the shock-capturing process.

6.1 *SESF* algorithm for 2D flows

To illustrate the algorithmic features of the SESF (see Fig.6.1), a 2D computational domain and a compressible flow-field featuring shocks and contact discontinuities are considered. For the purpose of illustrating the algorithm, reference will be made to a shock-wave, but contact-discontinuities can also be handled by using the appropriate jump relations and a simulation involving a contact discontinuity will be presented in Sect. 6.2.3. The example of Fig. 6.2a shows the Cartesian grid used to simulate a regular shock reflection from a flat plate. Firstly, a S-C calculation is run, since it supplies the initial solution and approximate shock location to the SESF simulation. SESF makes use of both a *background* mesh, which coincides with the grid used in the S-C calculation, and a *shock-mesh*. In 2D the shock-mesh is made up of a collection of points, the *shock-points*, marked with triangles in Fig. 6.2a, which are mutually joined to form an ordered sequence of connected straight segments, the *shock-edges*. In 3D, see the next Chapter, the shock-mesh is instead a triangulated surface. Regardless of the dimensionality of the physical space, a single set of dependent variables is associated with each cell of the background mesh, whereas two sets of values, corresponding to the upstream and downstream states, are assigned to each shock-point. The initial shape of the shock-front is obtained by post-processing the S-C solution by means of an automatic shock-detection technique, such as the ones described in [27, 164] or [36]. Contact discontinuities can also be detected by choosing the appropriate sensor, which for instance can be based on entropy or vorticity. When dealing with steady flows, it is not important that the initial shock-shape and the initial values of the dependent variables within the shock-points are accurately computed, because by enforcing the R-H jump relation across each pair of shock-points the algorithm is capable of driving the shock towards its steady-state location. The flowchart in Fig. 6.1 shows the seven algorithmic steps that allow to advance the solution and shock position from time t to $t + \Delta t$. By assuming the the dependent variables knowledge that at time t within all cells of the background mesh and all shock-points of the shock-mesh. A detailed description of each step will be given in the following paragraphs.

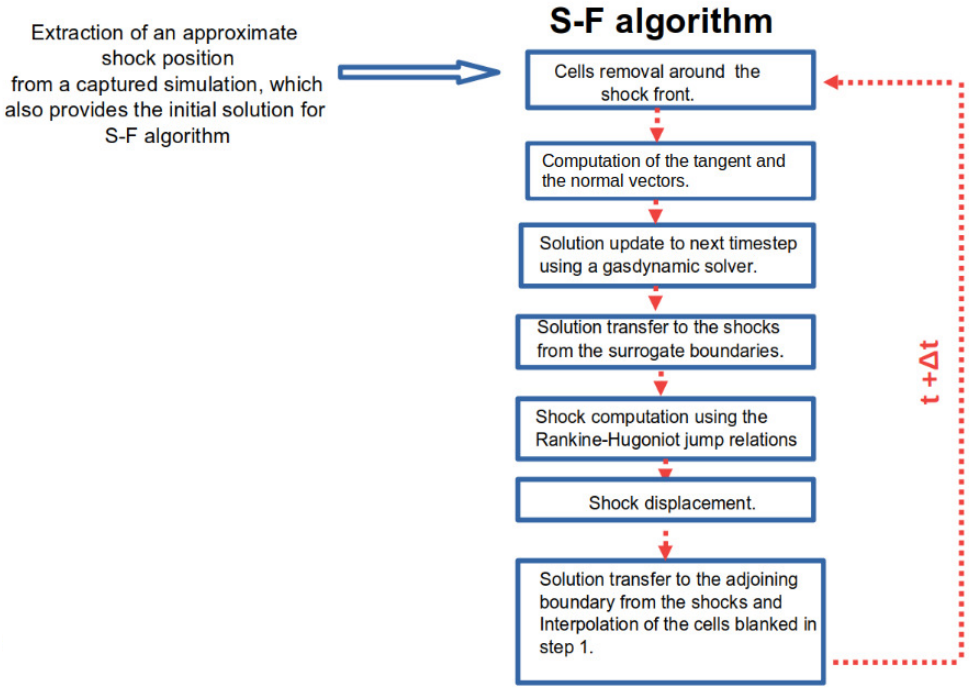
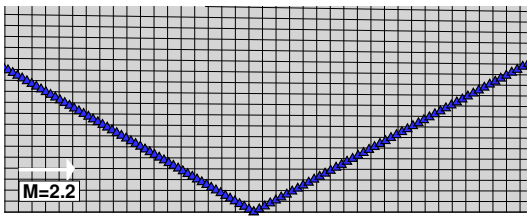


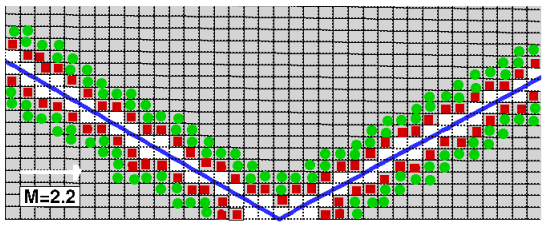
Figure 6.1. SESF algorithm flowchart

Step 1: Cell removal around the shock-front

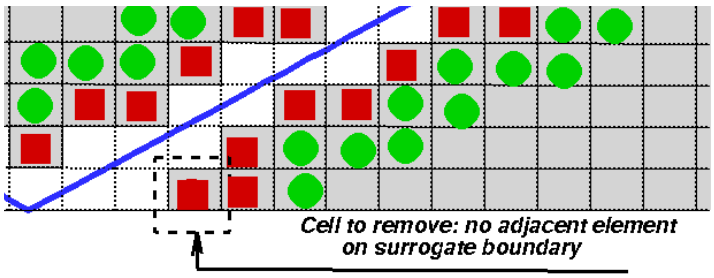
The first step consists in removing all those cells of the background mesh that are crossed by the discontinuity. More precisely, the algorithm checks for intersections between the edges of the background mesh and the shock-edges. Wherever this occurs, the two cells sharing the edge are blanked. By doing so, a hole enclosing the discontinuity is carved within the background mesh, as shown in Fig. 6.2b. The background grid with the blanked cells removed will be hereafter referred as *computational mesh* . Figure 6.2b also points out two important features of the proposed SESF technique: *i*) the cells marked by red squares represent the *adjoining boundary*, which is the collection of cells that have at least an adjacent blanked cell, and *ii*) the cells marked with green circles form the *surrogate boundary*, which is made up of those cells sharing at least one vertex with the cells on the adjoining boundary. Some further geometrical processing might be required to build the computational mesh from the background one. This is because, to be able to perform the extrapolation described in steps 5 and 7, the cells on the adjoining boundary need to be adjacent to at least one cell on the surrogate boundary. This may not be the case, however, in regions where different shocks mutually interact or a shock reaches a boundary: for example, the red cell highlighted by the dashed line in Fig. 6.2c has no neighbours belonging to the surrogate boundary and is therefore flagged for removal.



(a) Regular reflection on a plate: background Cartesian grid and shocks (marked by triangular points).



(b) Regular reflection on a plate: an example of computational mesh definition.



(c) Regular reflection on a plate: check on cavity boundary.

Figure 6.2. Regular reflection on a flat plate (M=2.2): creation of the computational grid.

Step 2: Computation of tangent and normal vectors

The procedure to compute both the tangent and normal vectors in each discontinuity point is the same described in Step 3 of the *UnDiFi-2D* algorithm, therefore please refer to Sec. 4.2.

Step 3: Solution update to time $t + \Delta t$ using a gas-dynamic solver

The solution within the computational mesh is advanced in time from t to $t + \Delta t$ using a CFD solver, without imposing any boundary condition on the adjoining boundary. In the present algorithm, the gas-dynamic code AFFS introduced in Sec. 2.1.1 and described in Ref. [69, 176, 70]) was used: it is a second-order-accurate FV solver based on Godunov's scheme, capable of simulating inviscid and viscous 2D/3D flows using structured, multi-block meshes. The choice of the structured-grid CFD solver is independent of the *SESF* algorithm described in this Section, as long as it is capable of dealing with blanked cells.

Step 4: Solution transfer from the surrogate boundaries to the discontinuities

The first transfer performed by the algorithm is required to update both the shock-upstream and shock-downstream values of the dependent variables at all shock-points. It is important to underline that data transfer towards the discontinuities is different depending on whether the upstream or the downstream side of the shock

is considered. The shock-upstream surrogate boundary behaves like a supersonic outflow: even though no boundary conditions are applied, the update to time $t + \Delta t$ performed by the gas-dynamic solver is correct and, therefore, extrapolation of all dependent variables from the shock-upstream surrogate boundary towards the shock-upstream side of the shock-mesh is used. On the shock-downstream surrogate boundary, however, three (in the 2D space) out of the four characteristic variables are convected downstream and away from the shock and only the Riemann variable (6.1) associated with the slow acoustic wave:

$$R_d{}^{t+\Delta t} = \tilde{a}_d^{t+\Delta t} + \frac{\gamma - 1}{2} \tilde{\mathbf{u}}_d^{t+\Delta t} \cdot \mathbf{n} \tag{6.1}$$

moves upstream towards the shock. The shock-downstream surrogate boundary behaves like a subsonic inflow boundary and three boundary conditions would be required. Since no boundary conditions are applied along the surrogate boundaries, the updated values computed by the CFD solver at time $t + \Delta t$ within the cells belonging to the shock-downstream surrogate boundary are incorrect (or provisional, hence the tilde in Eq. (6.1)), because of the three missing boundary conditions. However, due to the upwind nature of the discretization used in the CFD code, the Riemann variable (6.1), whose domain of dependence lies within the shock-downstream region, is correctly computed by the CFD code, even though the individual quantities $\tilde{a}_d^{t+\Delta t}$ and $\tilde{\mathbf{u}}_d^{t+\Delta t}$ may be wrong. It follows that on the shock-downstream side only the Riemann variable (6.1) is extrapolated from the shock-downstream surrogate boundary towards the downstream side of the shock. The values of the dependent variables on the downstream side of the shock-points will be subsequently corrected by enforcing the R-H jump relations, as described in step 5. The extrapolation process for both transfers is based on a Taylor series expansion truncated to the second term. By reference to Fig. 6.3, it follows:

$$\phi_j = \phi_A + (\nabla\phi)_A \cdot (\mathbf{r}_j - \mathbf{r}_A) \tag{6.2}$$

to compute the dependent variable, ϕ_j, in shock-point j using the cell-averaged value, ϕ_A, of a reference cell A that belongs to the surrogate boundary. In Eq. (6.2) $\mathbf{r}_A$ is the position vector of the centroid of cell A and $\nabla\phi$ is the gradient computed in cell A; to achieve second-order of accuracy in the calculation of ϕ_j, the approximation of the gradient in Eq. (6.2) only needs to be consistent, i.e. first-order accurate. In order to select the reference cell, firstly the cell on the adjoining boundary is identified as the closer element to shock-point j; this is cell 4 in the example of Fig. 6.3. Then, cell A is chosen among the cells sharing a vertex with cell 4, which also belong to the surrogate boundary; when multiple choices for A are available, the cell that is closer to the line passing through shock-point j and parallel to the shock-normal vector $\mathbf{n}_j$ is selected. Finally, gradient reconstruction in cell A is performed by means of the cell-based Green-Gauss formula, which reads:

$$\nabla\phi = \frac{1}{\Omega_A} \sum_{i=1}^{4} \phi_{e_i} \, \ell_i \, \mathbf{n}_i \tag{6.3}$$

In Eq. (6.3) the summation ranges over the four quadrilateral cells neighbouring cell A, $\mathbf{n}_i$ is the unit vector normal to the edge shared by cells A and i, and ϕ_{e_i}

the solution at the edge, which is computed using the arithmetic mean of the cell-averaged values of cells A and i:

$$\phi_{e_i} = \frac{\phi_A + \phi_i}{2}. \tag{6.4}$$

Whenever the i-th cell in Eq. (6.4) lies on the adjoining boundary, its ϕ_i value must be replaced using an alternative extrapolation, because the cells on the adjoining boundary will only be updated in step 7. This is the case, for instance, of cells 3 and 4 in Fig. 6.3 in which case ϕ_3 is extrapolated along the x-axis using values ϕ_A and ϕ_2, and ϕ_4 along the y-axis using ϕ_A and ϕ_1. In both cases the interpolation involves the cell-averaged value in A.

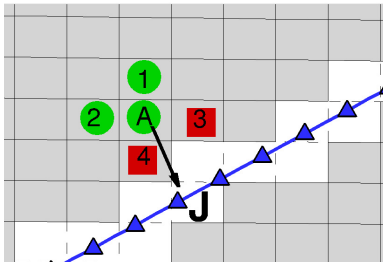


Figure 6.3. Extrapolation from the surrogate boundary to shock-point j.

In order to analyze the main properties of the gradient reconstruction, a grid-convergence analysis has been carried out and reported in the Appendix B 11.2.

Step 5: Shock computation enforcing the Rankine-Hugoniot jump relations

As stated before, the shock-downstream values of the dependent variables within the shock-points need to be corrected by enforcing the R-H relations. This amounts to solve a system of five (in the 2D space) non-linear algebraic equations for each shock point, which also supplies the local shock-speed, w_s. Specifically, the R-H jump relations are applied to all the shock points as previously described in step 5 of the *UnDiFi-2D* algorithm (please refer to Sec. 4.2 for further details).

Step 6: Shock displacement

The new shock position at time $t + \Delta t$ is obtained by displacing all shock-points using the local shock speed, w_s, computed in step 5 and the shock-normal unit vector, $\mathbf{n}$, computed in step 2. Also in this case, the same procedure for displacing shock points in step 6 of the *UnDiFi-2D* algorithm has been adopted, so that the reader is invited to refer to Sec. 4.2 for further details.

Step 7: Solution transfer from the shocks to the adjoining boundary

Once the dependent variables on the shock-downstream side of the shock-mesh have been correctly updated by enforcing the R-H jump relations, they can be used to update the cell-averaged values of those cells that belong to the adjoining boundaries,

as shown in Fig. 6.4. Once again, a generic cell A on the adjoining boundary is updated by means of a Taylor series expansion truncated to the second term using the closest shock-point, j, which reads:

$$\phi_A = \phi_j + (\nabla\phi)_j \cdot (\mathbf{r}_A - \mathbf{r}_j) \tag{6.5}$$

Although Eq. (6.5) looks specular to Eq. (6.2), it is important to highlight some

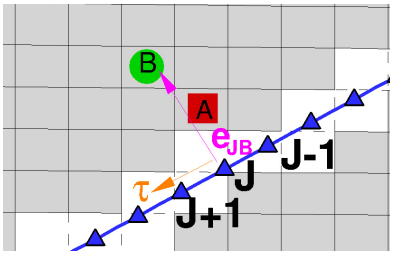


Figure 6.4. Adjoining boundary update using data computed in step 4 along the shock-downstream side of the shock-mesh

.

differences with respect to the data transfer described in step 4. First of all, in the current step data transfer only takes place on the downstream side of the shock, because the upstream side has already been updated in step 3. Secondly, the gradient reconstruction in shock-point j does not rely on the Green-Gauss formula, but it is instead computed along two (not mutually orthogonal) directions, $\boldsymbol{\tau}$ and $\mathbf{e}_{JB}$, shown in Fig. 6.4. The unit vector $\boldsymbol{\tau}$ is tangent to the discontinuity and has already been computed in step 2. The unit vector $\mathbf{e}_{JB}$ is parallel to the direction linking shock-point j to one of the cells on the surrogate boundary sharing a vertex with cell A. Also in this case, when multiple choices are available, the algorithm considers the cell that is closer to the line passing through shock-point j and parallel to the shock-normal vector $\mathbf{n}_j$; in the example shown in Fig. 6.4 this is cell B. The component of the gradient along the discontinuity uses a stencil of shock-points that belong to the range of influence of shock-point j:

$$\nabla\phi\cdot\boldsymbol{\tau} = \begin{cases} \frac{\phi_{j+1}-\phi_{j-1}}{|\Delta\mathbf{r}_{j+1/2}+\Delta\mathbf{r}_{j-1/2}|} & \text{if both } j+1 \text{ and } j-1 \text{ are in the range of influence of j} \\ \frac{\phi_j-\phi_{j-1}}{|\Delta\mathbf{r}_{j-1/2}|} & \text{if only } j-1 \text{ is in the range of influence of } j \\ \frac{\phi_{j+1}-\phi_j}{|\Delta\mathbf{r}_{j+1/2}|} & \text{if only } j+1 \text{ is in the range of influence of } j \end{cases} \tag{6.6}$$

The gradient along $\mathbf{e}_{JB}$, is instead calculated using the following finite difference formula:

$$\nabla\phi \cdot \mathbf{e}_{JB} = \frac{\phi_B - \phi_J}{|\mathbf{r}_B - \mathbf{r}_j|}. \tag{6.7}$$

Finally, the yet unknown Cartesian components of the gradient are obtained by solving a 2×2 linear system using the two known components computed using Eqs. (6.6) and (6.7).

A similar procedure has been used for updating the blanked cells. This is necessary because, as long as the shock keeps moving, those cells that have been

blanked during the current time step may be re-inserted into the computational mesh at a later time when they are not any longer crossed by the shock-mesh.

Upon completion of step 7, the dependent variables of all cells of the background mesh and of all shock-points of the shock-mesh have been updated to time $t + \Delta t$, the computational mesh can be discarded and the algorithm is restarted from step 1.

6.2 Numerical results

In the following sections the results obtained for three inviscid simulations of high-speed flows using both the S-C and SESF approaches will be analyzed and discussed. All numerical simulations have been run using the in-house S-C code described in Step 3, which has also been used in the smooth regions of the flow-field in the SESF simulations. The comparison of the results allows to highlight the differences between these two shock-modeling options by studying solution quality and order-of-convergence properties.

6.2.1 Circular cylinder

The first test-case consists in the supersonic flow ($M_\infty = 4$) past a circular cylinder. Despite its simplicity, it represents an important benchmark for the proposed technique, because the presence of a not uniform subsonic region around the stagnation point and the transition to supersonic flow across the sonic line may be particularly challenging for the extrapolation procedures employed in the SESF algorithm. This test-case has been simulated using three nested structured grids obtained by recursively coarsening the finest one to evaluate the qualitative and quantitative convergence properties of both the S-C and SESF techniques. The advantages of the SESF technique are clearly highlighted by Fig. 6.5, which shows the density flow-field computed using both the S-C and SESF approaches on all grid levels: the fitted bow shock is shown using a pink solid line in all three SESF solutions. By looking at the computed flow-fields, it is clear that the S-C solution on the coarsest grid is plagued by the carbuncle phenomenon [6], a numerical instability that affects captured shock waves and is probably due to the unphysical states that arise inside the captured shock [5, 177]. By modeling the bow shock as a true discontinuity, the SESF technique is not affected by this kind of numerical anomaly and provides high quality solutions even on coarse grids. Moreover, it can be seen that on all grid levels SESF provides a much cleaner density distribution within the shock-layer, compared to S-C.

A grid-convergence analysis of the two shock-modeling techniques is also presented by analyzing the global total temperature error. Indeed, since the flow is steady and the free-stream flow uniform, total temperature, T_0, should be preserved throughout the whole computational domain. Table 5 presents the L_1 norm of the global error ϵ computed on all pairs of grid levels according to Eq. 3.5 and the corresponding order-of-convergence, calculated as in Ref. [144]:

$$\tilde{n}_{i,i+1} = \frac{\log(\epsilon_i/\epsilon_{i+1})}{\log(r)} \tag{6.8}$$

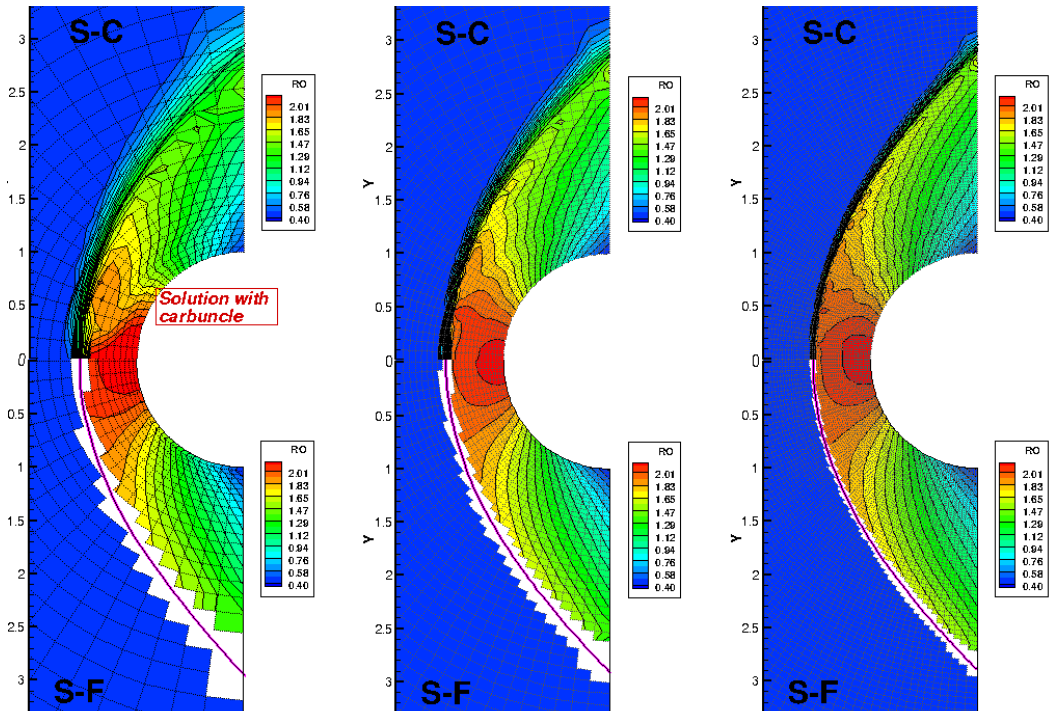


Figure 6.5. Supersonic flow past a circular cylinder ($M = 4$): comparison between the density flow-field computed on all the grid levels using the two different shock-modeling approaches.

In Eq. (6.8) the subscript $i = 0, 1, 2$ refers to the coarse, intermediate and fine grids, respectively, and the grid-refinement-ratio is constant and equal to $r = 2$. The computation of the global discretization error involves a summation over all cells of the computational mesh, i.e. excluding the blanked cells. This has been done not only for the SESF solution, but also for the S-C one because the hypotheses leading from a truncated Taylor series expansion to Eq. (6.8) do not hold close to a discontinuity. By looking at the data collected in Tab. 5 and shown in Fig.6.6, it can be seen that the numerical error of the S-C calculation is larger than that obtained with SESF on the same grid level and, maybe more importantly, the SESF results are characterized by a second-order convergence trend, whereas S-C drops slightly below first order.

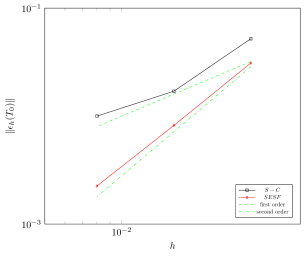


Figure 6.6. Convergence trend of L_1 norm.

Table 5. Supersonic flow past a circular cylinder: convergence analysis.

Mesh		SESF		S-C	
Grid level	h	L_1	$\tilde{n}$	L_1	$\tilde{n}$
0	0.032	$3.10 \cdot 10^{-2}$	-	$8.61 \cdot 10^{-2}$	-
1	0.016	$8.2 \cdot 10^{-3}$	1.91	$1.7 \cdot 10^{-2}$	-
2	0.008	$2.25 \cdot 10^{-3}$	1.86	$1.0 \cdot 10^{-2}$	0.96

A local grid convergence analysis has also been performed at stagnation point, where pressure and density, beside total temperature, can be analytically computed. This is because the bow shock is a normal shock at the point where it is crossed by the stagnation streamline. Using the results provided by both the SESF technique and the S-C approach on all three grid levels, and listed in Tab. 6, it is possible to estimate the observed order of accuracy [144, 152, 21], $\tilde{n}$, which is computed according to Eq. 3.1. Furthermore, the analysis can be further detailed by relying on the Grid Convergence Index (GCI) and the Richardson Extrapolated (RE) value $\tilde{u}_{RE}$, which are computed according respectively Eq. 3.3 and Eq. 3.4. For the finest grid level, GCI can be therefore evaluated as follows [144]:

$$\mathrm{GCI}_2 = 3\,\frac{|u_2 - u_1|}{1 - r^p}. \tag{6.9}$$

where p is the formal order of accuracy equal to 2. Specifically, GCI values referred to the coarser grid levels, can be recursively computed from the GCI on the finest level according to Eq. 6.10, as defined in Ref. [144]:

$$\mathrm{GCI}_i = r^p\,\mathrm{GCI}_{i+1} \quad \text{for } i = 0, 1 \tag{6.10}$$

In particular, GCI values, which play the role of a numerical uncertainty, are reported in Tab. 6 with respect to each simulated value (simulated value ± GCI). Furthermore, this table collects also the measured order of accuracy, the Richardson Extrapolated solution and the analytical values referred to the normalized pressure, density and total temperature. It is important to notice that the calculation of both the observed order of accuracy and the RE values cannot be carried out for the normalized pressure and density computed by the S-C method, because of their non monotonic convergence. This trend is due to the carbuncle which plagues the coarsest grid solution, especially regarding the pressure and density flow-field.
By comparing the data in Tab. 6, it is evident that the agreement between the SESF results on the finest grid and the analytical values is excellent and, for all three quantities, the exact solution falls within the uncertainty band defined by the GCI. As can be seen, the GCI data in Tab. 6 show that all GCI values are significantly smaller for the SESF solutions than the corresponding ones for S-C. This feature is particularly relevant when considering the solutions on the finest grid level, where for instance, the ratio between the S-F and S-C GCI values regarding the normalized total temperature is one-seventh. Table 6 also shows that the measured order of the SESF technique is close to design (second) order, which allows to reliably use the RE value as a close approximation of the corresponding analytical quantity, which is indeed confirmed when comparing values in the two bottom rows of Tab. 6. On the contrary, S-C observed order of accuracy computed with respect to the total temperature values is close to first order, as highlighted also by the global convergence analysis previously discussed.

6.2.2 Regular shock reflection

The proposed SESF technique has also been used to compute flows featuring shock-wall interactions, which is the case, for example, of the regular reflection of an

Table 6. Supersonic flow past a circular cylinder: local grid-convergence study.

	SESF			S-C		
Grid	P/P_∞	ρ/ρ_∞	T_0/T	P/P_∞	ρ/ρ_∞	T_0/T
coarse	20.669± 0.301	4.971± 0.088	4.156± 0.013	22.719± 1.25	5.451± 0.419	4.168± 0.024
intermediate	20.74± 0.075	4.992± 0.022	4.153± 0.003	20.662± 0.313	4.961± 0.105	4.159± 0.013
fine	20.759± 0.019	4.997±0.005	4.152± 0.001	20.741± 0.078	4.987±0.026	4.155± 0.007
$\tilde{n}$	1.93	1.91	1.86	-	-	0.94
$\tilde{u}_{RE}$	20.767	5.002	4.151	-	-	4.149
analytical values	20.765	5.000	4.152	20.765	5.000	4.152

oblique shock that impinges on a flat plate. In the chosen flow configuration the Mach number ahead of the incident shock is $M_\infty = 2.2$ and the flow undergoes a deflection $\alpha = 3.75°$ through the incident shock. Despite the simplicity of this test-case, which is characterized by three regions of uniform flow properties bounded by the incident and reflected shocks, it is impossible to recover the exact solution by modeling the shock-waves using S-C.

The computational grid is an H-grid made of 200×80 cells evenly spaced along each coordinate line in both directions, as shown in Fig. 6.7.

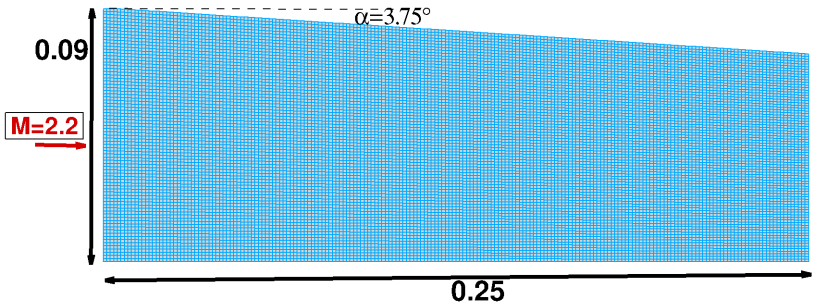
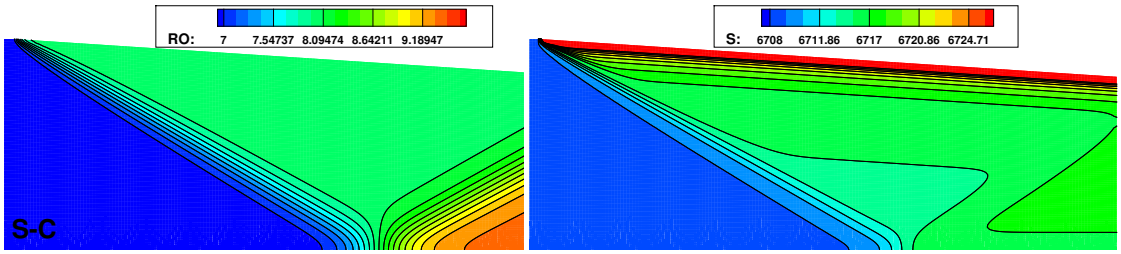


Figure 6.7. Regular reflection: free-stream conditions, domain and background grid (200×80 cells).

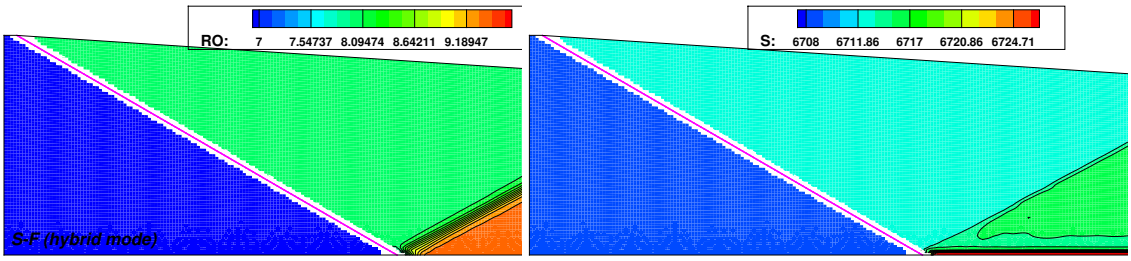
As already shown in previous publications [23, 36, 109, 33], S-F simulations can be performed in two different ways: fully-fitted or hybrid. In fully-fitted mode all discontinuities are fitted and the point(s) where different discontinuities mutually interact or interact with a solid boundary must also be modeled. In hybrid mode only some of the discontinuities are fitted and it is left to the S-C solver to capture all the other discontinuities as well as the interaction between fitted and captured discontinuities. Figures 6.8, 6.9 and 6.10 allow to compare the differences among S-C, hybrid and fully-fitted SESF. First of all, Fig. 6.8b clearly reveals that the corner of the upper wall, where the incident shock originates, is a large source of error in the S-C calculation. This anomaly is absent in the SESF simulations. Not surprisingly, solution quality improves as the number of fitted shocks increases. In the hybrid SESF simulation shown in Fig. 6.9 the incident shock has been fitted and the reflected one captured, which is sufficient to significantly reduce the (un-physical) width of the captured reflected shock with respect to the S-C calculation: compare Fig. 6.9a with 6.8a. As shown in Fig. 6.10a, solution quality further improves when

both shocks are fitted and their interaction point at the wall is modeled as described in [23]. In particular, the fully-fitted solution of Fig. 6.10a allows to recover the uniform, analytical solution, within the two regions downstream of the incident and reflected shocks. It may be argued that this is of little practical use given the simplicity of the shock-interaction pattern that is being analyzed. The test-case to be presented in Sect. 6.2.3 demonstrates that SESF outperforms S-C also when dealing with more complex shock-interaction topologies.



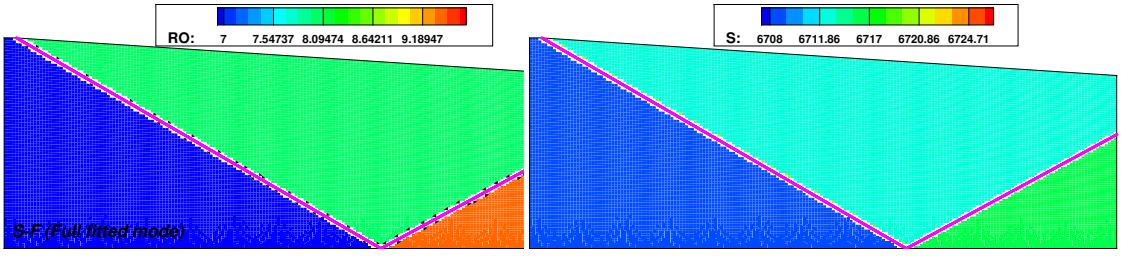
(a) Density flow-field. **(b)** Entropy flow-field.

Figure 6.8. Regular shock reflection: S-C computation.



(a) Density flow-field. **(b)** Entropy flow-field.

Figure 6.9. Regular shock reflection: hybrid S-F computations, fitted shock is depicted using pink solid line.



(a) Density flow-field. **(b)** Entropy flow-field.

Figure 6.10. Regular shock reflection: full S-F computation, fitted shocks are depicted using pink solid lines.

6.2.3 Mach reflection

This last test-case has been included in order to show that SESF is capable of dealing with complex shock-patterns, such as the triple-point that arises in a Mach reflection. The free-stream conditions and the computational domain are reported in Fig. 6.11: the computational mesh is a H-grid made of 232×102 cells evenly spaced, along each coordinate line, in both directions. The SESF simulation has been run in fully-fitted

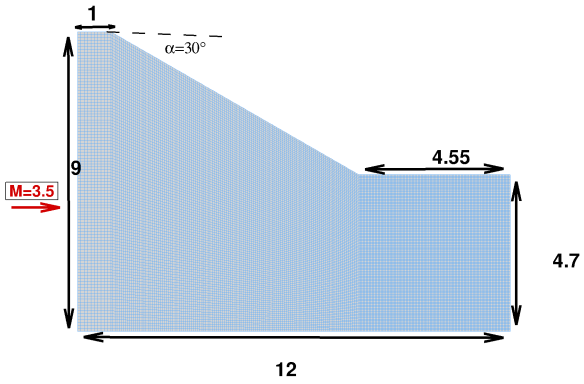
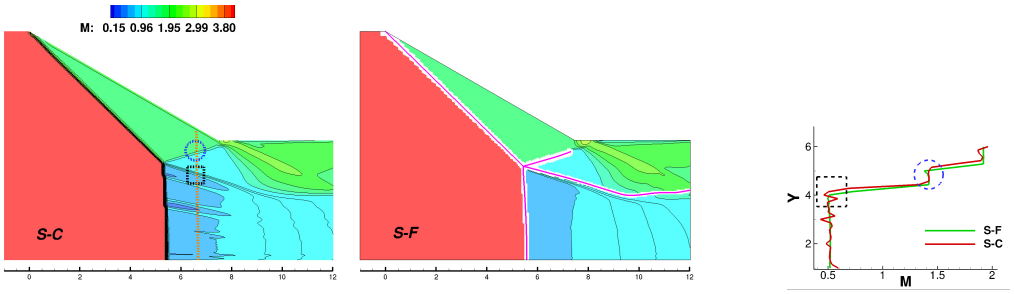


Figure 6.11. Mach reflection: free-stream conditions, domain and background grid (232×102 cells).

mode: all four discontinuities that meet at the triple-point, i.e. three shocks (the incident and reflected shocks and the Mach stem) and the contact discontinuity, have been fitted and the triple-point has been modelled as described in [178].

The Mach number flow-field computed with both S-C and SESF are displayed in Figs. 6.12a and 6.12b, respectively. It is clearly shown that the SESF solution exhibits a better quality with respect to the S-C simulation: the Mach-number distribution is smoother, especially downstream of the Mach stem and the reflected shock. This is evident when looking at the Mach distribution along the $x = 6.85$ line: Fig. 6.12c shows that the captured Mach stem gives rise to unphysical oscillations in the entire region downstream of the Mach stem, which are absent in the SESF calculation.



(a) S-C computation. **(b)** S-F computation. **(c)** Mach distribution along $x = 6.85$.

Figure 6.12. Mach reflection: Mach flow-field (fitted shocks are depicted using pink bold lines) and Mach distribution along the line $x = 6.85$ (orange dashes).

6.3 Summary

A new computational technique has been proposed for simulating high-speed compressible flows featuring discontinuous solutions (both shock-waves and contact discontinuities) using structured-grid solvers. The so-called Structured Extrapolated S-F (SESF), has been described in detail, highlighting its algorithmic features, novel features and its key advantages over the two S-F approaches, namely boundary and floating S-F, that had been developed for structured-grid solvers starting in the 1960s. SESF has been used to simulate three different inviscid 2D flows: the supersonic flow past a circular cylinder and both a regular and Mach reflection. The blunt-body flow has also been used to evaluate the local and global grid-convergence properties of the new technique, which shows an observed order-of-convergence very close to the theoretical one, both globally and at stagnation point. Moreover, the SESF solution is immune to the carbuncle phenomenon, that instead plagued the S-C solution on the coarsest mesh. Last but not least, the SESF algorithm has been designed with the aim of developing a technique that [165] "could be used as a black box in a variety of complicated problems". This is accomplished by treating as a black-box the structured-grid CFD solver which is used to discretize the governing PDEs in smooth regions of the flow-field, the only requirement being that the CFD code is capable of dealing with blanked cells. To summarize, the technique has been shown to be able to overcome typical problems encountered by S-C solvers (order-of-accuracy degradation and anomalous solutions) and to compute flows with shock-wall and shock-shock interaction, providing high quality solutions also on coarse meshes.

Chapter 7

SESF for 3D high speed flows

Contents

Over the last decades the simulations of compressible flows, especially those featured by shocks, have been one of the major driver for developing new computational algorithms and tools. However, the development of S-F techniques for computing 3D flows was held back by a main obstacle: the insertion and handling of shock surfaces mutually interacting within the background volume mesh. Even if these problems have been partially overcome using the S-F technique described in Chapter 5, S-F is still not able to deal with interacting discontinuities. It is worth to point out that these problems become even more complicated when structured grids are considered, because of their rigid topology. In these cases, it is not possible to directly transport the S-F technique described in Chapter 5, since it is strongly linked to the world of unstructured grids: the reason lies in the libraries used by the *UnDiFi-3D* code which are tailored for generating these types of grids. An alternative is proposed by the embedded approach for computing 2D flows on structured grids described in the Chapter 6, that allows the calculation of the shock-wave and the adjacent flow areas without the need of regenerating a computational grid around the shock. The high quality results obtained in Sec. 6.2 by the SESF techniques for computing 2D complex flows proves that it is possible to open a new path in the S-F techniques development and to break the strong link between meshes and solvers that has existed so far in all the S-F methods developed in the past. The extension of the *SESF* technique to 3D flows is therefore an obliged step of this new development. In this Chapter, the newly developed *SESF* technique for 3D flows and some of its applications will be presented: specifically, a particular attention will be paid to describe the methodological developments with respect to the algorithm presented in Chapter 6 since, in the 3D space, shock fronts are modelled by zero-thickness, double-

sided, triangulated surfaces instead of poly-lines. It is important to underline that the proposed technique is capable of computing for the first time 3D flows featuring shock-shock and shock-wall interactions thanks to the coupling with the MMG mesh generator [179], which is able to stitch together multiple shock/wall/discontinuity surfaces at the reflection/interaction lines, which can move during the computation, and to eventually re-mesh the different surfaces. Furthermore, the results obtained using both S-C and S-F in the simulations of several significant flows will be compared in order to highlight the differences between these two shock-modelling options and to provide a quantitative assessment of the order-of-convergence of the two techniques will also be presented.

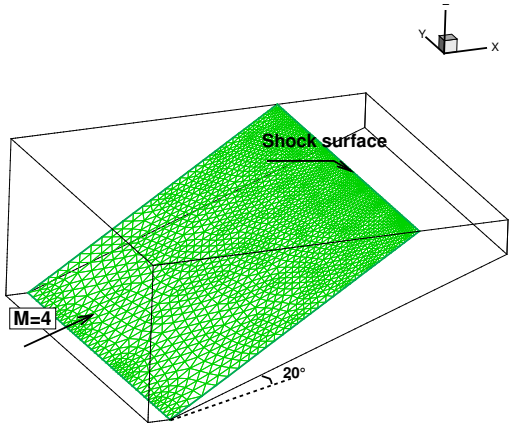
7.1 *SESF* algorithm for 3D flows

The algorithmic features of the $SESF$ technique for 2D applications have been deeply described in Chapter 6: the extension of the SESF technique to 3D cases requires some modifications especially regarding the extrapolation/interpolation processes between the shock surface and the background volume mesh. To illustrate the newly implemented features of the $SESF$, the algorithm steps listed in Fig. 6.1 will be described, by considering as an example the 3D supersonic flow past a wedge, as described in Figs. 7.1a and 7.1b. It is worth to notice that the S-F technique allows to represent the oblique shock originated at the compression corner as a triangulated surface, as pointed out by Fig. 7.1a. The initial shock position can be determined by evaluating a preliminary S-C calculation, which also provide the initial condition for the S-F computation: specifically, the extraction of a first attempt shock surface has been widely discussed in Chapter 5. Then, each shock front is evolved in time by the SESF algorithm until the correct steady position is reached: the procedure employed to evolve the solution from t to $t + \Delta t$ will be described in detail in the following sub-sections.

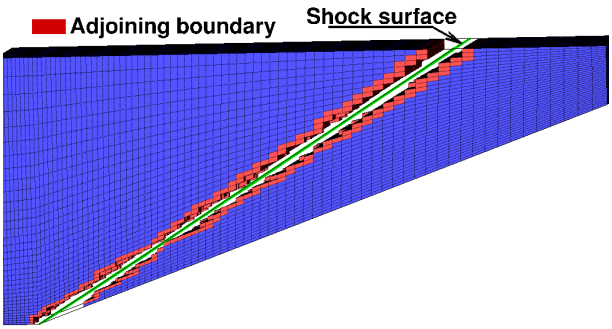
Step 1: Cell removal around the shock surface

The first step consists in removing all the elements of the volume mesh that are crossed by the shock surface: the removed elements of the background grid are identified as "ghost cells" whenever the distance d between one of their vertices and the nearest shock point P is lower than the characteristic grid size of the volume mesh. By doing so, a hole containing the shock front is carved within the background grid, as shown in Fig. 7.1b. Hereafter, the mesh without the blanked elements will be referred as *computational grid*: as in the 2D case, the collection of cells that have at least an adjacent blanked cell represent the *adjoining boundary*, as highlighted in Fig. 7.1b, whereas the *surrogate boundary* is made up of those cells sharing at least one vertex with the cells on the adjoining boundary. Some further geometrical processing might be required to build the computational mesh from the background one. As a matter of fact, to be able to perform the extrapolations described in steps 5 and 7, elements on the adjoining boundary require to be adjacent to at least one cell on the surrogate boundary. This may not be the case, however, in regions where different shocks mutually interact or whether a shock reaches a boundary:

for this reason, cavity boundaries need to be checked and those cells which have no neighbours belonging to the surrogate boundary are therefore flagged for removal.



(a) Shock surface and domain boundaries



(b) Computational grid example: cavity generation within the volume mesh.

Figure 7.1. Supersonic flow past a compression corner (M=4)

Step 2: Computation of tangent and normal vectors

In order to apply the R-H jump relations the normal $\mathbf{n}$ unit vector have to be calculated within each shock-point. This task is achieved by adopting the same procedure described in step 1 of the *UnDiFi-3D* algorithm (refer to Sec. 5.1 for details).

Step 3: Solution update to time $t + \Delta t$ using a gas-dynamic solver

The solution within the computational mesh is advanced in time from t to $t + \Delta t$ using the AFFS solver, which was coupled also with the 2D *SESF* technique, in 3D mode without imposing any boundary condition on the adjoining boundary.

Step 4: Solution transfer from the surrogate boundaries to the shock front

As in step 4 of Sec. 6.1, the first transfer performed by the algorithm is required to update both the shock-upstream and shock-downstream values of the dependent variables at all shock-points. It is important to underline that data transfer towards the discontinuities is different depending on whether the upstream or the downstream side of the shock is considered. The shock-upstream surrogate boundary behaves like a supersonic outflow: even though no boundary conditions are applied, the update to time $t + \Delta t$ performed by the gas-dynamic solver is correct and, therefore, extrapolation of all dependent variables from the shock-upstream surrogate boundary towards the shock-upstream side of the shock-mesh is used. On the contrary, the flow on the shock-downstream surrogate boundary has missing boundary conditions corresponding to the slow acoustic wave. For this reason, on the shock-upstream

side, all conserved variables are extrapolated from the surrogate boundary to the shock, whereas, on the shock-downstream side, only the Riemann variable in Eq. 6.1 is extrapolated.
The extrapolation process for both transfers is based on the truncated Taylor series expansion in Eq. 7.1. To provide an example, the updating process of shock-point J, according to the notation described in Fig. 7.2, is considered. Thus, the state vector associated to shock-point J, ϕ_J, is updated using the state stored in a reference element A, ϕ_A, which belongs to the surrogate boundary:

$$\phi_J = \phi_A + \nabla\phi \cdot \mathbf{d} \tag{7.1}$$

where $\mathbf{d}$ is the distance vector between shock-point J and cell A, and $\nabla\phi$ is the gradient computed on the correspondent surrogate boundary: it must be noticed, that Eq. 7.1 only needs to be consistent, i.e. first-order accurate. In order to select the reference cell, firstly the closest element to shock-point J is identified, so that it lies on the adjoining boundary (cell 3): in particular, the distance is computed with respect to the line corresponding to shock point J normal vector $\mathbf{n}_j$ (the dashed line in Fig. 7.2). Then, cell A is chosen among its adjacent elements which are also on the surrogate boundary. Specifically, when multiple choices for A are available, the algorithm takes the cell closer to shock point J normal vector. Finally, the gradient $\nabla\phi$ is evaluated by means of the cell-based Green-Gauss formula, which reads:

$$\nabla\phi = \frac{1}{Vol_A} \sum_{i=1}^{6} \Omega_{A_i}\, \phi_{f_i} \cdot n_i \tag{7.2}$$

where ϕ_{f_i} is the solution at the interface between cell A and the i-th adjacent element, and it is computed with an arithmetic mean between the two states:

$$\phi_{f_i} = \frac{\phi_A + \phi_i}{2} \tag{7.3}$$

if the i-th element lies on the adjoining boundary, its value must be replaced by a secondary extrapolation process, since those cells are not correctly updated: for example, in Fig. 7.2, the state of cell 3 has to be extrapolated along the k coordinate lines using both cells A and 1.

Step 5: Shock computation enforcing the Rankine-Hugoniot jump relations

In this step, shock-downstream values of the dependent variables are corrected by enforcing the jump relations across each pair of shock-points according the same procedure described in step 3 of Sec. 5.1.

Step 6: Shock displacement

The new shock position at time $t + \Delta t$ is obtained by displacing all shock-points using the local shock speed, w_s, computed in step 5 and the shock-normal unit vector, $\mathbf{n}$, computed in step 2, as described by step 4 of Sec. 5.1.

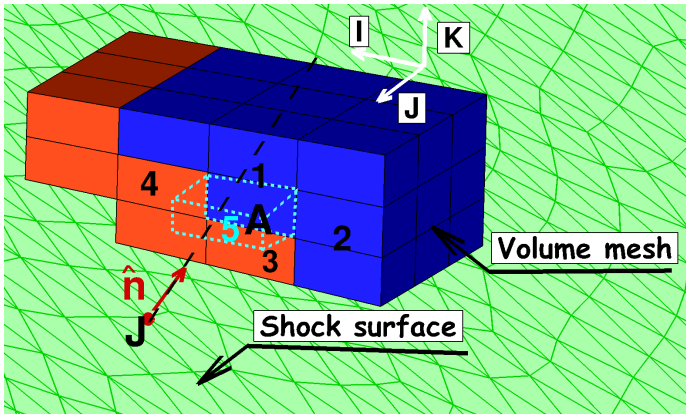


Figure 7.2. Solution transfer from the surrogate boundary to the shock front

Step 7: Solution transfer from the shock surface to the adjoining boundary

Once the shock states are correctly computed by enforcing the R-H jump relations, they can be used to update the elements belonging to the adjoining boundaries, as shown in Fig. 7.3. Once again, cell A on the adjoining boundary is updated by means of a Taylor series expansion truncated to the second term using the closest shock-point J, which contains A within its dependence domain. It reads:

$$\phi_{\mathbf{A}} = \phi_{\mathbf{J}} + \nabla\phi_J \cdot \mathbf{d} \tag{7.4}$$

the gradient is computed on a reference frame defined on J, which consists of the shock-point normal vector $\mathbf{n}_J$ and of two tangents ($\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$) to the shock surface in J, as described in Fig. 7.3.

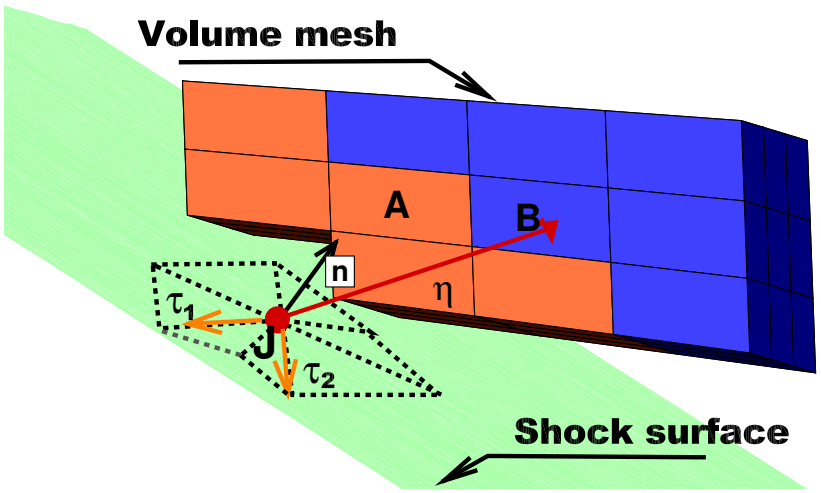


Figure 7.3. Solution transfer from the shock front to the adjoining boundary

In particular, the gradient $\nabla\phi_J$ in shock point J can be split into two components:

$$\nabla\phi = \nabla_{\mathbf{sh}}\phi_{\mathbf{J}} + \nabla_{\mathbf{n}}\phi_J \tag{7.5}$$

Where $\nabla_{\mathbf{sh}}\phi_J$ is defined locally tangent to the shock surface, whereas the component $\nabla_{\mathbf{n}}\phi_J$ is parallel the shock point J normal vector $\mathbf{n}_J$.
At this stage, $\nabla_{\mathbf{sh}}\phi$ is obtained by means of an area-weighted average of the fractions of the cell-wise constant gradients of all the shock cells that surround the shock point J according to:

$$\nabla_{\mathbf{sh}}\phi_J = \frac{1}{\sum_{J\ni T} A_T} \sum_{J\ni T} A_T \, \nabla_{\mathbf{sh}}\phi^T \tag{7.6}$$

where A_T is the area of each triangle which has J among its vertices. Moreover, the gradient $\nabla_{\mathbf{sh}}\phi^T$ on the triangle T belonging to the shock surface is calculated using the values of ϕ stored in each vertex of the triangles T and the inward unit normals to the triangle edges as stated below:

$$\nabla_{\mathbf{sh}}\phi_j^T = \frac{1}{2 \cdot A_T} \sum_{i=1}^{3} \phi_j \, \mathbf{n}_i \tag{7.7}$$

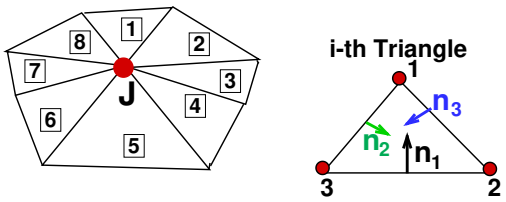


Figure 7.4. Gradient associated to the shock point J on the tangential directions.

At this point, the gradient normal component $\nabla_{\mathbf{n}}\phi$ is still an unknown. It can be evaluated using the derivative along the direction that links shock-point J to one of the adjacent elements of cell A (called B in Fig. 7.3) that belongs to the surrogate boundary, which is denoted by the unit vector $\boldsymbol{\eta}$. In particular, this is computed using the finite difference formula in Eq. 7.8, which allows to solve the scalar equation defined in Eq. 7.9, where the unknown term is marked in red.

$$\nabla\phi \cdot \eta = \frac{\delta\phi}{\delta\eta} = \frac{\phi_B - \phi_J}{d_{JB}} \tag{7.8}$$

$$\frac{\delta\phi}{\delta\eta} = (\nabla_{\mathbf{sh}}\phi \cdot \tau_{\mathbf{1}})(\tau_{\mathbf{1}} \cdot \boldsymbol{\eta}) + (\nabla_{sh}\phi \cdot \tau_{\mathbf{2}})(\tau_{\mathbf{2}} \cdot \boldsymbol{\eta}) + \nabla_{\mathbf{n}}\phi \cdot \boldsymbol{\eta} \tag{7.9}$$

Once the gradient reconstruction is completed, it is possible to compute the gradient vector components $\frac{\delta\phi}{\delta x}$, $\frac{\delta\phi}{\delta y}$, $\frac{\delta\phi}{\delta z}$ along the x, y and z axes, by solving the following system:

$$\begin{cases} \nabla\phi_{\tau_1}\tau_{1x} + \nabla\phi_{\tau_2}\tau_{2x} + \nabla\phi_n n_x = \frac{\delta\phi}{\delta x} \\ \nabla\phi_{\tau_1}\tau_{1y} + \nabla\phi_{\tau_2}\tau_{2y} + \nabla\phi_n n_y = \frac{\delta\phi}{\delta y} \\ \nabla\phi_{\tau_1}\tau_{1z} + \nabla\phi_{\tau_2}\tau_{2z} + \nabla\phi_n n_z = \frac{\delta\phi}{\delta z} \end{cases} \tag{7.10}$$

Where in particular:

$$\nabla\phi_{\tau_1} = \nabla_{\mathbf{sh}}\phi \cdot \tau_1$$

,

$$\nabla\phi_{\tau_1} = \nabla_{\mathbf{sh}}\phi \cdot \tau_1$$

and

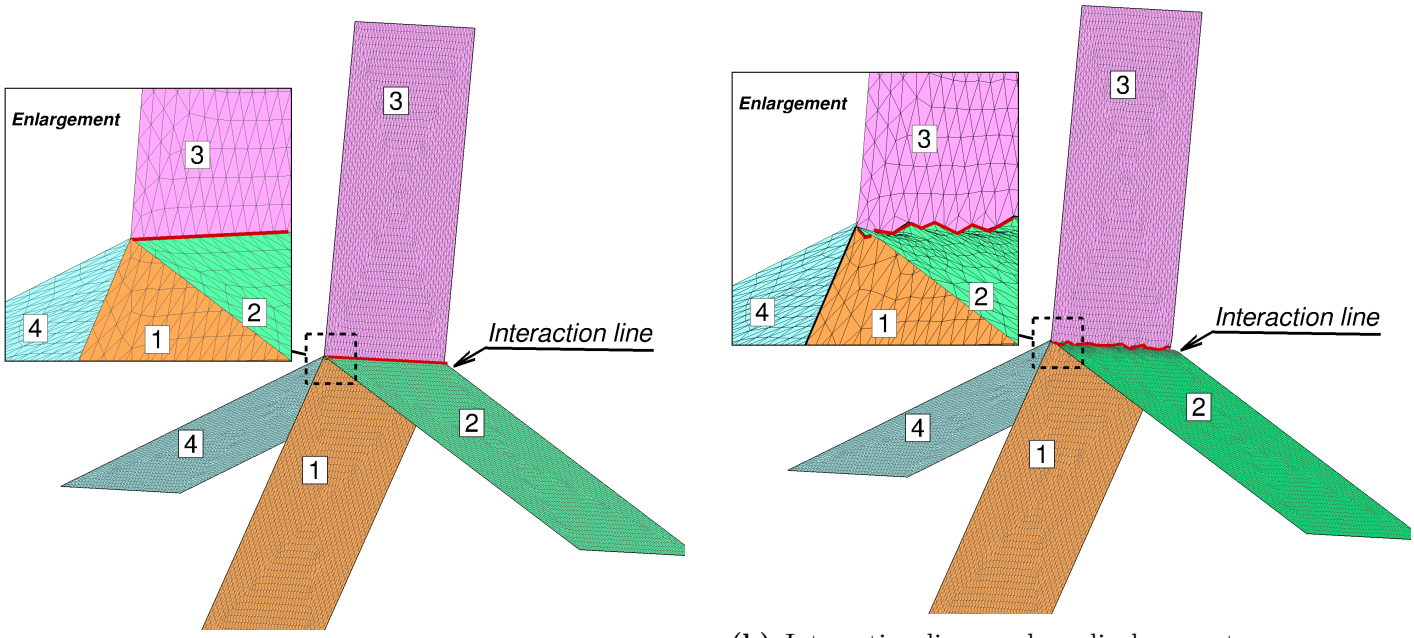
$$\nabla\phi_n = \nabla\phi \cdot \mathbf{n}$$

A similar procedure has been implemented also for blanked cells, so that all elements are updated at time $t + \Delta t$ and the algorithm can then restart from step 1.

7.1.1 Shock/shock interactions modelling

As mentioned earlier, in the past years the calculation of 3D flow configurations featuring interacting shocks using a full fitted approach, where all the shock surfaces and interaction lines are fitted, was hindered by the stumbling block of the generation and handling of multiple interacting shocks surfaces, which can move within the computational mesh. This obstacle has been easily overcome by coupling the SESF technique with the MMG mesh generator [179], which is able to stitch together multiple shock surfaces at the reflection/interaction lines, and to eventually re-mesh the shock fronts during their motion, until the final position is reached. To illustrate the important role of this software, now embedded in the SESF algorithm, let consider the four surfaces in Fig. 7.5a, that interact along the red line. This example can be seen as a prototype of a complex shock-shock interaction, such as a Mach reflection: in order to simulate what happens when the interaction line nodes are displaced, the interaction points have been randomly moved along the plane defined by the surface 1, as displayed in Fig. 7.5b. By doing so, the surfaces need to be re-meshed, without changing the node distribution along the interaction line. Fig. 7.5b shows that the MMG software is able to perform this task, by keeping stitched together the different shock fronts, therefore fulfilling the S-F technique requirements for computing 3D shock-shock interactions.

Moreover, when an interaction occurs within the flow-field, it is necessary to model the interaction line in order to determine the motion of these points. In the last decade, Paciorri and Bonfiglioli [23] have described how to model 2D interactions using an unstructured S-F technique , where shock-waves and discontinuity lines are represented by poly-lines and interactions are defined by points where the various discontinuities meet. Recently, the same interaction models have been used also in the *SESF* algorithm described in the previous Chapter to compute 2D interactions. The interaction models illustrated in Ref. [23] can be applied also to 3D compressible flows with some modifications: in 3D case, shock fronts are described using zero-thickness triangulated surfaces, so that their intersection results in a curve (see Fig. 7.6a), the points of which must be correctly modelled according to the interaction type. Specifically, the model described in Ref. [23] has been extended in 3D space, by drawing simple geometrical considerations. For instance, the case of a regular oblique shock reflection can be considered, which will be also analyzed in detail in Subsec. 7.2.2: the interaction line is highlighted by Fig. 7.6a, a detail of which is also shown by Fig. 7.6b. As can be seen, the interaction line consists of a list of special shock points: specifically, a generic special point IP belonging to this

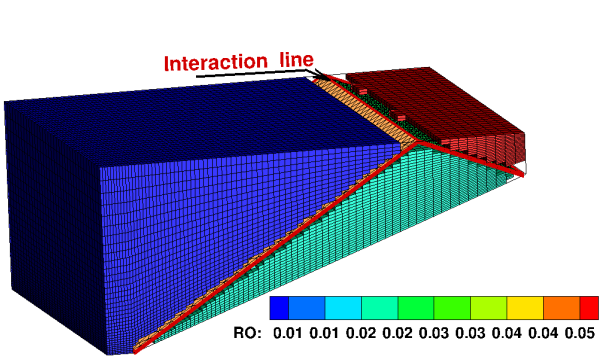


(a) Initial shocks surfaces

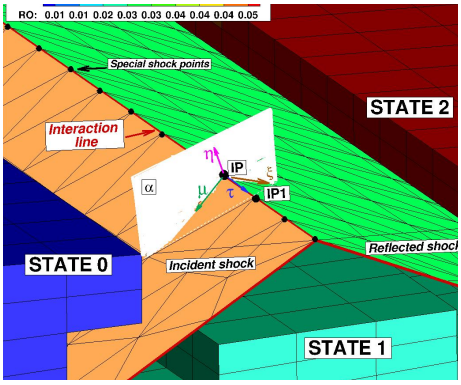
(b) Interaction line random displacement and shock surfaces remeshing

Figure 7.5. Shock-shock interaction example: role of the MMG software

line is computed on a plane α, which is locally orthogonal to the interaction line in IP. Once the velocity vectors are projected on this plane, it is possible to apply the same treatment proposed in Ref. [23] for the 2D shock-wall interaction case, which is solved on plane α as illustrated by the sketch in Fig. 7.7. In order to explain the reasons behind the plane α choice, it is worth to notice that this plane is normal to all the interacting shocks. As a matter of fact, let us consider the vectors τ and μ defined on Fig. 7.6b: the former is considered parallel to the vector tangent to the interaction curve in IP, whereas the latter is identified by the intersection between the incident shock surface and the plane α. The cross product between the two aforementioned vectors defines a vector η which is orthogonal to both μ and τ: in particular, since τ is also normal to plane α, it follows that η lies on plane α. At the same time, η is orthogonal to the incident shock surface, so it can be considered as the normal vector of the incident shock in IP. Similar considerations can be drawn for each special point and also for the reflected shock surface when considering the vector ξ instead of μ (see Fig. 7.6b), and therefore the plane α is locally normal to all the interacting shocks.
Because of this geometrical consideration, the solution of the shock-wall interaction on plane α as a 2D problem is correct: indeed, the R-H jump relations are applied along the shock normal vector, which lies on plane α, as proved before. Moreover, it should be noted that the aforementioned procedure can be applied to model all the special points analyzed in Ref. [23] for 2D shocks interactions. Thank to the MMG software and to the modelling of the interaction lines previously described, the *SESF* technique is capable of performing a full fitted computation of the regular reflection test-case discussed in Subsec. 7.2.2.



(a) S-F density flow-field



(b) Regular oblique shock reflection on a plate (M=4): interaction region detail.

Figure 7.6. Regular oblique shock reflection on a plate (M=4): interaction modelling.

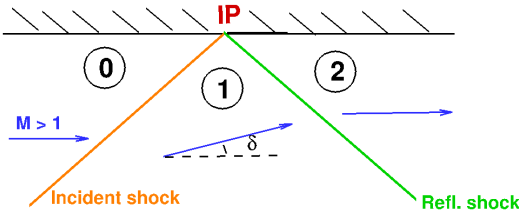


Figure 7.7. Regular oblique shock reflection on a plate: flow-field sketch

7.2 Numerical results

In the following sections the results obtained for inviscid simulations using both S-C and the SESF approach will be analyzed and discussed: it is worth to notice that all the numerical simulations have been performed using the in-house second-order S-C gasdynamic solver described in [69]: the same solver is also used in the smooth regions by the SESF technique. In particular, two different test cases will be proposed in the following subsections to highlight the differences between these two shock-modelling options in terms of solution quality and grid convergence properties: a supersonic flow past a circular cylinder and regular shock reflection on a plate.

7.2.1 Supersonic flow past a circular cylinder

In this section, a supersonic flow ($M_\infty = 3$) past a circular cylinder is taken into account: this flow configuration has been numerically computed on two nested volume meshes shown in Fig. 7.8, whose features are also reported in Table 7. Both meshes are hyperbolic structured grids, composed by evenly spaced points along the cylinder surface, whereas the nodes in the radial direction are distributed according to an exponential spacing varying from $\Delta_{min}S$ to $\Delta_{max}S$. In order to compare the solution quality provided by both approaches, it is worth to notice the differences between the density flow field computed on the coarse and fine grid levels, pointed out in Fig. 7.9a and 7.9b, with respect to the symmetry plane xz. For each grid,

Table 7. Nested structured grids features: number of nodes along the cylinder surface (I-lines), the radial direction (J-lines) and the cylinder axis (K-lines) and grid spacing size ΔS

Grid level	I	J	K	ΔS^I	ΔS^K	$\Delta_{min} S^J$	$\Delta_{max} S^J$
coarse	33	17	17	0.098	0.036	0.098	0.24
fine	65	33	33	0.046	0.018	0.046	0.12

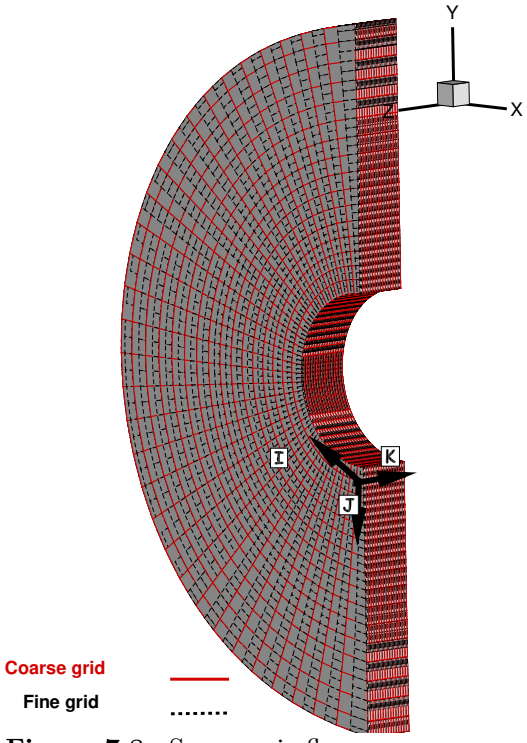
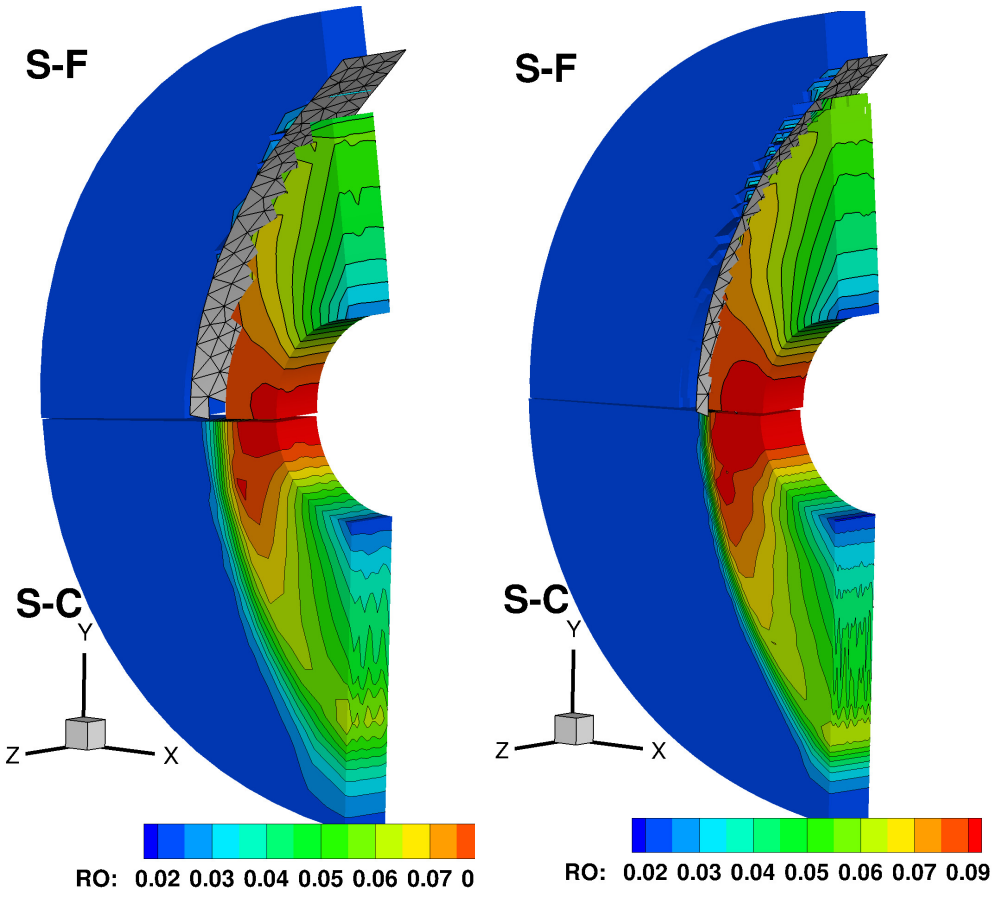


Figure 7.8. Supersonic flow past a cylinder (M=3): domain boundaries

the SESF solution is displayed in the upper side, whereas the S-C one is shown in the lower side. By fitting the bow shock, the spurious disturbances downstream the discontinuity are removed and the S-F solution exhibits a higher quality and a smoother density distribution within the shock layer. The availability of the exact solution allows to assess the grid convergence properties of the two methods. In particular, it is possible to evaluate the global error ϵ computed with respect to the total temperature T_0, since it should be constant in the whole computational domain and equal to its inlet value. Table 8 collects the discretization error L_1 norm computed on all grid levels and the measured order of convergence $\tilde{n}$ [144, 152, 21] evaluated according to Eq. 3.1. It is important to underline that all those cells crossed by the bow shock have been neglected for the global error measure: blanked cells are excluded from the computation of the discretization error for both S-C and S-F solutions. The convergence analysis confirms the results obtained in the 2D case. Specifically, in Table 8, the S-C discretization error is always larger than that obtained with SESF on the same grid level. Moreover, SESF computations are characterized by a convergence trend which is close to the designed order, $\tilde{n} = 2$, whereas S-C drops slightly below first order, as shown in Fig. 7.10. In order to analyze the 3D effects linked to this flow configuration, the nodes of the finest mesh level have been randomly displaced: in particular, those belonging to the cylinder have been only shifted along the z-axis to preserve the body surface (see Fig. 7.11a). Also in this case, numerical computations were run using both approaches: nevertheless,



(a) Coarse grid **(b)** Fine grid

Figure 7.9. Supersonic flow past a cylinder (M=3): density iso-countours

it was possible to perform only a first order accurate S-C simulation using this particular volume mesh (see Fig. 7.12a), since the second order solution featured a carbuncle, as highlighted in Fig. 7.12b. The origin of this numerical artefact appears to be related to the captured shocks thickness [5], due to the unphysical states that take place within the captured shock. By modelling shocks as discontinuities, S-F techniques avoid the presence of these unphysical states, so that they are immune to the carbuncle phenomenon.
Regarding the solution quality, which is possible to achieve using both the numerical approaches, the same considerations previously stated apply also in this case. By comparing the density flow field shown in Fig. 7.12, it is clear the positive effect of modelling the shocks via S-F.

Table 8. Global error L_1 norm

Grid level	S-C	S-F
0	$5.83 \cdot 10^{-2}$	$4.48 \cdot 10^{-2}$
1	$3.61 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$
Observed order $\tilde{n}$	0.69	1.75

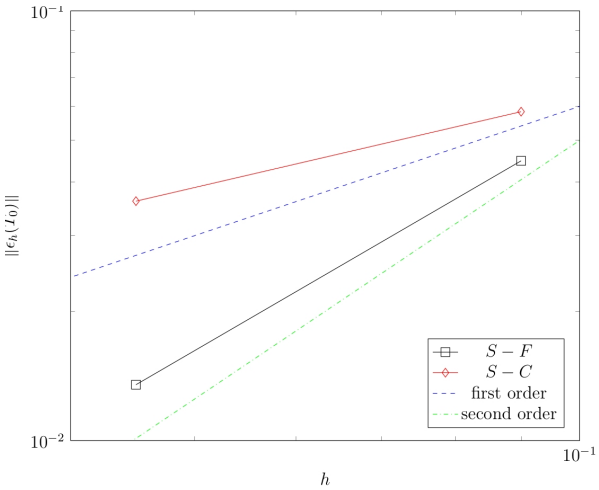
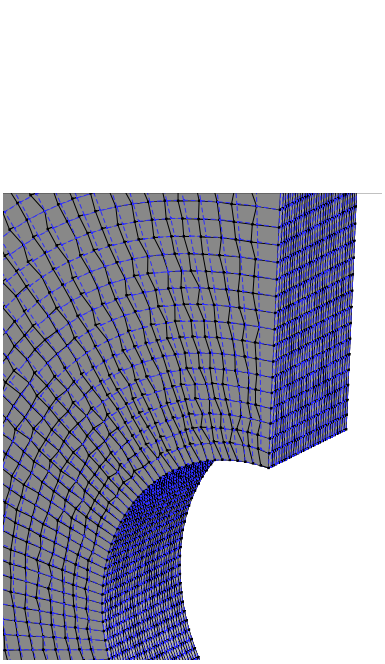
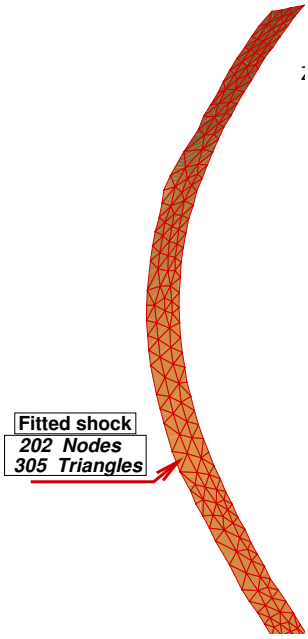


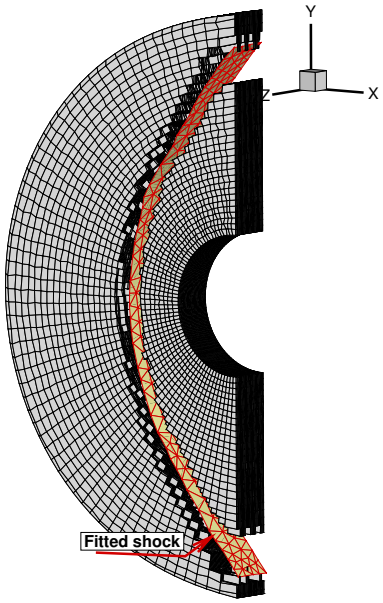
Figure 7.10. Convergence behaviour of L_1 norm



(a) Difference between the regular (blue dashed) and the randomly modified grids



(b) Shock surface



(c) Shock surface within the background volume mesh

Figure 7.11. Supersonic flow past a cylinder (M=3): domain boundaries and shock surface

7.2.2 Regular oblique-shock reflection

The SESF technique has been applied also to the case of a regular shock reflection on a plate ($M_\infty = 4$), located over a compression ramp ($\delta = 20°$). The computational grid, which is the same used for both S-F and S-C computations, is a h-grid of $77 \times 35 \times 35$ points distributed following an exponential law along the I and J

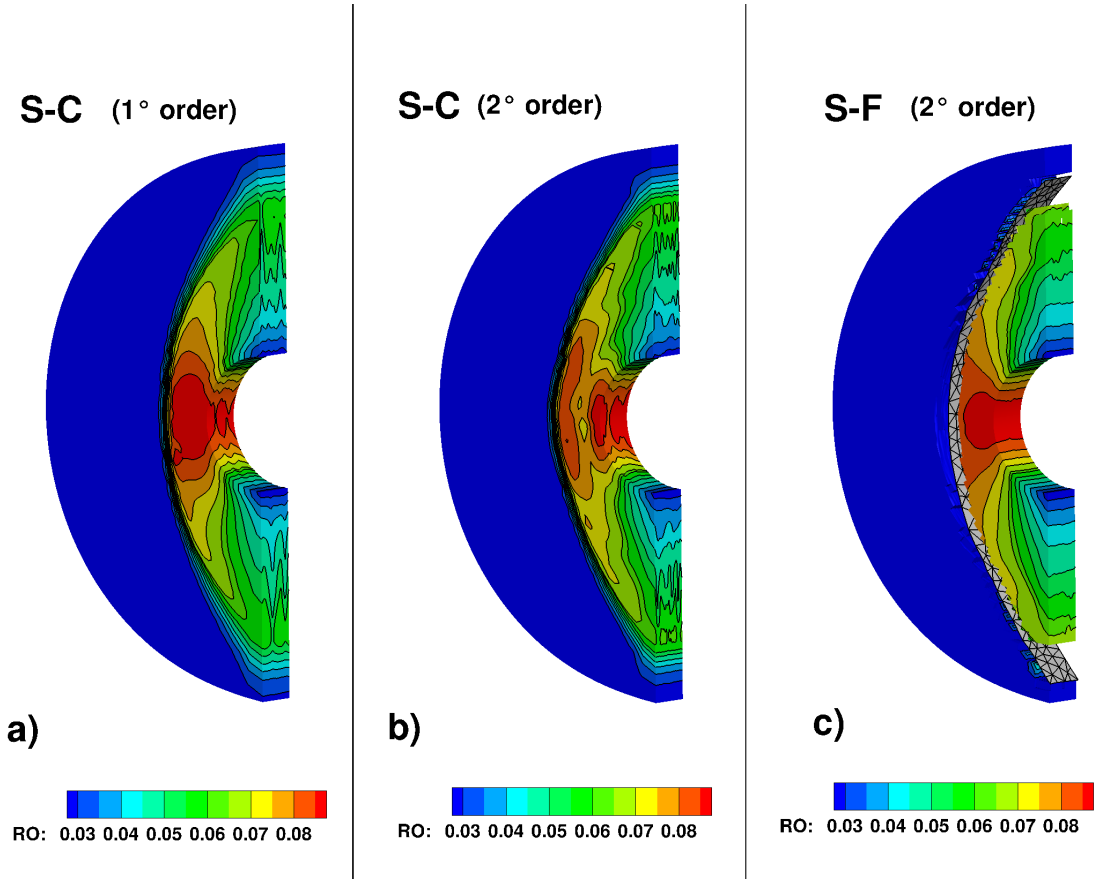


Figure 7.12. Supersonic flow past a cylinder (M=3): density iso-contour lines on the randomly modified grid

directions, as reported in Table 9 in order to cluster the cells around the corner region (see Fig. 7.13). This test case has been computed performing second-order-

Table 9. Compression corner structured grid features

I	J	K	$\Delta_{min}S^I$	$\Delta_{max}S^I$	$\Delta_{min}S^J$	$\Delta_{max}S^J$	ΔS^K
76	34	34	0.1	0.93	0.05	0.5	0.43

accurate simulations using both the S-C and the S-F approach. In particular, the S-F technique has been applied to both the incident and the reflected shock, which are modelled using the surfaces shown in Fig. 7.14: the two surfaces are stitched together along the interaction line that can float along the upper horizontal wall.

Moreover, the presented test case represents an important achievement for the development of the S-F techniques, since for the first time it was possible to perform a full fitted simulation of a 3D shock-shock interaction. A qualitative comparison between the S-C and S-F solutions is presented in Figs. 7.15 and 7.16, which respectively show the Mach number and the entropy flow-field provided by both techniques. Moreover, Table 10 reports also the amount of cells which were blanked

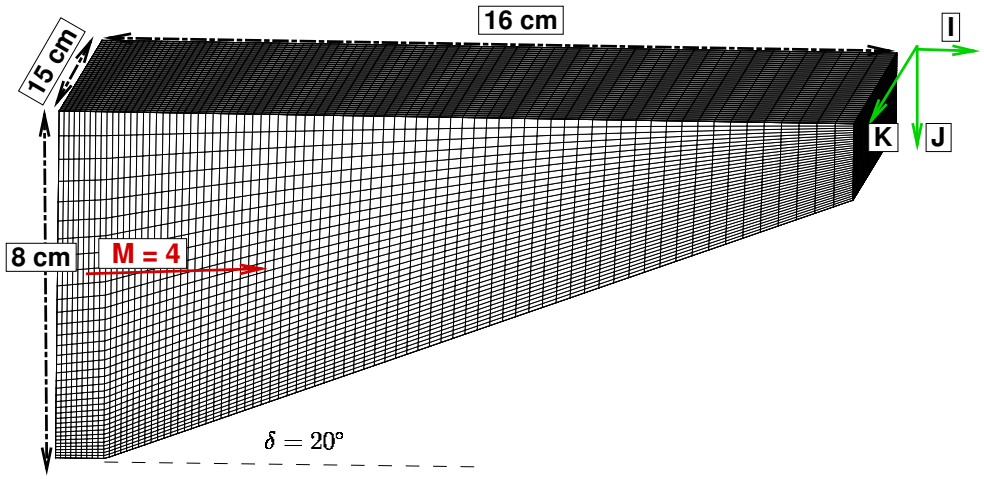


Figure 7.13. Regular oblique shock reflection on a plate (M=4). Domain boundaries and volume mesh.

Table 10. Blanked cells percentage

$N°$ elements	Blanked cells	% Blanked cells
94325	12870	13,5%

due to the shock surface insertions within the volume mesh and the percentage with respect to the total number of cells.

As in the previous test-case, by comparing the numerical solutions it is clearly pointed out the superior performance of the S-F approach. It is evident that the captured discontinuities spoil the quality of the final solution, due to their artificial thickness and because of the spurious disturbances that they generate. To improve the S-C computation quality, a finer and adapted mesh is required, but this also increase significantly the computational cost. Because of the shock modelling, the S-F solution exhibits a higher quality with respect to the S-C one computed on the same grid. Indeed, for this particular case, the numerical computation provided by SESF is equal to the analytical solution and therefore the entropy flow-field is significantly different from the one provided using a S-C approach, as displayed in Fig. 7.16. As it can be noticed by looking at the entropy flow-field in Figs. 7.16a and 7.16b, the S-F solution is able to remove the spurious disturbances which characterize the S-C computation, especially the high entropy production around the corner where the captured incident shock originates and downstream the interaction region.

7.3 Summary

A new structured shock-fitting technique for computing 3D high speed flows on structured meshes has been presented: this approach derives from the one proposed in Chapter 6 for 2D structured meshes and is able to overcome the main limitations of the previously developed S-F approaches in the structured grids framework,

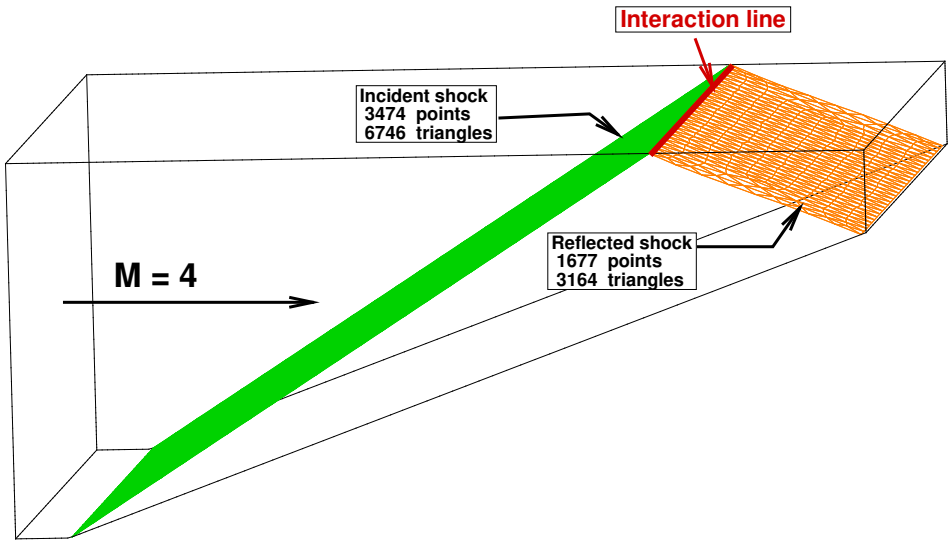
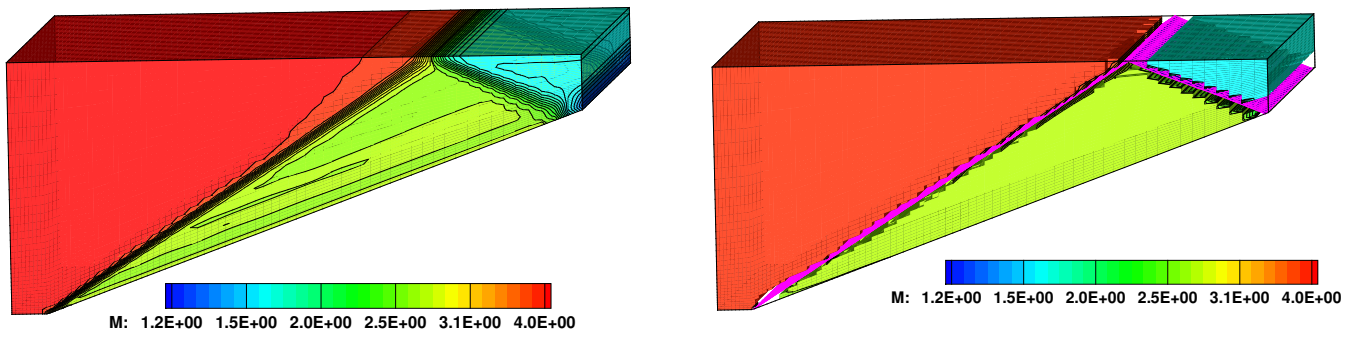


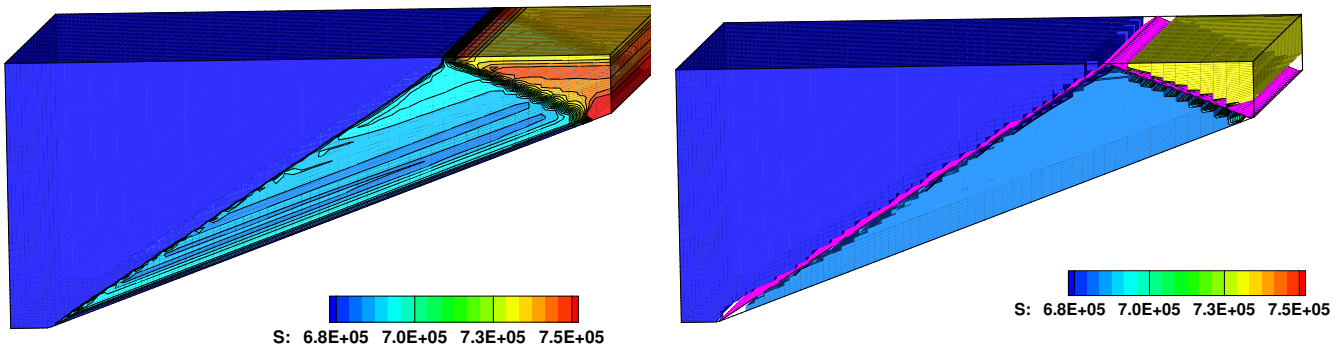
Figure 7.14. Regular oblique shock reflection on a plate (M=4). Shock surfaces.

namely boundary and floating S-F, that had been developed for structured-grid solvers starting in the 1960s. In particular, the capabilities of the novel structured shock-fitting have been highlighted by presenting two different test cases. The former describes a supersonic flow past a cylinder, which allowed to test the gradient reconstruction process described in Sec. 7.1. This flow configuration was analysed also to evaluate the global grid-convergence properties of the proposed technique, which shows an observed order-of-convergence very close to the theoretical one. Furthermore, this test case also proves that the SESF solution is immune to the carbuncle phenomenon, that instead plagued the S-C solution. The last test case considers a regular reflection on a plate, which was computed using both the shock modelling approaches. It is important to notice, that the SESF technique was applied for the first time to compute a shock-shock interaction in 3D space. This task has been achieved by coupling the SESF technique with the surface mesh generator MMG, able to handle mutually interacting surfaces which evolve in time until the steady state. The analysis of the solutions computed by both the SESF and the S-C method on the same volume mesh shows the significant improvements that can be achieved by fitting the discontinuities and it encourages a further development of the this technique.



(a) S-C computation **(b)** S-F computation

Figure 7.15. Regular oblique shock reflection on a plate (M=4): Mach flow-field



(a) S-C computation **(b)** S-F computation

Figure 7.16. Regular oblique shock reflection on a plate (M=4): Entropy flow-field

Part III

Applications

Chapter 8

Shock/boundary layer interactions

Contents

In this Chapter some examples of 2D shock waves/boundary layer interactions (SWBLIs) will be computed on unstructured grids using the S-F technique widely described in Sec. 4.2. Anyway, even though S-F techniques has already been used [180, 27] to simulate hypersonic viscous flows, those flow-cases did not involve SWBLIs, which require some algorithmic addiction with respect to what described in Chapter 4. Nevertheless, before considering the new algorithmic features that have been introduced to deal with SWBLIs, it is appropriate to briefly spend some words regarding the application of S-F techniques to viscous flows. From a mathematical modelling point of view, S-F treats shocks as zero thickness discontinuities and uses the Rankine-Hugoniot (R-H) jump relations to connect the flow states located on two sides of the discontinuity and compute its dynamics. In the real world, however, flows are viscous and the transition between the shock-upstream and shock-downstream states is continuous, even though it takes place over a distance (hereafter called the shock-width or shock-thickness) which in most cases is very small. Inside the shock the flow is dominated by viscous effects and only when the Reynolds number tends to infinity the shock wave becomes a true discontinuity, in the mathematical sense. For this reason, the application of the S-F technique to viscous flows may seem to be precluded on physical grounds, but it is not so. The S-F techniques can be used with great advantage with respect to S-C ones in the simulation of high-Reynolds number flows, as demonstrated by several examples, see e.g. [97, 95, 181, 182, 96, 183, 184]. Indeed, even when the effects of viscosity are accounted for, the R-H jump relations return the correct pre- and post-shock state,

see e.g. [185, 186, 56, 2]. Therefore, the errors incurred by S-F methods are due to: *i*) setting the shock-width to zero and *ii*) ignoring the [46] "physics at subgrid levels", i.e. the flow structure within the shock-width. This length-scale, however, is typically orders of magnitude smaller than any macroscopic fluid-dynamic length-scale [187] (such as the thickness of the boundary layer or the size of the eddies in a turbulent flow), and also much smaller than any "computationally affordable" mesh length. Two alternative possibilities therefore arise: either the flow physics inside the shock can be ignored, in which case S-F is adequate for shock-modeling, or [46] "subgrid physics cannot be ignored, and local mesh refinement must be used to resolve the internal structure of the discontinuity". This more complex scenario is however encountered only in fairly specific circumstances, one such example being weak steady Mach reflections [188, 189] in which case the local curvature of the shock near the triple-point becomes comparable to the shock-thickness. For most flows of practical engineering interest, such as those examined in this Chapter, the first scenario applies and according to Moretti and Salas [55]: "for Reynolds numbers of the order of 1000 or higher, the shock thickness can be neglected and a sharp discontinuity, satisfying the Rankine-Hugoniot conditions, can be assumed in a flow otherwise satisfying the Navier-Stokes equations". If, on the other hand, the shock is captured, rather than being fitted, it features a finite width which spans two or more computational cells, depending on the numerical details of the discretization scheme. When using "computationally affordable" mesh spacings, the width of a captured shock is therefore orders of magnitude larger than the "true" (physical) shock-width. Not only this introduces an error in the shock position which is of the order of the local mesh spacing [190], but also the internal shock-structure is grossly mis-represented [5]. The idea here is that it is better to neglect the shock-thickness and the internal shock-structure altogether, rather than compute those inaccurately. The alternative, which consists in bringing [165] "computational fluid dynamics to a molecular scale, where shocks will cease to be discontinuities and have a real thickness, full of viscous and thermodynamical effects" requires the ability to build a mesh capable of resolving the shock-width. This, in turn, requires the precise knowledge of the shock-location which is tantamount to tracking the shock anyway. Two such examples, spaced apart by fifty years of CFD developments, can be found in [55, 44]; as mentioned earlier, however, in the majority of cases this is an un-necessary and computationally unfeasible option. With regard to the validity of the jump relations and the applicability of the S-F technique to viscous flows, it is also possible to establish a parallelism with the upwind numerical schemes that are nowadays largely used in modern CFD solvers to simulate compressible viscous flows. Indeed, upwind schemes are built upon the theory of characteristics which applies to the case of inviscid flows, but in the case of viscous flows the characteristic curves/surfaces do not exist [191]. However, upwind schemes are used with great advantage also in the case of high-Reynolds number viscous flows, because the presence of the viscous terms in the governing PDEs does not suppress the wave propagation phenomena in the flow and, therefore, the Euler equations can be seen as a first order approximation of the Navier-Stokes equations. These considerations allow to construct numerical schemes and numerical boundary conditions [192] built upon the characteristic theory to be used also in the simulation of viscous flows. Using S-F techniques in the simulation of viscous flows is not only legitimate, for

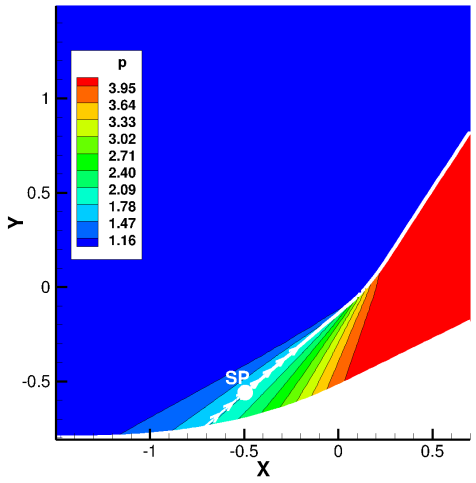
the reasons that have been stated above, but it has undoubted advantages compared to S-C techniques in terms of computational efficiency and quality of the numerical solutions. As mentioned earlier, both S-F and S-C schemes produce numerical artefacts when used in the simulation of high-Reynolds number viscous flows: when the shock is "fitted", its width and internal structure are ignored; when the shock is "captured", its width is exaggerated and the internal structure mis-represented. Which of these two numerical shock models is closer to reality is the key question to be addressed in this paper. The comparative analysis of the numerical results and their comparison with the experimental data show that the error induced by the excessive thickness of a captured shock has a negative impact on the quality of the S-C solution, in particular its order-of-convergence, which is degraded below design-order. Before concluding this introductory assessment concerning the use of S-F and S-C techniques in the simulation of viscous, high-Reynolds number flows, it is worth mentioning two issues related to the simulation of turbulent flows. In reference [56] Lele showed that the jump relations derived from the Reynolds-Averaged Navier-Stokes (RANS) equations are formally different from those that apply to the instantaneous Navier-Stokes equations, i.e. the equations that would be used in a Direct Numerical Simulation (DNS). The jump relations that apply to the RANS equations involve the Reynolds-averaged (or Favre-averaged) flow quantities, rather than the instantaneous ones, and include additional terms that must be modelled in order to close the problem. In this study the modeling of these extra terms will not be addressed and the closure of the jump relations will be done in a simplified way; it is however important to stress that using S-F it might be possible to model the extra terms in the jump relations for the RANS equations. S-C techniques apparently circumvent the problem, because they do not make use of the jump relations. However, also for these techniques the correctness of the numerical solution in the downstream side of the shock remains an open question and, unlike S-F, S-C does not easily allow to introduce extra-terms that should only affect the shock-wave. A similar problem arises in the Large Eddy Simulation (LES) by means of S-C schemes of a turbulent flow passing through a strong shock wave. Here, the numerical dissipation introduced by the upwind S-C scheme in the proximity of the shock wave can be higher than the turbulent dissipation modelled by the Sub-Grid Scale (SGS) model and this represents a source of inaccuracy which cannot be easily controlled [193]. Moreover, it is known [190, 99, 21] that the order-of-accuracy of a S-C calculation degrades to first order within the entire shock-downstream region, even if high-order-accurate schemes are used. In contrast, when all shocks are fitted, the discretization schemes are only used in smooth regions of the flow-field (provided that all discontinuities have been fitted) so that high-order-accurate schemes can be used without incurring into wiggles and retaining their formal order-of-convergence also downstream of the shock. These features make S-F an ideal candidate for simulating turbulent flow via DNS and LES, which typically rely on high-order-accurate discretization schemes, so that a further development of these techniques is highly encouraged.

8.1 Shock/Boundary layer interactions modelling via shock-fitting

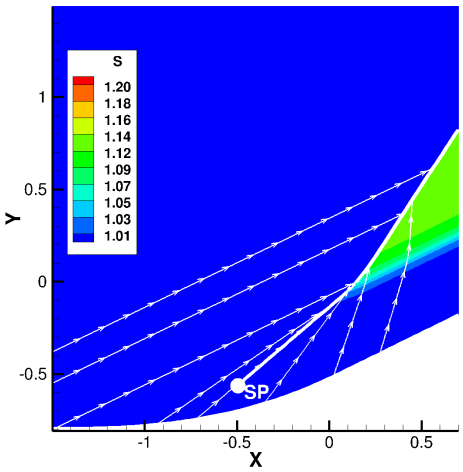
In order to correctly compute SWBLIs in laminar/turbulent regime using the S-F technique presented in Chapter 4, some adaptations are required to the algorithm currently included in the repository on GitHub and detailed in Sec. 4.2. In particular, two major features were implemented: the modelling of the interaction points and the enforcement of suitable jump relations for the turbulent variables, in order to model their behaviour across the discontinuity. First of all, the new features concerning the modelling of the shock end-points within the boundary layer are described. As a matter of fact, Equations (4.1a) and (4.1b) in algorithm Step 3 (please refer to Sec. 4.2) can be used at the interior shock-points of a fitted-shock, but special provisions have to be made at the end-points of the fitted-shock, where the use of one-sided formulae may violate the domain of dependence. Moreover, the flow behaviour close to an impermeable boundary is different, depending on whether an in-viscid or viscous fluid is being simulated. Indeed, in the inviscid case, the flow tangent to a solid surface can be supersonic, so that shock waves may reach the wall, and eventually be reflected off the wall. However, this is not any longer true for a viscous fluid, because the no-slip condition implies that the near-wall boundary layer is subsonic, which prevents the shock wave from reaching the wall. Therefore, the two fundamental interactions that may occur in the case of an in-viscid fluid, i.e. the wall reflection of a weak oblique shock-wave and the strong shock-wave that becomes a normal shock at the wall, are replaced by slightly more complex flow configurations in which the shock-wave either forms or terminates at a certain distance from the wall, within the supersonic part of the boundary layer. By analysing the different cases that may occur, see [194], the more common are essentially two:

1. The coalescence of compression waves in the supersonic part of the boundary layer, leading to the formation of a shock-wave.
2. The impingement of a weak oblique shock-wave on a wall, which penetrates the supersonic boundary layer, where it progressively bends because of the local Mach number decrease: correspondingly, its intensity weakens and it vanishes altogether when it reaches the boundary layer sonic line.

These two mechanism are also the only two encountered in the numerical examples proposed in this Chapter, so that their modelling will be clarified hereafter. Shock formation due to the steepening of compression waves also occurs in inviscid flows and therefore the inviscid test-case also used by Moretti [195, Fig. 9] is considered to describe how the formation of an embedded shock is handled using our S-F technique. Figure 8.1 shows a uniform, supersonic ($M_\infty = 2.3$) stream of air being deflected as it flows over a smooth convex wall: the coalescence of the compression waves leads to the formation of an embedded shock wave at a certain distance above the wall. The steady, supersonic Euler equations can be re-cast, see e.g. [196], into a set of characteristic compatibility equations whose characteristic curves are the streamlines and two different sets of curves having slope:



(a) Pressure (p/p_∞) field and C^+ characteristic curve passing through the SP.

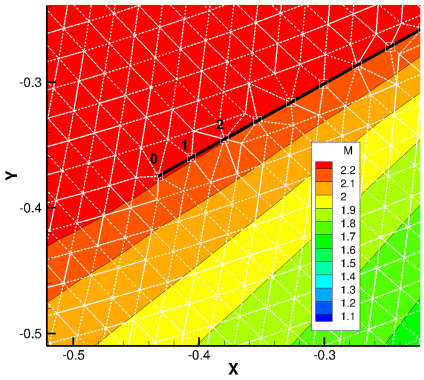


(b) Entropy (S/S_∞) field and C^+ characteristic curves.

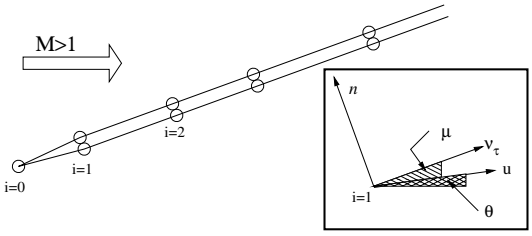
Figure 8.1. A shock embedded in a steady, supersonic flow.

$$\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)_{C^\pm} = \tan\left(\theta \pm \mu\right) = \frac{\beta\, v \pm u}{\beta\, u \mp v} \tag{8.1}$$

θ being the flow angle, μ the Mach angle and $\beta = \sqrt{M^2 - 1}$. The compression region shown in Fig. 8.1 is made up of C^+ characteristic curves, which are plotted in Fig. 8.1b on top of the entropy field. Even though the characteristic curves do not meet at a single focal point, the entropy field reveals that the shock forms near the origin of the reference frame in Fig. 8.1. The reader should not be surprised by the fact that the fitted shock-front (shown using a white solid line in Figs. 8.1a and 8.1b) starts from a starting point (SP) well ahead of the region where the characteristics merge because, in Moretti's words [160]: "Premature fitting of the shock in the region where compression waves tend to coalesce is not harmful at all, provided that the shock behaves as one of the characteristic surfaces coalescing into a finite discontinuity." Figure 8.2a shows a zoom of the first few shock-points along the fitted discontinuity and Fig. 8.2b schematically reproduces Fig. 8.2a by magnifying the width of the fitted discontinuity. Shock-points along the fitted discontinuity have been numbered as shown in Fig. 8.2. The role of shock-point 0, i.e. the SP, consists in opening up the fitted discontinuity, so that a single set of dependent variables is stored in the SP; all other shock-points store both the shock-upstream and shock-downstream values of the dependent variables. Figure 8.2a clearly reveals that the first shock-points lie along a steady C^+ characteristic-curve within a region of supersonic flow. Since the air stream flows from the left to the right of Fig. 8.1, the computation of the tangent vector $\boldsymbol{\tau}$ within these shock-points should be upwind-biased and rely on Eq. (4.1b). The three-points formula (4.1b) cannot, however, be used in shock-points 0 and 1, so that a different approach is required. Since the first shock-points are located in a smooth-flow region and lie along a characteristic curve,



(a) Mach iso-contour lines around the "origin" of the fitted shock.



(b) The fitted shock is an interior boundary; the smaller frame shows how the normal unit vector is computed at shock-point $i = 1$.

Figure 8.2. A shock embedded in a steady, supersonic flow: the "fitted" shock originates along a characteristic-curve.

it is required that $\boldsymbol{\tau}$ is tangent to the C^+ characteristic curve that goes through shock-point 1:

$$\boldsymbol{\tau}_1 \;=\; \frac{1}{a\,M^2}\left[(u\,\beta - v)\,\mathbf{e}_x + (v\,\beta + u)\,\mathbf{e}_y\right] \tag{8.2}$$

where a is the acoustic speed. The unit normal vector $\mathbf{n}_1$ is perpendicular to $\boldsymbol{\tau}_1$ and such that $\mathbf{u}\cdot\mathbf{n} < 0$, which gives:

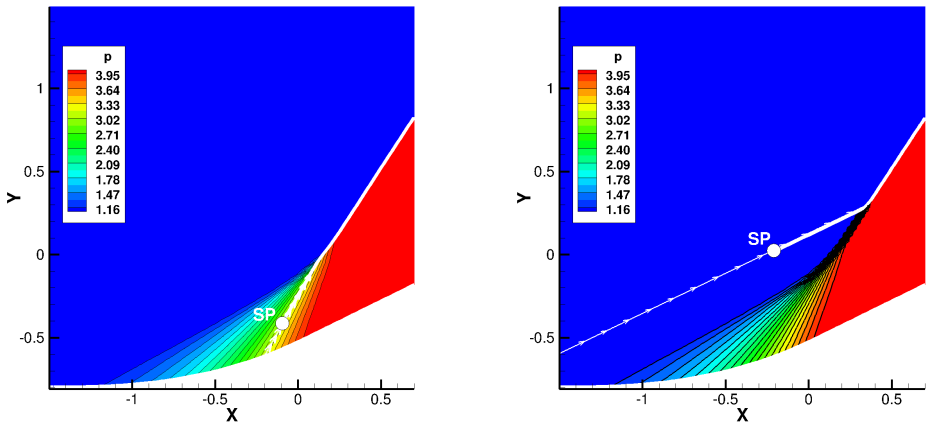
$$\mathbf{n}_1^{\pm} \;=\; \frac{1}{a\,M^2}\left[(\mp v\,\beta - u)\,\mathbf{e}_x + (\pm u\,\beta - v)\,\mathbf{e}_y\right] \tag{8.3}$$

where the $\pm$ notation used in Eq. (8.3) allows to also account for the possibility of characteristic curves of the C^- family. Concerning shock-point 0, there is no need to compute the unit normal vector there, because a single set of dependent variables is stored and, therefore, the R-H jump relations are not applicable. However, the location $\mathbf{r}_0$ of shock-point 0 is needed to compute the tangent vector in shock-point 2 using the upwind-biased formula (4.1b). Shock-point 0 is placed on the same characteristic curve passing through shock-point 1, which leads to:

$$\mathbf{r}_0 = \mathbf{r}_1 - \boldsymbol{\tau}_1\, l_{\frac{1}{2}} \tag{8.4}$$

It is worth mentioning that the location of the SP point has some impact on the computed solution. As long as SP lies within the region where the characteristics merge, differences are minimal, because the entire shock is going to be fitted: compare Figs 8.1a and 8.3a, which correspond to different locations of the SP, both inside the compression region. If, on the contrary, the SP lies outside the compression

region, which is the case of Fig. 8.3b, the shock will be captured, rather than being fitted, up to the location where the characteristic curve passing through SP joins the shock.



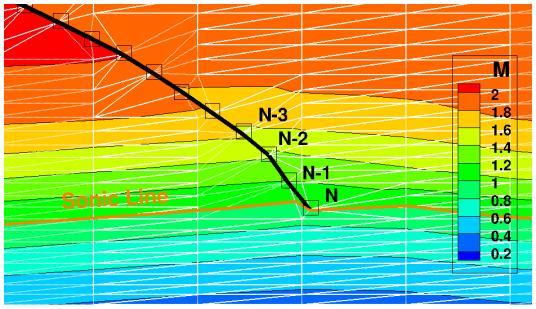
(a) The SP is inside the compression region.

(b) The SP is outside the compression region.

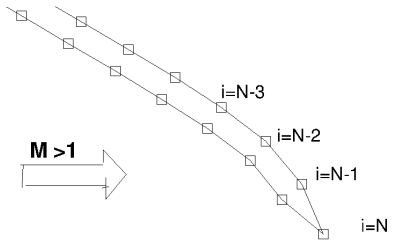
Figure 8.3. A shock embedded in a steady, supersonic flow. Pressure (p/p_∞) field and C^+ characteristic curve passing through the SP: comparison between different starting point locations.

The case of the weak (hence supersonic downstream flow) oblique shock-wave impinging on a boundary-layer may look similar to the previous case, but it is actually simpler. As shown in Fig. 8.4, the shock penetrates the supersonic boundary layer and weakens because the shock-upstream Mach number decreases as it approaches the sonic line. This second case does not require any ad-hoc treatment for calculating the shock-tangent vector $\boldsymbol{\tau}$ within the last two shock-points, because the one-sided FD formula (4.1b) can be used without violating the domain of dependence. Similarly to the previous case, end-point N, see Fig. 8.4, closes the fitted discontinuity, thus it is described using a single set of dependent variables, which are updated by the gas-dynamic solver and do not require to be corrected by enforcing the jump relations. This same algorithmic treatment has also been used in all test-cases addressed in Sect. 8.2 to model those end-points that terminate a fitted-shock outside the boundary layer.
Finally, when dealing with turbulent SWBLIs, some considerations are required regarding the modelling of the turbulent variables across the shock wave. Specifically, the Spalart-Allmaras (S-A) model [51] has been implemented within the `EulFS` solver used by the *UnDiFi-2D* code to perform computations in turbulent regime: thus, as far as the turbulent variable used in this model is concerned, in this Thesis it is followed the simplified approach suggested by the model developers [57], which consists in carrying the turbulent viscosity unchanged through the shock. In terms of jump relations, this amounts to set the shock-downstream value of the turbulent

kinematic viscosity equal to the corresponding shock-upstream value.



(a) Mach iso-contour lines around the end of the fitted shock.



(b) The fitted shock is an interior boundary.

Figure 8.4. S-F modeling of an oblique shock-wave impinging on a boundary-layer.

8.2 Numerical results

SWBLIs turn out to be an important test-bench to evaluate the ability of the proposed unstructured S-F technique to deal with viscous flows featuring shock-waves. This assessment will also be done by comparison with state-of-the-art S-C discretization. This kind of comparison is not new: figure 8.5, which is taken from a 1992 technical report by Marconi and Moretti [197], see also [20], shows the interaction between an oblique shock and a laminar boundary layer computed on structured grids using both S-F and S-C. Nearly thirty years later, and despite the enormous advances in

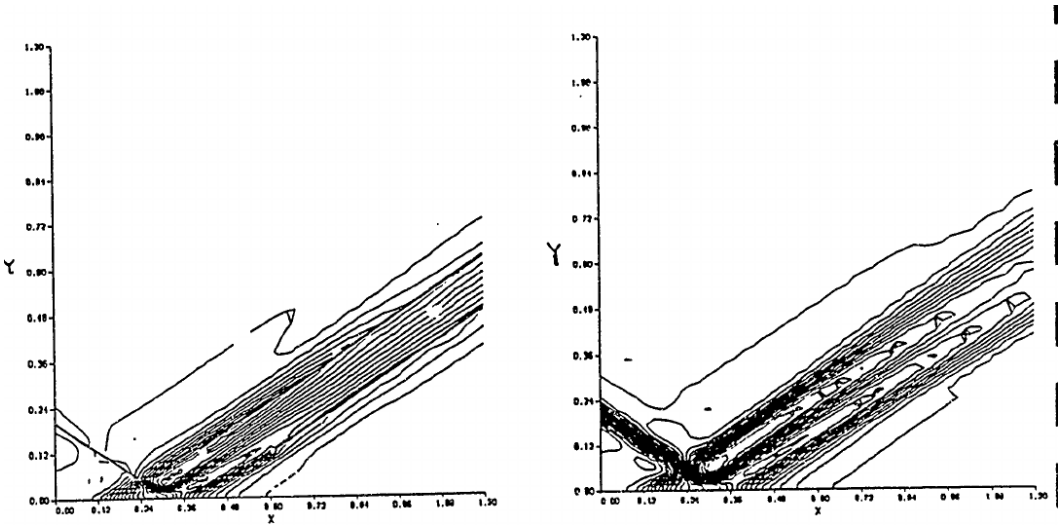


Figure 8.5. Interaction between an oblique shock and a laminar boundary-layer: isobars computed using S-F (left) and S-C calculation (right); re-printed from [197], with permission.

computing power that have meanwhile taken place, the predictive capabilities of the two different shock-modeling techniques have remained the same. In particular, as observed by Marconi and Moretti [197], the comparison between the two frames of Fig. 8.5 clearly " indicates the amount of shock spreading in the capturing result". In the following sections, in addition to a qualitative comparison between the two shock-modeling options, the computed results will be compared against the available experimental data and a quantitative assessment of the order-of-convergence of the two techniques will be made, using Richardson Extrapolation (RE) techniques.

8.2.1 Regular oblique-shock reflection

The interaction between an oblique shock and a boundary-layer, see Fig. 8.6, is labeled strong [198] whenever it causes boundary-layer separation. This is the case of the experimental and numerical study conducted by Degrez et al. [199], whose free-stream flow conditions are reported in Tab. 11.

Parameter	value
Mach	2.2
Re_∞	$1.2\,10^6$
Total temperature	293 K

Table 11. Oblique-shock reflection: free-stream conditions.

A sketch of the flow pattern is shown in Fig. 8.6: the region of recirculating flow extends from the separation point, S, to the re-attachment point, R.

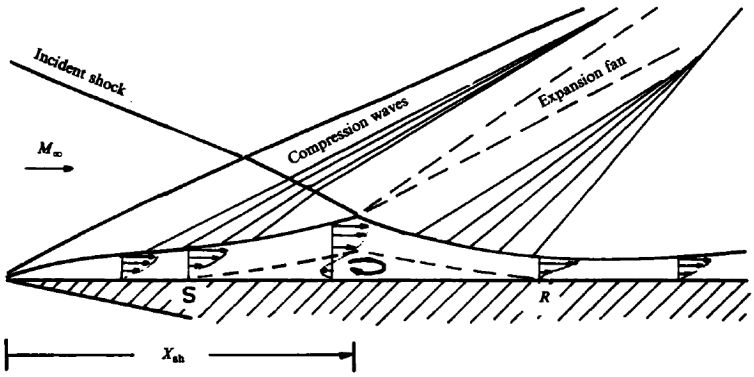
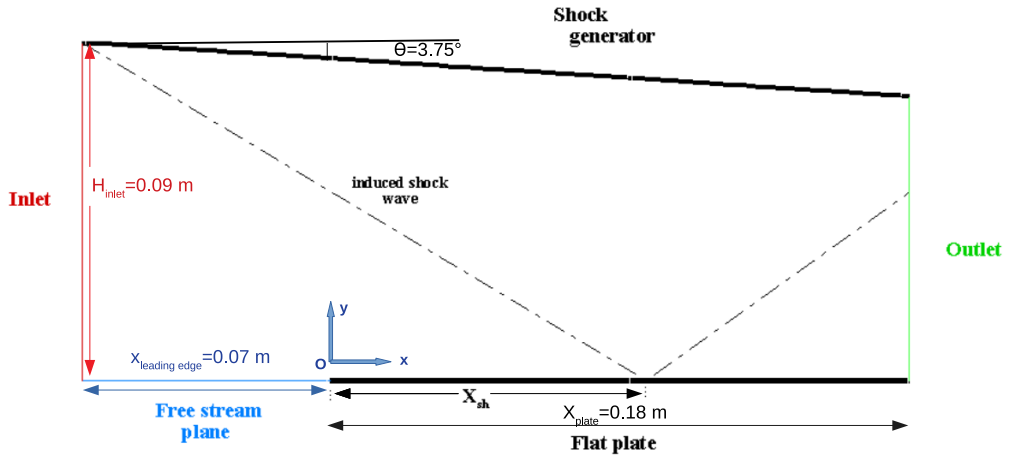


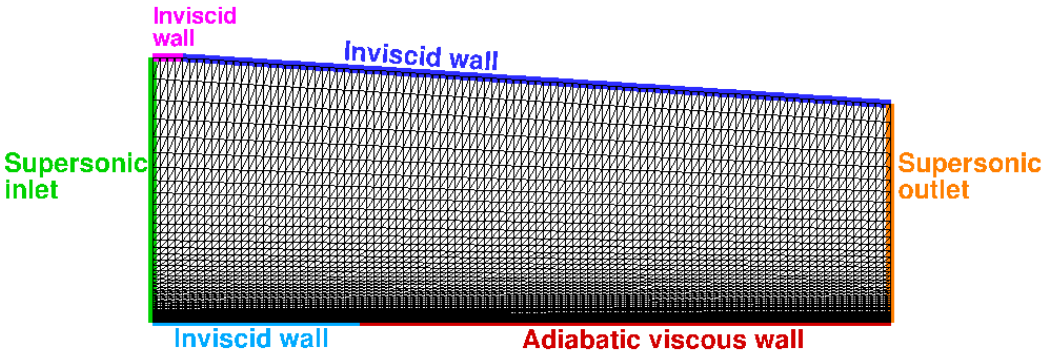
Figure 8.6. Schematic of the flow-field arising from the interaction between an oblique-shock and a boundary-layer. Re-printed from [199], with permission.

The incident shock, after crossing the leading edge (L.E.) shock, hits the boundary layer at $x = X_{sh}$ and is reflected as an expansion wave, because of the nearly constant pressure level within the bubble [198]. The flow deflection caused by the recirculating bubble leads to the formation of compression waves, both upstream of S and downstream of R. These compression waves coalesce at some distance from the flat plate to form the separation and re-attachment shocks. As far as

the computational setting is concerned, Fig. 8.7 shows the computational domain, the boundary conditions that have been prescribed along its boundaries in order to match the experimental setting and the coarsest mesh that has been used. In



(a) Computational domain geometry.



(b) Coarse mesh and labeling of the boundary patches.

Figure 8.7. Oblique-shock reflection: coarse mesh and numerical boundary conditions.

order to compare the grid-convergence properties of the S-C and S-F solutions, three nested grid levels have been obtained by recursively coarsening the finest one; their characteristic are summarized in Tab. 12. The column labeled "No. of quadrilateral cells" refers to the number of quadrilateral cells in the stream-wise and wall-normal directions. The unstructured triangular grids have been obtained by splitting each quadrilateral cell into two rectangular triangles, so that the overall number of triangles is twice the number of cells in Tab. 12. In order to adequately resolve the boundary-layer, the cells are stretched in the wall-normal direction (see Fig. 8.7b): the wall normal spacing of the first cell adjacent to the flat plate ranges between Δy_{min} and Δy_{max}, whereas Δx is the constant mesh spacing in the stream-wise direction normalized using the channel height (H=0.09 m).

The S-C calculations have been run first, using the three grids listed in Tab. 13. Then, a shock-detection algorithm, such as the one described in [164], has been

Table 12. Oblique-shock reflection: features of the three nested grid levels.

Grid level	No. of quadrilateral cells	$\frac{\Delta y_{min}}{H}$	$\frac{\Delta y_{max}}{H}$	$\frac{\Delta x}{H}$
coarse	100×50	$1.1 \cdot 10^{-3}$	$7.9 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$
intermediate	200×100	$5.5 \cdot 10^{-4}$	$3.9 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
fine	400×200	$2.7 \cdot 10^{-4}$	$1.9 \cdot 10^{-2}$	$5.5 \cdot 10^{-3}$

used to extract from the S-C calculation the approximate shape and location of the fitted discontinuities. The fitted discontinuities, and the flow solution computed by means of S-C, supply the initial condition to the S-F calculations. Table 13 allows to draw a comparison between the grids used in the S-C calculations (also used as background triangulation in the corresponding S-F calculation) and the S-F "computational" meshes. These latter feature a larger number of grid-points and triangles w.r.t. the corresponding S-C mesh, due to the addition of the shock-points[1], which is only partly compensated by the removal of the phantom points, and because local re-meshing around the fitted shock-front increases the number of triangles. The percentage increase in both the number of triangles and grid-points is lower on the finer meshes; this is a consequence of the fact that the shock-mesh has a lower dimensionality ($d-1$ in the d-dimensional space) than the computational mesh and re-meshing is limited to the neighbourhood of the fitted discontinuities.

Table 13. Oblique-shock reflection: comparison between the S-C meshes (also used as background triangulation in the corresponding S-F calculation) and the S-F "computational" meshes.

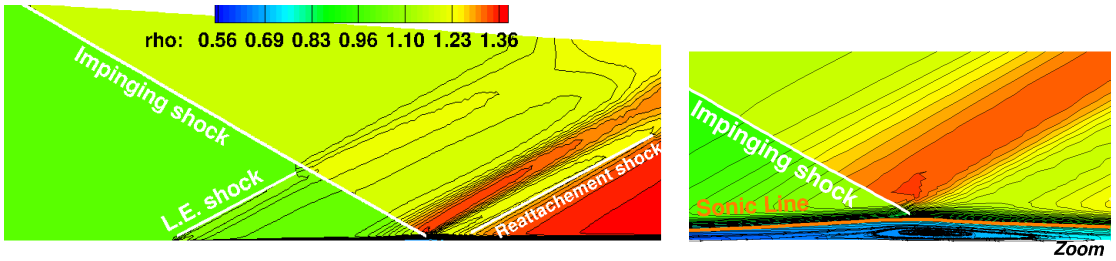
Grid level		S-C mesh	S-F mesh	% increase	Nof. shock-points
coarse	triangles	10000	11004	10.04	592
	grid-points	5151	5650	9.69	
intermediate	triangles	40000	42132	5.33	1253
	grid-points	20301	21364	5.24	
fine	triangles	160000	164260	2.66	2495
	grid-points	80601	82729	2.64	

The S-F technique has been used in an hybrid manner, whereby only the strongest discontinuities have been fitted, whereas the remaining ones have been captured. This is shown in Fig. 8.8, which shows the dimensionless density (ρ/ρ_∞) field computed by S-F on the coarsest mesh.

The fitted discontinuities, which have been highlighted using a white solid line, include the incident and leading edge (L.E.) shocks. The L.E. shock becomes a Mach-wave as soon as it moves away from the L.E.: its slope, which can be computed along the fitted shock-front using Eqs. (4.1), is very close to the free-stream Mach angle

[1] In Tab. 13 the shock-points are a subset of the grid-points.

$\mu_\infty = \sin^{-1} M_\infty^{-1} = 27°$. As shown in the sketch of Fig. 8.6, the two compression regions that originate close to the separation and re-attachment points coalesce into the compression and re-attachment shock at some distance from the plate, outside the boundaries of the computational domain. Therefore, the fitted re-attachment shock that appears in Fig. 8.8a is actually one of the C^+ characteristic curves (or Mach lines) that will merge into the re-attachment shock. As explained in Sect. 4.2 it is not harmful to fit a characteristic curve upstream of point where it actually runs into the shock. Finally, the discontinuities that have not been fitted are the shock and the slip-stream that arise at the point where the L.E. and incident shocks cross each other, see Fig. 8.6.

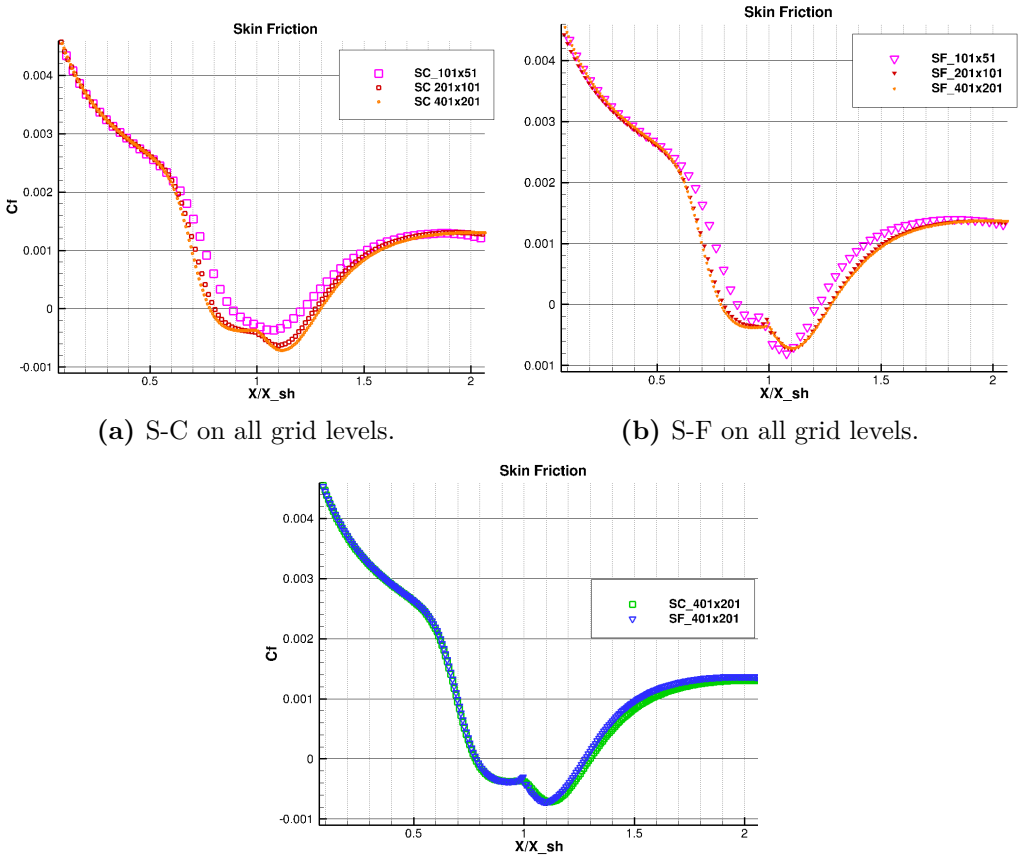


(a) Full computational domain **(b)** Recirculation region

Figure 8.8. Oblique-shock reflection: dimensionless density (ρ/ρ_∞) field computed using S-F on the coarsest mesh. Fitted shocks are shown using solid white lines.

A preliminary and qualitative grid-convergence study has been performed by analyzing the behavior of both the skin friction (Fig. 8.9) and wall pressure (Fig. 8.10) distributions computed on all grid levels using the two different shock-modeling options. Figures 8.9a and 8.10a respectively show the skin-friction and wall-pressure distributions computed using S-C on all three grid levels. The same quantities computed using S-F are displayed in Figs. 8.9b and 8.10b. When looking at the S-C and S-F calculations, both are seen to converge towards the fine-grid solution as the mesh is refined, but S-F appears to be converging faster. For instance, the S-F simulation already reveals on the coarsest grid level the skin friction and pressure plateau within the separation region. In the S-C calculations this same feature only becomes visible on the intermediate grid level. Figures 8.9c and 8.10c compare the skin-friction and wall-pressure distributions computed on the fine grid-level using S-C and S-F. As far as the wall pressure distribution is concerned, the agreement between the two shock-modeling options, and also with the experimental data available in [199], is good. Larger differences can be seen in the skin-friction distribution, and are more pronounced downstream of the point where the incident shock impinges on the boundary-layer, i.e. at $X/X_{sh} \approx 1$.

The comparison between the dimensionless density flow-field computed on all three grid levels using both S-C and S-F, see Fig. 8.11, clearly reveals the superior performance of the S-F technique: due to the finite width of the captured shocks, the SWBLI region is poorly resolved in the S-C calculations, particularly when the coarse and intermediate grid levels are used, see Figs. 8.11a and 8.11c. In contrast, S-F allows to represent the incident shock as a true discontinuity on all grid levels,



(a) S-C on all grid levels.

(b) S-F on all grid levels.

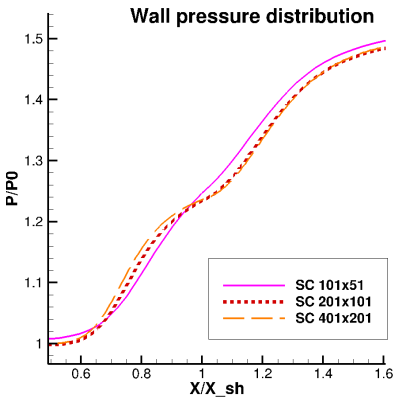
(c) S-C vs. S-F on the finest grid.

Figure 8.9. Oblique-shock reflection: skin friction distribution.

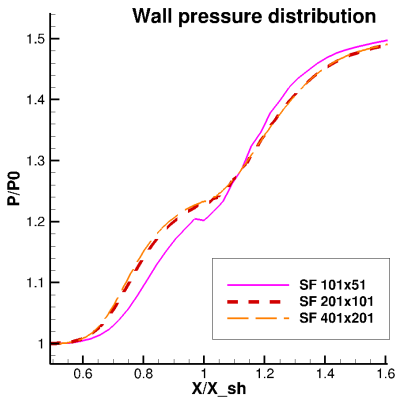
see Figs. 8.11b, 8.11d and 8.11f. Figure 8.11 is the unstructured-grid counterpart of Fig. 8.5: shock-smearing is further exacerbated, in the unstructured grid context of Fig. 8.11, by the use of highly elongated triangular cells and the fact that none of their edges is aligned with the incident shock.

A quantitative assessment of the grid-convergence properties of the two different shock-modeling options is presented in Tab. 14, which shows the location of the separation and re-attachment points and the size of the recirculation bubble computed using S-C and S-F on all three mesh levels. The separation and re-attachment points are identified as those points where the skin friction vanishes; their abscissae are measured relative to the L.E. and have been normalized by the distance, $\mathrm{X}_{sh} = 80$ mm, also measured from the L.E., where the shock wave impinges on the plate, see Fig. 8.7a. The bubble size is the difference between the abscissae of the re-attachment and separation points. Using the results computed on all three grid levels, it is possible to determine the observed order of accuracy, $\tilde{n}$ according to Eq. 3.1. Moreover, by evaluating the convergence monitor R (see Eq. 3.2), it can be noticed that both the S-C and S-F results exhibit monotone convergence, since $0 < R < 1$ for all three quantities [21].

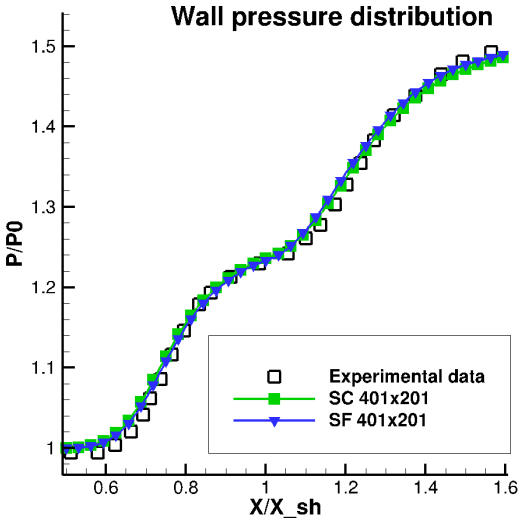
In particular, the observed order-of-accuracy of the S-F solutions is very close to



(a) S-C on all grid levels.



(b) S-F on all grid levels.



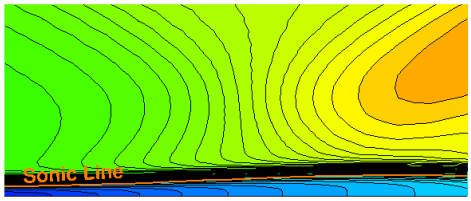
(c) S-C vs. S-F on the finest grid.

Figure 8.10. Oblique-shock reflection: wall pressure distribution.

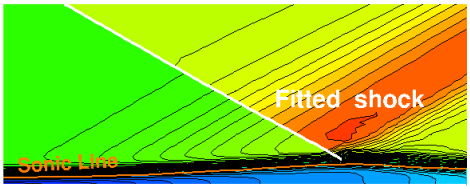
	Shock capturing			Shock fitting		
Grid	(X_{sep}/X_{sh})	(X_{re}/X_{sh})	Bubble size/X_{sh}	(X_{sep}/X_{sh})	(X_{re}/X_{sh})	Bubble size/X_{sh}
coarse	0.87±(0.13)	1.22±(0.12)	0.34 ±(0.25)	0.86±(0.11)	1.21±(0.09)	0.35 ±(0.20)
intermediate	0.80±(0.04)	1.27±(0.05)	0.47 ±(0.08)	0.80±(0.03)	1.26±(0.02)	0.46 ±(0.05)
fine	0.78±(0.01)	1.30±(0.02)	0.52 ±(0.03)	0.78±(0.01)	1.28±(0.01)	0.49 ±(0.01)
$\tilde{n}$	1.79	1.30	1.55	1.86	1.99	1.92
$\tilde{u}_{RE}$	0.77	1.31	0.54	0.78	1.28	0.50
experimental values	0.78±(0.01)	1.28±(0.02)	0.5 ± (0.04)	0.78 ±(0.01)	1.28±(0.02)	0.5 ± (0.04)

Table 14. Oblique-shock reflection: grid-convergence study and RE in the separation region.

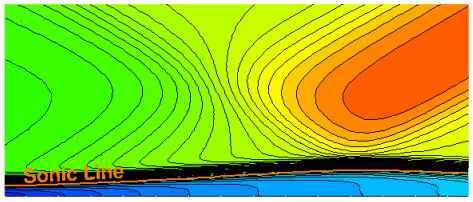
the design order ($n = 2$) of the discretization scheme, which is an indication [144, 152, 153] that the S-F solutions are in the asymptotic range of convergence. The analysis



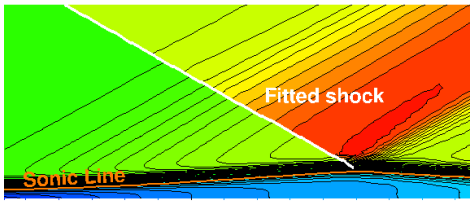
(a) S-C computation: coarse grid.



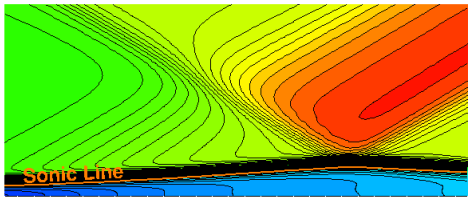
(b) S-F computation: coarse grid.



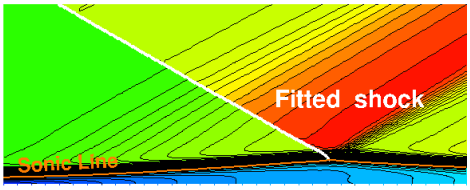
(c) S-C computation: intermediate grid.



(d) S-F computation: intermediate grid.



(e) S-C computation: fine grid.



(f) S-F computation: fine grid.

Figure 8.11. Oblique shock reflection dimensionless density (ρ/ρ_∞) field: zoom of the SWBLI region using the three grid levels and shock-modeling options.

can be further expanded by computing the Grid Convergence Index (GCI) [144, 152, 153] using Eq. 3.3 with a factor of safety F_s=1.25, as well as the Richardson Extrapolated (RE) value, defined in and Eq. 3.4.

It is worth to underline that the GCI is a measure of the numerical error, just like the experimental uncertainty. GCI values of the bubble size, separation and re-attachment locations are reported between brackets alongside each numerical value in Tab. 14; bracketed values beside the measured data refer to the experimental uncertainty [199]. The GCI data in Tab. 14 show that the S-F technique exhibits faster convergence than S-C: almost all GCI values are smaller for the S-F solutions than the corresponding ones for S-C. On the finest grid level, for instance, S-F returns GCI values that are half of the corresponding S-C values for both the re-attachment location and bubble size, whereas the GCI values of the separation location are equal.

Table 14 also includes the "Richardson-extrapolated" (RE) values of all three quantities computed using Eq. (3.4). The separation location and bubble size computed using the two different shock-modeling options fall within the experimental uncertainty; this observation also applies to the re-attachment location computed by S-F. In contrast, the "extrapolated" value of the separation location computed by means of S-C falls outside the experimental uncertainty. This is likely to be

explained by the fact that, for this particular quantity and shock-modeling option, the observed order is way below design order: 1.3 vs. 2. It is indeed known [152, 153], that the asymptotic solution can be reliably computed by means of RE only if the observed order is close to its theoretical value.

8.2.2 Hypersonic compression ramp

When an uniform hypersonic stream is deflected due to a geometrical discontinuity of the wetted surface (the ramp or corner in Fig. 8.12), a SWBLI takes place, which may eventually lead to boundary-layer separation. Among the several experimental setups studied by Holden [200, 201], the two flow configurations listed in Tab. 15 have been numerically reproduced, which only differ in the ramp deflection angle, θ_w. Both flows are assumed to be laminar, in agreement with the experimental evidence [200]. The computational domain and numerical boundary conditions are shown in Fig. 8.12.

Parameter	
Mach	14.1
Re_∞	236200
T_∞	88.88 K
T_w	297.2 K
θ_w	$15°, 24°$
Distance between L.E. and ramp corner (L)	0.4389 m

Table 15. Ramp flow: free-stream conditions

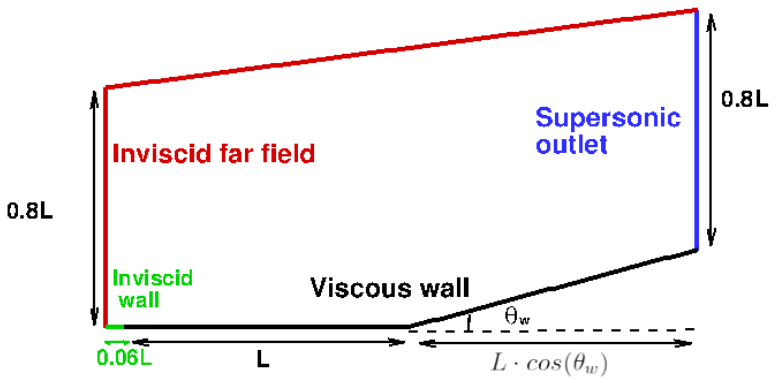
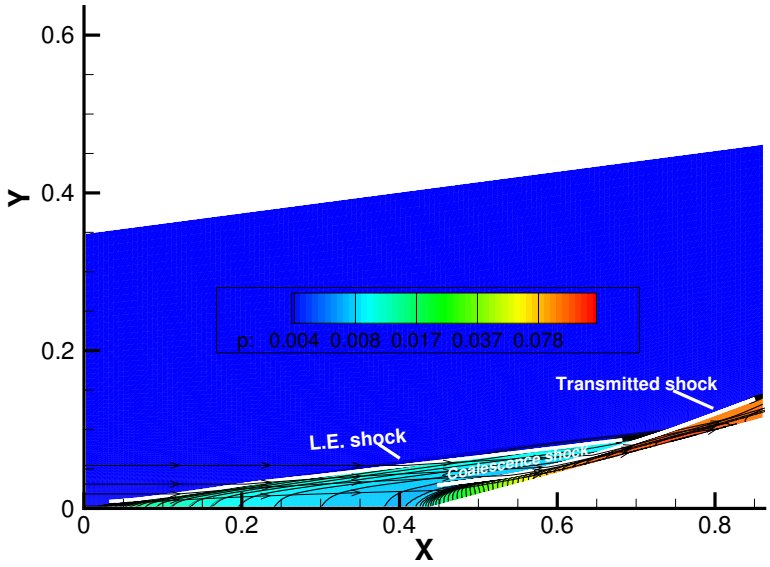


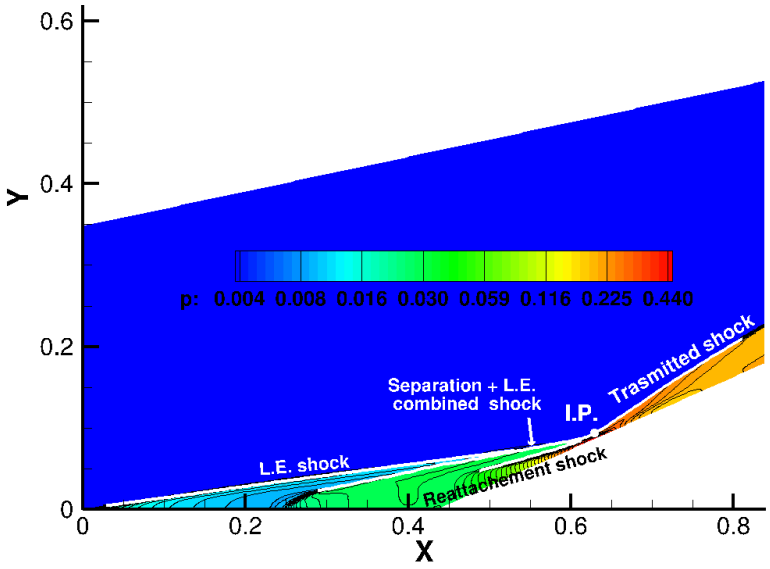
Figure 8.12. Ramp flow: computational domain and boundary conditions.

Before analyzing the numerical results, it is instructive to examine how the shock pattern is affected by the selected ramp deflection angle. This will be done by reference to the pressure fields computed by means of S-F, which are shown in Fig. 8.13a, resp. Fig. 8.13b, for the $\theta_w = 15°$, resp. $\theta_w = 24°$, ramp deflection angle. In both cases, a relatively weak shock is formed at the L.E.. In the $\theta_w = 15°$ case,

the boundary-layer remains attached. The flow deflection induced by the ramp gives rise to compression waves which merge into the "coalescence" shock of Fig. 8.13a at some distance above the wall and just downstream of the wall corner. The L.E. and "coalescence" shocks meet at an interaction point (I.P.); a "transmitted" shock, an expansion fan and a shear layer emerge from the I.P. but only the shock is clearly identifiable in Fig. 8.13a.



(a) $\theta_w = 15°$.



(b) $\theta_w = 24°$.

Figure 8.13. Ramp flow: pressure iso-contours in logarithmic scale.

Flow separation occurs when the ramp deflection angle is increased to $\theta_w = 24°$, which is the case shown in Fig. 8.13b. The presence of the separation bubble causes a

deviation of the streamlines both upstream of the separation point (which is located along the flat wall, upstream of the corner) and downstream of the re-attachment point (which is located along the ramp). The coalescence of these compression waves leads to the formation of a separation and a re-attachment shock, both clearly visible in Fig. 8.13b. The L.E. shock joins the separation shock upstream of the I.P. where the separation and re-attachment shocks merge, see also Fig. 8.14a. Similarly to the $\theta_w = 15°$ case, a "transmitted" shock, an expansion fan and a shear layer emerge from the I.P. The two different flow configurations have been computed using both the S-C and S-F approaches. A grid-convergence study, not reported here for brevity, allowed us to identify the grids listed in Tab. 16 as adequate to obtain grid-converged wall quantities. The nomenclature used in Tab. 16 is the same as that of Tab. 12.

θ_w	No. of cells quadrilateral cells	Δy_{min}	Δy_{max}	Δx
15°	140×180	$2 \cdot 10^{-4}$	$1 \cdot 10^{-2}$	$6.1 \cdot 10^{-3}$
24°	180×200	$2 \cdot 10^{-4}$	$1 \cdot 10^{-2}$	$4.7 \cdot 10^{-3}$

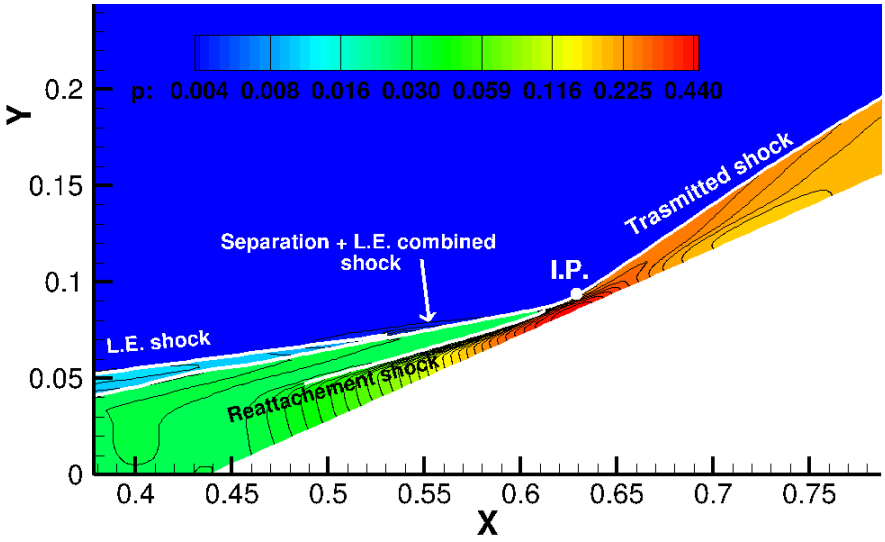
Table 16. Ramp flow: characteristics of the grids.

S-F calculations have been run in the so-called hybrid-mode, whereby most part of the various discontinuities has been fitted, but their mutual interaction has been captured. In the attached flow case ($\theta_w = 15°$) shown in Fig. 8.13a both the L.E. and the coalescence shocks have been fitted, but their mutual interaction has has been captured. In the separated flow case ($\theta_w = 24°$) shown in Fig. 8.14a the separation and transmitted shocks have been fitted as a single discontinuity and also the L.E. and the re-attachment shocks have been fitted. However, the L.E. shock has been fitted up to a point just upstream of where it impinges on the separation shock. Similarly, the re-attachment shock ceases to be a fitted discontinuity immediately upstream of the I.P.. It follows that both the interaction between the L.E. and the separation shocks and that between the separation and re-attachment shocks, have been captured, rather than fitted. All the aforementioned features are clearly visible in Fig. 8.14a.

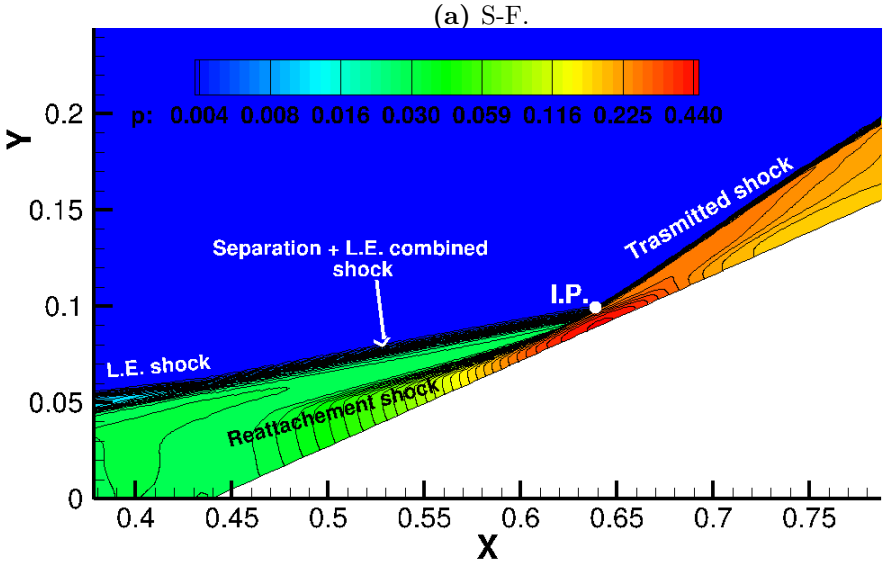
Figure 8.14b shows these same flow features obtained using a S-C simulation. The most striking difference between the two different shock-modeling options is in the shock-width. Observe, in particular, that in the S-F simulation the L.E. and separation shocks are two distinct discontinuities up to about $x = 0.55$, whereas in the S-C calculation these two shocks have already merged ahead of $x = 0.45$.

The dimensionless wall pressure and skin friction distributions corresponding to the different ramp deflection angles are plotted in Figs. 8.15a and 8.15b along with the experimental data measured by Holden [200]. The two frames of Fig. 8.15 show the results obtained using both S-C and S-F on nearly identical grids, as explained in Sect. 8.2.1.

In the attached flow case, the wall pressure distributions obtained by means of S-C and S-F are superimposed and both numerical results are also in fairly good



(a) S-F.

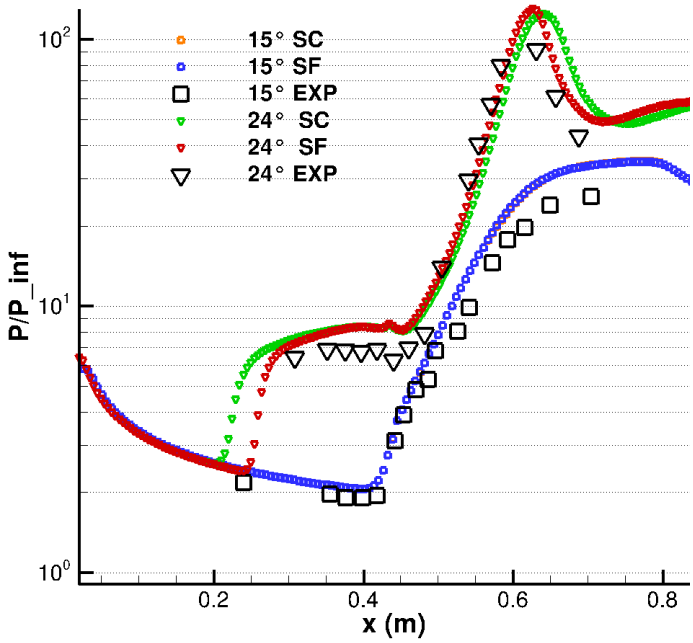


(b) S-C.

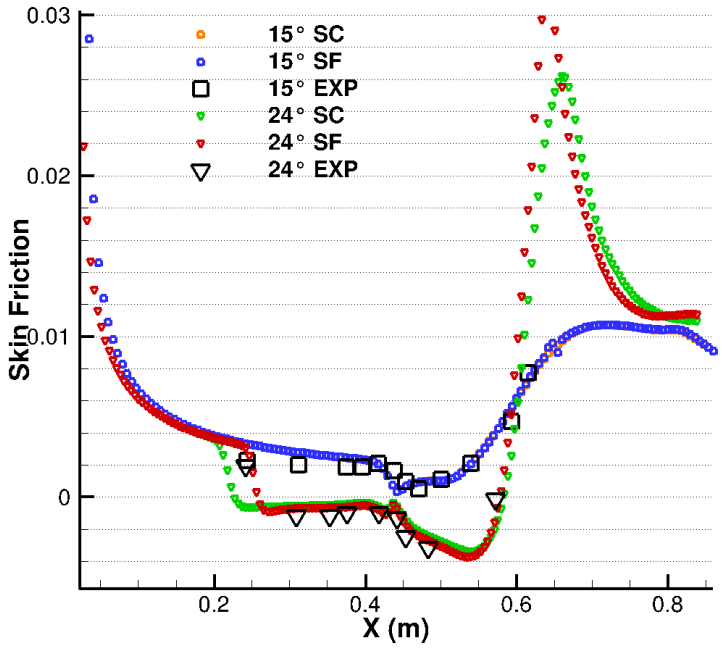
Figure 8.14. $\theta_w = 24°$ ramp-flow: pressure iso-contour lines in the region surrounding the I.P.

agreement with the data supplied by Holden [200], even though the computed values are slightly higher than those found in the experiment. This discrepancy could be due, however, to three-dimensional effects related to the finite span of the ramp, which cannot be accounted for in a two-dimensional simulation. Also the skin friction distributions computed using the two different shock-modeling options are mutually superimposed, see the circles in Fig. 8.15b.

The negative skin friction values in Fig. 8.15b reveal that flow separation takes place when the wedge angle is increased to $\theta_w = 24°$. The extent of the separation bubble is well predicted by the S-F technique, whereas the S-C calculation features a much larger bubble. Figure 8.14b highlights the reason behind the overestimate:



(a) Wall pressure distribution.



(b) Skin friction distribution.

Figure 8.15. Ramp-flow: comparison between experimental and numerical data at the wall.

when the separation shock is captured, its thickness is comparable to the distance between the I.P. and the wall. This causes a significant widening of the interaction region, that affects the position of the re-attachment point and, consequently, also the position of the separation point. It follows that the size of the separation bubble

in the S-C simulation is much larger than the one predicted by S-F or experimentally measured. Moreover, the pressure peak, due to the shocks interaction, is slightly shifted with respect to the experimental datum. The S-F technique predicts well also the extent of the high-pressure region. To summarize, the quality of the S-F solution is improved because, by reducing to zero the thickness of the separation shock, the extent of the interaction region is brought closer to its "true" (physical) size. It would be interesting to see if an even better result were obtained, if also the interaction points were fitted, as done in [23]. This is left for future work.

8.2.3 Transonic, turbulent flow past an airfoil

The transonic flow over a 18%-thick biconvex circular-arc airfoil has been computed to include a test-case where the SWBLI leads to fully turbulent, separated flow. The flow configuration experimentally studied by McDevitt et al. in [202, 203, 204, 205] has been reproduced; Tab. 17 shows the free-stream conditions and reference length.

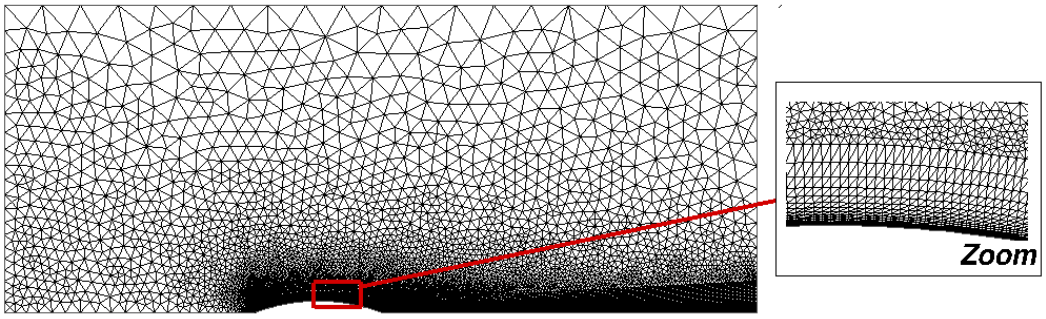
The computational domain and the numerical boundary conditions are shown in Fig. 8.16b: the channel width, h, has been chosen following Mcdevitt et al. [202], who state that blockage effects are negligible as long as the shock-height, z, to channel width ratio verifies: $z/h < 2/3$. Figure 8.16a shows the background triangulation,

Parameter	
α_∞	0°
M_∞	0.783
$\mathrm{Re}_\infty/\mathrm{c}$	10^7 m^{-1}
T_∞	290 K
c	20.3 cm

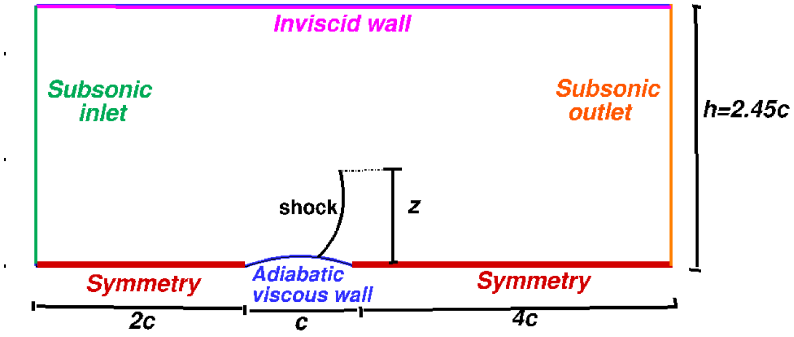
Table 17. Transonic, turbulent flow past an airfoil: free-stream flow conditions.

which is made of 17563 grid-points and 34646 triangles; 145 equally spaced grid-points have been placed along the airfoil's profile, whereas downstream of the TE the stream-wise spacing increases according to an hyperbolic tangent distribution. The smaller frame of Fig. 8.16a clearly reveals that the background triangulation has been built by merging a mesh made of rectangular triangles, which covers the near-wall and wake regions, with a fully unstructured triangulation. The height of the near-wall triangles is such that y^+ is around unity along the entire airfoil surface and exponential mesh-stretching has been used in the cross-flow direction to ensure adequate resolution both within the boundary layer and the wake.

Figure 8.17 shows the Mach contours computed by means of S-F; the flow-field is characterized by the presence of a supersonic pocket, which is bounded by the sonic line (shown using a dashed red line) on its upstream side and within the boundary layer and is terminated by a curved shock-wave. The smaller frames of Fig. 8.17 show a zoom of the tip (Fig. 8.17a) and the foot (Fig. 8.17b) of the fitted shock-wave, which is shown using a white solid line. C^- characteristic lines, drawn using orange lines in Fig. 8.17a are seen to be running into the shock from both of its sides. Similarly, C^+ characteristic lines, drawn using blue lines in Fig. 8.17b, also run into the shock from both of its sides. This is because the shock-downstream flow around



(a) Computational mesh.



(b) Boundary conditions.

Figure 8.16. Transonic, turbulent flow past an airfoil: computational domain.

the foot and the tip of the shock is supersonic, as shown in the smaller frames of Figs. 8.17 and also in Fig. 8.18. This mechanism, and its numerical treatment, is identical to that already described in Sect. 4.2, see Fig. 8.1b.

The availability of the pair of shock-upstream and shock-downstream flow states within each shock-point of the fitted discontinuity allows to perform analyses which would not possible when the shock-wave is captured. For example, Fig. 8.18 shows a flow-deflection, δ, versus shock-angle, σ, diagram. Symbols denote (σ, δ) values computed using all pairs of shock-upstream and shock-downstream flow-states, whereas the solid curves have been computed using the analytical oblique-shock relation, see e.g. [206, Eq. (138)], for selected values of the shock-upstream Mach number. Figure 8.18 allows to examine the features of the curved shock-wave while it is traversed from the foot to the tip. Close to the airfoil's surface, the shock is an oblique-shock of the weak type; in particular it is a Mach line at its foot, i.e. $\delta \approx 0$ and $\sigma \approx \mu$. Moving away from the airfoil, the shock-upstream Mach number remains nearly constant ($M_1 \approx 1.41$) at first, then starts decreasing and the transition from a weak to a strong shock occurs in the region indicated by the orange square, where $M_1 \approx 1.33$ and the downstream flow becomes subsonic. The shock-upstream Mach number further decreases and the shock becomes a normal shock , i.e. $\sigma = \pi/2$, when $M_1 \approx 1.17$ (see the circle in the Mach field). Finally, the flow deviation through the shock increases, the transition from a strong to a weak shock takes place and close

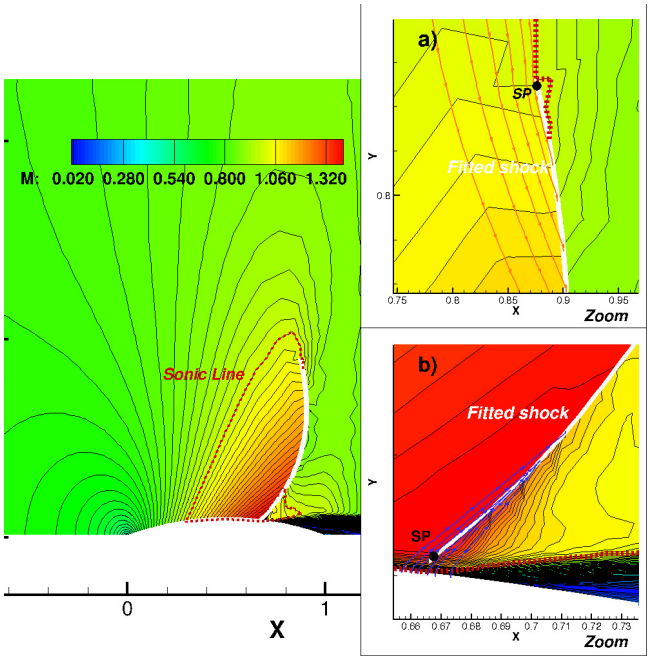


Figure 8.17. Transonic, turbulent flow past an airfoil: S-F calculation with detail of the foot and the tip of the shock. C^+, resp. C^- characteristic curves are drawn using blue, resp. orange, solid lines; the fitted-shock is the white solid line.

to the tip the shock "dies" as a Mach line.

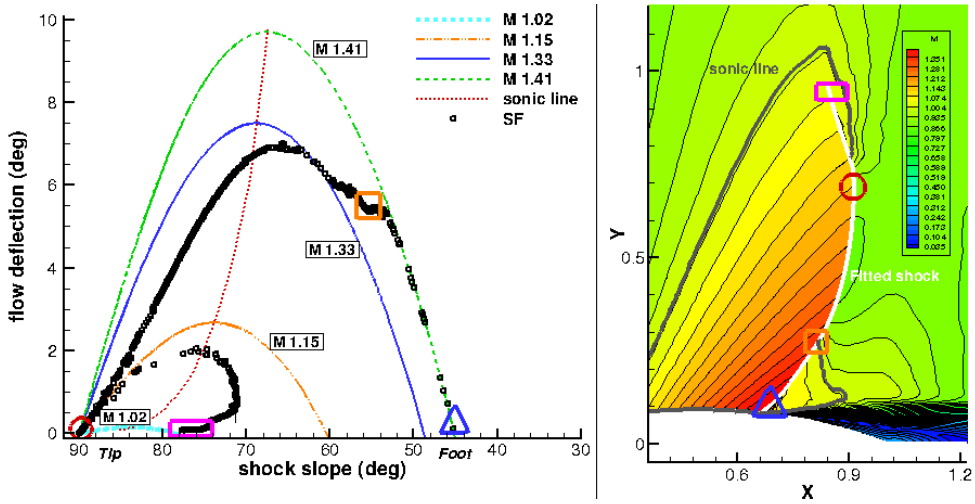
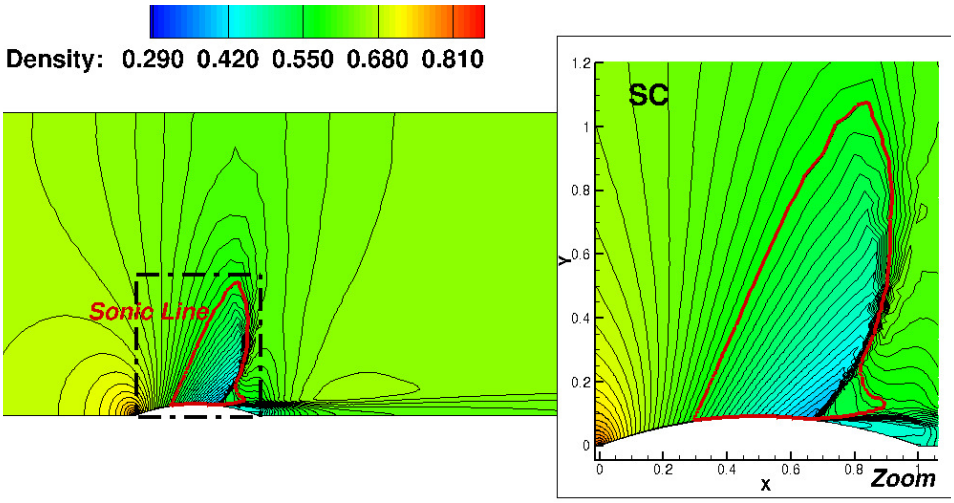


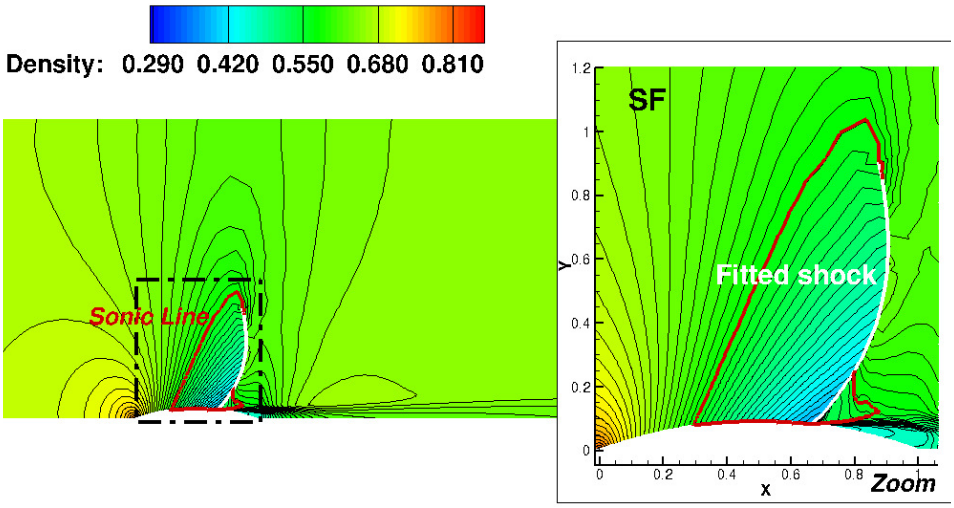
Figure 8.18. Transonic, turbulent flow past an airfoil: flow-deflection, δ, versus shock-angle, σ computed along the fitted-shock; detail of the Mach iso-contour lines within the supersonic pocket.

The comparison between the dimensionless density field (ρ/ρ^0_∞) computed using the two different shock-modeling options is reported in Figs. 8.19 and 8.21. Figure 8.19 shows the entire computational domain and a zoom centered around the supersonic pocket and Fig. 8.21 shows a detail of the region surrounding the foot of

the shock.



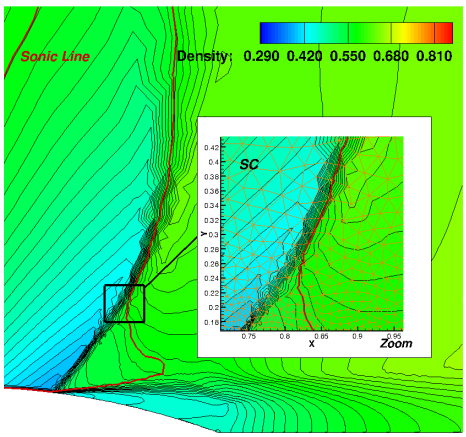
(a) S-C computation



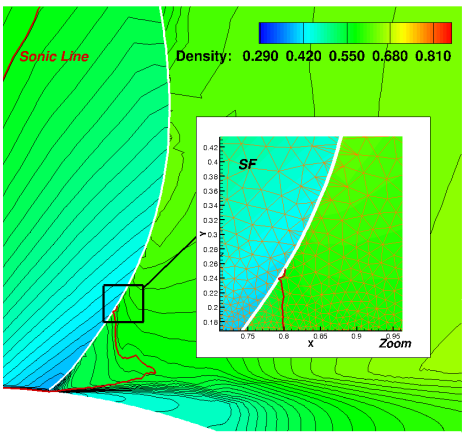
(b) S-F computation. The fitted shock is shown using a white line.

Figure 8.19. Transonic, turbulent flow past an airfoil: dimensionless (ρ/ρ^0_∞) density field.

The smaller frames of Figs. 8.20a and 8.20b allow to compare the different triangulation in the vicinity of the shock: it can be seen that the computational grid used in the S-F calculation is identical to the grid used in the S-C calculation (the background triangulation) except around the fitted shock, where local re-meshing has been used to accommodate the shock-mesh. Indeed the computational and background grids differ by less than 1% in terms of number of grid-points and triangles. The comparison between the smaller frames of Figs. 8.19 and 8.21 clearly shows the improvement in shock resolution provided by S-F. A more quantitative comparison between the S-C and S-F calculations is shown in Fig. 8.22, where the pressure coefficient distribution is plotted along the airfoil's surface. Also shown are the experimental data measured by McDevitt et al. [202, 203, 204, 205]. The



(a) S-C computation



(b) S-F computation. The fitted shock is shown using a white line.

Figure 8.21. Transonic, turbulent flow past an airfoil: dimensionless (ρ/ρ^0_∞) density field and underlying triangulation around the shock wave.

two different shock-modeling options show nearly identical distributions of the pressure coefficient, both of which are in fair agreement with the experimental data. A visible difference between the experimental and computational data has to do with the location of the shock wave, which is predicted a little bit downstream of the experimental location in both the S-C and S-F simulations. The origin of the discrepancy between simulations and experiments appears to be rooted in the turbulence model being used. This has been assessed by running two additional S-C calculations on the background mesh using the CFD++ commercial software and both the S-A and k-ϵ models. These results have also been plotted in Fig. 8.22 and show that the CFD++ calculation with the S-A model predicts the same shock position also found in our calculations, whereas the k-ϵ turbulence model locates the shock even further downstream.

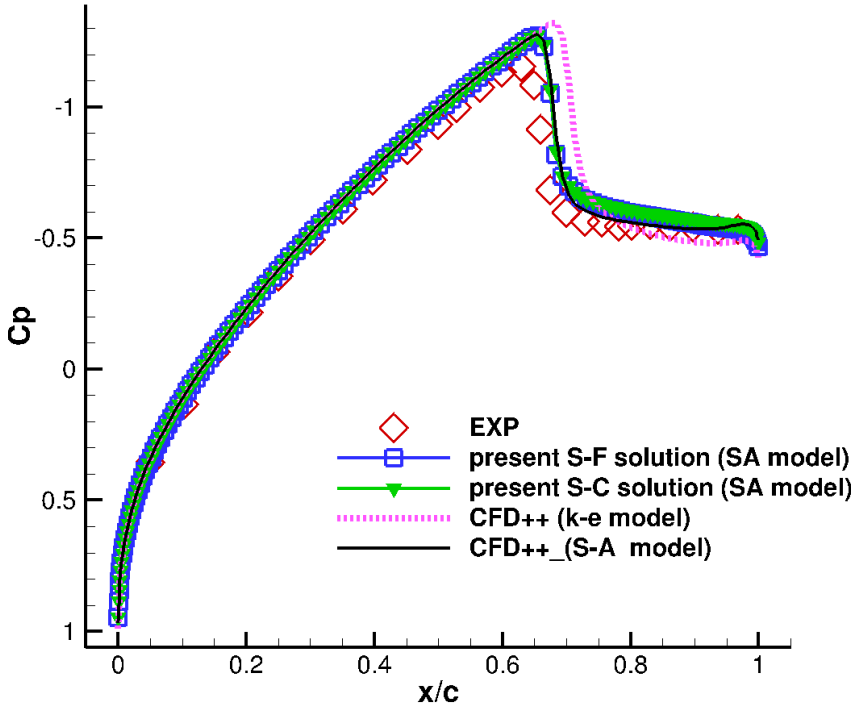


Figure 8.22. Transonic, turbulent flow past an airfoil: pressure coefficient distribution

8.3 Summary

The S-F technique for unstructured meshes presented in Chapter 4 has been successfully extended to laminar and turbulent viscous flows featuring SWBLIs. Compared to the technique originally developed for inviscid flows, the required changes are algorithmically simple and computationally inexpensive. The technique has been applied to three flow configurations representative of different SWBLI patterns, two laminar and one turbulent. Comparisons have been made, using nearly identical grids, between S-C and S-F; in all cases the computed results have also been compared against the available experimental data. A grid-convergence analysis, using Richardson Extrapolation, has been conducted for one of the laminar test-cases: it reveals that S-F features a measured order-of-convergence which is closer to design order than that observed in the S-C calculations. Shock resolution is also clearly enhanced in the S-F calculations as long as the comparison with S-C is made on meshes of nearly identical resolution.
In the laminar flow cases the comparison between the experimental data and the grid-independent S-F solutions shows very good agreement between simulations and measurements.
In the turbulent case the wall pressure distribution computed using the same turbulence model and the two different shock-modeling options shows negligible differences; indeed, numerical tests using different turbulence models seem to indicate that, at least for the selected test-case, turbulence modeling has a larger impact than shock-modeling on the computed wall pressure distributions, in particular the

shock-location. Two final comments are in order concerning turbulent flows. The first one has to do with shock-topology: even if the algorithmic ingredients described in this Chapter may be adequate for modeling most 2D SWBLIs, additional modeling might be required to cope with some more peculiar SWBLIs, such as those described in [207]. In certain circumstances, which depend upon the Mach and Reynolds numbers, the shock may exhibit ripplings, or even holes, and these features would certainly represent a challenging task for the S-F technique described here. In the context of RANS modeling, it would also be interesting to assess the impact of using the "true" jump relations [56], rather than the simplified approach followed in this Chapter.

Chapter 9

Analysis of the transonic flow past a NACA0012 using a shock-fitting technique

Contents

Slender profiles flying at nearly sonic speed exhibit a peculiar shock-pattern which is commonly referred to in the literature as the "fishtail" shock-structure. This can be clearly seen in Fig. 9.1, which has not been taken from a scientific paper, but from a 1950s documentary, titled "Transonic flight" [208]. Fishtail consists in two oblique shock waves that originate at the trailing edge (TE) and a nearly normal shock wave standing at some distance behind the TE: the nearly normal shock joins the two oblique shocks in two interaction points, each of these hereafter referred to as a "triple-point", because of the confluence of three different shock waves, as also shown in the sketch of Fig. 9.2.

The formation of a triple-point is more often associated with the reflection of a shock wave from a solid surface, a phenomenon that can be viewed under two different perspectives: either *i*) as a steady phenomenon, whereby an oblique shock wave impinges on a wall or as *ii*) an unsteady phenomenon, whereby a moving normal shock wave encounters a wedge. Although these two phenomena may seem very different, they are in fact quite similar, because it can be shown [210] that using a change of variables the unsteady problem can be turned into a pseudo-stationary one. Over 140 years ago, Ernst Mach, see the review by [211] on his and related experiments, showed that in the unsteady reflection of a shock wave

Figure 9.1. Transonic fishtail: experimental visualization re-printed from [208].

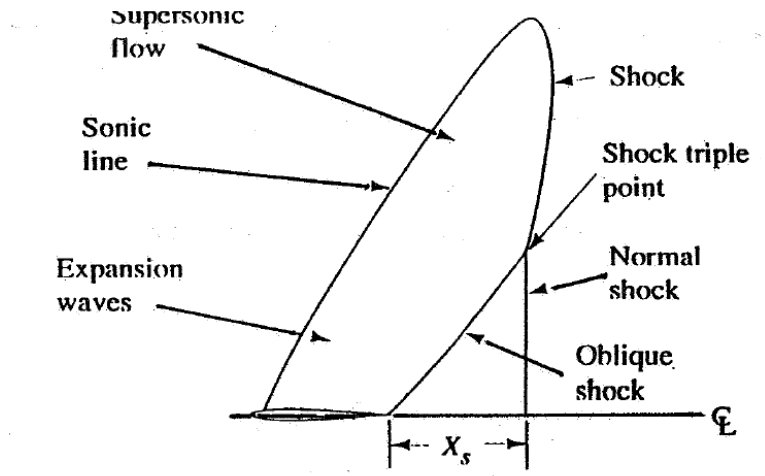


Figure 9.2. Transonic fishtail: sketch of the flow pattern and definition of test case AGARD03, reprinted from [209], with permission.

from a wedge there exist two different types of reflections: one regular and one irregular. In regular reflection, the oblique incident shock is reflected as an oblique shock wave of the opposite family, and the incident and reflected shocks share an interaction point located on the wall. On the contrary, the irregular reflection is characterized by a triple-point located close to (but not on) the wall where three different shock waves meet: the incident shock, the reflected shock and a strong shock, called the Mach stem, which reaches the solid surface. Von Neumann[212], see also [213], proposed an analytical model, hereafter referred to as the three-shock theory (3ST) to compute the flow around the triple-point. However, von Neumann's model is not able to explain some cases of weak[1] reflections where the presence of the triple-point is observed experimentally and numerically, but the 3ST fails to predict; this inconsistency has given rise in the literature to the so-called "von Neumann paradox", which has been the subject of several theoretical, experimental

[1] in the present context the term "weak"refers to the strength of the I-shock, see [50, § 11.11] for a more precise taxonomy.

and numerical studies that will be recalled in Sect. 9.4. The aim of this Chapter is to show that the conditions of the Von Neumann paradox also occur at the triple points of the transonic fishtail and to provide an accurate modelling of the interaction region using the S.F technique presented in Chapter 4.

9.1 Transonic fishtail and the three-shock theory (3ST)

Despite the fact that then fishtail shock-pattern is frequently encountered in transonic flows, only a limited number of experimental and numerical studies on the subject can be found in the literature. This is confirmed by the fact that it is not easy to find clean experimental pictures of the fishtail and Fig. 9.1 is several decades old. As far as computational studies are concerned, the transonic flow past a NACA0012 profile at free-stream conditions $M_\infty = 0.95$ and $AoA = 0°$, which gives rise to a fishtail shock-pattern, has been included as test-case AGARD03 among those proposed on the occasion of the ICASE/LaRC Workshop on Adaptive Grid Methods in 1995 [209]. Figure 9.2, which has been re-printed from the Workshop proceeding, shows a sketch of the flow-field. It is worth underlining that what is indicated in Fig. 9.2 as a "normal shock" cannot be a normal shock along its entire length and, therefore, it will be hereafter referred to as a "nearly normal" shock. The reason will be clarified in Sect. 9.4.1. Among the workshop contributors who engaged in test-case AGARD03, Richter and Leyland [214] examined the effect of the far-field location on the aerodynamic forces and the position, X_s, of the nearly normal shock. Maybe surprisingly, they found that even using feature-adapted meshes, the shock-induced drag was converged to four significant digits when the far-field is 300 chords away from the profile, but X_s keeps changing by almost 2% when the far-field is moved further away to 10,000 chords. Since then, because of its fairly complex shock-interaction pattern, test-case AGARD03 has been the subject of a number of computational studies, see e.g. [215], in particular those dealing with mesh adaptation. Since it can be proved that "Three shocks separating three zones of different continuous states are impossible" [185, § 129], in von Neumann's 3ST not only three shock-waves, but also a contact discontinuity, meet at the triple-point. This is schematically shown in Fig. 9.3, where the nomenclature is inspired by the Mach reflection: I and R are the incident and reflected shocks, respectively, M is the Mach stem and the dashed line is the contact-discontinuity, or slip-stream (SS). The same nomenclature is retained here, but the reader should be aware that the problem investigated in this Chapter does not deal with Mach reflections, but with the three-shocks confluence and triple-point sketched in Fig. 9.2.

The non-linear algebraic equations governing the 3ST, which can be found in either [185, § 135] or [216, § 1.3.2], consist in the Rankine-Hugoniot (R-H) jump relations for all three shocks, supplemented with the condition of parallel streams and equal pressure across the contact discontinuity. It is found that the number of unknowns matches that of the available equations once three parameters are given; these are: *i*) the adiabatic index, γ, of the gas (which is assumed hereafter to behave like a perfect gas); *ii*) the Mach number, M_1, ahead of the I-shock and *iii*) a measure of the I-shock strength, such as the I-shock angle, σ_{12}. A powerful graphical tool for finding solutions to von Neumann's 3ST consists in looking for intersections between

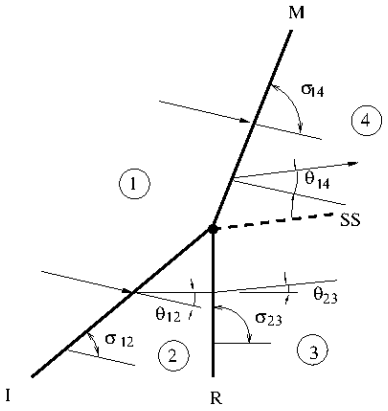


Figure 9.3. Flow in the neighborhood of a triple-point according to von Neumann's 3ST; all angles are relative to the direction of the shock-upstream flow.

the I- and R-shock polars in the pressure-deflection (p, θ) plane; this has been shown in the four insets of Fig. 9.4 for four different pairs of (M_1, σ_{12}) values. The two parameters γ and M_1 define the shape of the I-polar (shown using a dashed line in Fig. 9.4), whereas σ_{12} fixes the position along the I-polar where the knot of the R-polar (shown using a solid line in Fig. 9.4) has to be drawn. This is because there is a one-to-one correspondence between the shock-angle, σ, and the flow deflection through the shock, θ, see [206, Eq. (138)]. The point(s), if any, of mutual intersection between the two polars (marked using a square symbol in Fig. 9.4) identifies both state 3 and 4, see the sketch in Fig. 9.3, because it fulfils the conditions of equal pressure and parallel streams that hold across the slip-stream.

Even if graphical methods are very useful, they may not be quantitatively very accurate. Fortunately, Henderson [217] has shown that finding the intersection(s) between the I- and R-shock polars amounts to compute the real roots of a tenth-degree polynomial in the unknown pressure ratio $\xi_{41} = p_4/p_1 = p_3/p_1$ across the M-shock, i.e. the ordinate of the points marked with a square in the (p, θ) plane of the four insets of Fig. 9.4. Since two real roots are known and have multiplicity two, the polynomial order can be reduced to six. Moreover, not all real roots correspond to flow patterns that can be accepted on physical grounds and the number, m, of physically admissible solutions depends upon the aforementioned three parameters: γ, M_1 and σ_{12}. If a diatomic gas, i.e. $\gamma = 7/5$, is chosen, as done here, Henderson [217] has shown that $m = 0, 1, 2, 3$, depending on the values taken by the remaining two parameters: M_1 and σ_{12}. Therefore, a convenient way to look at the existence and features of the solutions to the 3ST consists in using the (M_1, σ_{12})-plane. This has been done in Fig. 9.4 where the abscissae are restricted to the M_1-range that is relevant to the current application and only a subset of the various curves has been drawn that mark the boundaries between regions where different shock-interaction patterns are observed. The features of the 3ST solution along the four lines drawn in Fig. 9.4, and inside the regions they bound, will now be briefly summarized, whereas the equations needed to draw the lines are given in Appendix C 11.3. The labelling of the various lines and regions follows [50, § 11]. On line 1 the I-shock is a Mach wave and, therefore, there can be no I-shock for points below line 1. On line 8a, the

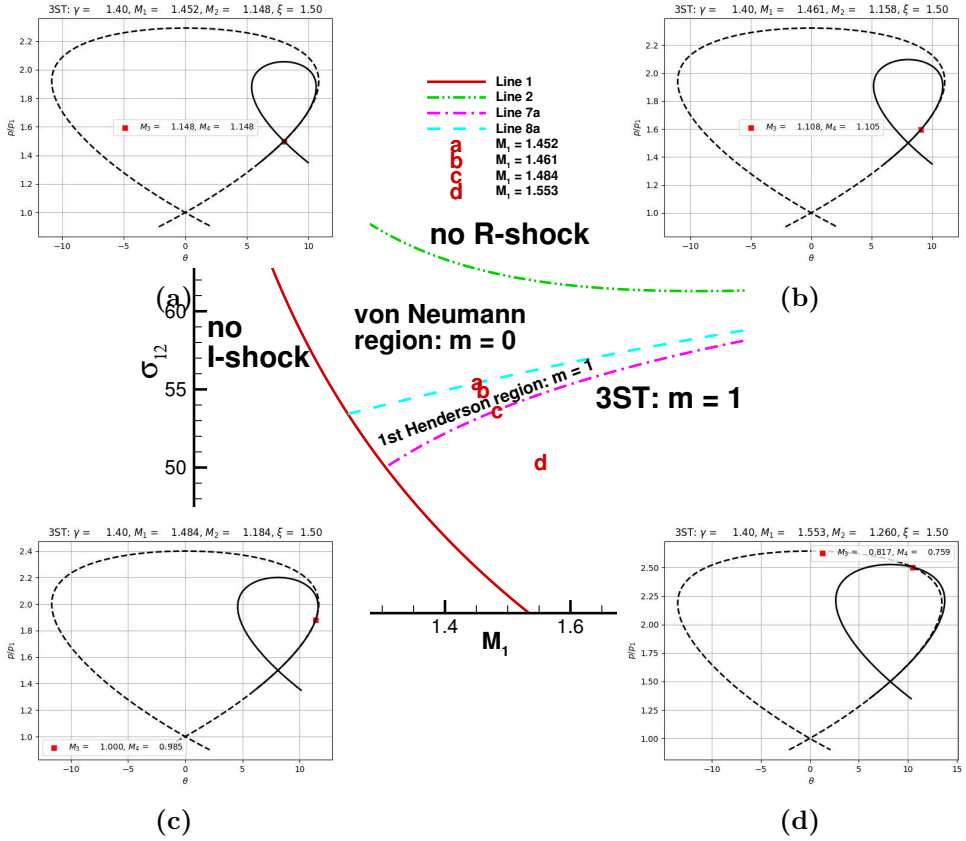


Figure 9.4. Domain of existence of solutions to the 3ST and transition boundaries between various shock-wave configurations in the (M_1, σ_{12}) plane; also shown are pressure-deflection diagrams corresponding to four different pairs of (M_1, σ_{12}) values; (a) on line 8a: the R-shock is a Mach wave; (b) inside the first Henderson region: the flow downstream of the R-shock is supersonic; (c) on the first Henderson line: the flow downstream of the R-shock is exactly sonic; (d) to the right of the first Henderson line: the flow downstream of the R-shock is subsonic.

intersection between the two polars takes place at the knot of the R-shock polar, see Fig. 9.4a. As M_1 increases, line 8a asymptotically approaches line 2, which is where the flow behind the I-shock is sonic. Therefore, shock-reflection is impossible above line 2. On line 7a, which is called the first Henderson line, the flow behind the R-shock is exactly sonic; see Fig. 9.4c. Henderson [217] has shown, see also [50, § 11.11], that there are no solutions to the 3ST, i.e. $m = 0$, inside the so-called von Neumann region, which is the region above line 1, below line 2 and bounded to the right by line 8a. Lines 1, 8a and 7a bound the first Henderson region, where $m = 1$ and the flow behind the R-shock is supersonic, see Fig. 9.4b; to the right of line 7a: $m = 1$ and the flow behind the R-shock is subsonic, see Fig. 9.4d. The case when there is more that one solution to the 3ST, i.e. $m > 1$, will not be discussed because it occurs when M_1 takes values larger than those of interest here. As mentioned earlier, the confluence of three shock-waves in a triple-point has been observed experimentally, see e.g. [218, 219], also for values of (M_1, σ_{12}) that fall inside the

von Neumann region. This contradiction between experimental evidence and von Neumann's 3ST is referred to in the literature as the von Neumann paradox [48].

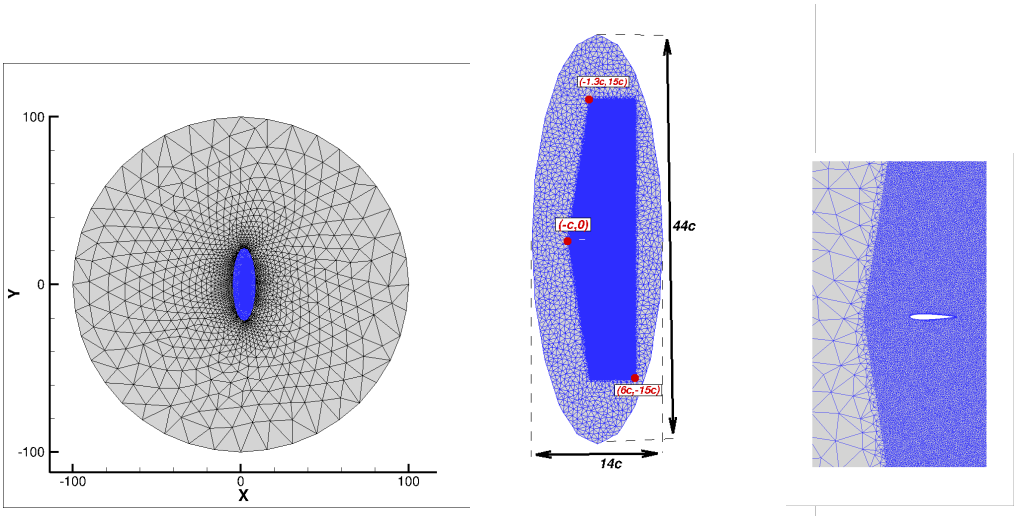
9.2 Numerical simulations of the transonic fishtail: an unexpected failure

The original goal of the present study consisted in simulating the transonic fishtail flow by means of the unstructured S-F technique described in Chapter 4, which allows also to employ a hybrid operational mode whereby only some of the discontinuities are fitted, while all others are captured. Even if hybrid simulations are not as accurate as the fully-fitted ones, see [36, 109], they turned out to be very useful for gaining a better understanding of the fishtail phenomenon, as will be explained in Sects. 9.3 and 9.4. In contrast with the hybrid S-F simulation, the term fully-fitted refers to a simulation whereby not only all discontinuities, but also the point singularities formed by the intersection of discontinuities, or where discontinuities meet a boundary, are fitted. Modelling the point of interaction amounts to define 2D Riemann problems with constant states within a finite number of wedges surrounding the point, see the sketch in Fig. 9.3. Different models need to be implemented to handle different topologies, such as the regular reflection from a wall, the collision between shocks of the same or different families, etc. In particular, for computing the AGARD03 test case was used the von Neumann's 3ST to model the triple-point arising in Mach reflections: examples and algorithmic details are given in [178, 23, 109]. Moreover, an approach similar the one proposed in the aforementioned studies for dealing with the triple-point arising in Mach reflections had long ago been proposed in the front-tracking community by [220].

For comparison purposes, the AGARD03 test-case was computed using both the S-F and S-C approaches. Figure 9.5 shows the computational domain and the triangular mesh. The leading edge (LE) of the NACA0012 airfoil is made to coincide with the origin of a Cartesian reference frame and the reference length is set equal to the chord, c, of the profile. The outer boundary is a circle of radius $R/c = 100$ centered at the origin; R was chosen based on the computations performed in [214] with the aim of finding a compromise between keeping a reasonable computational cost and obtaining an almost mesh-independent location of the nearly normal shock. As shown in the three frames of Fig. 9.5, three levels of decreasing mesh spacing have been used over the computational domain. The finest mesh spacing ($h/c = 0.0421$), see Figs. 9.5c and 9.5b, covers the region surrounding the airfoil and inside the supersonic pocket, whereas a coarser mesh spacing ($h/c = 0.6285$) has been used within the elliptical region of Fig. 9.5b. The Triangle code [221] has been used to create the triangulation shown in Figs. 9.5c and 9.5b, whereas the frontal/Delaunay mesh generator described in [222] has been used to bridge the boundary of the elliptical region with the far-field boundary, see Fig. 9.5a. The overall mesh is made of 139570 grid-points and 278753 triangles.

Three different sets of simulations were run using the `UnDiFi-2D` code: S-C, hybrid and fully-fitted S-F. The grids used in the three sets of calculations are nearly identical, differences being limited to the neighbourhood of the fitted discontinuities.

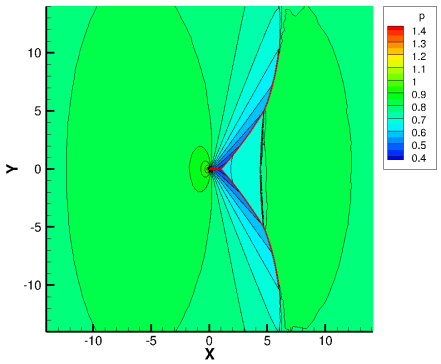
In the S-C calculation none of the shocks is fitted, but all shocks are captured.



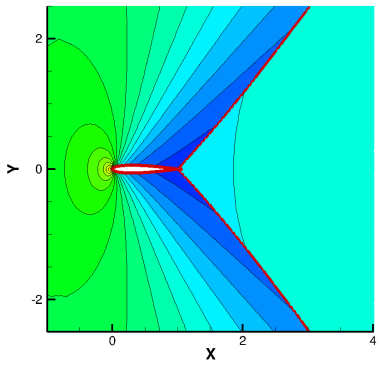
(a) Full computational domain. **(b)** Mesh inside the elliptical region. **(c)** Mesh around the airfoil.

Figure 9.5. Triangular mesh used for the AGARD03 test-case.

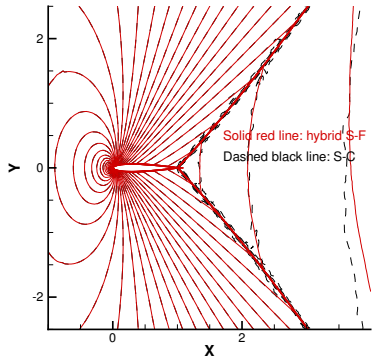
In the hybrid S-F calculation the two oblique shocks (each corresponds to both the I- and M-shocks of Fig. 9.3) originating at the TE are fitted, whereas the nearly normal shock (which corresponds to the R-shock of Fig. 9.3) are captured. It is important to emphasize that in the hybrid simulation von Neumann's 3ST is not used to model the triple-point; it is instead left to the CFD solver to "capture" the interaction between the nearly normal shock and the two "fitted" oblique shocks. This implies that the triple-point is not a geometrical point, but rather a region whose size is comparable to the numerical thickness of the nearly normal shock. The pressure field computed in the hybrid simulation is shown in Figs. 9.6a and 9.6b: to improve readability, the two fitted oblique shocks are marked using a solid (red) line even if their numerical thickness is zero. In contrast, the nearly normal shock downstream of the TE can be identified in Fig. 9.6d by the build-up of iso-pressure contour lines, which reveal that the numerical thickness of the captured shock is finite, spanning two to three cells. The comparison between the S-C (dashed lines) and hybrid S-F (solid lines) calculations is shown in Figs. 9.6c and 9.6d: it clearly reveals that the pressure iso-contour lines obtained from the two different simulations are superimposed, except along the oblique shock, because of its very different thickness in the two sets of calculations. Finally, in the fully-fitted S-F calculation, none of the discontinuities is captured, but all three shocks and the contact discontinuity that meet at each triple-point are fitted and the motion of the triple-points is modelled Since the fish-tail structure is a steady pattern, it is expected that the triple-point velocity computed by the model asymptotically vanishes and a steady configuration is reached, similar to those computed by either S-C or hybrid S-F. Despite numerous attempts, however, no stationary solution was obtained using the fully-fitted mode: when re-starting the calculation from the hybrid S-F solution, the two triple-points start moving along the two oblique shocks towards the TE without ever stopping.



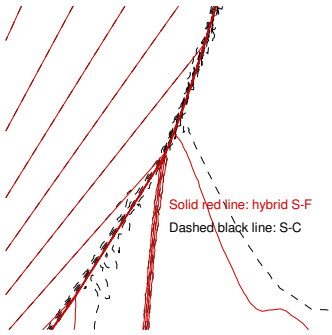
(a) Hybrid S-F: overall view.



(b) Hybrid S-F: close-up around the airfoil.



(c) Hybrid S-F vs. S-C: close-up around the airfoil.



(d) Hybrid S-F vs. S-C: close-up around the triple-point.

Figure 9.6. AGARD03 test-case: pressure field computed by S-C and hybrid S-F.

9.3 The reasons for the failure

The reasons behind the S-F technique failure to perform a full-fitted computation become clear by evaluating intersections in the (p, θ) plane between the polars of the oblique shock and the nearly normal shock in the neighbourhood of the triple-point, using data obtained from the hybrid S-F calculation. As already mentioned in Sect. 9.2, in the hybrid S-F simulation the triple-point is not a geometrical point modelled using von Neumann's 3ST, but rather a region of finite width resulting from the interaction between the fitted oblique shock and the captured nearly normal shock, respectively labelled I- and R-shock in Fig. 9.7a. Figure 9.7a shows the computed pressure field in the vicinity of the triple-point; square symbols mark the location of the grid-points inside the computational domain and on the fitted shock. This is because in the unstructured S-F algorithm (see Sect. 4.2) the discontinuities are made up of pairs of grid-points that share the same geometrical location, but store different values of the dependent variables, corresponding to the upstream

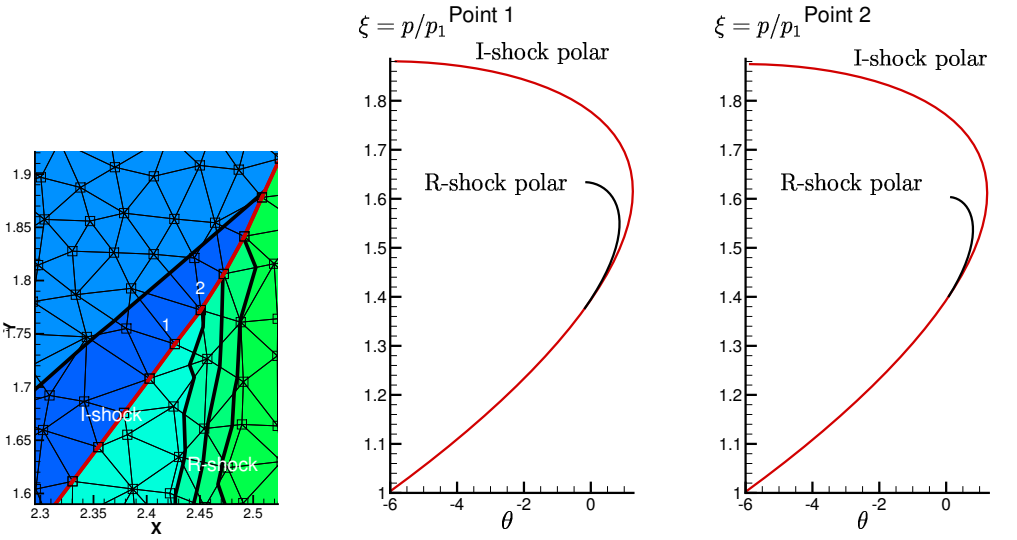
and downstream side. On each of the two sides of a discontinuity, grid-points are connected by straight edges, thus forming an internal boundary of zero thickness: the solid red line in Fig. 9.7a. Within each pair of grid-points along a fitted discontinuity the unit normal vector is computed using finite difference formulae [108] and the upstream and downstream values of the dependent variables satisfy the R-H jump relations. Therefore, all shock-related quantities, such as upstream and downstream Mach numbers, flow deflection through the shock and shock slope, are readily available along a fitted discontinuity. Since the location of the triple-point cannot be unambiguously defined in the hybrid S-F simulation, two grid-points are picked up (1 and 2 in Figs. 9.7a) along the fitted I-shock that are close to (and upstream of) the location where the captured quasi-normal R-shock (which can be identified by the black iso-pressure lines in Fig. 9.7a) joins the I-shock. Inspection of the I- and R-shock polars in both grid-points, shown in Figs. 9.7b and 9.7c, reveals that there is no intersection between the two curves, despite the fact that both the S-C and hybrid S-F simulations feature a shock-interaction pattern that resembles the confluence of three shock-waves. This is the same situation leading to the von Neumann paradox in the case of shock reflection problems. Lack of a point of intersection will be further confirmed in Sect. 9.4.1 by looking at the location within the (M_1, σ_{12}) plane of those grid-points that are located along the fitted oblique shock.
The anomalous behaviour of the fully fitted solution is also understandable. Since there is no steady solution to von Neumann's 3ST, when operating in fully fitted mode the `UnDiFi-2D` code finds an unsteady solution characterized by a non-zero triple-point velocity, $\mathbf{w}_{\mathrm{TP}}$, which moves the triple-point along the oblique shock towards the TE. Consequently, the relative Mach number:

$$M_1 = \frac{|\mathbf{u}_1 - \mathbf{w}_{\mathrm{TP}}|}{a_1}$$

increases, thus moving the point representative of the flow conditions in 1 into a region of the (M_1, σ_{12}) plane where a solution to the 3ST exists.

9.4 Solutions to the von Neumann paradox

Over the past 60 years, various researchers have proposed explanations of the von Neumann paradox and possible solutions to solve it. For instance, Sternberg [188] suggested that in the parameter range where the von Neumann paradox occurs, the effects of viscosity cannot be neglected around the triple-point and, therefore, the shock waves and the mixing layer that forms between the M- and R-shocks cannot be treated as discontinuities of negligible thickness. Other researchers, see e.g. [223, 224, 225] proposed, giving different motivations, to relax the conditions of parallel flow and identical pressure that hold across the slipstream in the 3ST. However, all the aforementioned solutions are anchored with the topology proposed by von Neumann which foresees the presence of three shock waves and one slip-stream branching at the triple-point. As mentioned in Ref. [226], Bargmann and Montgomery [227] were the first to propose the addition of an isentropic expansion at the triple-point to obtain a solution to those cases not covered by the 3ST. Two years later, the four-waves theory (4WT, the fourth wave being the centred expansion fan)

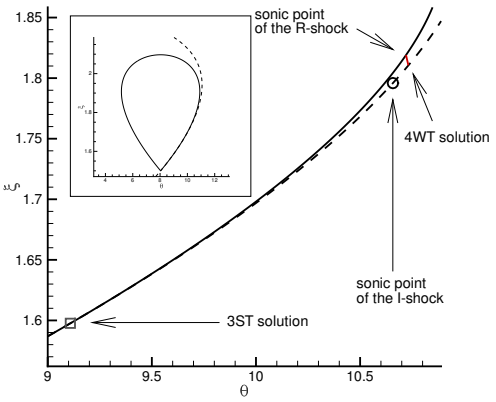


(a) Pressure field in the triple-point region. **(b)** I- and R-shock polars of grid-point 1. **(c)** I- and R-shock polars of grid-point 2.

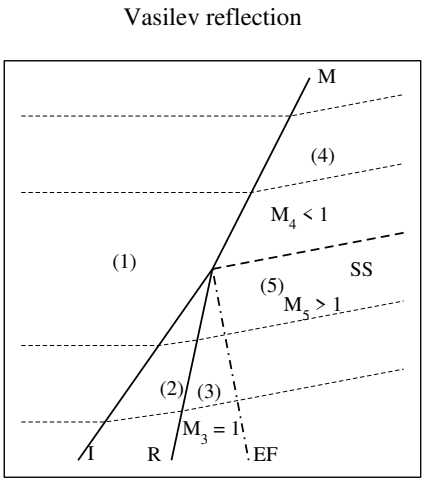
Figure 9.7. AGARD03 case: hybrid S-F calculation; pressure field and shock polars close to the triple-point.

was further elaborated in Ref. [49]. Unfortunately, the two aforementioned technical reports seem to be unavailable nowadays, but the four-waves model re-surfaced years later in Guderley's book on transonic flow [228, pag. 147], where it is suggested that a complex supersonic patch develops behind the R-shock, originating from the expansion fan centered at the triple-point. Guderley's homentropic model did not include the contact discontinuity which was added, fifteen years later, in Ref. [229]. This is the earliest occurrence that was found in the literature where the modified Guderley's model (i.e. the non-homentropic 4WT including the contact discontinuity) is described. As far as the conditions prevailing ahead of the expansion fan are concerned, it is stated in [229] that "We may therefore conclude, with some confidence, that the tail[2] shock flow will be sonic at the triple-point." Therefore, the flow-pattern at the triple-point according to the modified Guderley's 4WT can be sketched as in Fig. 9.8b and 9.8d: the slope of the R-shock is such that the downstream flow is exactly sonic and an expansion fan centred at the triple-point accelerates the flow from sonic to supersonic speed on one side of the contact discontinuity. On the other side of the contact discontinuity the flow can be either subsonic or supersonic; in the former case the flow pattern is referred in Ref. [216] as a Vasil'ev reflection (VR), whereas in the latter as a Guderley reflection (GR). The non-linear algebraic equations governing the 4WT are reported in Appendix 11.3, where the two sets of data used to draw the sample VR and GR of Fig. 9.8 are listed and it is also described how to compute the boundary between the GR and VR. The two shock-patterns drawn in Figs. 9.8b and 9.8d reproduce the actual slopes computed using the 4WT;

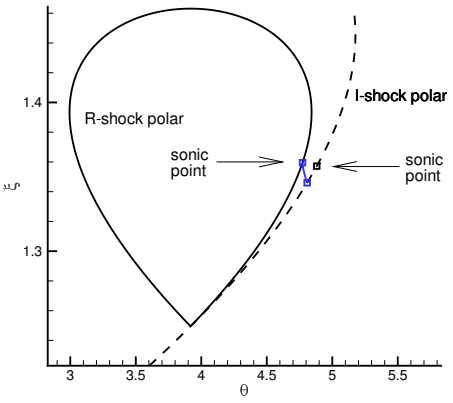
[2] the tail shock corresponds to the R-shock of Fig. 9.3.



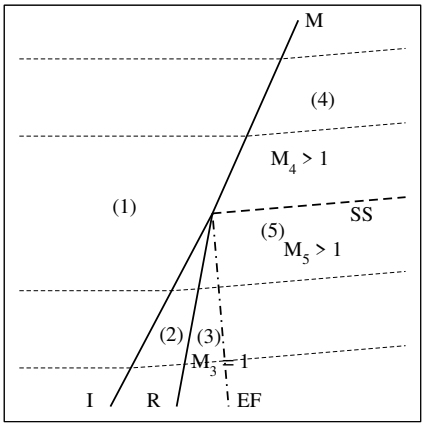
(a) Vasil'ev reflection: $M_4 < 1$.



(b) Shock-pattern in the Vasil'ev reflecction.



(c) Guderley reflection: $M_4 > 1$.



(d) Shock-pattern in the Guderley reflection.

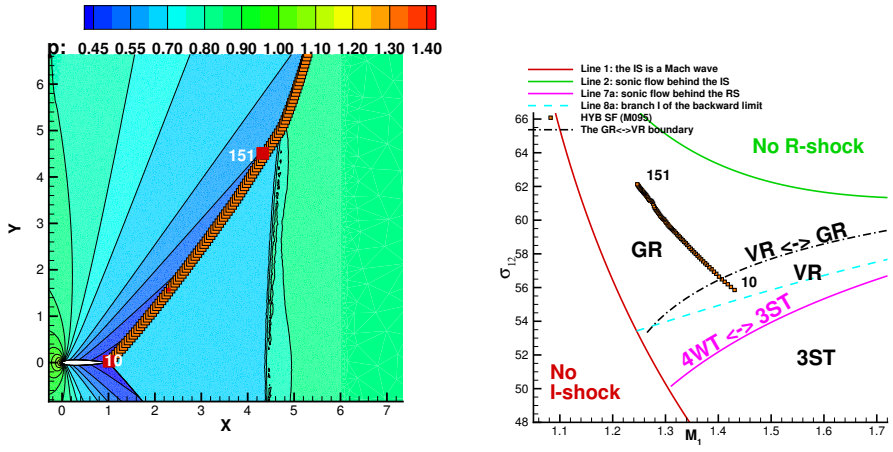
Figure 9.8. Vasil'ev and Guderley reflection: shock polars in the (p, θ) plane and shock pattern in the (x, y) plane.

observe that the head and the tail of the expansion fan (EF) cannot be distinguished, because the flow undergoes a tiny deviation across the EF and that the head of the EF is perpendicular to the streamlines in region 3, because the flow is sonic behind the R-shock. In the (p, θ) plane the 4WT amounts to connect the sonic point of the R-shock polar with the I-shock polar using a Prandtl-Mayer expansion; this is shown in Figs. 9.8a and 9.8c for the VR and GR, respectively. According to [216] and [230], the transition between the 3ST and 4WT takes place along the first Henderson line, or line 7a in Fig. 9.4, where the Mach number behind the R-shock in the 3ST is sonic. Inside the first Henderson region, where the flow behind the R-shock is supersonic, both 3ST and 4WT solutions are possible; an example is shown in Fig. 9.8a, which refers to point b in Fig. 9.4. According to the same authors [216, 230], however, the

3ST solutions inside the first Henderson region should be discarded as un-physical, because of the supersonic Mach number behind the R-shock, and should be replaced by the 4WT. Therefore, the 3ST applies to the right of the first Henderson line and the 4WT to its left. Moreover, solutions to the 4WT also exist to the left of line 8a, i.e. inside the von Neumann region, which implies that the 4WT offers a solution to the von Neumann paradox. Wind-tunnel visualizations obtained by [231], see also [232, 233], using an ad-hoc designed experimental facility seem to confirm the complex flow structure of Guderley's model, in particular the existence of the expansion fan. The earliest calculations showing a Guderley-type reflection pattern are those performed by [234, 235] solving the Euler equations. Similar results were obtained for the Euler equations in [236, 237] and calculations showing an interaction pattern compatible with Guderley's model have also been obtained when numerically solving the unsteady transonic small disturbance equations, see [238], the nonlinear wave system (a simplified version of the isentropic Euler equations), see [239], and the shallow waters equations, see [240, 241]. Two remarkable features of Guderley's solution are the smallness of the supersonic patch, which possibly explains why it has for long gone unnoticed both in experiments and simulations, and the apparent existence of not a single, but rather an array of supersonic patches of decreasing size, see [240, 237]. These features make the numerical simulation of the GR/VR very challenging and it is therefore not surprising that some of the aforementioned authors choose shock-fitting/front-tracking algorithms, rather than shock-capturing, to handle some of the discontinuities involved in the Guderley-type reflection.

9.4.1 Analysis of the hybrid shock-fitting solutions

Making use of the taxonomy previously discussed, the positioning of the hybrid S-F solutions of the fishtail problem within the (M_1, σ_{12}) plane can be evaluated. By extracting the (M_1, σ_{12}) values along the I-shock (the oblique shock), starting near the TE of the airfoil and up to a location (labelled 151 in Fig. 9.9a) close to where the captured R-shock (the nearly normal shock) joins the fitted I-shock (the oblique shock). The numbering of the grid-points along the I-shock is shown in Fig. 9.9a on top of the computed pressure field. Also shown in Fig. 9.9b is the first Henderson line, which marks the boundary between the 3ST (to its right) and the 4WT (to its left) and the boundary separating the Vasil'ev ($M_4 < 1$) and Guderley ($M_4 > 1$) solutions to the 4WT. Inspection of Fig. 9.9b clearly reveals that the 3ST cannot be applied to describe the three-shocks confluence taking place at the triple-point of the fishtail. Indeed, those points along the I-shock that are closer to the triple-point (e.g. grid-point 151 in Fig. 9.9a) are those further away from the first Henderson line, the leftmost boundary of the (M_1, σ_{12}) plane where (physically plausible) solutions to the 3ST exists. Figure 9.9b suggests that the three shock confluence that characterizes the fishtail structure should instead be modeled using the 4WT. Therefore, the solution to the 4WT was computed using the (M_1, σ_{12}) values at point 151 of Fig. 9.9a; the results are reported in Appendix B 11.3, as well as in Fig. 9.8c, where the 4WT solution in shown in the (p, θ)-plane. The same analysis described so far for the AGARD03 test-case has been repeated for decreasing values of the free-stream Mach number down to $M_\infty = 0.91$. The corresponding Mach iso-lines computed using the hybrid S-F approach are shown

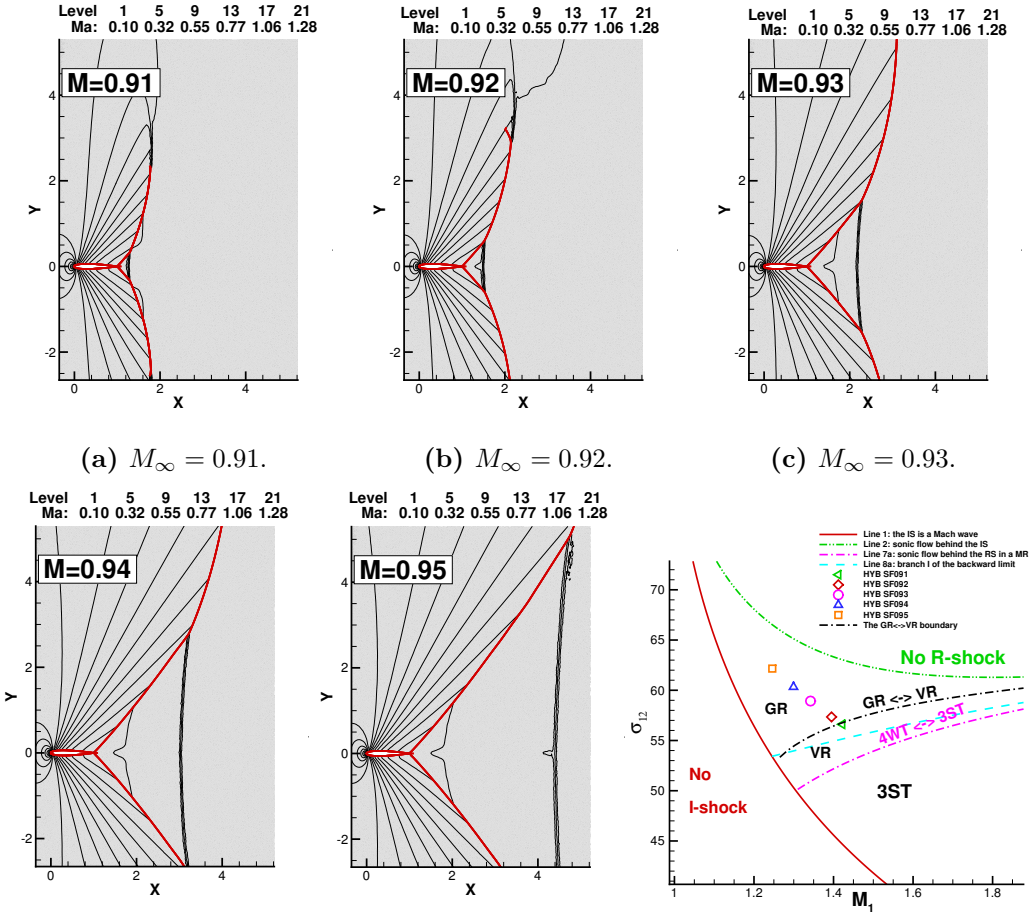


(a) Pressure field and numbering of the grid-points along the I-shock.

(b) Location within the (M_1, σ_{12}) plane of the values computed along the I-shock.

Figure 9.9. AGARD03 test-case: position in the (x, y) and (M_1, σ_{12}) planes of the points located along the fitted I-shock.

in the five frames of Fig. 9.10: it can be seen that the nearly normal shock moves upstream towards the TE of the airfoil when the free-stream Mach number decrease. Similarly to Fig. 9.9b, Fig. 9.10f shows the position within the (M_1, σ_{12}) plane of only one grid-point, picked-up along the I-shock and close to the triple-point, for each of the five different free-stream Mach numbers examined Figure 9.10f reveals that by decreasing the free-stream Mach number the point on the I-shock close to the triple-point approaches, but never reaches line 8a, always remaining inside the von Neumann region. It follows that regardless of the free-stream Mach number, the triple-point of the fishtail cannot be described using the 3ST, but the 4WT should be used instead. Figure 9.10f also shows that four out of the five points corresponding to different values of the free-stream Mach number fall inside the GR and only the one corresponding to $M_\infty = 0.91$ inside the VR region. However, this does not allow to draw definitive conclusions about the type of reflection taking place at the triple-point, because its location is only approximate in the hybrid S-F simulation and a fully-fitted calculation (implementing the 4WT) would be needed to gain further insight. Furthermore, it is possible to provide explanations on why the R-shock is referred as a nearly normal shock, rather than "normal shock", as it is done in Fig. 9.2. Using the 4WT, the slope of the R-shock is such that the downstream flow is exactly sonic; when M_2 is slightly larger than one, as it happens to be the case here, $\sigma_{23} \approx \pi/2 - \sqrt{\left(M_2^2 - 1\right)/2}$, see [50, Eq. (6.4.38)], and [235] has shown that σ_{23} increases very rapidly near the triple-point. Moreover, as long as the airfoil is symmetric and at zero angle-of-attack, the R-shock must be a normal shock on the x-axis. It follows that moving along the R-shock, starting at the triple-point towards the symmetry axis, the corresponding point on the shock-polar moves from the sonic point towards the normal shock point. All these observations point to the fact that the normal shock of Fig. 9.2 is close to being a normal shock along most of



(a) $M_\infty = 0.91$. (b) $M_\infty = 0.92$. (c) $M_\infty = 0.93$.

(d) $M_\infty = 0.94$. (e) $M_\infty = 0.95$. (f) (M_1, σ_{12}) plane.

Figure 9.10. Hybrid SF solutions for different free-stream Mach numbers and position in the (M_1, σ_{12}) plane of the point along the oblique shock closest to the triple-point; fitted shocks are marked using red solid lines.

its length, except close to the triple-point.

9.4.2 Analysis of the full fitted shock-fitting solution

By analysing the Chapman plane and the shock polars it was possible to investigate the nature of the interaction point between the oblique shocks and the quasi-normal shock wave in the fishtail configurations previously described. Specifically, reasons behind the S-F failure to perform a full fitted simulation as done in Ref. [23] were also clarified: indeed, no steady solution can be provided by the S-F technique using the triple point modelling implemented in the *UnDiFi-2D* code, since it is based on the 3ST and it does not correctly represent the Guderley interaction, which is described by the 4WT.

For this reason, to compute the AGARD03 case using the S-F technique presented in Chapter 4, a new interaction point model was added with respect to those

described in Ref. [23] and included in the *UnDiFi-2D* public repository. Basically, the modelling is similar to the one implemented in the *UnDiFi-2D* solver for dealing with the interaction between two shocks of the same family [23]. In particular, the new modelling approach is similar to the one adopted for describing the quadruple point (QP) and it includes the presence of a sonic state upstream the expansion fan: specifically, by referring to Fig. 9.8d, the model involves a total amount of 26 unknowns, which are reported hereafter.

- **State 1,2,3,4,5** 20 variables
- **Slopes of I/R/M/EF**[3] **shocks** 4 variables
- **Quadrupole point velocity components** 2 variables

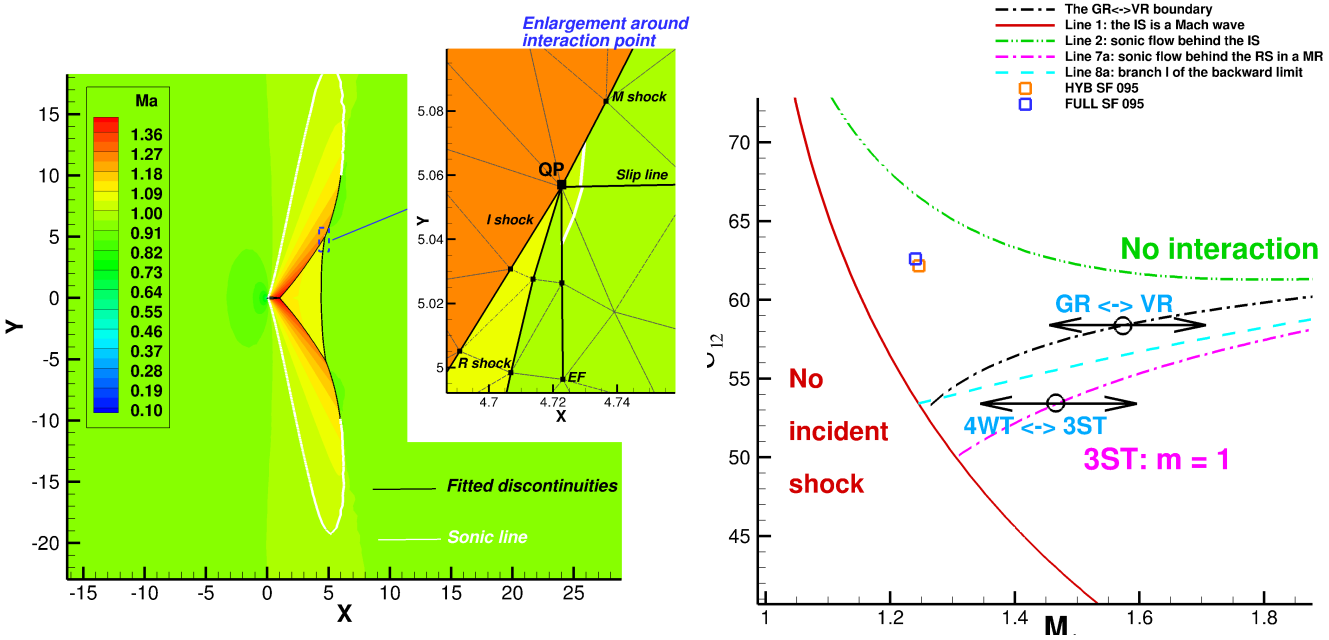
However, by knowing the upstream state (1) and the I-shock slope, the problems unknowns are reduce to 21 variables, which are computed using a two step procedure. The first stage consists of computing state (2) and the R-shock by applying the R-H relations, considering state (1) as the upstream region and state (2) as the downstream one: by doing so, it is possible to partially determine also the QP velocity vector. Indeed, the QP velocity vector $\mathbf{w}_{\mathrm{QP}}$ can be decomposed into two components, resp. normal and tangential to the I shock: this particular choice allows to evaluate the normal component $\mathbf{w}_{\mathrm{QP_n}}$ applying the R-H relations between state (1) and (2), since it is posed equal to I-shock local speed, whereas the tangential component is left unknown. In particular, some consideration can be drawn regarding the R-shock slope: indeed, it is determined when computing the state and the position of the nearest R-shock point to the QP and since state (3) is sonic, this geometrical slope cannot be modified without violating the dependence domain. Therefore, R-shock slope is assigned. By doing so, the effective number of unknowns in the QP model is 15: in detail, the states 3, 4 and 5 (12 unknowns), the slopes of the EF and M shock (2 unknowns) and the QP tangential velocity (1 unknown) computations are required.
At this stage, the QP problem can be solved by applying the following relations:

1. **R-H jump relations (4 eqs.)** between state 1 and 5
2. **R-H jump relations (4 eqs.)** between state 2 and 3
3. **R-H jump relations (4 eqs.)** between state 3 and 4
4. **Imposition of** $M_3 = 1$ **(1 eq.)**
5. **Jump relations for the slip-stream (2 eqs.)** between state 4 and 5, so that $p_4 = p_5$ and $|u_4 \times u_5| = 0$

The aforementioned modelling approach allowed to perform a full fitted computation of the AGARD03 test case and to obtain a steady solution for this flow configuration. The Mach flow field provided by the S-F solution is illustrated in Fig. 9.10, where the fitted discontinuities are depicted using a black solid line. Furthermore, the sonic line is also shown using the white colour. As can be seen in

[3]The expansion fan is modelled as a nearly negligible strength shock (*expansion shock*)

the enlargement around the upper interaction point, the S-F technique is able to reproduce the physical behaviour of the Guderley reflection, whose sketch in also available in Fig. 9.12c, with both the supersonic pockets up and down the slip line. Furthermore, full fitted simulation availability allows to draw final considerations about the type of reflection which occurs the AGARD03 fishtail configuration, by evaluating exactly the QP position within the (M_1, σ_{12}) plane. Indeed, the hybrid S-F computation (see Sec. 9.4.1) provided only an estimation of the interaction point position in the Chapman plane by considering a I-shock point near to the QP, since the R-shock was captured and, as a consequence, the interaction was spread over a region instead of being a point. Anyway, by plotting the QP data computed by the full S-F simulation, it is evident the good agreement with the estimation provided by the hybrid one, as shown in Fig. 9.11b



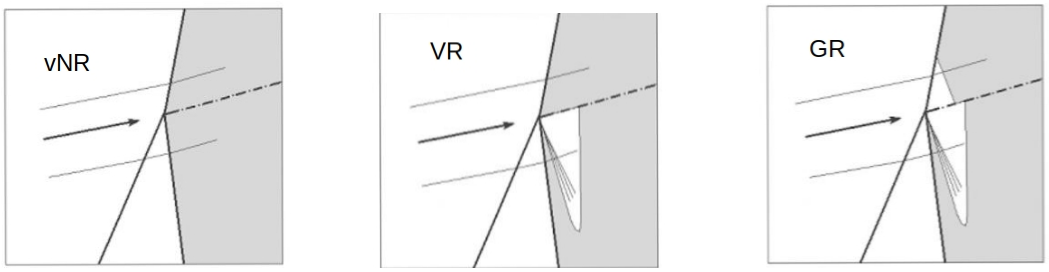
(a) S-F full fitted solution. Mach field and detail around an interaction region)

(b) Chapman plane analysis

Figure 9.11. AGARD03 test-case: S-F full fitted solution and Chapman plane analysis

9.5 Summary

This analysis was motivated by a seemingly incomprehensible failure of the unstructured S-F CFD code (`UnDiFi-2D`) to computing the transonic flow past a NACA0012 airfoil. In previous publications, see [178, 23], `UnDiFi-2D` code successfully used the to simulate triple points of Mach reflections using von Neumann's 3ST. However, `UnDiFi-2D` code, equipped with the same triple-point model, was not able to simulate the two interaction points occurring in the $M_\infty = 0.95$ flow past a NACA0012 profile. On the contrary, solutions of the same test-case are obtained when all three shocks



(a) vNR reflection (3ST) **(b)** Vasil'ev reflection **(c)** Guderley reflection

Figure 9.12. Reflections' sketches (Re-printed from Ref. [216], with permission): grey regions denotes subsonic flow

were captured, but also when two shocks are fitted and the third one is captured. In these latter two calculations, however, the triple-point was not modelled, but captured. Shock-polar analysis revealed that von Neumann's three-shocks theory does not admit solution in the parameter range characterizing the triple-points, whereas the Guderley's four-waves model, which includes a centred expansion fan in addition to these shock waves, is able to provide a solution. Indeed, the model based on the Guderley's 4WT, described in this Chapter and implemented in the `UnDiFi-2D`, allowed to obtain a full fitted simulation of the transonic flow past a NACA0012. It is worth underling that the comparison with the S-C and the hybrid S-F calculations shows almost identical solutions: nevertheless, only the fully S-F computation reveals the presence of a weak expansion fan in the interaction region.

Chapter 10

Conclusions and future work

The primary purpose of this study is to contribute to the development of a new class of gas-dynamic solvers based on shock fitting techniques, able to be an effective option with regards to the shock capturing methods for computing flows featuring shocks.
A part of this Thesis is dedicated to the presentation of different shock fitting techniques, developed and implemented during the PhD program: specifically, different shock-fitting algorithms were described and used in conjunction with unstructured or structured gas-dynamic solvers. As a matter of fact, one of the main features of the shock fitting approaches is the need of a data transfer between the fitted discontinuities and the computational grid, so that the shock-fitting algorithms must be generally tailored on the basis of the grid topology. In the unstructured grids context, analysed in Chapters 4 and 5, the flexibility of these meshes allows to easily constrain fitted discontinuities to be a part of the computational triangulation/tethraelization, so that shocks/contact discontinuities act as special boundary conditions between the smooth regions of the computational domain. This procedure is no longer practicable when structured grid are considered and therefore, in order to design novel shock fitting techniques for structured solvers a different approach was proposed in the present Thesis. This new shock-fitting technique for 2D/3D structured grids, inspired by the shifted boundary methods [31], is widely explained in Chapters 6 and 7. According to this approach, the fitted discontinuity carves a mesh-less hole in the computational domain over which it is floating and data transfer between the discontinuity and the boundaries of the hole and vice-versa relies on extrapolation/interpolation via truncated Taylor series expansions. The main advantage of these new technique is the capability of tracking the discontinuities and computing the surrounding smooth-flow areas without re-meshing around the discontinuity, as required by the shock-fitting schemes for unstructured grids described in Chapters 4 and 5. By doing so, it was possible to compute numerical simulations involving also complex shock interactions both in 2D and 3D space, where for the first time a full fitted computation of a regular oblique shock reflection has been performed.
Furthermore, these advancements in shock fitting methods open up possibilities to simulate several fluid dynamic phenomena by applying the shock-fitting techniques, such as turbulent flows and acoustic sound propagation. In addition to

the advantages highlighted in this Thesis with respect to shock capturing methods in terms of solution quality and accuracy, it is worth to underline that in general shock-fitting allows to achieve a better understanding of the physics behind shock-shock or shock/boundary layer interactions. Specifically, this study collects two particular applications of the unstructured shock fitting technique reported in Chap. 4: the former considers 2D shock/boundary layer interactions both in laminar and turbulent regime, which have been computed for the first time on unstructured grids using a shock fitting technique. In particular, the test cases in Chapter 8 underline that shock fitting is able to provide high quality solutions also on coarse grid levels: moreover, qualitative and quantitative grid convergence analyses performed in this Chapter show the significant achievements in terms of reduced discretization error and improved order of convergence that shock fitting delivers, compared to shock-capturing.
The latter application field regards a deep investigation of a shock/shock interaction characterized by the Von Neumann paradox, where the presence of the triple points is observed experimentally and numerically, even if the three shocks theory fails to predict them. In this regard, Chapter 9 reports an analysis about the transonic flow past a NACA0012, where shocks pattern exhibits a fishtail configuration. By modelling the interaction points via shock fitting, it was possible to verify that the conditions of the Von Neumann paradox occur at the triple points of the transonic fishtail and to compute explicitly for the first time a solution using shock-fitting technique based on the Guderley's four-waves model.
Despite these undoubted steps forward, an issue remains unsolved that still prevents the use of shock fitting solvers as an alternative to shock capturing ones: the capability of treating changes of shocks configuration topology. As pointed out in the previous Chapters, all the shock fitting techniques described in this Thesis require as input the a-priori knowledge of the discontinuity configuration topology, including the discontinuities and shocks interactions types within the flow field. Nevertheless, when dealing with unsteady flows, one must consider possible changes of both the shocks interactions topology and the number of discontinuities, due to the disappearance of weakening fronts and the appearance of new shock fronts. In order to be able to manage all these topological changes it will be necessary to develop new algorithmic tools capable of detecting the occurrence of a change in the shock topology and modify accordingly the fitted discontinuities and their mutual interactions. However, the design of this kind of tool is not trivial and will require to merge different skills and expertise field, as done for the 2D detection technique proposed in the past years by Paciorri and Bonfiglioli in Ref. [164]. Although this problem, several applications of different shock fitting methods developed during this PhD program have been presented in this Thesis, demonstrating that these methods can be effectively used to simulate complex 2D and 3D shocked flows. Specifically, shock fitting techniques have shown to be able to overcome typical problems encountered by shock capturing solvers both in the structured and unstructured context: for instance, the order of accuracy degradation and the presence of spurious waves in the shock downstream regions, as well as the numerical problems linked to the shock capture process. Moreover, the shock fitting techniques described in this Thesis were able to compute numerous inviscid/viscous and laminar/turbulent flows with shock-wall and shock-shock interactions, ensuring results that are qualitatively

superior to those that shock capturing methods deliver, especially when using coarse grids. Therefore, the benefits stated above highly encourage a revival of shock fitting methods and further advancements of these techniques. Moreover, the interesting results, together with the algorithmic development proposed in this Thesis, suggest that the shock fitting approach could become in a short time a viable option to shock capturing one for the simulation of complex shocked flows of aerospace interest.

Chapter 11

Appendices

11.1 Appendix A. Shock-tangent calculation

Derivation of the finite difference approximations

Using a parametric description of the shock-front:

$$\begin{aligned} x &= x(s) \\ y &= y(s) \end{aligned}$$

where s is the curvilinear abscissa (or arc-length) of the curve, the unit tangent vector can be computed as:

$$\boldsymbol{\tau} = \left[x'(s)\,\mathbf{e}_x + y'(s)\,\mathbf{e}_y\right] \tag{11.1}$$

Equations (4.1) are obtained by approximating the first derivatives in Eq. (11.1) using FD formulae. Taylor-expanding about shock-point i and taking into account that:

$$s_i - s_{i-1} = l_{i-1/2} \tag{11.2a}$$
$$s_i - s_{i-2} = l_{i-1/2} + l_{i-3/2} \tag{11.2b}$$
$$s_{i+1} - s_i = l_{i+1/2} \tag{11.2c}$$

one obtains:

$$x_i - x_{i-1} = x'_i\, l_{i-1/2} - \frac{1}{2!}\, x''_i\, l^2_{i-1/2} + \frac{1}{3!}\, x'''_i\, l^3_{i-1/2} + HOT \tag{11.3a}$$
$$x_i - x_{i-2} = x'_i \left(l_{i-1/2} + l_{i-3/2}\right) - \frac{1}{2!}\, x''_i \left(l_{i-1/2} + l_{i-3/2}\right)^2 + \frac{1}{3!}\, x'''_i \left(l_{i-1/2} + l_{i-3/2}\right)^3 + HOT \tag{11.3b}$$
$$x_{i+1} - x_i = x'_i\, l_{i+1/2} + \frac{1}{2!}\, x''_i\, l^2_{i+1/2} + \frac{1}{3!}\, x'''_i\, l^3_{i+1/2} + HOT \tag{11.3c}$$

where HOT denotes the high order terms. The following shorthand notation has also been used:

$$x_i = x(s_i) \qquad x'_i = x'(s_i) \qquad x''_i = x''(s_i) \qquad x'''_i = x'''(s_i)$$

By combining Eqs. (11.3a) and (11.3c) one obtains the centered approximation:

$$\begin{aligned} \boldsymbol{\tau}_i &= \frac{l_{i+1/2}}{l_{i-1/2}\left(l_{i-1/2}+l_{i+1/2}\right)}\boldsymbol{\Delta r}_{i-\frac{1}{2}} + \frac{l_{i-1/2}}{l_{i+1/2}\left(l_{i-1/2}+l_{i+1/2}\right)}\boldsymbol{\Delta r}_{i+\frac{1}{2}} \\ &- \frac{1}{3!}\left(x'''_i\mathbf{e}_x + y'''_i\mathbf{e}_y\right)\, l_{i-1/2}\, l_{i+1/2} + HOT \end{aligned} \tag{11.4}$$

whereas Eqs. (11.3a) and (11.3b) provide the one-sided approximation:

$$\begin{aligned} \boldsymbol{\tau}_i &= \frac{l_{i-1/2}+l_{i-3/2}}{l_{i-1/2}\, l_{i-3/2}}\boldsymbol{\Delta r}_{i-\frac{1}{2}} \\ &- \frac{l_{i-1/2}}{l_{i-3/2}\left(l_{i-1/2}+l_{i-3/2}\right)}\left(\boldsymbol{\Delta r}_{i-\frac{1}{2}} + \boldsymbol{\Delta r}_{i-\frac{3}{2}}\right) \\ &+ \frac{1}{3!}\left(x'''_i\mathbf{e}_x + y'''_i\mathbf{e}_y\right)\, l_{i-1/2}\left(l_{i-1/2}+l_{i-3/2}\right) + HOT \end{aligned} \tag{11.5}$$

Equations (11.4) and (11.5) reveal that the truncation error of the two FD approximations is proportional to l^2, l being the curvilinear mesh spacing along the shock front, regardless of whether l is constant or not.

Results

We have performed a grid-convergence study using an analytical curve, the Cissoid of Diocles (see fig. 11.1):

$$\left(x^2+y^2\right)x = 2ay^2 \qquad a = 2 \tag{11.6}$$

The reason for choosing this particular curve is that it resembles a fishtail shock, but also because it features a cusp in the origin, which means that the second and higher derivatives of its parametric representation are un-bounded in the origin.

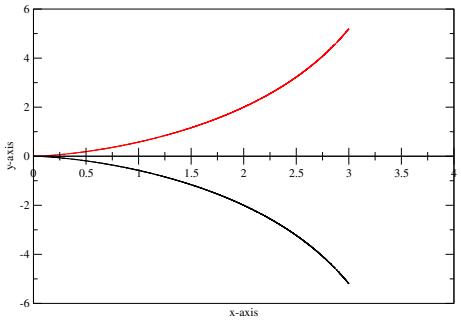


Figure 11.1. The cissoid of Diocles when $a = 2$.

We have performed a mesh-convergence study using a sequence of ten nested grid levels: the coarsest one is made of $N_1 = 30$ shock-edges and the finest grid level has $N_{10} = 15360$ edges. The grid-refinement-ratio, $r = N_{k+1}/N_k$, is constant and equal to $r = 2$.

Grid-points along the curve have been generated by subdividing the $[x_0, x_1]$ range of the x axis into N_k intervals with constant $\Delta x = (x_1 - x_0)/N_k$ spacing; we then compute the location of the shock-points as:

$$\mathbf{r}_i = x_i \, \mathbf{e}_x + y\,(x_i)\ \mathbf{e}_y$$

where:

$$y\,(x) = \pm\sqrt{\frac{x^3}{2a - x}}$$

easily follows from (11.6)[1]. Even if Δx is constant the grid-points along the curve are not equally spaced in the curvilinear abscissa s.

The L_2-norm of the error on grid level k is computed as follows:

$$e_k = \sqrt{\frac{\sum_{i=2}^{i_{\max}} |\mathbf{n}_i - \mathbf{n}_i^k|^2}{N_k + 1}}. \tag{11.7}$$

In Eq. (11.7) $\mathbf{n}_i^k$ is the numerical approximation of the normal vector to the curve in grid-point i of grid level k, computed using either Eq. (11.4) or (11.5) and $\mathbf{n}_i$ is the analytically computed normal unit vector. Depending on whether we use the one-sided or the centered approximation, $i_{\max} = N_k$ or $i_{\max} = N_{k-1}$.

Once the L_2-norm of the error has been computed on all grid levels, the measured order $\tilde{n}$ can be estimated globally, for each pair of grid levels, as follows:

$$\tilde{n} = \frac{\log\,(e_{k-1}/e_k)}{\log r} \tag{11.8}$$

The results are summarized in Tab. 18. When the cusp is not included in the grid convergence study, i.e. we use $x_0 = 0.1$, design order is recovered, see Tab. 18a. When the cusp is included in the grid convergence study, i.e. we use $x_0 = 0$, the discretization error "globally" exhibits only first order convergence, see Tab. 18b. This is because, close to the origin, the higher order derivatives of the parametric description of this particular curve grow unbounded. This pathologic behavior is however localized close to the cusp and shows up in Tab. 18b, because the error norm measured with Eq. (11.7) has a global nature. If a local grid convergence analysis is performed in $x = 0.2$, which is the shock-point closer to the cusp that exists on all grid levels, design order is recovered. Moreover, it is not at all obvious that a cusp-like shape may ever been found in a real shock-wave.

[1] In the grid-convergence study we have only considered the branch of the curve with positive y values.

(a) $x_0 = 0.1; x_1 = 3$

	one-sided			centered		
k	e_k	e_{k-1}/e_k	$\tilde{n}$	e_k	e_{k-1}/e_k	$\tilde{n}$
1	$.17176\,10^{-2}$			$.97105\,10^{-3}$		
2	$.52368\,10^{-3}$	3.280	1.714	$.28366\,10^{-3}$	3.423	1.775
3	$.14830\,10^{-3}$	3.531	1.820	$.77714\,10^{-4}$	3.650	1.868
4	$.39807\,10^{-4}$	3.726	1.897	$.20420\,10^{-4}$	3.806	1.928
5	$.10339\,10^{-4}$	3.850	1.945	$.52390\,10^{-5}$	3.898	1.963
6	$.26363\,10^{-5}$	3.922	1.971	$.13272\,10^{-5}$	3.947	1.981
7	$.66574\,10^{-6}$	3.960	1.985	$.33403\,10^{-6}$	3.973	1.990
8	$.16728\,10^{-6}$	3.980	1.993	$.83788\,10^{-7}$	3.987	1.995
9	$.41928\,10^{-7}$	3.990	1.996	$.20982\,10^{-7}$	3.993	1.998
10	$.10495\,10^{-7}$	3.995	1.998	$.52500\,10^{-8}$	3.997	1.999

(b) $x_0 = 0; x_1 = 3$

	one-sided			centered		
k	e_k	e_{k-1}/e_k	$\tilde{n}$	e_k	e_{k-1}/e_k	$\tilde{n}$
1	$.39480\,10^{-2}$	0.000		$.27329\,10^{-2}$	0.000	
2	$.19297\,10^{-2}$	2.046	1.033	$.13345\,10^{-2}$	2.048	1.034
3	$.95603\,10^{-3}$	2.018	1.013	$.66019\,10^{-3}$	2.021	1.015
4	$.47617\,10^{-3}$	2.008	1.006	$.32846\,10^{-3}$	2.010	1.007
5	$.23767\,10^{-3}$	2.003	1.002	$.16384\,10^{-3}$	2.005	1.003
6	$.11874\,10^{-3}$	2.002	1.001	$.81828\,10^{-4}$	2.002	1.002
7	$.59348\,10^{-4}$	2.001	1.001	$.40891\,10^{-4}$	2.001	1.001
8	$.29668\,10^{-4}$	2.000	1.000	$.20440\,10^{-4}$	2.001	1.000
9	$.14833\,10^{-4}$	2.000	1.000	$.10218\,10^{-4}$	2.000	1.000
10	$.74161\,10^{-5}$	2.000	1.000	$.51088\,10^{-5}$	2.000	1.000

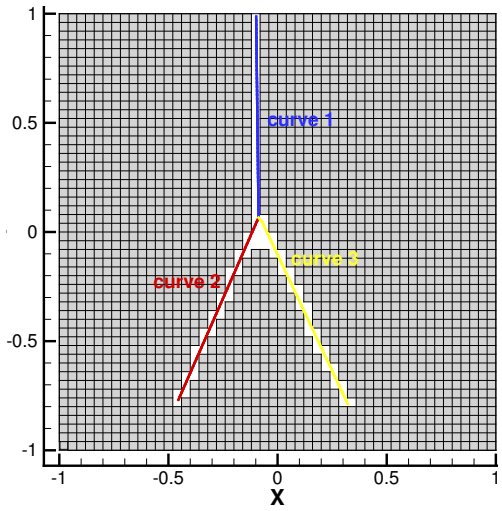
Table 18. Grid convergence study using both the centered and one-sided approximations.

11.2 Appendix B. Order of accuracy study of the *SESF* extrapolation process

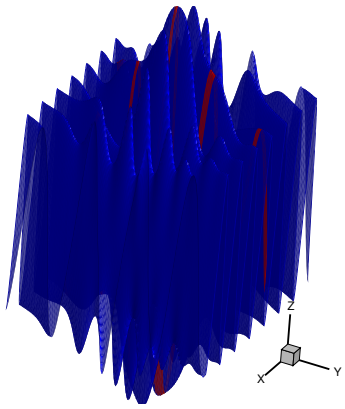
In order to analyze the order-of-accuracy of the truncated Taylor series expansion(6.2) described in step 4 of Sec. 6.1 a simple test-case has been set-up. It consists in extrapolating the following analytical function:

$$u_0(x,y) = -\cos(4\pi(\sqrt{3}x - y))\, e^{0.7e^{(-x^2-4y^2)}} \cos(7\pi xy) \tag{11.9}$$

from the surrogate boundaries to three different curves, which play the role of the shock-fronts in the SESF simulations. In order to perform a grid-convergence analysis, three nested grid levels have been used; the coarsest one and the three curves, marked using solid lines, are shown in Fig. 11.2a, whereas the carpet plot of the analytical function is shown in Fig. 11.2b. We studied the grid-convergence trend of the discretization error, ϵ_h, which is the difference between the extrapolated



(a) Computational domain and curves definition.



(b) Carpet plot of Eq. (11.9) computed on the finest grid: red strips refer to the blanked regions in Fig. 11.2a.

Figure 11.2. Convergence test: computational domain and exact solution.

solution u_h and the exact solution (11.9):

$$\epsilon_h = u_h - u_0. \tag{11.10}$$

The grid-convergence properties of Eq. (7.1) are shown in Tab. 19, which collects the L_1-norm (computed at all points along each curve) of the discretization error on all grid levels and the measured order-of-convergence $\tilde{n}_{i,i+1}$ for each pair of grid levels, computed using Eq. (3.1). All three tests confirm that the discretization error decays at design (second) order, as also clearly highlighted by the plot in Fig. 11.3.

Table 19. Convergence analysis for each curve in Fig. 11.2a

		Curve 1		Curve 2		Curve 3	
Grid level	h	L_1	$\tilde{n}$	L_1	$\tilde{n}$	L_1	$\tilde{n}$
0	0.032	1.54	-	3.56	-	4.92	-
1	0.016	$3.92 \cdot 10^{-1}$	1.98	0.81	2.1	1.18	2.05
2	0.008	$1.08 \cdot 10^{-1}$	1.95	0.21	1.95	0.30	1.94

11.3 Appendix C. Boundaries in the (M_1, σ_{12}) plane.

We list here the equations required to draw those lines in the (M_1, σ_{12}) plane, see Fig., which we referred to throughout the paper. This is only a small subset of the numerous lines that bound regions of the (M_1, σ_{12}) where different shock-interaction patterns are observed. The interested reader is referred to [210, 50, 242] for a more extensive discussion. See also [217] and [229] where these same boundaries are drawn in the (M_1, θ_{12}) and (M_1, ξ_{21}) planes. The labelling of the lines used here follows [50].

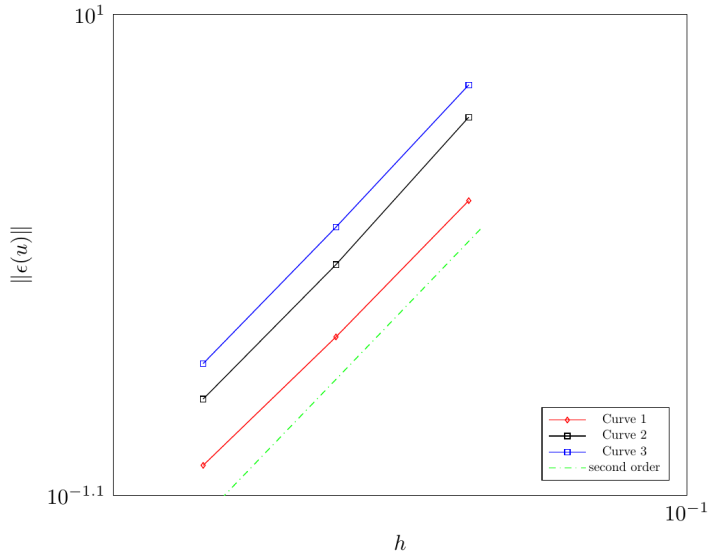


Figure 11.3. Convergence behavior of the L_1-norm for each of the three curves shown in Fig. 11.2a.

Source codes for computing the various boundaries in the 3ST will be made available on a public repository at the time of publication.

- Line 1: the incident shock (IS) is a Mach wave

  On line 1 the I-shock has zero strength. i.e. it is a Mach wave; therefore, the I-shock angle is the Mach angle:

$$\sigma_{12} = \sin^{-1}\left(\frac{1}{M_1}\right) \tag{11.11}$$

- Line 2: Sonic flow behind I-shock

  Shock-shock interaction is possible only as long as the flow behind the I-shock is supersonic, i.e. $M_2 > 1$. The I-shock angle corresponding to the limit condition of sonic flow behind the I-shock can be computed from [206, Eq.(167)] using the given value of M_1.

- Line 7a (the first Henderson line): sonic flow behind the R-shock in the 3ST.

  The flow behind the R-shock in the 3ST is sonic along two distinct lines (the first and second Henderson line) of the (M_1, σ_{12}) plane, but only the first Henderson line is of interest when the I-shock is weak. Points on line 7a are

solutions to the following 4x4 non-linear system of algebraic equations:

$$\sin^2\sigma_{14} = \frac{(\gamma-1)(\gamma+1)+4\,[\gamma M_1^2\sin^2\sigma_{12}-\delta]\,[\gamma M_2^2\sin^2\sigma_{23}-\delta]}{2\gamma\,(\gamma+1)M_1^2} \tag{11.12a}$$

$$\theta(M_1,\sigma_{14}) = \theta(M_1,\sigma_{12})+\theta(M_2,\sigma_{23}) \tag{11.12b}$$

$$M_2^2 = \frac{\left(1+\delta M_1^2\sin^2\sigma_{12}\right)^2+\left(\frac{1}{2}(\gamma+1)\,M_1^2\sin\sigma_{12}\cos\sigma_{12}\right)^2}{\left(1+\delta M_1^2\sin^2\sigma_{12}\right)\left(\gamma M_1^2\sin^2\sigma_{12}-\delta\right)} \tag{11.12c}$$

$$\sin^2\sigma_{23} = \frac{(\gamma-3)+(\gamma+1)M_2^2+\sqrt{(\gamma+1)\left[(\gamma+9)+2(\gamma-3)\,M_2^2+(\gamma+1)M_2^4\right]}}{4\gamma M_2^2} \tag{11.12d}$$

Equations (11.12a) and (11.12b) respectively translate the conditions of equal pressure and flow direction across the SS. Equation (11.12c) is [206, Eq. (132)] and gives the Mach number behind the I-shock as a function of M_1 and the I-shock angle. Equation (11.12d) is [206, Eq. (167)] and gives the R-shock angle corresponding to sonic flow behind the R-shock, i.e. $M_3 = 1$. The flow deflection θ in Eqs. (11.12b) is computed from [206, Eq. (138)]. The four unknowns in the system (11.12) are: $M_2, \sigma_{12}, \sigma_{23}, \sigma_{14}$, whereas M_1 is given. The (M_1, σ_{12}) coordinates of the point where the first Henderson line joins line 1 can be computed as described in [242].

- Line 8a: branch I of the backward limit in the 3ST.

  Physically, the backward limit corresponds to a three-shocks-system in which the RS is a Mach wave, which implies that the IS and MS form a unique shock. A simple closed-form expression that allows to draw the backward limit is attributed by [217, Eqs. (15-16)] to [243] and it is repeated here:

$$M_1^2 = \frac{1+\xi_i/c}{\xi_i\,U_\pm}-\frac{2}{\gamma-1} \qquad c=\frac{\gamma-1}{\gamma+1} \tag{11.13a}$$

  where:

$$U_\pm = 1+\frac{2\,c\,\xi_i}{1+\xi_i\pm\sqrt{(1+c\,\xi_i)(1+\xi_i/c)}} \tag{11.13b}$$

  In Eqs. (11.13) $\xi_i = \xi_{12}^{-1}$ i.e. the inverse pressure ratio across the I-shock. There are two branches of the backward limit, but only branch I, which corresponds to the plus sign in Eq. (11.13b), is of interest in the range of Mach number considered here. Line 8a is drawn by taking values of ξ_i ranging between 1 (which corresponds to the point where line 8a joins line 1) and a pre-set value smaller than 1. M_1 is then computed from Eq. (11.13a) and the I-shock angle, σ_{12}, follows from [206, Eq. (128)].

Four waves theory (4WT).

The set of non-linear algebraic equations governing the 4WT can be found in [230]. It is however worth recalling it here. Given the free-stream Mach number, M_1 and I-shock angle, σ_{12}, which are the same two input parameters also used in the 3ST, the

following quantities behind the I-shock can be easily computed: the flow deflection, θ_{12}, and pressure ratio, ξ_{21}, across the I-shock and the Mach number, M_2, in the region bounded by the I- and R-shocks. Since the flow behind the R-shock is sonic in the 4WT, i.e. $M_3 = 1$, see Figs. 9.8b and 9.8d, the R-shock angle, σ_{23}, can be computed from [206, Eq. (167)] using the known value of M_2. The pressure ratio across the R-shock, ξ_{32}, follows from [206, Eq. (128)] using the known values of M_2 and σ_{23}. The flow across the EF is isentropic, therefore:

$$\frac{p_5}{p_3} = \left[\frac{1+\delta\, M_5^2}{(\gamma+1)/2}\right]^{-\frac{\gamma}{\gamma-1}} \qquad \delta = \frac{\gamma-1}{2}$$

The two conditions that hold across the SS, i.e. equal flow direction and equal pressure, can be translated into the following two by two non-linear system of algebraic equations:

$$\theta_{14}\left(M_1, \sigma_{14}\right) - \nu\left(M_5\right) = \theta_{12} + \theta_{23} \tag{11.14a}$$

$$\xi_{41}\left(M_1, \sigma_{14}\right)\left[\frac{1+\delta\, M_5^2}{(\gamma+1)/2}\right]^{\frac{\gamma}{\gamma-1}} = \xi_{32}\,\xi_{21} \tag{11.14b}$$

where terms on the r.h.s. are known and the only two unknowns appear on the l.h.s. These are: the (supersonic) Mach number, M_5, in the region bounded by the tail of the EF and the SS and the M-shock angle, σ_{14}. In Eqs. (11.14) θ and ξ are the flow deflection and pressure ratio through an oblique shock, which can be computed using [206, Eq. (138)] and [206, Eq. (128)], respectively; $\nu\left(M\right)$ is the Prandtl-Meyer function [206, Eq. (171c)].

Two sample applications of the 4WT are listed in Tab. 20. The left column of Table 20, which refers to Fig. 9.8a, shows both the 3ST and 4WT solutions for point *b* in Fig. 9.4. According to [216], it is the 4WT solution which should be chosen inside the first Henderson region. The right column of Tab. 20, which refers to Fig. 9.8c, lists the 4WT solution at point 151 of Fig. 9.9a, which is the point along the fitted I-shock that is closest to the triple-point region of the fishtail in the AGARD03 test-case.

The boundary between the VR and GR in the 4WT

The boundary between the Vasilev ($M_4 < 1$) and Guderley ($M_4 > 1$) reflection corresponds to the condition of sonic flow behind the M-shock in the 4WT. i.e. $M_4 = 1$. Using M_1 as the free parameter, the shock-angle, σ_{14}, and flow deflection across the M-shock, θ_{14}, must be those corresponding to sonic flow behind an oblique shock and can be readily computed from [206, Eq. (167)] and [206, Eq. (138)].

Table 20. Features of the VR and GR shown in Fig. 9.8; input data are M_1 and σ_{12}.

Vasilev reflection of Fig. 9.8a		
	3ST	4WT
ξ_{53}	-	0.99532
M_5	-	1.00401
$\nu(M_5)$	-	$1.14187\ 10^{-2}$
ξ_{41}	1.59762	1.81053
σ_{14}	57.335	63.02324
M_4	1.1052	0.99229
θ_{14}	9.10869	10.73184
ξ_{21}	1.5	1.5
σ_{12}	54.9069	54.90695
M_2	1.1581	1.15810
θ_{12}	8.03923	8.03923
ξ_{32}	1.06508	1.21269
σ_{23}	62.5286	69.86787
M_3	1.10812	1.00000
θ_{23}	1.06946	0.04680
M_1	1.46077	1.46077

Guderley reflection of Fig. 9.8c	
	4WT
ξ_{53}	0.99006
M_5	1.00854
$\nu(M_5)$	.03531
ξ_{41}	1.34599
σ_{14}	66.04472
M_4	1.00727
θ_3	4.80818
ξ_{21}	1.24948
σ_{12}	62.15677
M_2	1.07076
θ_1	3.91812
ξ_{32}	1.08805
σ_{23}	75.58431
M_3	1.00000
θ_{12}	0.01492
M_1	1.24600

The system on non-linear algebraic equations to be solved is the following:

$$\theta_{14} = \theta_{12}(M_1, \sigma_{12}) + \theta_{23}(M_2, \sigma_{23}) + \nu(M_5) \tag{11.15a}$$

$$\xi_{41} = \xi_{21}(M_1, \sigma_{12})\,\xi_{32}(M_2, \sigma_{23}) \left[\frac{1+\delta M_5^2}{(\gamma+1)/2}\right]^{-\frac{\gamma}{\gamma-1}} \tag{11.15b}$$

$$\sin^2\sigma_{23} = \frac{(\gamma-3) + (\gamma+1)M_2^2 + \sqrt{(\gamma+1)\left[(\gamma+9) + 2(\gamma-3)\,M_2^2 + (\gamma+1)M_2^4\right]}}{4\gamma M_2^2} \tag{11.15c}$$

$$M_2^2 = \frac{\left(1+\delta M_1^2\sin^2\sigma_{12}\right)^2 + \left(\frac{1}{2}(\gamma+1)\,M_1^2\sin\sigma_{12}\cos\sigma_{12}\right)^2}{\left(1+\delta M_1^2\sin^2\sigma_{12}\right)\left(\gamma M_1^2\sin^2\sigma_{12} - \delta\right)} \tag{11.15d}$$

Equations (11.15a) and (11.15b) translate the conditions of parallel streams and equal pressure across the SS; the various functions involved have already been defined by reference to Eq. (11.14). Equations (11.15c) and (11.15d) have also been already introduced by reference to Eq. (11.12).

The l.h.s. of Eqs.(11.15a) and(11.15b) is known for the reasons stated above, so that the four unknowns in(11.15) are: $M_2, M_5, \sigma_{12}, \sigma_{23}$.

Bibliography

[1] D. Baker, North American X-15 Owner's Workshop Manual: 1954-1968 X15, X-15b & Delta Wing Models: 1954-1968 (X-15A, X-15B & Delta Wing models), Haynes Pubns, 2016.

[2] M. D. Salas, A shock-fitting primer, CRC Press, 2009.

[3] J. J. Quirk, A contribution to the great riemann solver debate, in: Upwind and High-Resolution Schemes, Springer, 1997, pp. 550–569. doi:10.1007/978-3-642-60543-7_22.

[4] M. H. Carpenter, J. H. Casper, Accuracy of shock capturing in two spatial dimensions, AIAA journal 37 (9) (1999) 1072–1079.

[5] D. Zaide, P. Roe, Shock capturing anomalies and the jump conditions in one dimension, in: Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2011, pp. –. doi:10.2514/6.2011-3686.

[6] M. Pandolfi, D. D'Ambrosio, Numerical instabilities in upwind methods: analysis and cures for the "carbuncle" phenomenon, Journal of Computational Physics 166 (2) (2001) 271–301. doi:10.1006/jcph.2000.6652.

[7] P. K. Sweby, High resolution tvd schemes using flux limiters, in: Large-scale computations in fluid mechanics, 1985.

[8] T. K. Lee, X. Zhong, Spurious numerical oscillations in simulation of supersonic flows using shock-capturing schemes, AIAA journal 37 (3) (1999) 313–319.

[9] A. Dadone, B. Grossman, A multi-dimensional upwind scheme for the euler equations, in: Thirteenth international conference on numerical methods in fluid dynamics, Springer, 1993, pp. 95–99.

[10] P. Gnoffo, Multi-dimensional inviscid flux reconstruction for simulation of hypersonic heating on tetrahedral grids, in: A. I. of Aeronautics, R. Astronautics (Eds.), Aerospace Sciences Meetings, 2009.
URL https://doi.org/10.2514/6.2009-599

[11] N. K. Yamaleev, M. H. Carpenter, On accuracy of adaptive grid methods for captured shocks, Journal of Computational Physics 181 (1) (2002) 280–316.

[12] H. W. Emmons, The numerical solution of compressible fluid flow problems, NACA-TN 932, NASA (1944).
URL https://apps.dtic.mil/dtic/tr/fulltext/u2/b814416.pdf

[13] M. Abbett, G. Moretti, A time-dependent computational method for blunt body flows., AIAA Journal 4 (12) (1966) 2136–2141. doi:10.2514/3.3867.

[14] Y. Yamamoto, K. Karashima, Floating shock fitting for three-dimensional inviscid supersonic flows, AIAA Journal 20 (1) (1982) 9–17. doi:10.2514/3.7897.

[15] M. Valorani, M. Onofri, B. Favini, F. Sabetta, Nonequilibrium hypersonic inviscid steady flows, AIAA journal 30 (1) (1992) 86–93.

[16] M. Valorani, M. Di Giacinto, Fast moving sub-subsonic shocks in closed-end tubes, Journal of Computational Physics 88 (2) (1990) 409–432.

[17] K. J. Weilmuenster, A hybridized method for computing high-Reynolds-number hypersonic flow about blunt bodies, National Aeronautics and Space Administration, 1979.

[18] X. Zhong, High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition, Journal of Computational Physics 144 (2) (1998) 662–709.

[19] R. Marsilio, G. Moretti, Shock-fitting method for two-dimensional inviscid, steady supersonic flows in ducts, Meccanica 24 (4) (1989) 216–222. doi:10.1007/BF01556453.

[20] G. Moretti, F. Marconi, M. Onofri, Shock-boundary layer interaction by shock fitting, in: Thirteenth International Conference on Numerical Methods in Fluid Dynamics, Springer, 1993, pp. 345–349.

[21] A. Bonfiglioli, R. Paciorri, Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids, AIAA J. 52 (7) (2014) 1404–1416. doi:10.2514/1.J052567.

[22] R. Paciorri, A. Bonfiglioli, A shock-fitting technique for 2d unstructured grids, Computers & Fluids 38 (3) (2009) 715 –726. doi:10.1016/j.compfluid.2008.07.007.

[23] R. Paciorri, A. Bonfiglioli, Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique, Journal of Computational Physics 230 (8) (2011) 3155–3177. doi:https://doi.org/10.1016/j.jcp.2011.01.018.

[24] L. Campoli, P. Quemar, A. Bonfiglioli, M. Ricchiuto, Shock-fitting and predictor-corrector explicit ale residual distribution, in: M. Onofri, R. Paciorri (Eds.), Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, Springer International Publishing, Cham, 2017, pp. 113–129. doi:10.1007/978-3-319-68427-7_5.

[25] A. Bonfiglioli, M. Grottadaurea, R. Paciorri, F. Sabetta, An unstructured, three-dimensional, shock-fitting solver for hypersonic flows, Computers & Fluids 73 (0) (2013) 162–174. doi:10.1016/j.compfluid.2012.12.022.

[26] A. Lani, T. Quintino, D. Kimpe, H. Deconinck, S. Vandewalle, S. Poedts, The coolfluid framework: design solutions for high performance object oriented scientific computing software, in: International Conference on Computational Science, Springer, 2005, pp. 279–286.

[27] A. Lani, V. De Amicis, Sf: An open source object-oriented platform for unstructured shock-fitting methods, in: M. Onofri, R. Paciorri (Eds.), Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, Springer International Publishing, Cham, 2017, pp. 85–112. doi:10.1007/978-3-319-68427-7_4.

[28] M. Ricchiuto, R. Abgrall, Explicit Runge-Kutta residual distribution schemes for time dependent problems: Second order case, Journal of Computational Physics 229 (16) (2010) 5653 – 5691. doi:10.1016/j.jcp.2010.04.002.

[29] M. Ricchiuto, An explicit residual based approach for shallow water flows, Journal of Computational Physics 280 (2015) 306–344. doi:https://doi.org/10.1016/j.jcp.2014.09.027.
URL https://www.sciencedirect.com/science/article/pii/S0021999114006639

[30] A. Bonfiglioli, R. Paciorri, L. Campoli, Unsteady shock-fitting for unstructured grids, International Journal for Numerical Methods in Fluids 81 (4) (2016) 245–261. doi:10.1002/fld.4183.

[31] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, Journal of Computational Physics 369 (2018) 45–79. doi:10.1016/j.jcp.2018.04.052.

[32] M. Ciallella, M. Ricchiuto, R. Paciorri, A. Bonfiglioli, Extrapolated shock tracking: Bridging shock-fitting and embedded boundary methods, Journal of Computational Physics 412 (2020) 109440. doi:https://doi.org/10.1016/j.jcp.2020.109440.

[33] M. Ciallella, M. Ricchiuto, R. Paciorri, A. Bonfiglioli, Extrapolated DIscontinuity Tracking for complex 2D shock interactions, Computer Methods in Applied Mechanics and Engineering 391 (2022) 114543. doi:10.1016/j.cma.2021.114543.

[34] J. Liu, D. Zou, A Shock-Fitting Technique for ALE Finite Volume Methods on Unstructured Dynamic Meshes, Springer International Publishing, Cham, 2017, pp. 131–149. doi:10.1007/978-3-319-68427-7_6.

[35] D. Zou, C. Xu, H. Dong, J. Liu, A shock-fitting technique for cell-centered finite volume methods on unstructured dynamic meshes, Journal of Computational Physics 345 (2017) 866 – 882. doi:https://doi.org/10.1016/j.jcp.2017.05.047.

[36] S. Chang, X. Bai, D. Zou, Z. Chen, J. Liu, An adaptive discontinuity fitting technique on unstructured dynamic grids, Shock Waves 29 (8) (2019) 1103–1115. doi:10.1007/s00193-019-00913-3.

[37] D. Zou, A. Bonfiglioli, R. Paciorri, J. Liu, An embedded shock-fitting technique on unstructured dynamic grids, Computers & Fluids 218 (2021) 104847. doi:10.1016/j.compfluid.2021.104847.

[38] A novel stabilization method for high-order shock fitting with finite element methods, Journal of Computational Physics (2021) 110096doi:10.1016/j.jcp.2020.110096.

[39] M. Zahr, P.-O. Persson, An optimization-based approach for high-order accurate discretization of conservation laws with discontinuous solutions, Journal of Computational Physics 365 (2018) 105 – 134. doi:10.1016/j.jcp.2018.03.029.

[40] M. J. Zahr, A. Shi, P.-O. Persson, Implicit shock tracking using an optimization-based high-order discontinuous galerkin method, Journal of Computational Physics (2020) 109385doi:10.1016/j.jcp.2020.109385.

[41] M. J. Zahr, J. M. Powers, High-order resolution of multidimensional compressible reactive flow using implicit shock tracking, AIAA Journal 59 (1) (2021) 150–164. doi:https://doi.org/10.2514/1.J059655.

[42] T. Huang, M. J. Zahr, A robust, high-order implicit shock tracking method for simulation of complex, high-speed flows, Journal of Computational Physics (2022) 110981doi:https://doi.org/10.1016/j.jcp.2022.110981.

[43] A. T. Corrigan, A. D. Kercher, D. A. Kessler, D. Wood-Thomas, Application of the moving Discontinuous Galerkin method with interface condition enforcement to shocked compressible flows, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-4272.

[44] A. T. Corrigan, A. Kercher, D. A. Kessler, The moving Discontinuous Galerkin method with interface condition enforcement for unsteady three-dimensional flows, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-0642.

[45] A. Corrigan, A. D. Kercher, D. A. Kessler, A moving discontinuous Galerkin finite element method for flows with interfaces, International Journal for Numerical Methods in Fluids 89 (9) (2019) 362–406. doi:10.1002/fld.4697.

[46] I.-L. Chern, J. Glimm, O. Mcbryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics, Journal of Computational Physics 62 (1) (1986) 83 – 110. doi:10.1016/0021-9991(86)90101-4.

[47] D. W. Zaide, C. F. Ollivier-Gooch, Inserting a surface into an existing unstructured mesh, Int. J. Numer. Methods Eng 106 (6) (2016) 484–500. doi:10.1002/nme.5132.

[48] G. Birkhoff, Hydrodynamics: A study in logic, fact, and similitude, Bull. Amer. Math. Soc 57 (1951) 97–499.

[49] K. G. Guderley, Considerations of the structure of mixed subsonic-supersonic flow patterns, Air material command tech. report, F-TR-2168-ND, ATI No. 22780, U.S. Wright-Patterson Air Force Base, GS-AAF-Wright Field 39, Dayton, Ohio. (Oct. 1947).

[50] C. J. Chapman, High speed flow, Cambridge University Press, 2000.

[51] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th aerospace sciences meeting and exhibit, 1992, p. 439.

[52] J. Anderson, Fundamentals of Aerodynamics, McGraw-Hill Education, 2010. URL `https://books.google.it/books?id=xwY8PgAACAAJ`

[53] H. N. (as K. Masatsuka), I Do Like CFD, 2013, iSBN 9781304827937, `www.cfdbooks.com/`.

[54] M. Onofri, R. Paciorri, Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, Springer, 2017.

[55] G. Moretti, M. D. Salas, Numerical analysis of viscous one-dimensional flows, Journal of Computational Physics 5 (3) (1970) 487–506.

[56] S. K. Lele, Shock-jump relations in a turbulent flow, Physics of Fluids A: Fluid Dynamics 4 (12) (1992) 2900–2905. `doi:doi:10.1063/1.858343`.

[57] S. R. Allmaras, F. T. Johnson, Modifications and clarifications for the implementation of the spalart-allmaras turbulence model, in: Seventh international conference on computational fluid dynamics (ICCFD7), Vol. 1902, ICCFD7-1902 Big Island, Hawaii, 2012.

[58] M. D. Salas, Shock fitting method for complicated two-dimensional supersonic flows, Aiaa Journal 14 (5) (1976) 583–588.

[59] J. VonNeumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of applied physics 21 (3) (1950) 232–237.

[60] P. D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Communications on pure and applied mathematics 7 (1) (1954) 159–193.

[61] S. Godunov, I. Bohachevsky, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Matematičeskij sbornik 47 (3) (1959) 271–306.

[62] S. K. Godunov, A. Zabrodin, M. I. Ivanov, A. Kraiko, G. Prokopov, Numerical solution of multidimensional problems of gas dynamics, Moscow Izdatel Nauka (1976).

[63] B. Van Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method, Journal of computational Physics 32 (1) (1979) 101–136.

[64] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, iii, in: Upwind and high-resolution schemes, Springer, 1987, pp. 218–290.

[65] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1) (1994) 200–212.

[66] J. Gottlieb, C. P. Groth, Assessment of riemann solvers for unsteady one-dimensional inviscid flows of perfect gases, Journal of Computational Physics 78 (2) (1988) 437–458.

[67] P. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (2) (1981) 357–372. doi:10.1016/0021-9991(81)90128-5.

[68] P. Colella, P. R. Woodward, The piecewise parabolic method (ppm) for gas-dynamical simulations, Journal of computational physics 54 (1) (1984) 174–201.

[69] M. Geron, R. Paciorri, F. Nasuti, F. Sabetta, Flowfield analysis of a linear clustered plug nozzle with round-to-square modules, Aerospace Science and Technology 11 (2-3) (2007) 110–118. doi:https://doi.org/10.1016/j.ast.2006.08.004.

[70] R. Paciorri, F. Sabetta, A. Bonfiglioli, Turbulence modeling of base drag on launcher in subsonic flight, Journal of Spacecraft and Rockets 51 (5) (2014) 1673–1680. doi:https://doi.org/10.2514/1.A32757.

[71] S. Pirozzoli, Numerical methods for high-speed flows, Annual Review of Fluid Mechanics 43 (1) (2011) 163–194. doi:10.1146/annurev-fluid-122109-160718.

[72] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, Journal of computational physics 126 (1) (1996) 202–228.

[73] Y.-T. Zhang, C.-W. Shu, Eno and weno schemes, in: Handbook of Numerical Analysis, Vol. 17, Elsevier, 2016, pp. 103–122.

[74] L. Gosse, A two-dimensional version of the godunov scheme for scalar balance laws, SIAM Journal on Numerical Analysis 52 (2) (2014) 626–652.

[75] B. Wendroff, A two-dimensional hlle riemann solver and associated godunov-type difference scheme for gas dynamics, Computers & Mathematics with Applications 38 (11) (1999) 175–185. doi:https://doi.org/10.1016/S0898-1221(99)00296-5.
URL https://www.sciencedirect.com/science/article/pii/S0898122199002965

[76] R. Abgrall, M. Ricchiuto, High-Order Methods for CFD, John Wiley & Sons, Ltd, 2017, pp. 1–54. doi:10.1002/9781119176817.ecm2112.

[77] H. Deconinck, M. Ricchiuto, Residual Distribution Schemes: Foundations and Analysis, John Wiley & Sons, Ltd, 2017, pp. 1–53. doi:10.1002/0470091355.ecm054.

[78] R. Abgrall, K. Mer, B. Nkonga, A Lax–Wendroff type theorem for residual schemes, in: M. Haffez, J. Chattot (Eds.), Innovative methods for numerical solutions of partial differential equations, World Scientific, 2002, pp. 243–266. doi:10.1142/9789812810816.

[79] R. Abgrall, P. L. Roe, High order fluctuation schemes on triangular meshes, Journal of Scientific Computing 19 (1) (2003) 3–36. doi:10.1023/A:1025335421202.

[80] H. Deconinck, P. Roe, R. Struijs, A multidimensional generalization of Roe's flux difference splitter for the euler equations, Computers & Fluids 22 (2-3) (1993) 215–222. doi:10.1016/0045-7930(93)90053-c.

[81] A. Bonfiglioli, Fluctuation splitting schemes for the compressible and incompressible euler and navier-stokes equations, International Journal of Computational Fluid Dynamics 14 (1) (2000) 21–39. doi:10.1080/10618560008940713.

[82] A. Csík, M. Ricchiuto, H. Deconinck, A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws, Journal of Computational Physics 179 (1) (2002) 286–312. doi:https://doi.org/10.1006/jcph.2002.7057.

[83] M. Ricchiuto, A. Csík, H. Deconinck, Residual distribution for general time-dependent conservation laws, Journal of Computational Physics 209 (1) (2005) 249–289. doi:10.1016/j.jcp.2005.03.003.

[84] M. Ricchiuto, R. Abgrall, Explicit runge–kutta residual distribution schemes for time dependent problems: Second order case, Journal of Computational Physics 229 (16) (2010) 5653–5691. doi:https://doi.org/10.1016/j.jcp.2010.04.002.
URL https://www.sciencedirect.com/science/article/pii/S0021999110001786

[85] L. Arpaia, M. Ricchiuto, R. Abgrall, An ale formulation for explicit runge–kutta residual distribution, Journal of Scientific Computing 63 (2) (2015) 502–547.

[86] R. Abgrall, Toward the ultimate conservative scheme : Following the quest, Journal of Computational Physics 167 (2) (2001) 277–315. doi:10.1006/jcph.2000.6672.

[87] E. van der Weide, H. Deconinck, Positive matrix distribution schemes for hyperbolic systems, in: Computational Fluid Dynamics, Wiley, New York, 1996, pp. 747–753.

[88] R. Abgrall, M. Mezine, Construction of second-order accurate monotone and stable residual distribution schemes for steady problems, Journal of Computational Physics 195 (2) (2004) 474 – 507. doi:https://doi.org/10.1016/j.jcp.2003.09.022.

[89] P. L. Roe, Linear advection schemes on triangular meshes, Tech. Rep. CoA 8720, Cranfield Institute of Technology (1987).

[90] P. L. Roe, "optimum" upwind advection on a triangular mesh, Tech. Rep. ICASE 90-75, ICASE (1990).

[91] P. Roe, D. Sidilkover, Optimum positive linear schemes for advection in two and three dimensions., SIAM J. Numer. Anal. 29 (6) (1992) 1542–1568. doi:10.1137/0729089.

[92] H. Paillere, H. Deconinck, Multidimensional upwind residual distribution schemes for the 2d euler equations, in: H. Deconinck, B. Koren (Eds.), Notes on Numerical Fluid Mechanics, Vieweg-Verlag, Braunschweig, Germany, 1997, pp. 51–112.

[93] H. Deconinck, H. Paillère, R. Struijs, P. L. Roe, Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws, Comput. Mech. 11 (1993) 323–340.
URL https://doi.org/10.1007/BF00350091

[94] M. Najafi, K. Hejranfar, V. Esfahanian, Application of a shock-fitted spectral collocation method for computing transient high-speed inviscid flows over a blunt nose, Journal of Computational Physics 257 (2014) 954–980.

[95] X. Zhong, M. Tatineni, High-order non-uniform grid schemes for numerical simulation of hypersonic boundary-layer stability and transition, Journal of Computational Physics 190 (2) (2003) 419–458.

[96] P. Rawat, X. Zhong, Numerical simulation of shock-turbulence interactions using high-order shock-fitting algorithms, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 114.

[97] F. Nasuti, M. Onofri, Analysis of unsteady supersonic viscous flows by a shock-fitting technique, AIAA J. 34 (1996) 1428–1434. doi:10.2514/3.13249.

[98] G. Moretti, M. Valorani, Detection and fitting of two-dimensional shocks, Notes Numer. Fluid Mech. 20 (1988) 239–246.

[99] P. S. Rawat, X. Zhong, On high-order shock-fitting and front-tracking schemes for numerical simulation of shock-disturbance interactions, Journal of Computational Physics 229 (19) (2010) 6744 – 6780. doi:10.1016/j.jcp.2010.05.021.

[100] J. Glimm, J. W. Grove, X. L. Li, K.-m. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM Journal on Scientific Computing 19 (3) (1998) 703–727. doi:10.1137/S1064827595293600.

[101] J. Glimm, E. Isaacson, D. Marchesin, O. McBryan, Front tracking for hyperbolic systems, Advances in Applied Mathematics 2 (1) (1981) 91–119. doi:https://doi.org/10.1016/0196-8858(81)90040-3.

[102] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, L. Wu, A simple package for front tracking, Journal of Computational Physics 213 (2) (2006) 613–628. doi:10.1016/j.jcp.2005.08.034.

[103] C. Klingenberg, B. Plohr, An introduction to front tracking, in: Multidimensional Hyperbolic Problems and Computations, Springer, 1991, pp. 203–216. doi:10.1137/S106482750343028X.

[104] C. Helzel, M. J. Berger, R. J. LeVeque, A high-resolution rotated grid method for conservation laws with embedded geometries, SIAM Journal on Scientific Computing 26 (3) (2005) 785–809. doi:10.1137/S106482750343028X.

[105] I.-L. Chern, P. Colella, A conservative front tracking method for hyperbolic conservation laws, LLNL Rep. No. UCRL-97200, Lawrence Livermore National Laboratory 2 (1987) 83–110.

[106] A. Harten, J. M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws, Journal of computational physics 50 (2) (1983) 235–269. doi:10.1016/0021-9991(83)90066-9.

[107] D.-k. Mao, A treatment of discontinuities for finite difference methods in the two-dimensional case, Journal of Computational Physics 104 (2) (1993) 377–397. doi:10.1006/jcph.1993.1038.

[108] A. Assonitis, R. Paciorri, A. Bonfiglioli, Numerical simulation of shock/boundary-layer interaction using an unstructured shock-fitting technique, Computers & Fluids 228 (2021) 105058. doi:10.1016/j.compfluid.2021.105058.

[109] L. Campoli, A. Assonitis, M. Ciallella, R. Paciorri, A. Bonfiglioli, M. Ricchiuto, Undifi-2d: an unstructured discontinuity fitting code for 2d grids, Computer Physics Communications 271 (2022) 108202. doi:10.1016/j.cpc.2021.108202.

[110] W. H. Reed, T. R. Hill, Triangular mesh methods for the neutron transport equation, Tech. rep., Los Alamos Scientific Lab., N. Mex.(USA) (1973).

[111] J.-F. Remacle, J. E. Flaherty, M. S. Shephard, An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems, SIAM review 45 (1) (2003) 53–72.

[112] I. S. Raju, The discontinuous galerkin finite element method, Tech. rep. (2022).

[113] A. Assonitis, R. Paciorri, M. Ciallella, M. Ricchiuto, A. Bonfiglioli, Extrapolated shock fitting for two-dimensional flows on structured grids, AIAA Journal 60 (11) (2022) 6301–6312.

[114] A. Assonitis, M. Ciallella, R. Paciorri, M. Ricchiuto, A. Bonfiglioli, A shock-fitting technique for 2d/3d flows with interactions using structured grids, in: AIAA Aviation Forum 2022, 2022, pp. AIAA 2022–4123. doi:10.2514/6.2022-4123.

[115] T. J. Baker, Mesh generation: Art or science?, Progress in Aerospace Sciences 41 (1) (2005) 29–63.

[116] T. I.-P. Shih, R. Bailey, H. Nguyen, R. Roelke, Algebraic grid generation for complex geometries, International journal for numerical methods in fluids 13 (1) (1991) 1–31.

[117] B. V. Saunders, et al., The application of numerical grid generation to problems in computational fluid dynamics, CONTEMPORARY MATHEMATICS 275 (2001) 95–106.

[118] W. M. Chan, Hyperbolic methods for surface and field grid generation, 1996.

[119] D. Mavriplis, Unstructured grid techniques, Annual Review of Fluid Mechanics 29 (1) (1997) 473–514.

[120] A. Katz, V. Sankaran, Mesh quality effects on the accuracy of cfd solutions on unstructured meshes, Journal of Computational Physics 230 (20) (2011) 7670–7686.

[121] T. Shih, Y. Qin, A posteriori method for estimating and correcting grid-induced errors in cfd solutions part 1: Theory and method, in: 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007, p. 100.

[122] J. Tu, G.-H. Yeoh, C. Liu, Chapter 4 - cfd mesh generation: A practical guideline, in: J. Tu, G.-H. Yeoh, C. Liu (Eds.), Computational Fluid Dynamics (Third Edition), third edition Edition, Butterworth-Heinemann, 2018, pp. 125–154. doi:https://doi.org/10.1016/B978-0-08-101127-0.00004-0. URL https://www.sciencedirect.com/science/article/pii/B9780081011270000040

[123] M. F. Siladic, Numerical grid generation and potential airfoil analysis and design, Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH (1988).

[124] P. R. Eiseman, R. E. Smith, Mesh generation using algebraic techniques, Numerical Grid Generation Tech. (1980).

[125] J. F. Thompson, F. C. Thames, C. W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, Journal of computational physics 15 (3) (1974) 299–319.

[126] J. THOMPSON, Z. Warsi, Three-dimensional grid generation from elliptic systems, in: 6th Computational Fluid Dynamics Conference Danvers, 1983, p. 1905.

[127] J. L. Steger, R. L. Sorenson, Use of hyperbolic partial differential equations to generate body fitted coordinates, NASA. Langley Research Center Numerical Grid Generation Tech. (1980).

[128] S. Nakamura, Marching grid generation using parabolic partial differential equations, in: Numerical grid generation; Symposium on Numerical Generation of Curvilinear Coordinate Systems and Their Use in the Numerical Solution of Partial Differential Equations, 1982.

[129] S. Takahashi, I. Monjugawa, K. Nakahashi, Unsteady flow computations around moving airfoils by overset unstructured grid method, Transactions of the Japan Society for Aeronautical and Space Sciences 51 (172) (2008) 78–85.

[130] F. Cougherty, J. A. Benek, J. L. Steger, On applications of chimera grid schemes to store separation, Tech. rep. (1985).

[131] F. DOUGHERTY, J.-H. KUAN, Transonic store separation using a three-dimensional chimera grid scheme, in: 27th Aerospace Sciences Meeting, 1989, p. 637.

[132] S. Zhang, X. Zhao, Computational studies of stage separation with an unstructured chimera grid method, in: 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2007, p. 5409.

[133] S. Lo, A new mesh generation scheme for arbitrary planar domains, International journal for numerical methods in engineering 21 (8) (1985) 1403–1426.

[134] E. Seveno, et al., Towards an adaptive advancing front method, in: 6th International Meshing Roundtable, Citeseer, 1997, pp. 349–362.

[135] L. P. Chew, Constrained Delaunay triangulations, Algorithmica 4 (1-4) (1989) 97–108.

[136] D. Cohen-Steiner, E. C. De Verdiere, M. Yvinec, Conforming delaunay triangulations in 3d, in: Proceedings of the eighteenth annual symposium on Computational geometry, 2002, pp. 199–208.

[137] D. Rufat, https://rufat.be/triangle/ (2017).

[138] P. R. Eiseman, Adaptive grid generation, Computer Methods in Applied Mechanics and Engineering 64 (1) (1987) 321–376. doi:https://doi.org/10.1016/0045-7825(87)90046-6.
URL https://www.sciencedirect.com/science/article/pii/0045782587900466

[139] S. Pirzadeh, An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement, in: 14th Computational Fluid Dynamics Conference, 1999, p. 3255.

[140] M. J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of computational Physics 82 (1) (1989) 64–84.

[141] J. Tu, G.-H. Yeoh, C. Liu, Chapter 8 - some advanced topics in cfd, in: J. Tu, G.-H. Yeoh, C. Liu (Eds.), Computational Fluid Dynamics (Second Edition), second edition Edition, Butterworth-Heinemann, 2013, pp. 349–394. doi:https://doi.org/10.1016/B978-0-08-098243-4.00008-1.
URL https://www.sciencedirect.com/science/article/pii/B9780080982434000081

[142] N. Qin, X. Liu, Flow feature aligned grid adaptation, International journal for numerical methods in engineering 67 (6) (2006) 787–814.

[143] M. Rai, D. Anderson, The use of adaptive grids in conjunction with shock capturing methods, in: 5th Computational Fluid Dynamics Conference, 1981, p. 1012.

[144] P. J. Roache, Quantification of uncertainty in computational fluid dynamics, Annual review of fluid Mechanics 29 (1) (1997) 123–160. doi:https://doi.org/10.1146/annurev.fluid.29.1.123.

[145] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of computational physics 25 (3) (1977) 220–252.

[146] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[147] W.-X. Huang, F.-B. Tian, Recent trends and progress in the immersed boundary method, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 233 (23-24) (2019) 7617–7636.

[148] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Progress in Aerospace Sciences 65 (2014) 1–21.

[149] Simulating flows with moving rigid boundary using immersed-boundary method, Computers & Fluids 39 (1) (2010) 152–167. doi:https://doi.org/10.1016/j.compfluid.2009.07.011.
URL https://www.sciencedirect.com/science/article/pii/S0045793009001108

[150] R. Ghias, R. Mittal, T. Lund, A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries, in: 42nd AIAA aerospace sciences meeting and exhibit, 2004, p. 80.

[151] P. J. Roache, Perspective: a method for uniform reporting of grid refinement studies (1994).

[152] A. Di Mascio, R. Paciorri, B. Favini, Truncation error analysis in turbulent boundary layers, J. Fluids Eng. 124 (3) (2002) 657–663. doi:https://doi.org/10.1115/1.1478564.

[153] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, Procedure for estimation and reporting of uncertainty due to discretization in cfd applications, Journal of fluids Engineering-Transactions of the ASME 130 (7) (2008).

[154] D. C. Wilcox, et al., Turbulence modeling for CFD, Vol. 2, DCW industries La Canada, CA, 1998.

[155] J. R. Shewchuk, Triangle: A two-dimensional quality mesh generator and delaunay triangulator. (2005). arXiv:http://www.cs.cmu.edu/~quake/triangle.html.

[156] J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: M. C. Lin, D. Manocha (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Vol. 1148 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 203–222, from the First ACM Workshop on Applied Computational Geometry. doi:10.1007/BFb0014497.

[157] John Burkardt's home page, last accessed: January 19, 2021.
URL https://people.sc.fsu.edu/~jburkardt/index.html

[158] A. Bonfiglioli, R. Paciorri, A mass-matrix formulation of unsteady fluctuation splitting schemes consistent with Roe's parameter vector, International Journal of Computational Fluid Dynamics 27 (4-5) (2013) 210–227. doi:10.1080/10618562.2013.813491.

[159] L. Arpaia, M. Ricchiuto, R. Abgrall, An ALE formulation for explicit Runge-Kutta residual distribution, Journal of Scientific Computing (2014) 1–46doi:10.1007/s10915-014-9910-5.

[160] G. Moretti, On the matter of shock fitting, in: R. D. Richtmyer (Ed.), Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics, Springer, Berlin, Heidelberg, 1975, pp. 287–292. doi:10.1007/BFb0019764.

[161] J. D. Müller, Grid generation tools: Delaunay triangulation with delaundo.
URL http://www.ae.metu.edu.tr/tuncer/ae546/prj/delaundo/

[162] M. S. Ivanov, A. Bonfiglioli, R. Paciorri, F. Sabetta, Computation of weak steady shock reflections by means of an unstructured shock-fitting solver, Shock Waves 20 (4) (2010) 271–284.
URL https://doi.org/10.1007/s00193-010-0266-y

[163] F. Grasso, S. Pirozzoli, Shock-wave-vortex interactions: Shock and vortex deformations, and sound production, Theoretical and Computational Fluid Dynamics 13 (6) (2000) 421–456. doi:10.1007/s001620050121.

[164] R. Paciorri, A. Bonfiglioli, Accurate detection of shock waves and shock interactions in two-dimensional shock-capturing solutions, Journal of Computational Physics 406 (2020) 109196. doi:10.1016/j.jcp.2019.109196.

[165] G. Moretti, Thirty-six years of shock fitting, Computers & Fluids 31 (4-7) (2002) 719–723. doi:10.1016/s0045-7930(01)00072-x.

[166] H. Si, TetGen a quality tetrahedral mesh generator and a 3D Delaunay triangulator, http://tetgen.berlios.de/ (2011).

[167] P. J. Frey, Yams: a fully automatic adaptive isotropic surface remeshing procedure, Rapport technique n° 0252, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France), iSSN 0249-0803 (2001).
URL http://www.ann.jussieu.fr/~frey/software.html

[168] I. Tecplot, Tecplot360 user's manual (2013).

[169] T. J. Tautges, Cgm: A geometry interface for mesh generation, analysis and other applications, Engineering with Computers 17 (3) (2001) 299–314.

[170] S. Gosselin, C. Ollivier-Gooch, Constructing constrained delaunay tetrahedralizations of volumes bounded by piecewise smooth surfaces, International Journal of Computational Geometry & Applications 21 (05) (2011) 571–594.

[171] F. G. Blottner, Accurate navier-stokes results for the hypersonic flow over a spherical nosetip, Journal of spacecraft and Rockets 27 (2) (1990) 113–122.

[172] C. Ollivier-Gooch, R. Paciorri, A. Assonitis, A. Bonfiglioli, An unstructured shock-fitting technique for three-dimensional flows with shock interactions, in: WCCM-ECCOMAS Congress, Vol. 200, ESP, 2021, pp. 1–12. doi:10.23967/wccm-eccomas.2020.065.

[173] E. Houtman, W. Bannink, B. Timmerman, Experimental and numerical investigation of the high-supersonic flow around an axi-symmetric blunt-cylinder-flare model, in: Aerothermodynamics for space vehicles, Vol. 367, 1995, p. 517.

[174] B. Edney, Anomalous heat transfer and pressure distributions on blunt bodies at hypersonic speeds in the presence of an impinging shock., Tech. rep., Flygtekniska Forsoksanstalten, Stockholm (Sweden) (1968).

[175] A. Assonitis, C. Ollivier-Gooch, R. Paciorri, A. Bonfiglioli, 3d flow computations over blunt bodies at hypersonic speeds using a shock-fitting technique, in: AIAA Aviation Forum 2022, 2022. doi:10.2514/6.2022-3989.

[176] M. Pizzarelli, F. Nasuti, R. Paciorri, M. Onofri, Numerical analysis of three-dimensional flow of supercritical fluid in cooling channels, AIAA journal 47 (11) (2009) 2534–2543. doi:https://doi.org/10.2514/1.38542.

[177] D. W. Zaide, P. L. Roe, A second-order finite volume method that reduces numerical shockwave anomalies in one dimension, in: Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2013, pp. –. doi:10.2514/6.2013-2699.

[178] M. S. Ivanov, A. Bonfiglioli, R. Paciorri, F. Sabetta, Computation of weak steady shock reflections by means of an unstructured shock-fitting solver, Shock Waves 20 (4) (2010) 271–284. doi:10.1007/s00193-010-0266-y.

[179] M. platform website, Mmg platform website, https://www.mmgtools.org/ (2017).

[180] A. Bonfiglioli, R. Paciorri, Comparative study of stagnation point anomalies by means of shock capturing and shock fitting unstructured codes, in: Proceeding of The 6th European Symposium on Aerothermodynamics for Space Vehicles, 3–6 November 2008, Versailles, France, Vol. 659, 2009.

[181] S. Rehman, J. Eldredge, X. Zhong, J. Kim, An evaluation of shock-capturing methods on a hypersonic boundary layer receptivity problem, in: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, 2009, p. 941.

[182] F. Nasuti, M. Onofri, Viscous and inviscid vortex generation during startup of rocket nozzles, AIAA J. 36 (5) (1998) 809–815. doi:10.2514/2.440.

[183] E. Johnsen, J. Larsson, A. V. Bhagatwala, W. H. Cabot, P. Moin, B. J. Olson, P. S. Rawat, S. K. Shankar, B. Sjögreen, H. Yee, X. Zhong, S. K. Lele, Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves, Journal of Computational Physics 229 (4) (2010) 1213 – 1237. doi:10.1016/j.jcp.2009.10.028.

[184] P. Rawat, X. Zhong, Direct numerical simulations of turbulent flow interactions with strong shocks using shock-fitting method, in: 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2011, p. 649.

[185] R. Courant, K. O. Friedrichs, Supersonic Flow and Shock Waves, Fourth Edition 1963 Edition, Vol. I of Pure and Applied mathematics, Interscience Publishers, New York, 1948.

[186] W. HAYES, D. 1960 gasdynamic discontinuities.

[187] J. Ryu, D. Livescu, Turbulence structure behind the shock in canonical shock–vortical turbulence interaction, Journal of Fluid Mechanics 756 (2014).

[188] J. Sternberg, Triple-shock-wave intersections, The Physics of Fluids 2 (2) (1959) 179–206. doi:10.1063/1.1705909.

[189] M. Sichel, Structure of weak non-hugoniot shocks, The Physics of Fluids 6 (5) (1963) 653–662.

[190] M. H. Carpenter, J. H. Casper, Accuracy of shock capturing in two spatial dimensions, AIAA Journal 37 (9) (1999) 1072–1079. doi:10.2514/2.835.

[191] T. Levi-Civita, G. Lampariello, Caratteristiche dei sistemi differenziali e propagazione ondosa, 1931.

[192] T. J. Poinsot, S. Lelef, Boundary conditions for direct simulations of compressible viscous flows, Journal of computational physics 101 (1) (1992) 104–129.

[193] D. V. Kotov, H. Yee, A. A. Wray, A. Hadjadj, B. Sjögreen, High order numerical methods for the dynamic sgs model of turbulent flows with shocks, Communications in Computational Physics 19 (2) (2016) 273–300.

[194] H. Babinsky, J. K. Harvey, Shock wave-boundary-layer interactions, Vol. 32, Cambridge University Press, 2011.

[195] G. Moretti, Circumspect exploration of multidimensional imbedded shocks, AIAA Journal 14 (7) (1976) 894–899.

[196] K. Masatsuka, I do like cfd, vol. 1, version 2.6 (2018).

[197] F. Marconi, G. Moretti, Highly accurate prediction of unsteady viscous flows, technical report, DTIC Document (1992).

[198] J. Délery, J.-P. Dussauge, Some physical aspects of shock wave/boundary layer interactions, Shock waves 19 (6) (2009) 453–468.

[199] G. Degrez, C. Boccadoro, J. F. Wendt, The interaction of an oblique shock wave with a laminar boundary layer revisited. an experimental and numerical study, Journal of fluid mechanics 177 (1987) 247–263.

[200] M. S. Holden, Boundary-layer displacement and leading-edge bluntness effects on attached and separated laminar boundary layers in a compression corner. ii-experimental study, AIAA journal 9 (1) (1971) 84–93.

[201] M. S. Holden, Theoretical and experimental studies of the shock wave-boundary layer interaction on compression surfaces in hypersonic flow, Tech. rep., Cornell Aeronautical Lab Inc Buffalo NY (1970).

[202] J. B. McDevitt, L. L. Levy Jr, G. S. Deiwert, Transonic flow about a thick circular-arc airfoil, AIAA Journal 14 (5) (1976) 606–613.

[203] J. MCDEVITT, L. LEVY, JR, G. DEIWERT, Transonic flow about a thick circular-arc airfoil, in: 8th Fluid and PlasmaDynamics Conference, p. 878.

[204] M. W. Rubesin, A. F. Okuno, L. L. Levy Jr, J. B. McDevitt, H. L. Seegmiller, An experimental and computational investigation of the flow field about a transonic airfoil in supercritical flow with turbulent boundary-layer separation, Tech. rep. (1976).

[205] J. B. McDevitt, Supercritical flow about a thick circular-arc airfoil, Tech. rep. (1979).

[206] A. R. Staff, Equations, tables, and charts for compressible flow, Tech. rep.

[207] D. A. Donzis, Shock structure in shock-turbulence interactions, Physics of Fluids 24 (12) (2012) 126101.

[208] Shell Film Unit, Transonic flight, YouTube, uploaded by jasduffy (1950).
URL `https://www.youtube.com/watch?v=bELu-if5ckU`

[209] J. C. J. South, J. L. Thomas, J. Van Rosendale (Eds.), ICASE/LaRC Workshop on Adaptive Grid Methods, 1995.
URL `https://ntrs.nasa.gov/citations/19960011635`

[210] H. Hornung, Regular and mach reflection of shock waves, Annual Review of Fluid Mechanics 18 (1) (1986) 33–58. doi:10.1146/annurev.fl.18.010186.000341.

[211] P. Krehl, M. van der Geest, The discovery of the mach reflection effect and its demonstration in an auditorium, Shock Waves 1 (1) (1991) 3–15. doi:10.1007/BF01414863.

[212] J. Von Neumann, Oblique reflection of shocks, Bureau of Ordinance, Explosives Research Report (1943).

[213] J. Von Neumann, Theory of shock waves, Tech. rep. (1943).
URL https://apps.dtic.mil/sti/pdfs/ADB805024.pdf

[214] R. Richter, P. Leyland, Auto-adaptive finite element meshes, in: NASA Langley Research Center, ICASE (LaRC Workshop on Adaptive Grid Methods, 1995, pp. 219–232.
URL https://ntrs.nasa.gov/citations/19960011650

[215] D. Ait-Ali-Yahia, G. Baruzzi, W. G. Habashi, M. Fortin, J. Dompierre, M.-G. Vallet, Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. part II. structured grids, International Journal for Numerical Methods in Fluids 39 (8) (2002) 657–673. doi:10.1002/fld.356.

[216] G. Ben-Dor, Shock wave reflection phenomena, Springer Verlag, 2007.

[217] L. F. Henderson, On the confluence of three shock waves in a perfect gas., Aeronautical Quarterly 15(2) (1964) 181–197. doi:10.1017/S0001925900003115.

[218] W. R. Smith, Mutual reflection of two shock waves of arbitrary strengths, The Physics of Fluids 2 (5) (1959) 533–541. doi:10.1063/1.1705945.

[219] P. Colella, L. F. Henderson, The von Neumann paradox for the diffraction of weak shock waves, Journal of Fluid Mechanics 213 (1990) 71–94. doi:10.1017/S0022112090002221.

[220] J. Glimm, C. Klingenberg, O. McBryan, B. Plohr, D. Sharp, S. Yaniv, Front tracking and two-dimensional riemann problems, Advances in Applied Mathematics 6 (3) (1985) 259 – 290. doi:10.1016/0196-8858(85)90014-4.

[221] J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: M. C. Lin, D. Manocha (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Vol. 1148 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 203–222, from the First ACM Workshop on Applied Computational Geometry.

[222] J.-D. Müller, P. L. Roe, H. Deconinck, A frontal approach for internal node generation in Delaunay triangulations, International Journal for Numerical Methods in Fluids 17 (3) (1993) 241–255. doi:10.1002/fld.1650170305.

[223] M. Olim, J. M. Dewey, A revised three-shock solution for the Mach reflection of weak shocks ($1.1 < M < 1.5$), Shock Waves 2 (3) (1992) 167–176. doi:10.1007/bf01414639.

[224] S. Kobayashi, T. Adachi, T. Suzuki, Examination of the von Neumann paradox for a weak shock wave, Fluid dynamics research 17 (1) (1995) 13–25. doi:10.1016/0169-5983(95)00020-E.

[225] G. P. Shindyapin, Mach reflection and interaction of weak shock waves under the von Neumann paradox conditions, Fluid dynamics 31 (2) (1996) 318–324. doi:10.1007/BF02029694.

[226] H. J. Pain, E. W. E. Rogers, Shock waves in gases, Reports on Progress in Physics 25 (1) (1962) 287–336. doi:10.1088/0034-4885/25/1/308.

[227] V. Bargmann, D. Montgomery, Report 5011, O.S.R.D. (1945).

[228] K. G. Guderley, Theory of Transonic Flow, Pergamon Press, 1962.

[229] G. T. Kalghatgi, B. L. Hunt, The three-shock confluence problem for normally impinging, overexpanded jets, Aeronautical Quarterly 26 (2) (1975) 117–132. doi:10.1017/S0001925900007265.

[230] E. I. Vasil'ev, T. Elperin, G. Ben-Dor, Analytical reconsideration of the von Neumann paradox in the reflection of a shock wave over a wedge, Physics of Fluids 20 (4) (2008) 046101. doi:10.1063/1.2896286.

[231] B. W. Skews, J. T. Ashworth, The physical nature of weak shock wave reflection, Journal of Fluid Mechanics 542 (2005) 105–114. doi:10.1017/S0022112005006543.

[232] B. W. Skews, G. Li, R. Paton, Experiments on Guderley mach reflection, Shock Waves 19 (2) (2009) 95–102. doi:10.1007/s00193-009-0193-y.

[233] A. Cachucho, B. W. Skews, Guderley reflection for higher mach numbers in a standard shock tube, Shock Waves 22 (2) (2012) 141–149. doi:10.1007/s00193-011-0349-4.

[234] E. I. Vasil'ev, High resolution simulation for the Mach reflection of weak shock waves, in: K. D. Papailiou (Ed.), Proc. of the ECCOMAS 4th Computational Fluid Dynamics Conference, Vol. 1, John Wiley & Sons, Ltd., 1998, pp. 520–527, athens, Greece (7-11) Sept. 1998.

[235] E. I. Vasil'ev, Four-wave scheme of weak Mach shock wave interaction under von Neumann paradox conditions, Fluid Dynamics 34 (3) (1999) 421—427.

[236] A. R. Zakharian, M. Brio, J. K. Hunter, G. M. Webb, The von Neumann paradox in weak shock reflection, Journal of Fluid Mechanics 422 (2000) 193–205. doi:10.1017/S0022112000001609.

[237] A. M. Tesdall, R. Sanders, N. Popivanov, Further results on Guderley Mach reflection and the triple point paradox, Journal of Scientific Computing 64 (3) (2015) 721–744. doi:10.1007/s10915-015-0028-1.

[238] A. M. Tesdall, J. K. Hunter, Self-similar solutions for weak shock reflection, SIAM Journal on Applied Mathematics 63 (1) (2002) 42–61. doi:10.1137/S0036139901383826.

[239] A. M. Tesdall, R. Sanders, B. L. Keyfitz, The triple point paradox for the nonlinear wave system, SIAM Journal on Applied Mathematics 67 (2) (2007) 321–336. doi:10.1137/060660758.

[240] A. Defina, F. M. Susin, D. P. Viero, Numerical study of the Guderley and Vasilev reflections in steady two-dimensional shallow water flow, Physics of Fluids 20 (9) (2008) 097102. doi:10.1063/1.2972936.

[241] A. Defina, D. P. Viero, F. M. Susin, Numerical simulation of the Vasilev reflection, Shock Waves 18 (3) (2008) 235–242. doi:10.1007/s00193-008-0159-5.

[242] C. A. Mouton, Transition between regular reflection and Mach reflection in the dual-solution domain, Ph.D. thesis, California Institute of Technology, Pasadena, California, USA (2007). arXiv:https://thesis.library.caltech.edu/36/1/ThesisRootFinal.pdf.

[243] F. Wecken, Grenzlagen gegabelter verdichtungsstöße, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 29 (1949) 147–155.