Dario Isola

Ph.D. Thesis

2012

AN INTERPOLATION-FREE TWO-DIMENSIONAL CONSERVATIVE ALE SCHEME OVER ADAPTIVE UNSTRUCTURED GRIDS FOR ROTORCRAFT AERODYNAMICS

Politecnico di Milano
Dipartimento di Ing.
Aerospaziale

POLITECNICO DI MILANO
Department of Aerospace Engineering
Doctoral Programme In Aerospace Engineering

An Interpolation-Free Two-Dimensional Conservative ALE Scheme over Adaptive Unstructured Grids for Rotorcraft Aerodynamics

Doctoral Dissertation of:
Dario Isola
Mat. 738536

Supervisor:
Prof. Alberto Guardone

Tutor:
Prof. Pierangelo Masarati

The Chair of the Doctoral Program:
Prof. Sergio Ricci

2012–XXIV cycle

A Lorenza

Summary

This thesis present a novel method for compressible gasdynamics for two-dimensional rotorcrafts applications. The arbitrary Lagrangian-Eulerian formulation of the Euler equations is solved within the finite-volume framework over adaptive grids. The modifications to the topology of the grid, resulting from the adaptation step, are locally interpreted in terms of continuous deformation of the finite volumes built around the nodes. This allows to compute the flow variable over the new grid by simply integrating the arbitrary Lagrangian-Eulerian formulation of the Euler equations, without any explicit interpolation step.

The new, adapted, grids are obtained resorting to a suitable mix of mesh deformation, edge-swapping, node insertion and removal. The adaptation procedure is driven by a sensor that depends on both the geometry/configuration of the domain, i.e. the position of the boundaries at a give time, and the solution, e.g. error estimators based on the gradient or the Hessian matrix. A fixed-point approach to the adaptation problem in the unsteady case has been adopted, by iterating until both grid and solution have converged at each timestep. Both steady and unsteady simulations over adaptive grids are presented that demonstrate the validity of the proposed approach.

The adaptive scheme outlined above is used to tackle typical two-dimensional problems for rotorcraft blade sections, where mesh adaptation is of primary importance to perform efficient unsteady computations while highlighting relevant flow features, such as shocks, wakes or vortices. The developed adaptive scheme is used to carry out high-resolution computations over three selected problems of interest for rotorcraft aerodynamics: an oscillating airfoil, an impulsively started airfoil and parallel blade-vortex interaction.

The material used to compile this thesis has been produced during the course of the Ph.D. research at Politecnico di Milano. Part of this material has been already published in refereed international journals [60, 74, 76] and presented at scientific conferences [89, 93, 59, 58, 91, 92, 75, 56, 151, 57, 77, 90].

Sommario

In questo lavoro è presentato un metodo innovativo per lo studio della gasdinamica comprimibile per applicazioni bidimensionali di tipo elicotteristico. La formulazione Arbtirariamente lagrangiana-euleriana delle equazioni di Eulero è affrontata con una discretizzazione ai volumi finiti su griglie adattive. Le modifiche topologiche del reticolo dovute alla adattazione sono interpretate in chiave continua come deformazione dei volumi finiti costruiti attorno ai nodi. Tale approccio consente di calcolare la soluzione sulla nuova griglia semplicemente integrando le equazioni di governo, senza interpolazione.

L'adattazione di griglia è effettuata utilizzando tecniche di deformazione di griglia, scambio delle diagonali, inserimento e rimozione dei nodi. Un apposito sensore è utilizzato per guidare l'adattazione che dipende contemporaneamente dalla geometria del dominio, in particolare dalla distanza dai contorni, e dalla soluzione, per esempio dal gradiente o dall'essiano del numero di Mach. Un approccio a punto-fisso è stato seguito per estendere le classiche tecniche di adattazione di griglia al caso non stazionario, cioé iterando fino a che sia la griglia che la soluzione sono arrivate a convergenza ad ogni passo temporale. La validità dell'approccio proposto è dimostrata tramite simulazioni su griglie adattive sia nel caso stazionario sia instazionario.

Lo schema di adattazione descritto è quindi utilizzato per studiare correnti bidimensionali tipiche per profili di pala di elicottero, dove il controllo della spaziatura di griglia è chiave per poter cattuarare in maniera efficiente alcune peculiarità della corrente, come onde d'urto, scie o vortici. Lo schema presentato è usato per effettuare simulazioni ad alta risoluzione di tre problemi di tipo elicotteristico: un profilo oscillante, l'avviamento impusivo di un profilo e l'interazione parallela tra vortice e pala.

Il materiale usato per scrivere questa tesi è stato prodotto durante il corso di dottorato presso il Politecnico di milano. Parte di questo materiale è già stata publicata in riviste internazionali [60, 74, 76] e presentato presso conferenze scientifiche [89, 93, 59, 58, 91, 92, 75, 56, 151, 57, 77, 90].

Contents

Chapter 1

Introduction

The present chapter provides an account of the relevant literature on the subject of computational aerodynamics, arbitrary Lagrangian-Eulerian (ALE) methods and mesh adaptation techniques. The reader should be aware that such account is far from being a complete review. It only mentions the literature that has had a major impact on the present work.

In section 1.1 the milestones of numerical methods in aerodynamics are described, in section 1.2 the arbitrary Lagrangian-Eulerian formulation is briefly described together with the most important works that are based on it and in section 1.3 the most popular mesh adaptation strategies for steady and unsteady applications are presented.

1.1 A brief history of numerical methods for aerodynamic applications

In the present section a brief description of the most important numerical methods in fluid dynamics and aerodynamics is presented. The extent of the subject are such that it would be impossible to give a comprehensive description of it. Therefore only the major achievement have been covered with a specific focus on the works that are specifically relevant to the present research.

Incompressible inviscid flows

In the pioneering era of aviation the engineers could only rely on analytical methods and wind-tunnel tests to evaluate the characteristics of the low-speed airfoil sections they designed. In fact, at the time, neither automatic calculators nor numerical methods for partial differential equations were available. To this purpose Theodorsen and Garrick [176] developed a complex potential flow theory to compute the velocity field and pressure around arbitrary airfoil sections. The idea was to map circles into arbitrary airfoil shapes in the complex space. All in all the method was easy to implement and produced results in agreement with those obtained by the experiments.

However this method could not be applied to wings since a complex space is not available in three dimensions and only when the first computers became available and

the idea of solving the three-dimensional potential flow equations by means of integral methods took shape. One of the main advantages of using integral methods is the fact that only the body surface has to be discretized, and not the flow field, and this was vital given the limitations in computational resources. Amongst the most popular panel methods, the one from Morino [137] and Hess [85] covered the bodies with panels and then computed the integrals on the panels itself, i.e. so-called influence coefficients. Once the influence coefficients were computed and stored, the numerical solution simply required the solution of a linear system with a full matrix [100]. Due to the increase in speed of the airplanes, testified by the braking of the so-called sound barrier in 1947 [150], a numerical solution in the transonic regime was needed and the panel method were extended to the transonic regime. Field points needed to be included in the regions of the flow field significantly affected by compressibility, with a sensible increase in the level of complexity of the method. The increase in computational capabilities of the 60s, allowed researchers to move to numerical methods based on the direct solution of partial differential equations.

Full potential flows and the advent of Computational Fluid Dynamics

With aircraft flying at transonic speed, the capability of the flow model to capture shock waves appeared to be paramount. To this end the first efforts were directed to solve the full potential equation for compressible inviscid flows rather than the Euler equations due to limitations in computer power. Nonetheless the adopted finite-difference methods were usually hampered by instabilities, because of the mixed elliptic-hyperbolic nature of the equation in the transonic regime. As later pointed out by Murman and Cole [140], central differences schemes are indeed unsuitable to solve hyperbolic equations and up-winding schemes, which evaluated the differences according to the flow direction, are to be preferred. The first application of the upwind scheme was used to solve the transonic small disturbance equation [140, 86], a simpler version of the full potential model which was later solved by Jameson [94] thanks to the devised the rotated difference scheme. All in all for discontinuous flow problems the finite-volume method proved to be a better choice than the finite-differences [86]. Indeed, if integrated in over a control volume, the governing equations are casted in the form of a conservation law and they can formally handle discontinuities. Application of the method demonstrated its high potential and computations of the flow around wings appeared. A description of the finite volume method can be found in [86], and a more rigorous exposition is found in [112] and [113]. The introduction of finite-volume and finite-difference methods as tools to describe the dynamic of fluids defined a whole new branch of research. The first Computational Fluid Dynamics (CFD) conference consecrated this new era and, as pointed out by Jameson in [95], "The AIAA First CFD Conference, held in Palm Springs in July 1973, signified the emergence of computational fluid dynamics as an accepted tool for airplane design". In the period that followed the research efforts where enthusiastically directed to investigate better numerical schemes, faster solution methods for the equations or flexible ways of generating computational meshes.

Euler equations, artificial dissipation and reconstruction schemes

The first multi dimensional Euler solver where developed at the beginning of the 80s, thanks to improved numerical knowledges and increased computational resources. A

central discretization plus artificial dissipation scheme that uses a Runge-Kutta pseudo-time marching procedure to reach steady state was devised by Jameson, Schmidt and Turkel [97]. Such scheme, commonly referred to as JST scheme, is very effective and is still in use today.

A rather different approach is followed by the so called Godunov method [73] that considers the flux between two cells to be similar to that obtained by solving the Riemann problem of gas-dynamics, i.e. the solution of the Euler equations starting from a piecewise constant initial data having a single discontinuity. In the Godunov method it is not necessary to introduce artificial dissipation, which is naturally provided by upwinding. However, it is only first-order accurate, thus not very attractive in practice. Second-order accuracy was achieved by Van Leer, who introduced a reconstruction procedure that was named Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) [186]. MUSCL schemes are still Godunov-like schemes, in that they use a Riemann solver, but a linear reconstruction of the primitive variables at the cell interface is performed. With reconstruction-based schemes high-accuracy is obtained in smooth flows, but unwanted oscillations in the solution are usually introduced when discontinuities or sharp gradients are present.

To preserve the monotonicity of the scheme the reconstructed variables had to be "limited" in order to enforce a monotonic behavior. Amongst the several limiters that have been introduced by researchers, the Van Albada limiter [185] and the Van Leer [186] are widely used.

Aside from the reconstruction methods, a slightly different class of high-resolution schemes has been developed that avoid the arise of spurious oscillations near shocks [186], Such schemes limit the fluxes instead of the reconstructed variables to ensure the total variation diminishing (TVD) property [113]. In general, when the limiter of the schemes that act on the system fluxes the flux limiter is used, while when the limiter acts on system state, e.g. velocity, density and pressure, the slope limiter is preferred. Note that both flux limiters and slope limiters have the effect of bounding the solution gradient near shocks or discontinuities.

To reduce the computational burden of the original Godunov method, which requires an iterative Newton method to compute the "exact" solution of the Riemann problem, many approximate schemes where developed. Amongst the several solvers that have been introduced, Roe's approximate Riemann solver [157] and Osher's scheme [143] are widely used. A detailed description of the several reconstruction-based and dissipation-based methods developed and their applications can be found in [109].

Unstructured grid solvers

The main advantage of finite-volume scheme is indeed the fact that they are a natural framework for the discretization of conservation laws. However, another interesting feature of such methods is the ability of discretizing the equations on hybrid and unstructured grids, i.e. meshes made by elements of different types and without any particular ordering of the nodes. Other methods exist that may be used for the implementation of unstructured flow solvers, e.g., the finite-element method and the spectral element method.

Structured grid generation may be extremely time consuming and eventually impossible, especially around complex configurations such as complete aircraft. Compared to structured solvers, unstructured solvers usually require a much larger amount of memory

for the storage of several quantities such as the gradient, which cannot be computed on-the-fly as in the case of structured solvers. Moreover computational power is needed in a larger amount because an unstructured solver performs in general more operations than a structured one. E.g., the solution of the linear system is complicated by the fact that the matrix is sparse and it generally requires preconditioning techniques which require a substantial amount of preparatory work.

The constant growth in computer performances allowed to start investigating unstructured solvers in the mid 80s. In 1988 the Euler equations were solved on an unstructured median-dual mesh, obtained by the union of the nodes with the elements center, by Désidéri and Dervieux [47] with the so-called upwind triangle scheme, which was the first attempt to implement reconstruction schemes on unstructured meshes. Barth [12] investigated the implementation of reconstruction schemes and solution methods suitable for median-dual meshes and his work was cardinal in the further progress of unstructured solvers. He has also shown the equivalence of the finite-volume method on the median-dual mesh to one type of finite-element method (FEM), that was already well spread in structural analysis [86, 203].

The first computations of the flow around complete aircraft configurations appeared in the second half of the 80s. The 3D transonic potential flow around the complete Dassault Falcon by means of finite elements was computed [28] and an unstructured FEM solver was used to compute the transonic Euler flow around the Boeing 747 by Jameson, Baker and Weatherill [96]. The numerical scheme used by Jameson et al. was essentially an adapted version of his dissipation scheme and the FEM he used was equivalent to the finite-volume method on the median-dual mesh.

During the 90s several solution techniques, specifically developed for structured solvers, have been successfully applied to unstructured solvers [126]. E.g. the multigrid method has been largely studied by Marviplis [129] as well as by other authors [196, 101, 84, 170, 138, 97], and the lower-upper symmetric Gauss-Seidel (LU-SGS) method [98] has been refined by several authors [41, 121]. Amongst the others two comprehensive review articles on unstructured solvers have been written by Venkatakrishnan [188] and Mavriplis [130].

Very high order schemes

Since the second half of the 90s edge-based solvers have become a standard implementation strategy of numerical methods for fluid dynamics, within both the aeronautical industry and research centers. To name but a few of those solvers: the Funcode of NASA Langley, the Edge code of the FOI, the elsA from ONERA, the TAU code of the DLR and the NSU code of Mavriplis.

During the last decade the research on numerical methods for fluid dynamics have moved toward different approaches which, for the most part, relies on very high-order space integration. For example Discontinuous have been developed to combine features of the finite element and the finite volume framework and have been successfully applied to hyperbolic, elliptic and parabolic problems arising from a wide range of applications. For a complete review on the Discontinuous Galerkin methods the reader is referred to [8, 13, 14]. Another approach that has received much attention during the last decade is the so-called Residual Distribution (RD) method, which allows to obtain very high accuracy in space. The mathematical foundations of RD have been primarily driven by a large number of publications by Abgrall and co-workers [1, 3, 2]. The Ph.D. thesis by

Ricchiuto provides a very complete description of the fundamental mathematics behind Residual Distribution methods [156].

To further development and extension of adaptive higher order variational methods for aerospace applications is dedicated the ADIGMA project. The reader is referred to [102] for an overview over the latest developments and achievements in the field of numerical methods for fluid dynamics.

1.2 Arbitrary Lagrangian-Eulerian methods in continuum mechanics

The numerical simulation for fluid-dynamic applications and nonlinear solid mechanics often requires to deal with strong distortions of the continuum under consideration while allowing for a clear delineation of free surfaces and fluid-fluid, solid-solid, or fluid-structure interfaces. The choice of the kinematic description of the material determines how the continuum is related to the computational mesh. This has a major impact over the overall accuracy of the method and over the capabilities to complete the simulation, e.g. mesh invalidation due to an excessive distortion.

Two natural description of the motion exists in continuum mechanics: the Lagrangian and the Eulerian one [40, 124]. The mesh nodes of the Lagrangian algorithms move at the local velocity of material particles during the motion and are mostly adopted in structural mechanics. Such approach is specifically well-suited to track interfaces between different materials but it is limited by the instability to follow large mesh distortion, mostly caused by shear movements and vorticity. Eulerian algorithms find natural applications in fluid dynamics, in which the mesh is fixed and the continuum material "flows inside" the mesh. No problems in terms of mesh distortion arise but it is generally harder to identify accurately the interfaces, where present, and an higher level of error could be obtained due unresolved regions.

The arbitrary Lagrangian-Eulerian description of the fluid motion is an intermediate formulation developed to combine the advantages of the other two classical approaches and possibly minimize their respective drawbacks. In the ALE description of the motion, the grid nodes are moved in some arbitrarily specified way in order to control the quality of the mesh and, therefore, the accuracy of the solution. Comprehensive reviews of the ALE approach have been written by Hirt, Amsden and Cook [87] and by Donea [49].

Note that by the ALE method two similar, yet different, type of approaches are intended. As pointed out by Shashkov et al. [104] one is the variation of Lagrangian hydrodynamics which avoids mesh distortion by rezoning and remapping and another uses a mesh smoothly moving in a predefined way, typically determined by moving boundaries rather than by fluid motion. The underlying idea is applied in two different ways.

Decoupled ALE approach

In their earlier applications ALE algorithms where used to tackle solid mechanics problems. To this extent already existing Lagrangian solvers for fluid or solid motion are usually adapted to use the ALE formulation in order to increase accuracy and performances. Those type of algorithm usually features a three steps procedure:

Lagrangian update: during this phase the equations of motion are explicitly updated.

In fluid dynamics the governing equations, e.g. Euler equations or Navier Stokes equations, are casted and solved in the Lagrangian framework where each mesh cell is considered as a fluid particle moving along with the grid, i.e. with no mass flux between the cells. This phase can be followed by an implicit iteration of the momentum equation with the equation of state. Indeed when the material velocities are much smaller than the fluid sound speed, the optional implicit phase allows sound waves to move many cells per cycle, thereby significantly improving computational efficiency.

The Lagrangian phase can be used to simulate the behavior of more sophisticated materials, e.g. the Quotidian EOS. The hyperbolic Lagrangian system is treated by methods that conserve total energy [35, 34], and artificial viscosity schemes can be implemented into the difference scheme [103].

Rezone: during the rezone phase the mesh quality is improved where necessary, so that the computation can continue with desired precision. This can be done in several ways, mesh regularization/smoothing as the Winslow approach [197] or combining geometry-based node placement with numerical optimization of a quality indicator which allows to recover invalid elements [183].

Remap: after the rezone phase has been performed the computational grid is different from the one over which the solution had be computed with the Lagrangian step. The solution has therefore to be interpolated from the old grid to the new one. This operation must conserve mass, momentum, and total energy, must preserve the monotonicity of the solution should be as accurate as possible. The ALE formulation is used to effectively obtain these requirements. Many approaches exists, depending on how the ALE equations are discretized, on perform an interpolation weighted by the swept-area followed by a repair step, which prevent under/overshoots [105, 67]. Other techniques combine low-order inter cell fluxes with some portion of higher-order fluxes, in a flux-limiter fashion, e.g. the Flux-Corrected Remapping [184].

As described above this type of ALE methods are inherently Lagrangian, in that the governing equations of the fluid or solid are resolved in a Lagrangian fashion and a conservative interpolation is performed resorting to the ALE framework.

It has to be noted that the "Lagrangian update" step can be replaced by a classic Eulerian update, where the fluid governing equations are solved, and the above scheme reduces to a classic three-step procedure for unsteady adaptive computations: time-advance, mesh deformation and interpolation. Differently from the Lagrangian ALE approach, however, the Eulerian ALE approach poses some limitations, e.g. the interpolation step must be repeated more than once if multi-step schemes are used to advance in time in the first phase.

Embedded ALE approach

Whether the update phase is carried out in a Lagrangian way or in an Eulerian one, it is always decoupled from the ALE, i.e. remap, phase. Another approach can be obtained if the update and the remap phase are performed simultaneously.

Many problems of interests involve the relative motion of bodies surrounded by a continuum material, i.e. aeroelastic problems or turbomachinery. In this cases the

extension of a classic Eulerian algorithm to the case of moving meshes by means of an ALE formulation appears more natural, since aerodynamic solvers are usually Eulerian, and leads to a class of schemes that are slightly different than the one described above. The flow equations are recasted in the ALE framework which allows to automatically take into account the fact that the mesh points are moving, thus avoiding any explicit interpolation step. Mavriplis and Yang showed how to extend a fixed gird Euler solver to deforming meshes [132], for both Backward Differences Formulæ (BDF) and implicit Runge-Kutta schemes. Michler, De Sterck and Deconinck recasted a residual distribution scheme in the ALE framework [134]. Geuzaine, Grandmont and Farhat have extend a Navier-Stokes solver to moving grids focusing on the design of ALE time-integrators [69].

Recasting Eulerian schemes in the ALE framework is fairly straightforward and usually requires minor modifications to the algorithm. The methods obtained with this approach to the ALE formulation are generally are simpler than the one obtained with the decoupled approach and many key properties, such as conservativeness or monotonicity, of the original schemes are naturally inherited. In this approaches the remap phase is naturally "embedded in" but also "hidden by" the governing equations and this might be misleading. It has to be noted, for example, that the time-space discrete counterpart of the ALE formulation could be interpreted as an interpolation, albeit no explicit interpolation phase occurs.

Geometric Conservation Law

The issue of preserving the time-accuracy in the ALE extension of existing Eulerian codes has been investigated in details only recently [141, 69, 53, 128]. In this respect, Nobile [141] showed that a naïve extension of fixed-grid methods to flows in moving domains does not preserve numerical accuracy and may possibly lead to numerical instabilities. Therefore, care is to be taken in both the evaluation of the local grid velocities and the definition of the geometric quantities, which are necessary to compute the fluxes across a given portion of the domain, which cannot be chosen independently [52]. Thomas and Lombard [177] proposed to supply the discrete statement of the problem with the additional constraint of reproducing a uniform flow field exactly. This condition, known as the Geometric Conservation Law (GCL), is demonstrated to be sufficient to achieve a first order time accuracy [166] but it is neither necessary nor sufficient for higher order accuracy [69]. Moreover, satisfying the GCL is a necessary and sufficient condition to guarantee the nonlinear stability of the integration scheme [147]. An updated review of the literature on the subject can be found in [51].

1.3 Adaptive mesh strategies for PDEs

The accuracy of CFD analysis depends on the spatial discretization of the flow domain, namely, by the grid used in the computations. With an inappropriate distribution of the nodes within the grid predictions of the flow field, in terms of both integral quantities (aerodynamic coefficients) and local flow field features (shock waves, slip lines, etc.) may be inaccurate. When steady problems are tackled it is possible to obtained suitably spaced grids by means of a grid adaptation procedure, i.e. an iterative process that employs information obtained from the solution itself, at either a previous time (unsteady flows) and/or on a coarser grid (steady flows), to generated a computational

grid. Successful implementation of the adaptive strategy can increase the accuracy of the numerical approximations and also decrease the computational cost. In the past two decades, there has been important progress in developing mesh methods for PDEs, including the variational approach of Winslow [198], Brackbill [26], and Brackbill and Saltzman [27]; finite element methods by Miller and Miller [135] and Davis and Flaherty [44]; the moving mesh PDEs of Li and Petzold[116], Cao, Huang, and Russell [31], Stockie, Mackenzie, and Russell [174], and Ceniceros and Hou [39]; and moving mesh methods based on harmonic mapping of Dvinsky [50] and Li, Tang, and Zhang [114, 115].

A key factor for adaption strategy is the definition of a suitable indicator of the error that is used to provide an estimate of the required local grid spacing. The most popular strategies modify the grid to improve the capturing of local flow features or to better computed the flow integral quantities, e.g. the lift coefficient of a complete aircraft. Error sensor developed for adaption to local flow features depends on the gradient or undivided differences of a relevant variable of the flow field [10, 117, 149, 193], or on the Hessian matrix of the solution [37, 80, 146, 195, 193, 200]. Indicators that are based on the norm of the FEM residual the have also been experimented [117, 168] for steady flow applications. As far as integral estimators are concerned, many approaches have been proposed [16, 17, 61, 70, 110, 122, 145, 152, 187]. For highly anisotropic flows, such as boundary layer flows, anisotropic adaption is to be preferred. Generally it requires the specification of a suitable Riemannian measure [38] of the space, to be used in evaluation of the element dimensions. To determine the grid length in the transformed Riemann space, a metric tensor [25, 37, 46, 187] is computed from the solution, e.g. the Hessian matrix of a given physical variable. The grid is then altered to produce elements characterized by the same dimension in the Riemannian metric [68].

A suitable algorithm is used to generate a grid that satisfies the "target" spacing distribution function of the error indicator. The adapted grid can be produced from scratch or obtained from the previous one via subsequent modifications. The first approach is related to the grid generation problem, the subject has been extensively covered, e.g. [179, 65, 131, 154], the second approach has been extensively covered as well and approaches have been followed to perform the alteration step [10]. For example, Peraire and collaborators [146] presented a method, suitable for Delaunay triangulations, based on the remeshing of the whole computational domain, allowing for a direct control over the element sizes and resulting in grids which are virtually independent from the initial one.

Mesh adaptation via local modification

In many cases of practical interest a complete remesh might be undesirable and a grid that satisfy the imposed size-distribution can be obtained by just performing local changes by means of a combination of node movement or r-refinement [133], topology modification e.g. edge-swapping [64], node insertion/deletion or h-refinement [118] and local remeshing [48].

The displacement of the vertices towards desired regions allows to reduce the size of the elements while the number of grid nodes is untouched, however the obtained mesh could feature excessively stretched elements. Element refinement techniques has been first implemented by Berger and Jameson [21] and by Dannenhoffer and Baron [43]. To avoid an excessive growth of the grid size it is possible to balance the number of inserted nodes by deleting those placed in low error regions, i.e. the so called grid derefinement.

Many flow solvers require the grid to be conformal, i.e. without hanging nodes, if so local grid topology reconstruction technique might be necessary after performing a insertion/removal of a node [10, 169]. Common practices are the insertion of new nodes in the center of an element or in the middle point of an edge; the local grid topology is then recovered by means of a coherent splitting of the elements interested by node insertion [48, 127, 168, 195]. In particular, Mavriplis [127] proposed an element based algorithm in which grid conformity is enforced by adding new elements where nodes have been inserted.

An algorithm to perform grid derefinement by element splitting or by removal after a suitable configuration is reached by means of edge-swapping has been presented by Sonar [168]. An edge collapsing derefinement strategy has been applied to triangular anisotropic and isotropic grids by Webster [195] and Dolejsi [48], respectively. In Mavriplis [127] and Webster et al. [195] the topology reconstruction is performed after the a node has been inserted by resorting to a look-up table for each possible element refinement pattern. Instead, in [48] the local grid reconstruction is performed every time a new node is inserted, by connecting all nodes belonging to the two elements sharing the refined edge. This technique, which does not required specific element refinement patterns, is viable only for grids made of triangles.

Unsteady adaptation

As described above the idea of the mesh adaptation method for steady-state configurations is to converge toward a desired fixed point for the pair formed by the mesh and the solution. In unsteady simulations there is no fixed state to converge to and for capturing a time-dependent phenomenon the solution at the each subsequent time step must be somehow predicted.

In this context, few works on mesh adaptation have been done and a very popular approach consists in carrying out the adaptation step after a given number of time iterations [118, 144, 153] . With this approach, the grid is adapted to the solution computed at the time t and the solution at the time $t + \Delta t$ is computed over the previously adapted grid. Thus no predictions in the evolution of the phenomena are carried out and the capabilities of capturing transient solutions depends severely on the Courant number.

Alauzet et al. [7] proposed a transient fixed-point approach to the adaptation problem by introducing a new loop in the main adaptation loop. The same fixed-point approach used for steady computations is repeated at every time t but instead of computing the solution as a steady state it is computed as the unsteady solution at the time $t + \Delta t$. To furthermore improve the capability of the adaptation scheme to capture the transitory, Alauzet et al. [7] introduced a time-step dependent error estimator which is given by the intersection of the error sensor computed between two subsequent times.

1.4 Thesis goals and outline

This thesis present a novel method for compressible gasdynamics to solve the arbitrary Lagrangian-Eulerian formulation of the Euler equations is solved within the finite-volume framework over adaptive grids. Thanks to the interpetation of the grid topology modifications as of continuous deformation of the finite volumes, the solution over the new

(adapted) grid is computed by simply integrating the arbitrary Lagrangian-Eulerian formulation of the Euler equations, without any explicit interpolation step.

An adaptation strategy is proposed for unsteady problems. A suitable mix of mesh deformation, edge-swapping, node insertion and removal is performed to adapt the grid to a new configuration of the geometry or to redistribute the discretization error amongst the grid elements.

Both steady and unsteady simulations over adaptive grids are presented that demonstrate the validity of the proposed approach. In the final chapter typical two-dimensional problems for rotorcraft blade sections are tackled. The adaptive ALE scheme is used to perform high-resolution computations over three selected problems of interest for rotorcraft aerodynamics: an oscillating airfoil, an impulsively started airfoil and parallel blade-vortex interaction. Resorting to mesh adaptation is of primary importance to perform unsteady computations in such problems, where the need of efficiency is combined to the necessity to highlight relevant flow features, such as shocks, wakes or vortices.

Chapter 2: Finite volume ALE scheme for adaptive meshes

The Navier-Stokes equations provide a complete mathematical description of the behavior of flows past aerodynamic bodies. Due to the multi-scale nature of the turbulence that characterize the majority of the flows of interest their solution is still an extreme challenging task. In many problems of practical interest, however the Navier-Stokes equations can be replaced by the more simple Euler equations neglecting the effect of the viscosity and thermal conductivity. The computational costs required to solve the Euler equations are dramatically lower than the ones required to solve the so-called Reynolds-Averaged Navier-Stokes (RANS) coupled with the most simple turbulence models [99].

Section 2.1 - Euler equations: The numerical solution of the RANS equations requires very fine meshes which are characterized by a huge increase of the mesh density approaching the wall in the boundary layer region. On the other hand the Euler equations can predict accurately enough the wave and the induced component of the drag since the viscosity is neglected. As such, their use is limited to specific problems, e.g. high-Reynolds number flows over aerodynamic bodies, and some crucial information cannot be correctly evaluated, e.g. aircraft efficiency.In the present work the Euler equations are the model of choice. A detailed description of the equations and their properties can be found in [86, 113].

Section 2.2 - ALE formulation of the Euler equations: To perform computations over dynamic grids, the Arbitrary Lagrangian-Eulerian approach, in which the control volumes are allowed to change in shape and position as time evolves, is to be preferred to the classic Eulerian one. In the ALE approach an additional contribution to the fluxes is present, that is proportional to the velocity component normal to the interface of the control volume. A particular emphasis is placed on the derivation procedure and on the eigenstructure of Jacobian matrix. Indeed the differences between the Eulerian and the ALE formulation of the governing equations can be reduced to a modification of the eigenvalues of the Jacobian matrix, while the eigenvectors are unchanged. An additional constraint over the interface velocities exists within the ALE framework, the so called Geometric Conservation Law.

Section 2.3 - Space discretization: The flow solver is based upon an unstructured finite-volume formulation used to discretize the ALE formulation of the Euler equations

on the median-dual mesh [12]. The median-dual mesh can be easily drawn on top of the original mesh, connecting each cell center to its edge mid-points. The result of the construction is a mesh where the control volumes can be located on the nodes of the original mesh.

Interesting feature of the median-dual discretization is the so called transparency of the computational mesh [82]. In fact the edge-based data structure used within the solver makes no distinction between 2D and 3D or between different types of elements. There is always a single edge, only with a normal vector linked to it. To automatically satisfy the GCL constraint the velocity integrated along the cell boundary is computed as the time derivative of the area swept by the interface during the movement.

The flow solver employs a Flux Limiter approach in which a second-order centered flux contribution is blended with a first-order one. To avoid numerical oscillations a van Leer limiter [186] is used, which enforces monotonicity in the solution. A piecewise constant representation of the solution is used, therefore the variables have different values across the control volume interfaces and therefore a discontinuity exists. Roe's approximate Riemann solver [158] is used to evaluate the flux given the discontinuous states. The boundary conditions are imposed in a weak form, i.e. by evaluating the boundary fluxes on a suitable boundary state which depends on the type of boundary condition to be enforced. Particular emphasis is placed on the effects of mesh movement on the boundary conditions.

Section 2.4 - Time integration: A standard BDF scheme [106] is adopted to approximate the derivatives with respect to the time. To enforce the time-discrete version of the Geometric Conservation Law the same BDF scheme is adopted to compute the grid velocity, which is therefore proportional to the area swept by the interfaces during each time-step. The solution of the flow equations at a given time is found by means of an implicit dual-time stepping scheme. The scheme is obtained as an application of the defect correction method to the semi-discrete form of the equations [101]. A local pseudo-time step is added to each control volume and is gradually increased to infinity according to the level of convergence of the solution. In practice, the scheme coincides with a backward Euler method that uses an approximate Jacobian [126].

At each pseudo-time step the flow solution is updated by solving a sparse linear system of equations. The solution is obtained iteratively by means of a Symmetric Gauss-Seidel (SGS) procedure. The system is therefore solved by a forward sweep on the nodes for the lower-triangular part of the matrix, followed by a backward sweep for the upper-triangular part. The linear iterations are stopped when the norm of the linear residuals vector is around one order of magnitude smaller than the norm of the residuals vector. The method has some resemblance with the LU-SGS method [98, 121], of whom SGS may be seen as an improvement.

Section 2.5 - Test cases for steady problems: The edge-based finite volume solver is tested on standard fluid dynamic and aerodynamic test cases in order to assess the properties of the numerical scheme, i.e. steady flow past a transonic airfoil, steady low Mach number flow around a cylinder and the steady supersonic flow facing a positive slope. Grid convergence is checked for the presented test cases and the convergence curve obtained with the SGS scheme are compared with the one obtained with a GMRES solver and LU solver.

Section 2.6 - ALE scheme for adaptive grids: the implicit unsteady finite-volume scheme described in section 2.4 is extended to the case of adaptive grids. The application

of local modifications of the mesh topology, e.g. edge-swapping and mesh de/refinement, causes a modification in the shape of the cells. In principle, due to this modification the solution at new time is to be interpolated over the new grid. The changes in the topology are interpreted as continuous deformations of the finite volumes happening in the time lapse from t to $t + \Delta t$. This is reflected in the application of a correction term to the interface velocities given by the grid movement. The solution over the new grid can be therefore computed by simply integrating the governing equations.

To conserve the solution an additional flux term is to be included in the system for every edge that has been removed from the mesh. Similarly an additional governing equation has to be taken into account for every deleted cell. The nature of such additional contributions depends strongly on the type of time integration scheme adopted.

Chapter 3: Mesh update strategy

In the third chapter the strategy followed to perform the mesh adaptation of a grid made of triangles is briefly described. A suitable mix of very standard techniques is adopted, given by mesh deformation, edge-swapping and node insertion/deletion.

Section 3.1 - Mesh deformation: A mesh deformation technique is adopted to permit the movement of the boundaries, while maintaining high level of grid quality reducing numerical errors. A two steps procedure is carried out: first each boundary node is displaced as prescribed by a given law, then the position of the inner nodes is modified accordingly. The used internal node-displacement algorithm extends to idea of the elastic analogy that represent each element as a deformable body presented by Belytschko, Liu and Moran [18]. The obtained algorithm works well with grids made of triangles and the provided examples demonstrate both its robustness and computational efficiency.

Section 3.2 - Edge-swapping: For large displacements of the mesh nodes the topology is altered with an edge-swapping technique [64], which allow to preserve the total number of grid nodes. The edge-swap consists in altering the connectivity of a given couple of triangular elements by deleting the edge connecting the two vertexes shared by the two elements and by adding a new edge connecting the other two vertexes.

A suitable element quality measure is defined to decide whether an edge must be swapped. A scale-invariant quality measure Q has been chosen among those presented in [165] for two-dimensional grids, which indicate how close an element is to the equilateral triangle. This allow to improve the condition number of the stiffness matrix associated with the fluid problem. Presented numerical examples show that large grid deformations can be achieved with a suitable combination of mesh deformation and edge-swapping.

Section 3.3 - Node insertion and deletion: Several type of element refinement techniques can be found in literature, even for simple triangular meshes [63]. A simple barycentric node insertion is performed on domain elements, while the boundary elements are split in half. Domain and boundary nodes removal are performed reconstructing the local connectivity in order to locally maximize the element quality.

In the present work the spacing distribution is controlled in two ways: as proportional to the distance from the boundaries and as proportional to a suitably defined error indicator. The first approach is specifically suited for aerodynamic applications, indeed in most of the cases of interest the smaller elements are gathered close to the solid walls and the size smoothly increase as the maximum dimension is prescribed on the outer

boundary. The second approach aims to capture the local features of the flow-field locally changing the grid spacing according to the principle of error equidistribution [174, 63]. In this way, if a computational region has a numerical error higher than the average, it will be refined using a technique of node insertion; on the other hand, regions with a lower numerical error will be derefined thanks to a deletion node procedure. An estimator is defined to identify regions with high and low numerical error which is usually a function of the flow gradients [10, 117, 148, 193], of the Hessian matrix $\mathcal{H}$ of a convenient sensor variable [37, 200, 195], of the vorticity or of the substantial derivative of the density [181].

Standard fluid dynamics flow fields usually present features with different intensities, e.g. the gradients near the shock wave and the rarefaction fan that are computed as solution of a one dimensional Riemann problem. In order to capture flow features with lower intensities a multiple passages technique is adopted to suitably modify the list of nodes that participate in the computation of the de/refinement threshold [4].

Section 3.4 - Grid adaptation for unsteady applications: For steady problems is computed the implementation of an adaptation scheme is quite straightforward and it essentially seeks the convergence of both the solution and the computational grid. The application of grid adaptation to the unsteady case different approaches are possible.

Mohammadi and Hecht [136] use a "forward" approach in which the grid is adapted over the solution computed at the time t, the solution at $t + \Delta t$ is computed over the new grid and so on. However Such procedure is suitable only if very small time steps are adopted since, as pointed out by Alauzet et al.[7]: "the mesh is late over the solution".

In the present work a an approach similar to the one proposed by Alauzet et al.[7] is followed, which consists in a iterative procedure which alternate a solution prediction phase with the adaptation one, when the grid has converged to a stationary state a new time step is tackled. By doing so the grid is essentially adapted over the solution computed at a current time, instead of the previous one. To further increase the grid-convergence rate and the overall efficiency of the algorithm an error interpolation technique has also implemented that allows to repeatedly apply the adaptation procedure without computing the solution over the new grid.

Chapter 4: Simulations of reference compressible flows

Numerical experiments are carried out to test the accuracy of the proposed solution procedure. Both steady and unsteady cases are tackled to demonstrate the correctness of the used adaptation strategy.

Section 4.1 - Oblique shock problem: First the computational efficiency of the implemented adaptation strategies is evaluated against the standard steady oblique-shock test case where no complications in the domain geometry are present and the analytical solution is simple enough that only two shocks are featured in the flow field [180].

Section 4.2 and 4.3 - Steady flow past AGARD 02 and NACA 0012 airfoil sections: Steady flow simulations around the AGARD-02 and NACA 0012 airfoil are performed to assess the solver capabilities at simultaneously capturing relevant flow features of transonic flows, including shock waves, shear lines and regions of smooth flows.

Section 4.4 - Sod Problem: The shock tube problem first proposed by Sod [167] is tackled to test the capabilities of the adaptive scheme to describe a transient solution.

The Sod problem features simultaneously a shock waves, a contact discontinuity and a rarefaction fan moving inside the domain and an analytical/exact solution is available. It is thus well suited to study different error sensors in the unsteady case.

Section 4.5 - Forward facing step : The supersonic forward-facing step problem made famous by Woodward and Colella [199] is also tackled. The forward-facing step test case is particularly well suited to test the solver capabilities in capturing very complex flow fields made by curved shocks that interacts with the wall and with other discontinuities.

Chapter 5: Simulations of compressible flows around rotorcraft blade sections

Helicopter simulation is a challenging problem due to the complexity of the flow field generated by the rotor disk, and the interaction between vortices, blades and fuselage. Prediction of helicopter performance strongly dependend on the accuracy of the prediction of the transonic flow past the blade on the advancing side of the rotor and on the resolution of blade-vortex and blade-wake interactions. Therfore a robust and accurate compressible CFD solver is essential in computing the flow around rotor blades.

As briefly discussed in section 1.1, the stability of the majority of the CFD solvers is ensured by the introduction of a certain amount of numerical dissipation, which is generally proportional to the mesh size. To this purpose a method that captures the vortical structures in order to properly resolve a helicopter wake is crucial, since excessive numerical dissipation may lead to wrong preditction of the intensity of the wake or the vortices. To this purpose the idea of Vorticity Confinement, first proposed by Steinhoff, has shown to be effective in treating concentrated vortical regions in coarse grids [191, 119, 29]. Another approach that can be adopted consists in increase the grid resolution in the region of the domain that surrounds the vortex core [175, 142].

In this chapter typical two-dimensional problems for rotorcraft blade sections are presented, appling the adaptive scheme outlined in chapter 3 and 2 to perform high-resolution computations involving pitching motion, airfoil start-up and parallel blade-vortex interaction.

Section 5.1 - Pitching Airfoil : Compressible inviscid subsonic and transonic flow computations around a pitching NACA 0012 airfoil are carried out. The variation of the angle of attack in time is prescribed analytically as a sinusoidal function. This type of pitching movement is similar to the one that is performed by a blade section for an helicopter rotor in forward flight, where a variation the in cyclic pitch angle is used to generate a thrust force. A critical difficulty of this type of problem is the fact that a shock wave is continuously moving from the upper to the lower side of the airfoil and, because of the variation of the entropy increase, a shear wake is intermittently shed. The adaptation scheme must efficiently capture all this flow features.

Section 5.2 - Start-up vortex from the NACA 0012 airfoil: The 2D unsteady flow past the NACA 0012 airfoil impulsively set into motion is also tackled. The generation of the vortical structure is not due to the viscous effects but to the singularity in the trailing edge, therefore the Euler equations can correctly represent the space-time evolution of the start-up vortex in the computational domain. The dynamics of the start-up vortex strongly influence the time history of the force coefficient over the airfoil. Since both the flow field around the airfoil and the start-up vortex need to be captured accurately at the same time, due to the Kelvin's circulation theorem, it is mandatory to adapt the computational grid in an unsteady fashion to follow the vortex dynamics and accurately

compute the flow field close to the airfoil.

Section 5.3 - Parallel blade-vortex interaction: The blade-vortex interaction (BVI) phenomenon occurs when a rotor blade passes within a close proximity of the shed tip vortices from a previous blade. The parallel BVI is the most critical configuration and occurs when the axis of the filament-like vortex is aligned with the axis of the blade. One of the major challenges faced when simulating an AVI is to preserve the vortex structure accurately as it convects through the solution and minimize the numerical dissipation that is inherent in CFD simulations.

The work of Oh et al. [142] addressed this problem by the use of adaptive unstructured meshes to dynamically concentrates mesh points in the region surrounding the vortex core. In the present work a similar approach has been followed reserving a special treatment to the nodes surrounding the vortex core. To better study the capabilities of the adaptive flow solver the simulation vortices transport within the flow field, the free-vortex advection problem is first tackled on both fixed and adaptive grids.

Chapter 2

Finite volume ALE scheme for adaptive meshes

In many cases of aerodynamic interest the Euler equations are sufficient to describe the dynamic behavior of the air. More generally the approximation introduced when deriving these equations are valid for sufficiently dense gases, i.e. the continuum hypothesis must hold, for high values of the flow Reynolds number, i.e. for "thin" boundary layers, for attached flows, i.e. no regions of separation are present in the flow field, and for low values of the flow Mach number, i.e. to avoid possible hypersonic effects.

Since in the present work the aim is to compute the pressure distribution on the surface of aerodynamic bodies, e.g. airfoil sections, that face high-Reynolds transonic air flows, the Euler equations can be considered a good approximation in terms of predictive capabilities.

The present chapter is organized as follows: in section 2.1 the Euler equations are introduced, in section 2.2 the arbitrary Lagrangian-Eulerian formulation is derived, in section 2.3 the finite-volume discretization of the governing equations for grid with constant connectivity is presented, in section 2.4 the time integration scheme is described, with particular emphasis on the ALE formulation, in section 2.4.2 the iterative implicit solver for the equations is introduced, in section 2.5 the result obtained on some standard test cases for steady flows are shown and in section 2.6 the finite-volume scheme is extended to the case of grids with variable connectivity and number of nodes.

2.1 Euler equations

The governing equations for a two-dimensional compressible inviscid fluid flow represent the balance of mass, momentum and total energy of a fixed control volume, i.e.

$$\int_{\mathcal{C}} \frac{\partial \mathsf{u}}{\partial t}\, d\boldsymbol{x} + \oint_{\partial \mathcal{C}} \mathbf{f}(\mathsf{u}) \cdot \boldsymbol{n}\, ds = 0, \qquad \forall \mathcal{C} \subseteq \Omega, \tag{2.1}$$

where $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^2$ is the position vector and $t \in \mathbb{R}^+$ is time and where $\mathsf{u} : \Omega \times \mathbb{R}^+ \to \mathbb{R}^4$ is the vector of the conservative variables, i.e.

$$\mathsf{u} = (\rho, \boldsymbol{m}, E^{\mathrm{t}})^{\mathrm{T}},$$

where ρ is the density, $\boldsymbol{m}$ is the momentum vector and E^{t} is total energy per unit volume. The flux function $\mathbf{f} \in \mathbb{R}^4 \times \mathbb{R}^4$ of Eq. (2.1) is defined as

$$\mathbf{f}(\mathsf{u}) = \left(\boldsymbol{m},\ \boldsymbol{m} \otimes \boldsymbol{m}/\rho + P(\mathsf{u})\ \boldsymbol{I}^2,\ \left[E^{\mathrm{t}} + P(\mathsf{u})\right] \boldsymbol{m}/\rho\right)^{\mathrm{T}}, \tag{2.2}$$

where $\boldsymbol{I}$ is the 2×2 identity matrix and P is the pressure. The outward unit vector normal to the boundary $\partial\mathcal{C}$ of the control volume $\mathcal{C}$ is termed as $\boldsymbol{n}(s) \in \mathbb{R}^2$ and it is a function of the curvilinear coordinate s along $\partial\mathcal{C}$. The dot product between the flux function and the normal vector of Eq. (2.1) is defined as $\mathbf{f} \cdot \boldsymbol{n} = \mathsf{f}_x n_x + \mathsf{f}_y n_y$. Eq. (2.1) is to be made complete by specifying suitable initial and boundary conditions on the boundary $\partial\Omega$, see e.g. [72].

The expression of the fluxes of the Euler equations, i.e. Eq. (2.2), depends on the thermodynamic model chosen for the gas. The polytropic ideal gas is the most simple model and is, with no doubts, the most adopted. For the aerodynamic problems of interest is suitable and well describes the thermodynamics behavior of the air in the lower atmosphere. The equation of state for perfect gas, which reads

$$P(\mathsf{u}) = (\gamma - 1)\left(E^{\mathrm{t}} - \frac{|\boldsymbol{m}|^2}{2\rho}\right), \tag{2.3}$$

fully describe the thermodynamics for the Euler equations, where γ is the ratio of specific heats, which for air is about 1.4. The speed of sound, for a polytropic ideal, gas depends on the temperature only and it can be expressed as a function of the conserved variables as

$$c = \sqrt{\gamma \frac{\gamma - 1}{\rho}\left(E^{\mathrm{t}} - \frac{|\boldsymbol{m}|^2}{2\rho}\right)}.$$

2.1.1 Non dimensional variables

Dimensionless variables are used for the solution of the equations. A reference value for the pressure and the temperature is chosen, i.e. P_{ref} and T_{ref} respectively. Other reference variables can be obtained through dimensional analysis of the problem, i.e.

$$\rho_{\mathrm{ref}} = \frac{P_{\mathrm{ref}}}{R T_{\mathrm{ref}}}, \qquad \boldsymbol{m}_{\mathrm{ref}} = \sqrt{\rho_{\mathrm{ref}} P_{\mathrm{ref}}}, \qquad E^{\mathrm{t}}_{\mathrm{ref}} = P_{\mathrm{ref}}, \qquad t_{\mathrm{ref}} = \sqrt{\frac{\rho_{\mathrm{ref}}}{P_{\mathrm{ref}}}},$$

where R is the specific gas constant, that for air is about 287.04 $\frac{\mathsf{J}}{\mathsf{kg\,K}}$. The non dimensional counterparts of Eq. (2.1), (2.2) and (2.3) have the same form, but the non equations of state for a polytropic ideal gas read

$$\hat{P} = \hat{\rho}\hat{T} \qquad \text{and} \qquad \hat{E}^{\mathrm{t}} = \hat{c}_{\mathrm{v}}\, \hat{\rho}\hat{T} + \frac{|\hat{\boldsymbol{m}}|^2}{2\hat{\rho}},$$

where the non-dimensional quantities are built as $\hat{P} = P/P_{\mathrm{ref}}$ and the reference value of both the specific gas constant R and the specific the specific heat at constant volume c_{v} is chosen as R. In the following, all the variables are assumed to be dimensionless, the "hat" however is not used since it can be safely dropped without causing confusion.

2.2 ALE formulation of the Euler equations

The Arbitrary Lagrangian-Eulerian formulation of the Euler equations expresses the balance of the conservative variables with respect to a moving control volume. For later convenience, the ALE equations for an arbitrary control volume $\mathcal{C}(t)$ moving with a velocity $\boldsymbol{v}(s,t)$ specified on the boundary $\partial\mathcal{C}(t)$ are now derived by introducing a suitable mapping between the current volume, i.e grid configuration, $\Omega_x(t)$ and the reference or initial one Ω_X, namely

$$\begin{aligned}\boldsymbol{\psi} : \Omega_X \times \mathbb{R}^+ &\longrightarrow \Omega_x \times \mathbb{R}^+ \\ (\boldsymbol{X}, t) &\longmapsto \boldsymbol{\psi}(\boldsymbol{X}, t) = (\boldsymbol{x}, t)\end{aligned}$$

where $\boldsymbol{X}$ indicate the reference domain coordinates. The well-known Reynolds transport theorem provides

$$\frac{d}{dt}\int_{\mathcal{C}_x(t)} \mathsf{u}(\boldsymbol{x},t)\,d\boldsymbol{x} = \frac{d}{dt}\int_{\mathcal{C}_X} J(\boldsymbol{X},t)\,\mathsf{U}(\boldsymbol{X},t)\,d\boldsymbol{X} = \int_{\mathcal{C}_X}\frac{\partial}{\partial t}[J\,\mathsf{U}]\,d\boldsymbol{X}. \tag{2.4}$$

with $\mathsf{U}(\boldsymbol{X},t) = \mathsf{u}(\boldsymbol{x}(\boldsymbol{X},t),t)$ and $d\boldsymbol{x} = J(\boldsymbol{X},t)\,d\boldsymbol{X}$, and where

$$J(\boldsymbol{X},t) = \left|\frac{\partial \boldsymbol{x}(\boldsymbol{X},t)}{\partial \boldsymbol{X}}\right| > 0, \tag{2.5}$$

is the Jacobian of the mapping $\boldsymbol{\psi}$ at each time instant. Note that the shape and position of the reference or initial volume $\mathcal{C}_X$ does not depend on time and therefore the time derivative operator commutes with integration over $\mathcal{C}_X$. From [182], one also has

$$\frac{\partial J}{\partial t} = J\,\boldsymbol{\nabla}_x\cdot\boldsymbol{v}, \tag{2.6}$$

where $\boldsymbol{v}(\boldsymbol{x},t) = \partial\boldsymbol{x}(\boldsymbol{X},t)/\partial t$ is the local velocity of the volume and $\boldsymbol{\nabla}_x\cdot(\,\cdot\,)$ is the divergence operator in Ω_x. Substituting Eq. (2.6) and the identity $\partial\mathsf{U}/\partial t = \partial\mathsf{u}/\partial t + \boldsymbol{v}\cdot\boldsymbol{\nabla}_x\mathsf{u}$ in Eq. (2.4) in the current configuration one obtains

$$\frac{d}{dt}\int_{\mathcal{C}_x(t)} \mathsf{u}\,d\boldsymbol{x} = \int_{\mathcal{C}_x(t)}\left[\frac{\partial \mathsf{u}}{\partial t} + \boldsymbol{\nabla}_x\cdot(\mathsf{u}\boldsymbol{v})\right]\,d\boldsymbol{x}.$$

The governing equations in the ALE framework for an arbitrary control volume $\mathcal{C}(t) \in \Omega_x(t)$ moving with mesh velocity $\boldsymbol{v}$ are finally obtained as follows

$$\frac{d}{dt}\int_{\mathcal{C}(t)} \mathsf{u}\,d\boldsymbol{x} + \oint_{\partial\mathcal{C}(t)}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}\big]\cdot\boldsymbol{n}\,ds = 0, \qquad \forall\mathcal{C}(t)\subseteq\Omega(t), \tag{2.7}$$

where differently from equation (2.1), here $\boldsymbol{n}$ and $\boldsymbol{v}$ are functions of both s and t.

Relation (2.6) can be interpreted as a constraint that the control volume must comply with during the movement [62, 51]. Indeed setting $\mathsf{u}\equiv\mathsf{U}\equiv 1$, integrating (2.6) in time over $\mathcal{C}_X$ from the initial condition $J(0)=1$ and substituting (2.4), one has

$$\int_{\mathcal{C}_X}\left[\frac{\partial J}{\partial t} - J\boldsymbol{\nabla}_x\cdot\boldsymbol{v}\right]\,d\boldsymbol{X} = 0 \quad\Leftrightarrow\quad \frac{d}{dt}\int_{\mathcal{C}_x(t)} d\boldsymbol{x} = \oint_{\partial\mathcal{C}_x(t)}\boldsymbol{v}\cdot\boldsymbol{n}\,ds, \tag{2.8}$$

a condition that is usually referred to as the Geometric Conservation Law. Relation (2.8) represents a constraint that must be satisfied by the interface velocity to ensure that the ψ mapping is regular. The role of the GCL for ALE formulations has been the subject of a large number of investigations, generating controversial opinions. It is worthwhile recalling here some basic issues related to the GCL condition (2.8) and to its discrete version. In general, the GCL is neither a necessary nor a sufficient condition to preserve time accuracy [69], but schemes that violate it usually are polluted by spurious oscillations [132]. Additionally, Shyy et al. [166] showed that errors up to $O(1)$ can be introduced if the GCL is not enforced. It has been shown that using a time integration scheme that is of p-th order on fixed meshes, compliance with the GCL condition up to the p-th order is a sufficient condition to obtain a scheme that is at least first order accurate [54]. Therefore, it is generally accepted that enforcing the GCL results in improving the accuracy and the stability of the numerical scheme. An updated review of the literature on the subject can be found in [51].

2.2.1 Jacobian matrix and eigenstructure

Due to the hyperbolic nature of the problem the Euler fluxes and their derivative with respect to the solution, i.e. Jacobian matrix, are of key importance for the development of numerical schemes. In the ALE framework the Jacobian matrix computed along a given direction $\mathsf{A}(\mathsf{u}, \boldsymbol{n}, \boldsymbol{v}) \in \mathbb{R}^4 \times \mathbb{R}^4$ reads

$$\mathsf{A}(\mathsf{u}, \boldsymbol{n}, \boldsymbol{v}) = \frac{\partial \mathbf{f}(\mathsf{u}) \cdot \boldsymbol{n}}{\partial \mathsf{u}} + (\boldsymbol{v} \cdot \boldsymbol{n}) \boldsymbol{I}^4. \tag{2.9}$$

A is therefore given by the sum of the Jacobian matrix computed in the Eulerian framework and a correction term that is proportional to the projection of the velocity vector along $\boldsymbol{n}$.

Of primary importance for the description of the wave phenomena is the eigenstructure of the Jacobian matrix. $\mathsf{A}(\mathsf{u}, \boldsymbol{n}, \boldsymbol{v})$ can be therefore decomposed in the product of three matrices namely the left eigenvectors $\mathsf{L}(\mathsf{u}, \boldsymbol{n}) \in \mathbb{R}^4 \times \mathbb{R}^4$, the right eigenvectors $\mathsf{R}(\mathsf{u}, \boldsymbol{n}) \in \mathbb{R}^4 \times \mathbb{R}^4$ and the eigenvalues diagonal matrix $\Lambda(\mathsf{u}, \boldsymbol{n}, \boldsymbol{v}) \in \mathbb{R}^4 \times \mathbb{R}^4$.

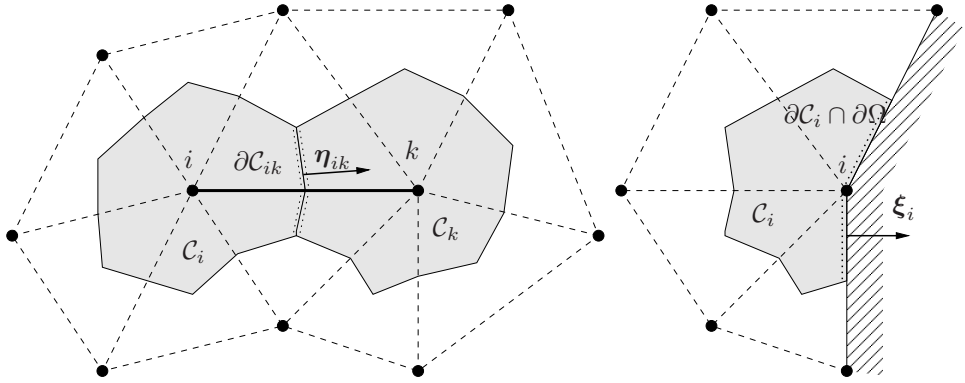
In the ALE framework the expression of the eigenvalues in the 2D case reads

$$\Lambda(\mathsf{u}, \boldsymbol{n}) = \left(\frac{\boldsymbol{m} \cdot \boldsymbol{n}}{\rho} - \boldsymbol{v} \cdot \boldsymbol{n} \right) \mathsf{I}^4 - \operatorname{diag}(c, 0, 0, -c)^{\mathrm{T}},$$

which correspond to the eigenvalues of the equations in the Eulerian framework plus a correction term proportional to $\boldsymbol{v} \cdot \boldsymbol{n}$. Since the last term of Eq. (2.9) does not depend on u, the matrices of the eigenvectors $\mathsf{L}(\mathsf{u}, \boldsymbol{n})$ and $\mathsf{R}(\mathsf{u}, \boldsymbol{n})$ do not depend on the velocity of the control volume and correspond to the ones computed in the Eulerian framework. For a more detailed explanation the reader is referred to [113, 76].

2.3 Space discretization

The discrete representation of the ALE Euler equation (2.7) over a dynamic grid with no topology modifications is now obtained by selecting a finite number of non overlapping volumes $\mathcal{C}_i(t) \subset \Omega(t)$, with boundary $\partial\mathcal{C}_i(t)$, such that $\bigcup_i \mathcal{C}_i(t) \equiv \Omega(t)$. According to the node-centered finite volume approach considered here, each finite volume $\mathcal{C}_i$



(a) Nodepair discretization (b) Boundary discretization

Fig. 2.1: Nodepair discretization of the control volumes. The shaded regions are the finite volumes $\mathcal{C}_i$ and $\mathcal{C}_k$; dashed lines indicate the underlying triangulation. (a) Edge i-k associated with the finite volume interface $\partial\mathcal{C}_{ik} = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$ and metric vector $\boldsymbol{\eta}_{ik}$ (integrated normal) in two spatial dimensions. (b) Discretization along the boundary interface $\partial\mathcal{C}_i \cap \partial\Omega$ and metric vector $\boldsymbol{\xi}_i$ (integrated normal) in two spatial dimensions.

surrounds a single node i of the triangulation of Ω, as shown in fig. 2.1. For each finite volume, equation (2.7) reads

$$\frac{d[V_i\,\mathsf{u}_i]}{dt} + \oint_{\partial\mathcal{C}_i(t)} \big[\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}\big]\cdot\boldsymbol{n}_i\,ds = 0, \quad \forall i \in \mathcal{K}, \tag{2.10}$$

where

$$V_i(t) = \int_{\mathcal{C}_i(t)} d\boldsymbol{x} \qquad and \qquad \mathsf{u}_i(t) = \frac{1}{V_i(t)}\int_{\mathcal{C}_i(t)} \mathsf{u}(\boldsymbol{x},t)\,d\boldsymbol{x},$$

$\mathcal{K}$ is the set of all nodes of the triangulation and $\boldsymbol{n}_i = \boldsymbol{n}_i(s,t)$ denotes the outward normal with respect to the volume $\mathcal{C}_i$, see fig. 2.1. The set of all the nodes located along the boundary of the mesh is termed as $\mathcal{K}_\partial$. In the following expressions the variables of integration $d\boldsymbol{x}$ and ds are not indicated for brevity. The second term of (2.10) is now rearranged to put into evidence the internal and boundary contributions, namely,

$$\frac{d[V_i\,\mathsf{u}_i]}{dt} + \sum_{k\in\mathcal{K}_{i,\neq}} \int_{\partial\mathcal{C}_{ik}(t)} \big[\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}\big]\cdot\boldsymbol{n}_i + \int_{\partial\mathcal{C}_i(t)\cap\partial\Omega(t)} \big[\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}\big]\cdot\boldsymbol{n}_i = 0, \tag{2.11}$$

where $\mathcal{K}_{i,\neq} = \{k \in \mathcal{K}, k \neq i | \partial\mathcal{C}_i \cap \partial\mathcal{C}_k \neq \emptyset\}$ is the set of the indexes k of the finite volumes $\mathcal{C}_k$ sharing a portion of their boundary with $\mathcal{C}_i$, $\mathcal{C}_i$ excluded. The set $\partial\mathcal{C}_{ik}(t) = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$ is often referred to as the cell interface between the volumes $\mathcal{C}_i(t)$ and $\mathcal{C}_k(t)$, see fig. 2.1(a). Each interface $\partial\mathcal{C}_{ik}$ is associated to the corresponding edge i-k connecting nodes i and k of the triangulation of Ω. The second term of Eq. (2.11) is the contribution of the boundary fluxes to the conservation equation of the i-th finite volume, which is different from zero provided that $\partial\mathcal{C}_i \cap \partial\Omega \neq \emptyset$, namely, node i is a boundary node, see fig. 2.1(b).

A suitable integrated numerical flux, i.e. $\Phi^{\mathrm{II}}_{ik} \in \mathbb{R}^4$, approximating the flux across the cell interface $\partial\mathcal{C}_{ik}$, is now introduced [113]. Considering for example a centered approximation [113] of the unknown and of the flux function at the cell interfaces, the domain contribution reads

$$\begin{aligned}\int_{\partial\mathcal{C}_{ik}(t)} [\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}] \cdot \boldsymbol{n}_i &\simeq \frac{\mathbf{f}(\mathsf{u}_i) + \mathbf{f}(\mathsf{u}_k)}{2} \cdot \boldsymbol{\eta}_{ik}(t) - \frac{\mathsf{u}_i + \mathsf{u}_k}{2}\,\nu_{ik} \\ &= -\Phi^{\mathrm{II}}_{ik}(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}),\end{aligned} \tag{2.12}$$

where $\boldsymbol{\eta}_{ik}$ and ν_{ik} are the integrated outward normal and interface velocity, respectively, defined as follows

$$\boldsymbol{\eta}_{ik} = \int_{\partial\mathcal{C}_{ik}(t)} \boldsymbol{n}_i \qquad \text{and} \qquad \nu_{ik} = \int_{\partial\mathcal{C}_{ik}(t)} \boldsymbol{v} \cdot \boldsymbol{n}_i. \tag{2.13}$$

and where $\eta_{ik} = |\boldsymbol{\eta}_{ik}|$ and $\hat{\boldsymbol{\eta}}_{ik} = \boldsymbol{\eta}_{ik}/\eta_{ik}$. The conservativity of the scheme [113] imposes that $\boldsymbol{\eta}_{ik} = -\boldsymbol{\eta}_{ki}$ and $\nu_{ik} = -\nu_{ki}$. Thanks to the piecewise constant representation of the unknown in the finite volume framework, u is constant along the boundary $\partial\mathcal{C}_i \cap \Omega$ and the corresponding flux contribution of Eq. (2.11) simplifies to

$$\begin{aligned}\int_{\partial\mathcal{C}_i(t)\cap\partial\Omega(t)} [\mathbf{f}(\mathsf{u}) - \mathsf{u}\,\boldsymbol{v}] \cdot \boldsymbol{n}_i &\simeq \mathbf{f}(\mathsf{u}_i) \cdot \boldsymbol{\xi}_i - \mathsf{u}_i\,\nu_i \\ &= -\Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i),\end{aligned} \tag{2.14}$$

where $\boldsymbol{\xi}_i$ and ν_i are the integrated outward normal and interface velocity of the i-th boundary node, respectively, defined as

$$\boldsymbol{\xi}_i(t) = \int_{\partial\mathcal{C}_i(t)\cap\partial\Omega(t)} \boldsymbol{n}_i(s,t) \qquad \text{and} \qquad \nu_i(t) = \int_{\partial\mathcal{C}_i(t)\cap\partial\Omega(t)} \boldsymbol{v}(s,t) \cdot \boldsymbol{n}_i(s,t)\,, \tag{2.15}$$

and where $\xi_i = |\boldsymbol{\xi}_i|$ and $\hat{\boldsymbol{\xi}}_i = \boldsymbol{\xi}_i/\xi_i$. Moreover, for the finite volume to be closed, the following relation holds

$$\boldsymbol{\xi}_i(t) + \sum_{k\in\mathcal{K}_{i,\neq}} \boldsymbol{\eta}_{ik}(t) = \mathbf{0}.$$

For a general, namely, non-centered approximation of the numerical fluxes a general expression of Eq. (2.10) can be formally rewritten as, i.e.

$$\frac{d}{dt}\,[V_i\,\mathsf{u}_i] = \sum_{k\in\mathcal{K}_{i,\neq}} \Phi^{\mathrm{II}}(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}) + \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i). \tag{2.16}$$

The calculation of the domain and boundary fluxes will be addressed in section 2.3.1 and 2.3.2 respectively.

The same finite volume approach can be adopted to compute the interface integrals appearing in the Geometric Conservation Law, thus Eq. (2.8), i.e.

$$\frac{d}{dt}\int_{\mathcal{C}_i(t)} = \int_{\partial\mathcal{C}_i(t)\cap\partial\Omega(t)} \boldsymbol{v}\cdot\boldsymbol{n}_i \;+ \sum_{k\in\mathcal{K}_{i,\neq}} \int_{\partial\mathcal{C}_{ik}(t)} \boldsymbol{v}\cdot\boldsymbol{n}_i\;,$$

becomes

$$\frac{dV_i(t)}{dt} = \nu_i(t) + \sum_{k \in \mathcal{K}_{i,\neq}} \nu_{ik}(t). \tag{2.17}$$

A natural way to satisfy Eq. (2.17) is to split the derivative of the cell volume into contribution pertaining to the so called Interface Volumes, i.e. the volumes (areas in two dimensions) swept by the different parts of the interfaces, which satisfy the following

$$\nu_{ik} = \frac{dV_{ik}}{dt} \qquad \text{and} \qquad \nu_i = \frac{dV_{i,\partial}}{dt}, \tag{2.18}$$

where $V_{ik}(t)$ and $V_{i,\partial}(t)$ are the volumes swept by the interfaces $\partial\mathcal{C}_{ik}(t)$ and $\partial\mathcal{C}_i(t) \cap \partial\Omega(t)$ respectively. Such definition of the integrated normal velocities allows to automatically satisfy the GCL constraint and is therefore the most natural choice [132, 166], however it is not unique. Indeed, there is a $\dim \mathcal{K}_{i,\neq} - 1$ parameter family of interfaces velocities satisfying condition (2.17), where $\dim$ indicates the number of elements of a given set. Additionally, it can be observed that one can think of a deforming control volume as a set of points moving with a given velocity field, which is defined by the imposed grid velocity. As said, the GCL is satisfied by an infinite combination of interface velocities, but only one of them is the one that is computed as the result of subsequent geometry configurations that includes at every time the same (material) points.

The differential relations of Eq. (2.18), which for an assigned mesh motion allows one to compute the interface velocities $\nu_{ik}(t)$ and $\nu_i(t)$, complements the ODE system of the Euler equations (2.16), namely

$$\begin{cases} \dfrac{dV_i \mathsf{u}_i}{dt} = \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i) + \displaystyle\sum_{k \in \mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}), & i \in \mathcal{K} \\ \dfrac{dV_{ik}}{dt} = \nu_{ik}, & k \in \mathcal{K}_{i,\neq} \\ \dfrac{dV_{j,\partial}}{dt} = \nu_j, & j \in \mathcal{K}_{\partial} \end{cases} \tag{2.19}$$

In the system (2.19) the unknowns are $4N_V$, with $N_V = \dim \mathcal{K}$, the boundary conditions are applied to $N_{V,\partial} = \dim \mathcal{K}_{\partial}$ boundary nodes, the numerical fluxes are evaluated over each of the N_E edges, with $N_E = \frac{1}{2}\sum_{i \in \mathcal{K}} \dim \mathcal{K}_{i,\neq}$, and $N_E + N_{V,\partial}$ interface velocity consistency conditions are defined.

2.3.1 Domain Fluxes

The evaluation of numerical fluxes of Eq. (2.16) requires particular care, due to the hyperbolic nature of the equations and, possibly, due to the need to achieve second-order accuracy in space.

The domain fluxes are discretized resorting to an approximate Riemann solver, which gives the flux that would be obtained by solving the well-known Riemann problem of gas dynamics, i.e. evolution of the solution from a discontinuous initial state. Riemann based solvers are well suitable to describe wave propagations but lead to first order spatially accurate schemes [113]. A high-resolution Total Variation Diminishing expression for the integrated numerical flux is used in the to compute flows with shock waves [113].

Centered Approximation

The integrated numerical flux can be calculated using the mean of the "natural" fluxes evaluated on the grid nodes, i.e.

$$\Phi_{ik}^{\mathrm{II}} = \frac{\mathbf{f}(\mathsf{u}_i) + \mathbf{f}(\mathsf{u}_k)}{2} \cdot \boldsymbol{\eta}_{ik} - \frac{\mathsf{u}_i + \mathsf{u}_k}{2} \nu_{ik}. \tag{2.20}$$

This centered approximation is found to be second-order accurate [158].

Roe Solver

It is well known, see e.g. [113], that the use of the second-order integrated numerical flux of Eq. (2.20) may lead to the appearance of spurious oscillations in advection dominated flows and in particular near discontinuities of the flow variables; in this case, the following first-order upwind approximation due to Roe [158], namely

$$\Phi_{ik}^{\mathrm{I}} = \frac{\mathbf{f}(\mathsf{u}_i) + \mathbf{f}(\mathsf{u}_k)}{2} \cdot \boldsymbol{\eta}_{ik} - \frac{\mathsf{u}_i + \mathsf{u}_k}{2} \nu_{ik} - \frac{1}{2} |\widetilde{\mathsf{A}}| \, (\mathsf{u}_k - \mathsf{u}_i), \tag{2.21}$$

is to be preferred. The Roe matrix, i.e. $\widetilde{\mathsf{A}} \in \mathbb{R}^4 \times \mathbb{R}^4$, is the Jacobian of the flux vector projected along the normal vector $\hat{\boldsymbol{\eta}}_{ik}$ evaluated at the Roe state $\widetilde{\mathsf{u}} = \widetilde{\mathsf{u}}(\mathsf{u}_i, \mathsf{u}_k)$, namely $\widetilde{\mathsf{A}} = \mathsf{A}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}, \nu_{ik})$. The Roe state can be computed as

$$\widetilde{\rho} = \sqrt{\rho_i \rho_k}, \qquad \widetilde{\boldsymbol{m}} = \frac{\boldsymbol{m}_i \sqrt{\rho_k} + \boldsymbol{m}_k \sqrt{\rho_i}}{\sqrt{\rho_i} + \sqrt{\rho_k}} \qquad \text{and} \qquad \widetilde{h} = \frac{h_i \sqrt{\rho_i} + h_k \sqrt{\rho_k}}{\sqrt{\rho_i} + \sqrt{\rho_k}},$$

where $h = (E^{\mathrm{t}} + P)/\rho$ is the specific total enthalpy.

The absolute value of the Jacobian matrix $|\mathsf{A}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}, \nu_{ik})|$ is computed resorting to its spectral decomposition, namely

$$|\widetilde{\mathsf{A}}| = \widetilde{\mathsf{R}} |\widetilde{\Lambda}| \widetilde{\mathsf{L}},$$

where $\widetilde{\mathsf{R}} = \mathsf{R}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{ik})$ and $\widetilde{\mathsf{L}} = \mathsf{L}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{ik})$ are matrices of the right and left eigenvectors of A, introduced in section 2.2.1, evaluated in the Roe state, respectively. $\widetilde{\Lambda} = \widetilde{\Lambda}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}, \nu_{ik})$ is the diagonal matrix built with the absolute values of the eigenvalues, namely $|\widetilde{\Lambda}| = \mathrm{diag}\,(|\widetilde{\lambda}_{(1)}|, \ldots, |\widetilde{\lambda}_{(4)}|)$, where

$$\widetilde{\lambda}_{(1)} = \eta_{ik} \left(\frac{\widetilde{\boldsymbol{m}} \cdot \hat{\boldsymbol{\eta}}_{ik}}{\widetilde{\rho}} - c(\widetilde{\mathsf{u}}) \right) - \nu_{ik},$$

$$\widetilde{\lambda}_{(2)} = \widetilde{\lambda}_{(3)} = \frac{\widetilde{\boldsymbol{m}} \cdot \boldsymbol{\eta}_{ik}}{\widetilde{\rho}} - \nu_{ik},$$

$$\widetilde{\lambda}_{(4)} = \eta_{ik} \left(\frac{\widetilde{\boldsymbol{m}} \cdot \hat{\boldsymbol{\eta}}_{ik}}{\widetilde{\rho}} + c(\widetilde{\mathsf{u}}) \right) - \nu_{ik},$$

are the eigenvalues integrated along the interface $\partial \mathcal{C}_{ik}(t)$.

The Roe flux is essentially a central term plus a matrix dissipation term, see Eq.(2.21). If the introduced numerical dissipation is too small, i.e. if the eigenvalues are close to zero, non-physical solutions may appear. In order to avoid such behavior the a correction,

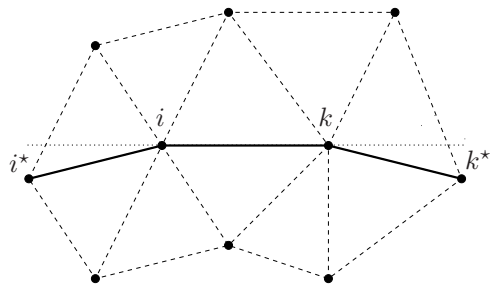


Fig. 2.2: Extended edge structure $i^\star$-i-k-$k^\star$ for high-resolution.

usually termed as entropy fix, to the eigenvalues is performed in order to ensure that the introduced dissipation is not below a given threshold [113].

In the present work a modification of the entropy fix first proposed by Selmin is adopted [162] and the eigenvalues used to compute the Roe matrix are replaced by

$$\widehat{\lambda}_h = \begin{cases} |\widetilde{\lambda}_{(h)}| & \text{if} \quad \widetilde{\lambda}_{(h)} > \widetilde{\delta}, \\ \dfrac{\widetilde{\lambda}^2_{(h)} + \widetilde{\delta}^2}{2\widetilde{\delta} + \varepsilon} & \text{if} \quad \widetilde{\lambda}_{(h)} < \widetilde{\delta}, \end{cases} \tag{2.22}$$

where $\varepsilon = 10^{-12}$ is a small positive parameter chosen in order to avoid divisions by zero of a null quantity, $\widetilde{\delta} = 0.2(\mathrm{M}(\widetilde{\mathsf{u}}_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}, \nu_{ik}) + 1)\eta_{ik}$ is adopted as a threshold and

$$\mathrm{M}(\mathsf{u}, \hat{\boldsymbol{\eta}}, \eta, \nu) = \frac{1}{c(\widetilde{\mathsf{u}})} \left(\frac{\widetilde{\boldsymbol{m}} \cdot \hat{\boldsymbol{\eta}}}{\widetilde{\rho}} - \frac{\nu}{\eta} \right) \tag{2.23}$$

is the relative Mach number projected along the $\hat{\boldsymbol{\eta}}_{ik}$ direction.

Flux Limiter

Following [186], a high-resolution expression for the integrated numerical flux is now obtained by resorting to the Total Variation Diminishing approach, in which the second order approximation Φ^{II}_{ik} is replaced by its first order counterpart Φ^{I}_{ik} near flow discontinuities; the switch is controlled by a suitable flux limiter Υ, $\Upsilon = \mathrm{diag}\,(\Upsilon_1, \ldots, \Upsilon_5)$. The resulting high-resolution integrated numerical flux reads

$$\Phi^{\mathrm{HR}}_{ik} = \Phi^{\mathrm{I}}_{ik} + \Upsilon \left[\Phi^{\mathrm{II}}_{ik} - \Phi^{\mathrm{I}}_{ik} \right] = \Phi^{\mathrm{II}}_{ik} + \frac{1}{2} \widetilde{\mathsf{R}} |\widetilde{\Lambda}| \, (\widetilde{\mathsf{w}} - \widetilde{\mathsf{v}}), \tag{2.24}$$

where $\widetilde{\mathsf{v}} = \widetilde{\mathsf{L}} \, (\mathsf{u}_k - \mathsf{u}_i)$ and $\widetilde{\mathsf{w}} = \Upsilon \, \widetilde{\mathsf{L}} \, (\mathsf{u}_k - \mathsf{u}_i)$. By substituting the expression of the limiter by Van Leer [186], the h-th component of the vector of the limited characteristic jumps $\widetilde{\mathsf{w}}$ reads

$$\widetilde{\mathsf{w}}_{(h)} = \frac{\widetilde{\mathsf{v}}_{(h)} |\widetilde{\mathsf{q}}_{(h)}| + |\widetilde{\mathsf{v}}_{(h)}| \widetilde{\mathsf{q}}_{(h)}}{|\widetilde{\mathsf{v}}_{(h)}| + |\widetilde{\mathsf{q}}_{(h)}| + \epsilon}, \tag{2.25}$$

where ϵ is a small positive parameter introduced here to avoid division by zero of a zero quantity ($\epsilon = 10^{-12}$ here) and where the h-th component of the vector of "upwind"

jumps $\widetilde{\mathsf{q}}$ is given by

$$\widetilde{\mathsf{q}}_{(h)} = \begin{cases} \dfrac{\hat{\boldsymbol{\eta}}_{ik} \cdot (\boldsymbol{x}_k - \boldsymbol{x}_i)}{\hat{\boldsymbol{\eta}}_{ik} \cdot (\boldsymbol{x}_{k^\star} - \boldsymbol{x}_k)} \widetilde{\mathsf{L}}_{(h)} (\mathsf{u}_{k^\star} - \mathsf{u}_k) & \text{if} \quad \widetilde{\lambda}_{(h)} > 0, \\ \dfrac{\hat{\boldsymbol{\eta}}_{ik} \cdot (\boldsymbol{x}_k - \boldsymbol{x}_i)}{\hat{\boldsymbol{\eta}}_{ik} \cdot (\boldsymbol{x}_i - \boldsymbol{x}_{i^\star})} \widetilde{\mathsf{L}}_{(h)} (\mathsf{u}_i - \mathsf{u}_{i^\star}) & \text{if} \quad \widetilde{\lambda}_{(h)} \leq 0. \end{cases}$$

In the above definition, the nodes $i^\star$ and $k^\star$ are the extension nodes belonging to edge i-k of the triangulation and $\widetilde{\mathsf{L}}_{(h)}$ is the h-th row of matrix $\widetilde{\mathsf{L}}$. Note the above high-resolution version of the scheme requires the definition of an extended edge data structure that includes also the extension nodes $i^\star$ and $k^\star$, that are needed in the evaluation of the limiter function Υ. Following [194], the extension nodes belong to the two edges best aligned with i-k, see Fig. 2.2.

The computer implementation of the finite volume scheme described above is fairly straightforward and very efficient, see e.g. [163]. All computations are performed only over the edges of the mesh: edges are present in one-, two- and three-dimensional grids and therefore the extension to different spatial dimension requires only few modifications to the code, that are limited to the definition of the vector unknown and to the associated flux function.

2.3.2 Boundary Fluxes

Boundary conditions are enforced in a weak form, i.e. by evaluating the flux Φ_i^∂ of Eq. (2.14) in a suitable boundary state $\mathsf{u}^\partial = \mathsf{u}^\partial(\mathsf{u}_i, \hat{\boldsymbol{\xi}}_i, \xi_i, \nu_i)$. Two different boundary conditions are described: slip and non-reflection condition.

Slip boundary conditions are imposed by subtracting to u_i the component of the velocity normal to the wall in a reference frame that is moving with the body, i.e.

$$\mathsf{u}^{\mathrm{W}}(\mathsf{u}_i, \hat{\boldsymbol{\xi}}_i, \xi_i, \nu_i) = \mathsf{u}_i - \left[0, \left(\boldsymbol{m}_i \cdot \hat{\boldsymbol{\xi}}_i - \frac{\rho_i \, \nu_i}{\xi_i}\right) \hat{\boldsymbol{\xi}}_i, \frac{1}{2} \left| \frac{\boldsymbol{m}_i \cdot \hat{\boldsymbol{\xi}}_i}{\rho_i} - \frac{\nu_i}{\xi_i} \right|^2 \right]^{\mathrm{T}}. \tag{2.26}$$

The boundary flux corresponding to a slip wall condition, therefore is equal to

$$\Phi_i^\partial(\mathsf{u}^{\mathrm{W}}, \hat{\boldsymbol{\xi}}_i, \xi_i, \nu_i) = P(\mathsf{u}_i) \, \left[0, \xi_i \, \hat{\boldsymbol{\xi}}_i, \nu_i\right]^{\mathrm{T}},$$

which correspond to the pressure forces and power. The boundary state u_i^∂ at the far-field is computed via characteristic reconstruction from the nodal state vector u_i and the far-field state u_∞ as follows [78]

$$\mathsf{u}^\infty(\mathsf{u}_i, \boldsymbol{\xi}_i, \nu_i) = \mathsf{u}_i + \mathsf{R}(\mathsf{u}_i, \hat{\boldsymbol{\xi}}_i) \, \mathsf{N}(\mathsf{u}_i, \hat{\boldsymbol{\xi}}_i, \xi_i, \nu_i) \, \mathsf{L}(\mathsf{u}_i, \hat{\boldsymbol{\xi}}_i) \, \left[\mathsf{u}_\infty - \mathsf{u}_i\right], \tag{2.27}$$

where $\mathsf{N} = -\operatorname{diag}\left(\min\left(\lambda_{(h)}/|\lambda_{(h)}|, 0\right)\right)$, $h = 1, \ldots, 4$. Note that the expression above simplifies to $\mathsf{u}_i^\partial = \mathsf{u}_i$ for $\lambda_{(h)} < 0 \ \forall h$ and to $\mathsf{u}_i^\partial = \mathsf{u}_\infty$ for $\lambda_{(h)} > 0 \ \forall h$, i.e. a supersonic outflow and inflow respectively.

2.3.3 Metrics computation

In the present section, the expression for the volume (area in two dimensions) V_i and the metric quantities $\boldsymbol{\eta}_{ik}$ and $\boldsymbol{\xi}_i$, are given in terms of geometrical entities defined over

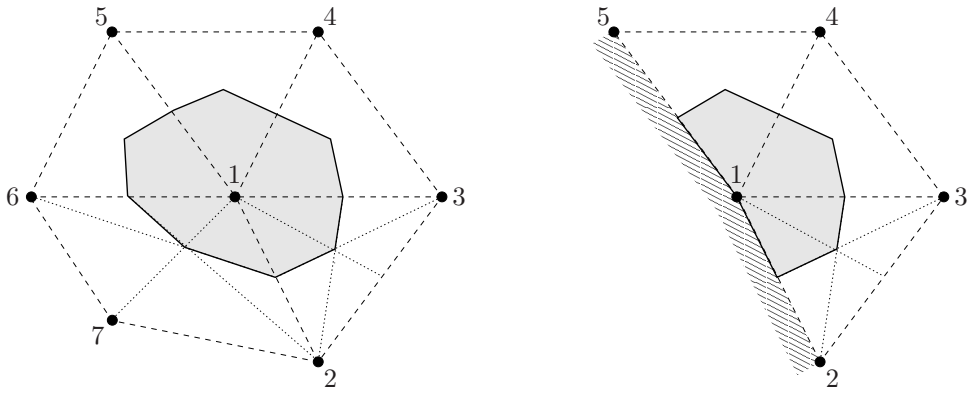


Fig. 2.3: Definition of the finite volume $\mathcal{C}_1$ (shaded region) from the underlying triangulation in two spatial dimensions. The nodes of the triangulation are indicated with the symbol •; the grid elements are indicated by dashed lines. a) Node 1 belongs to the interior of the computational domain Ω. b) Node 1 lies on the boundary of Ω, here represented by the segments 2–1 and 1–5. The medians of the triangle $(1, 2, 3)$ and of the quadrilateral element $(1, 6, 7, 2)$ are also indicated with dotted lines, cf. Fig. 2.3.

the underlying triangulation of the domain Ω. The latter is possibly a so-called hybrid triangulation, namely, it could be made of elements of different types. The computation of the velocities ν_{ik} and ν_i will be discussed in section 2.4

With reference to Fig. 2.3, the finite volume $\mathcal{C}_i$ is first split into the subsets $\mathcal{C}_{i,e} = \mathcal{C}_i \cap \Omega_e$, where Ω_e is the e-th element of the triangulation, so that $\mathcal{C}_i = \bigcup_{e\in\mathcal{E}_i} \mathcal{C}_i \cap \Omega_e$, with $\mathcal{E}_i$ the set of the elements sharing node i. The set $\mathcal{E}_i$ is often referred to as the element bubble of node i. Over each subset $\mathcal{C}_{i,e}$, the boundary $\partial\mathcal{C}_{i,e} = \partial\mathcal{C}_i \cap \Omega_e$ is made of the two segments connecting the center of gravity $\boldsymbol{x}_e$ of the element to the midpoints $\boldsymbol{x}_{ik}$ of the two edges from node i, as shown in Fig. 2.4. The elemental contribution $\boldsymbol{\eta}_{ik,e}$, see Fig. 2.4a, is therefore computed as

$$\boldsymbol{\eta}_{ik,e}(t) = \int_{\partial\mathcal{C}_{ik,e}(t)} \boldsymbol{n}_i = (\boldsymbol{x}_{ik}(t) - \boldsymbol{x}_e(t)) \times \hat{\boldsymbol{z}}, \tag{2.28}$$

where $\hat{\boldsymbol{z}}$ is the unit vector normal to the plane x-y. The metric vector $\boldsymbol{\eta}_{ik}$ is then computed as

$$\boldsymbol{\eta}_{ik}(t) = \sum_{e\in\mathcal{E}_i} \boldsymbol{\eta}_{ik,e}(t). \tag{2.29}$$

Similarly, the boundary metric vector $\boldsymbol{\xi}_i = \sum_{e\in\mathcal{E}_i^\partial} \boldsymbol{\xi}_{i,e}$, where $\mathcal{E}_i^\partial$ is the set of the boundary elements having node i in common, one has

$$\boldsymbol{\xi}_{i,e}(t) = \int_{\partial\mathcal{C}_{i,e}(t)} \boldsymbol{n}_i = (\boldsymbol{x}_e(t) - \boldsymbol{x}_i(t)) \times \hat{\boldsymbol{z}}. \tag{2.30}$$

Note that in two spatial dimensions the boundary elements are the edges of the domain element themselves, so that the center of gravity of the e-th boundary element is the

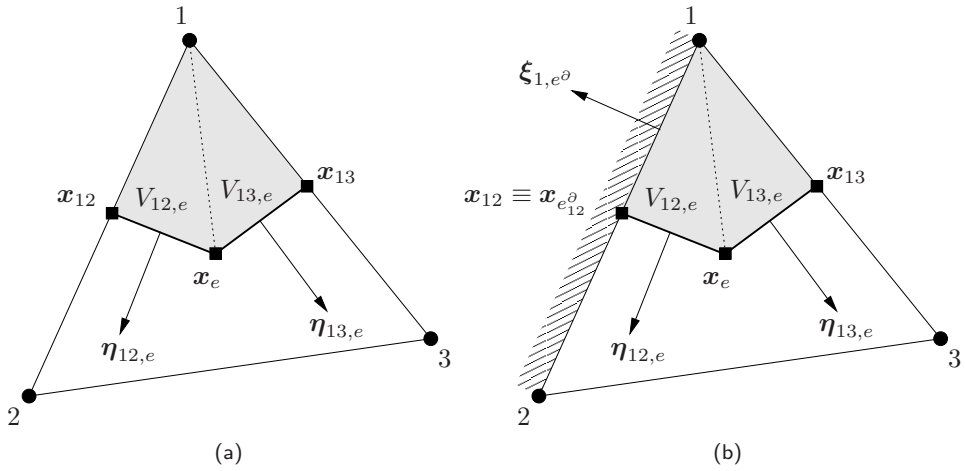


Fig. 2.4: Definition of the finite volume from the underlying triangulation in two spatial dimensions. The nodes of the triangulation are indicated with the symbol •, the barycenter $\boldsymbol{x}_{ik}$ and $\boldsymbol{x}_e$ of the edges and of the element, respectively, are indicated with ■. a) The portion of the perimeter of the finite volume $\mathcal{C}_1$ pertaining to the triangular element e with nodes $(1, 2, 3)$ is made of the two segments $(\boldsymbol{x}_{12}, \boldsymbol{x}_e)$ and $(\boldsymbol{x}_e, \boldsymbol{x}_{13})$. The contribution to the area of the finite volume is given by the sum of the area of the two triangles $(\boldsymbol{x}_1, \boldsymbol{x}_{12}, \boldsymbol{x}_e)$ and $(\boldsymbol{x}_1, \boldsymbol{x}_{13}, \boldsymbol{x}_e)$. b) Elemental contributions for triangular element lying on the domain boundary.

midpoint of the segment of the e-th element which lies on the boundary, as shown in Fig. 2.4b. The contribution $V_{i,e}$ of element e to the volume of $\mathcal{C}_i$ (area in the present two-dimensional case) is given by sum of the area of the two triangles having as vertices node i, $\boldsymbol{x}_e$ and each of the two midpoints $\boldsymbol{x}_{ik}$, see Fig. 2.4. In terms of edge contributions it the cell volume is computed as

$$V_i(t) = \sum_{e \in \mathcal{E}_i} V_{i,e}(t) = \sum_{e \in \mathcal{E}_i} \sum_{k \in \mathcal{K}_{i,\neq}} V_{ik,e}(t) \ ,$$

where

$$V_{ik,e}(t) = \frac{\boldsymbol{x}_e(t) - \boldsymbol{x}_i(t)}{2} \cdot \boldsymbol{\eta}_{ik,e}(t) \, . \tag{2.31}$$

All the definitions given above are applicable to elements of any kind, e.g. to hybrid triangulations of triangles and quadrilaterals, and guarantee that each finite volume is closed. It is worth noticing that, for fixed grids, Selmin [163] and Selmin and Formaggia [164] proved that a centered finite volume discretization built according to the above prescriptions is equivalent to a linear finite element approach in the case of two-dimensional grids of triangles or three-dimensional grids of tetrahedra, but for the boundary terms.

2.4 Time integration

The time discretization of the finite volume discretization of the ALE Euler equations, i.e. Eq. (2.19), here introduced. A generic Backward Differences Formulæ scheme [106] is adopted to approximate the derivatives with respect to the time, e.g. $dy/dt \simeq \Delta t^{-1} \sum_{q=-1}^{p} a_q y^{n-q}$ is an approximation of order $p{+}1$ of accuracy and a_q are the $p{+}1$ coefficients of the BDF [106]. The same scheme can be recasted in an incremental fashion, i.e. $dy/dt \simeq \Delta t^{-1} \sum_{q=-1}^{p-1} \alpha_q \Delta y^{n-q}$, with $\Delta y^n = y^n - y^{n-1}$ and $\alpha_q = \sum_{d=-1}^{q} a_d$. Eq. (2.19) is therefore rearranged as

$$\begin{cases} \displaystyle\sum_{q=-1}^{p} a_q V_i^{n-q} \mathsf{u}_i^{n-q} = \Big[\sum_{k \in \mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik})^{n+1} \\ \qquad\qquad + \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i)^{n+1} \Big] \Delta t, & i \in \mathcal{K} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{ik}^{n-q} = \nu_{ik}^{n+1} \Delta t, & k \in \mathcal{K}_{i,\neq} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{\ell,\partial}^{n-q} = \nu_{\ell}^{n+1} \Delta t, & \ell \in \mathcal{K}_{\partial} \end{cases} \tag{2.32}$$

where $\Phi(\,\cdot\,,\,\cdot\,)^{n+1} = \Phi(\,\cdot^{\,n+1},\,\cdot^{\,n+1})$. In system (2.32) all quantities are assumed to be known at time level n and all grid-dependent quantities, such as V_i, $\hat{\boldsymbol{\eta}}_{ik}$, η_{ik}, $\hat{\boldsymbol{\xi}}_i$ and ξ_i, are computed from the known positions of the grid nodes at time level $n+1$. The nonlinear system for the fluid variables u at time level $n+1$ is solved here by means of a modified Newton method, in which the Jacobian of the integrated flux function is approximated by that of the first-order scheme, and by resorting to a dual time-stepping technique [190], to improve the conditioning number of the Jacobian matrix.

The Geometric Conservation Law (2.17) is made discrete using the same BDF scheme adopted to discretize the governing equation, i.e.

$$\sum_{q=-1}^{p-1} \alpha_q \frac{\Delta V_i^{n-q}}{\Delta t} = \nu_i^{n+1} + \sum_{k \in \mathcal{K}_{i,\neq}} \nu_{ik}^{n+1}. \tag{2.33}$$

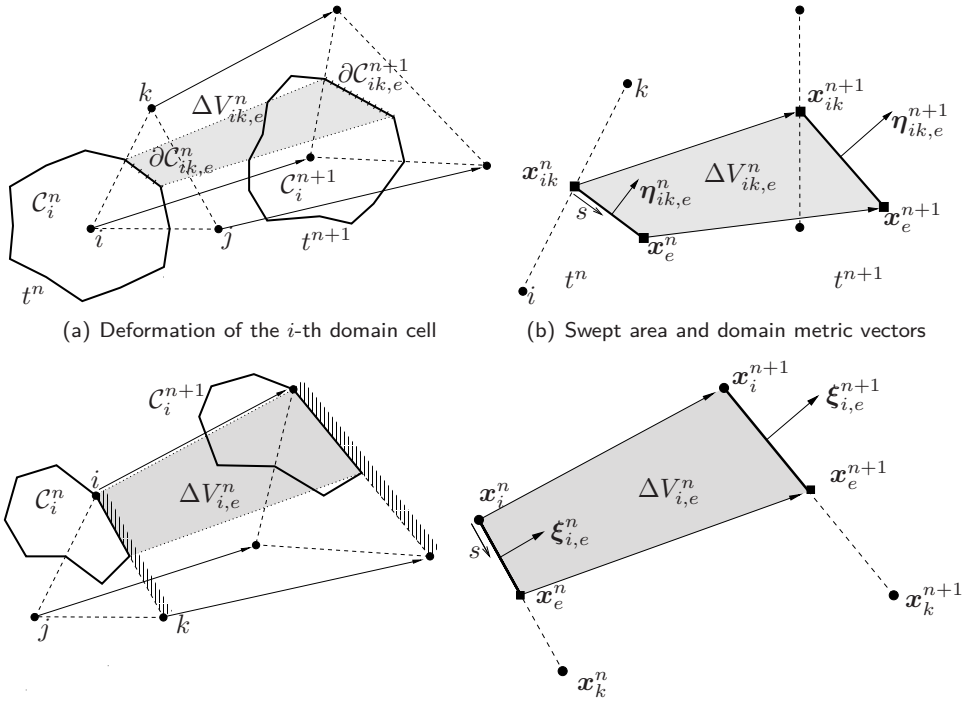
Condition (2.33) is usually referred to as the Discrete Geometric Conservation Law (DGCL) [54].

2.4.1 Interface Velocity Consistency Condition

The GCL compliant interface velocity ν_{ik}^{n+1} is computed using Eq. (2.32) from the values of ΔV_{ik}^{n+1} that represent the area swept by the interface $\partial \mathcal{C}_{ik}(t)$ during the time step from t^n to t^{n+1}, namely

$$\Delta V_{ik}^{n+1} = \int_{t^n}^{t^{n+1}} \int_{\partial \mathcal{C}_{ik}(t)} \boldsymbol{v}(t) \cdot \boldsymbol{n}(t).$$

By noticing that the interface $\partial \mathcal{C}_{ik}$, is the union of straight sides $\partial \mathcal{C}_{ik,e}$, see fig. 2.5(a), and assuming a linear dependence of the node position on time in $[t^n, t^{n+1})$, a simple



(a) Deformation of the i-th domain cell

(b) Swept area and domain metric vectors

(c) Deformation of the i-th boundary cell

(d) Swept area and boundary metrics

Fig. 2.5: Area swept by the domain and boundary interface $\partial\mathcal{C}_{ik,e}$ and $\partial\mathcal{C}_{i,e}$, respectively, during the interval $[t^n, t^{n+1})$

computation gives

$$\Delta V^{n+1}_{ik} = \sum_{e\in\mathcal{E}_i\cap\mathcal{E}_k} \frac{\left(\boldsymbol{x}^{n+1}_e - \boldsymbol{x}^{n+1}_{ik}\right) - \left(\boldsymbol{x}^{n}_e - \boldsymbol{x}^{n}_{ik}\right)}{4} \cdot \left(\boldsymbol{\eta}^{n+1}_{ik,e} + \boldsymbol{\eta}^{n}_{ik,e}\right),$$

where $\boldsymbol{x}_e$ and $\boldsymbol{x}_{ik}$ are the barycenter of the element e and of the edge i-k in fig. 2.5(b), respectively, and where $\boldsymbol{\eta}_{ik,e}$ is the contribution to $\boldsymbol{\eta}_{ik}$ pertaining to element e.

Similarly, see fig. 2.5(c), the integrated velocities of the boundary interfaces can be computed from the values of $\Delta V^{n+1}_{i,\partial}$, namely

$$\Delta V^{n+1}_{i,\partial} = \sum_{e\in\mathcal{E}^\partial_i\cap\partial\Omega} \frac{\left(\boldsymbol{x}^{n+1}_i - \boldsymbol{x}^{n+1}_e\right) - \left(\boldsymbol{x}^{n}_i - \boldsymbol{x}^{n}_e\right)}{4} \cdot \left(\boldsymbol{\xi}^{n+1}_{i,e} + \boldsymbol{\xi}^{n}_{i,e}\right),$$

where $\boldsymbol{x}_e$ is the center of mass of the boundary element e in fig. 2.5(d), respectively, and where $\boldsymbol{\eta}_{ik,e}$ is the contribution to $\boldsymbol{\eta}_{ik}$ pertaining to element e.

2.4.2 Iterative Implicit Solver

The governing equation for the steady state are solved adopting an implicit pseudo-time stepping scheme.

Due to the non-linear nature of the equations an iterative scheme is adopted which require the solution of the linear system. The matrix so generated by the finite-volume discretization of the equations is poorly diagonally dominant, this is caused by the elliptic-hyperbolic nature of the equations and the second-order space accuracy and poses several problems in solving the linear system. To improve the diagonal dominance of the system matrix a defect-correction method is used [101], i.e. the exact Jacobian of the numerical fluxes is with an approximated one, and a fictitious time step is introduced [190]. When the convergence is achieved, or when large time steps are used, the scheme is indeed an inexact Newton method with a linear convergence rate.

In section 2.4.3 the linear system solver is introduced. An iterative a symmetric Gauss-Seidel method [36] is used to produce the solution at each pseudo time step. The SGS method has been preferred over more standard iterative or direct sparse scheme because it increases the overall efficiency of the method by balancing the accuracy of the solution of the linear system with the accuracy of the solution of the non linear equations.

Without loosing generality, system (2.32) can be recasted as

$$\frac{d\mathbf{V}\mathbf{u}}{dt} = -\mathbf{q} \tag{2.34}$$

where $\mathbf{u} = [\mathsf{u}_1, \mathsf{u}_2, \ldots, \mathsf{u}_{N_V}]^{\mathrm{T}}$ is the solution vector, $\mathbf{V} = \mathrm{diag}\,(V_1 \mathsf{I}^4, \ldots, V_{N_V} \mathsf{I}^4)$ is the diagonal matrix built from the cells area and $\mathbf{q} = [\mathsf{q}_1, \mathsf{q}_2, \ldots, \mathsf{q}_{N_V}]^{\mathrm{T}}$ is the residual vector. The lower-case bold notation is used in the following sections to indicate column vectors of size $4N_V$, while upper case bold letters for matrices of size $4N_V \times 4N_V$. The i-th row of $\mathbf{q}$ is computed as the right hand side of the first equation of system (2.32), i.e.

$$\mathsf{q}_i = -\Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i)^{n+1} - \sum_{k \in \mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik})^{n+1}.$$

In the steady case system (2.32) can be reduced to $\mathbf{q} = 0$, that ideally could be resolved simply adopting a Newton-Raphson scheme. However, due to the non-linearity of the equations, such iterative methods generally fail to converge unless the initial guess is sufficiently close to the final solution. It is common practice to include a the pseudo-time derivative in the flow-equations and use the free-stream as a first guess. The solution that satisfies $\mathbf{q}(\mathbf{u}) = 0$ is therefore obtained as the steady state marching in the pseudo-time step, i.e.

$$\mathbf{V}\frac{\mathbf{u}^{m+1} - \mathbf{u}^m}{\Delta\tau} = -\mathbf{q}(\mathbf{u}^{m+1}). \tag{2.35}$$

The right hand side of Eq. (2.35) is then approximated by a first-order Taylor series expansion, i.e.

$$\left[\frac{\mathbf{V}}{\Delta\tau} + \frac{\partial\tilde{\mathbf{q}}}{\partial\mathbf{u}}\right]^m (\mathbf{u}^{m+1} - \mathbf{u}^m) = -\mathbf{q}^m. \tag{2.36}$$

where $\mathbf{q}^m = \mathbf{q}(\mathbf{u}^m)$. As common, in Eq. (2.36) the exact Jacobian of $\mathbf{q}$ has been replaced with an approximated on, i.e. $\tilde{\mathbf{q}}$, to increase the diagonally dominance of the linear system [101]. The convergence of the scheme is therefore more robust but unfortunately the convergence rate of an inexact Newton method is not quadratic, even when $\Delta\tau$ goes to infinity [12, 126, 189].

Local time stepping

The stability condition for linear hyperbolic equations states that the Courant-Friedrichs-Lewy number must not be greater than one, i.e. $\mathrm{Co} \leq 1$ [113]. In the present work a local time-stepping technique is obtained to choose the local pseudo-time step such that the CFL condition is locally satisfied at every step, i.e.

$$\Delta\tau_i = \frac{V_i \,\mathrm{Co}}{\lambda_{max}(\mathsf{u}_i, \boldsymbol{\xi}_i, \nu_i) + \sum_{k \in \mathcal{K}_{i,\neq}} \lambda_{max}(\mathsf{u}_i, \mathsf{u}_k, \boldsymbol{\eta}_{ik}, \nu_{ik})},$$

where λ_{max} is the largest of eigenvalues taken as positive [71]. As the solution $\mathbf{u}^m$ approaches the exact one, the CFL condition can be relaxed and the Courant number can be chosen as greater than one. To increase the convergence speed, the Co number is updated as

$$\mathrm{Co}^m = \min\left(\max\left[\gamma \frac{L_2(\mathbf{q}^{m-2})}{L_2(\mathbf{q}^{m-1})}, 1\right] \mathrm{Co}^{m-1}, \mathrm{Co}^{\max}\right),$$

where L_2 is a discrete norm of the residual, $\gamma = 0.8 \div 1$ is the increase ratio and $\mathrm{Co}^{\max}$ is the maximum allowed Courant number.

In steady state computations, to increase the robustness of the method, it is common practice to first obtain a first-order accurate solution which is then used as a starting guess to obtain a quasi second-order one. In the first-order computation the Roe fluxes are adopted and, due to the highly diffusive nature of the scheme, virtually no upper bound to the Co is required. The quasi second-order solution $\mathbf{u}$ is computed resorting to the flux-limiter approach and the maximum value of Co is chosen in the range $1 \div 10^3$.

2.4.3 Iterative solution of the linear system of equations

At each pseudo time-step τ^m, Eq. (2.36) implies the solution of a linear system $\mathbf{M}\mathbf{z} = \mathbf{q}^m$ where $\mathbf{M}$ is the linear system matrix of dimension $4N_V \times 4N_V$, i.e.

$$\mathbf{M} = \left[\frac{\mathbf{V}}{\Delta\tau} + \frac{\partial\tilde{\mathbf{q}}}{\partial\mathbf{u}}\right]^m.$$

The solution is therefore updated as $\mathbf{u}^{m+1} = \mathbf{u}^m + \mathbf{z}$. Due to the finite volume discretization of the governing equation, the matrix $\mathbf{M}$ is sparse and the solution of the linear system can become prohibitive as the number of nodes grows. Due to the non-linear nature of Eq. (2.36), however, both the system matrix and the residual $\mathbf{q}^m$ changes at every pseudo-time step, i.e. at every non-linear iteration. As it is common practice, in the present work Eq. (2.36) is solved using an iterative scheme without seeking full convergence and performing only a "suitable" number of so called linear iterations [189, 190, 202].

The exact solution of Eq. (2.36) is replaced with an approximated one, i.e. $\mathbf{z}^k$, that can be computed with the following three-step procedure

$$\mathbf{q}_{\text{lin}} = \mathbf{q}^m - \mathbf{M}\mathbf{z}^k, \quad \Delta\mathbf{z} = \mathbf{P}_{\text{M}}^{-1}\mathbf{q}_{\text{lin}}, \quad \text{and} \quad \mathbf{z}^{k+1} = \mathbf{z}^k + \Delta\mathbf{z} \tag{2.37}$$

where $\mathbf{q}_{\text{lin}}$ is the residual of the linear system and $\mathbf{P}_{\text{M}}$ is the preconditioner of $\mathbf{M}$. The procedure of Eq. (2.37) is started choosing $\mathbf{z}^0 = \mathbf{u}^m - \mathbf{u}^{m-1}$.

As already introduced, in the present work a Symmetric Gauss-Seidel scheme is adopted in which the system matrix is decomposed as $\mathbf{M} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where $\mathbf{L}$, $\mathbf{U}$ and $\mathbf{D}$ are the strictly lower, the strictly upper and the diagonal parts of the matrix, respectively. It can be shown [36] that the preconditioner corresponding to the SGS scheme is $\mathbf{P}_{\mathsf{M}} = (\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})$.

Eq. (2.37) represent a linear iteration of the SGS scheme and it is repeated until the ratio between the norm of the linear residual and the norm of the non-linear residual is smaller below a given threshold, i.e.

$$L_2(\mathbf{q}_{\text{lin}}{}^k) \leq 0.1\, L_2(\mathbf{q}^m)$$

A drop of a factor 10 in the linear residual has been shown to be a reasonable compromise between computational costs and convergence rate. [36] Very low values of tolerance are not useful since no improvements in the convergence rate can be expected due to the use of an approximate Jacobian.

The efficiency of any iterative solver depends on "how well" the preconditioner approximate the system matrix and how easily can it be inverted. If the first requirement is not sufficiently satisfied the convergence can be poor and in some cases it can stall but usually the more effective the preconditioner, the more expensive its application is. As usual a trade-off between effectiveness and cost is necessary.

Eq. (2.37) is solved by computing the matrix-vector products $\mathbf{M}\mathbf{z}^k$ and applying the preconditioner to $\mathbf{q}_{\text{lin}}{}^k$. Both tasks may be efficiently carried out by taking into account the topology of the approximate Jacobian as described in the next section.

The approximate Jacobian

The Jacobian $\partial\tilde{\mathbf{q}}/\partial\mathbf{u}$ of Eq. (2.36) is approximate, in that only the first-order upwind term of Eq. (2.24) is considered and its derivation is not computed exactly, i.e. $\widetilde{\Phi}_{ik} = \Phi^{\mathrm{I}}_{ik}$ and $\partial\widetilde{\Phi}_{ik}/\partial \mathsf{u}_i \approx \partial\Phi_{ik}/\partial \mathsf{u}_i$. Since the adopted first-order fluxes depend only on the solution on the nodes i and k, i.e. no extended structure is necessary, the approximated Jacobian can be constructed simply looping over the edges as it is done for the residual.

Using the upper-lower decomposition for the system matrix adopted in the SGS scheme, the contribution of each edge to $\mathbf{M}$ is

$$[ii] \quad \mathsf{D}_i = \mathsf{D}_i + \frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}, \qquad [ik] \quad \mathsf{L}_{ik} = \frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k},$$
$$[ki] \quad \mathsf{U}_{ki} = -\frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}, \qquad [kk] \quad \mathsf{D}_k = \mathsf{D}_k - \frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k},$$

The off-diagonal terms L_{ik} and U_{ki} do not need to be accumulated and the assembly of the diagonal terms should be completed with the boundary flux Jacobian and the area terms of Eq. (2.36), which have been neglected in above.

The numerical flux Jacobian for the Roe flux in Eq. (2.21) are obtained by taking the Roe matrix as a constant during the differentiation, which gives

$$\frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i} = \frac{\mathsf{A}(\mathsf{u}_i) + |\widetilde{\mathsf{A}}(\mathsf{u}_i, \mathsf{u}_k)|}{2} \quad \text{and} \quad \frac{\partial\widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k} = \frac{\mathsf{A}(\mathsf{u}_k) - |\widetilde{\mathsf{A}}(\mathsf{u}_i, \mathsf{u}_k)|}{2}. \tag{2.38}$$

The numerical boundary flux in Eq. (2.16) is differentiated considering the chain rule, i.e.

$$\frac{\partial\widetilde{\Phi}^{\partial}_{i}}{\partial \mathsf{u}_i} = \mathsf{A}\big(\mathsf{u}^{\partial}(\mathsf{u}_i)\big)\frac{\partial \mathsf{u}^{\partial}}{\partial \mathsf{u}_i}$$

where $\partial \mathsf{u}^\partial / \partial \mathsf{u}_i$ is computed exactly from the definition given by Eq. (2.26) and (2.27).

It is worth to note that the numerical dissipation and the pseudo-time step have a key role in the solution of the linear system. The complete expression of the diagonal terms of $\mathbf{M}$, including the time step and the boundary contributions, is

$$\mathsf{D}_i = \frac{V_i}{\Delta\tau_i}\mathsf{I}^4 + \frac{\partial \widetilde{\Phi}_i^\partial}{\partial \mathsf{u}_i} + \sum_{k \in \mathcal{K}_{i,\neq}} \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}.$$

For an internal node, taking into account Eq. (2.38) and the fact that the control volume is closed, the above equation reduces to

$$\mathsf{D}_i = \frac{V_i}{\Delta\tau_i}\mathsf{I}^4 + \frac{1}{2}\sum_{k \in \mathcal{K}_{i,\neq}} |\widetilde{\mathsf{A}}(\mathsf{u}_i, \mathsf{u}_k)|.$$

The above equation highlights that both the numerical dissipation and the pseudo time-step contribute to the diagonal dominance of the matrix. A small value of the local Courant number together with a large dissipation are therefore beneficial to increase the convergence properties of the scheme. The artificial dissipation introduced in the scheme depends directly on the (entropy fixed) eigenvalues and on η_{ik}, therefore a small mesh size may cause the diagonal dominance of the matrix to reduce at the point where the solution stalls. Thus if the mesh size is reduced it is generally necessary to lower the overall value the Courant number, which must be updated more slowly.

Matrix-vector products

The computation of the Matrix-vector products $\mathbf{Mz}$ is necessary in the solving procedure of Eq. (2.37). The i-th component of the resulting vector is given by

$$\mathbf{Mz}|_i = \left[\frac{V_i}{\Delta\tau_i}\mathsf{I}^4 + \frac{\partial \widetilde{\Phi}_i^\partial}{\partial \mathsf{u}_i} + \sum_{k \in \mathcal{K}_{i,\neq}} \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}\right] \mathsf{z}_i + \sum_{k \in \mathcal{K}_{i,\neq}} \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k}\mathsf{z}_k,$$

where the term inside the square bracket is equal to D_i. The first two terms depends only on the solution u_i and can be efficiently computed. The computation of the third and last term however require one loop on the nodes $i \in \mathcal{K}$ and another loop on the nodes $k \in \mathcal{K}_{i,\neq}$, which can be time consuming. A very efficient way to compute this terms can be derived if the edge-based data structure is used. Some algebra reveals that, neglecting the time-step contribution and the boundary term, the matrix-vector product can be assembled as

$$\begin{aligned} \mathbf{Mz}|_i &= \mathbf{Mz}|_i + \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}\mathsf{z}_i + \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k}\mathsf{z}_k, \\ \mathbf{Mz}|_k &= \mathbf{Mz}|_k - \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_i}\mathsf{z}_i - \frac{\partial \widetilde{\Phi}_{ik}}{\partial \mathsf{u}_k}\mathsf{z}_k, \end{aligned} \tag{2.39}$$

which involve a loop on the edges ik.

Symmetric Gauss-Seidel Preconditioning

The inversion of the SGS preconditioner of Eq. (2.37) is performed sweeping twice on the nodes of the matrix, back and forth. First the lower-triangular matrix $(\mathbf{D} + \mathbf{L})$ is inverted with a Gauss-Seidel algorithm, i.e. executing a forward sweep on the nodes of $\mathcal{K}$ as

$$\Delta \mathbf{z}^* = \mathsf{D}_i^{-1} \left(\mathsf{q}_{\mathsf{lin}_i} - \sum_{k \in \mathcal{K}_{i,<}} \mathsf{L}_{ik} \Delta \mathbf{z}_k^* \right), \qquad i \in \mathcal{K}.$$

A second (backward) sweep on the nodes of $\mathcal{K}$ is performed to invert the upper-triangular matrix $(\mathsf{I}^4 + \mathbf{D}^{-1}\mathbf{U})$, i.e.

$$\Delta \mathbf{z}_i = \Delta \mathbf{z}_i^* - \mathsf{D}_i^{-1} \sum_{k \in \mathcal{K}_{i,>}} \mathsf{U}_{ik} \Delta \mathbf{z}_k, \qquad i \in \{\mathcal{K},\ \text{backward}\}.$$

$\mathcal{K}_{i,<}$ and $\mathcal{K}_{i,>}$ are respectively the lower and upper subsets of $\mathcal{K}_{i,\neq}$, i.e.

$$\mathcal{K}_{i,<} = \{k \in \mathcal{K}_{i,\neq} : k < i\} \qquad \text{and} \qquad \mathcal{K}_{i,>} = \{k \in \mathcal{K}_{i,\neq} : k > i\}.$$

The preparatory work for the inversion of the preconditioner is limited to the inversion of the diagonal 4×4 matrix D_i, which is computed exactly. There is no need to store both $\mathbf{M}$ and $\mathbf{P}_\mathsf{M}$, but it is enough to store only $\mathbf{D}$, $\mathbf{L}$, $\mathbf{U}$, and $\mathbf{D}^{-1}$ at each non-linear iteration, thus halving the required memory.

2.4.4 Implicit solver for unsteady equations

The unsteady Euler ALE equations are solved resorting to the pseudo-time stepping approach, similarly to the steady case described in section 2.4.2. u^{n+1} is sought as the steady state solution in the fictitious time τ of a modified version of Eq. (2.35), i.e.

$$\mathbf{V}^{n+1} \left[\frac{a_{-1}}{\Delta t} + \frac{1}{\Delta \tau} \right] \Delta \mathbf{u}^{m+1} = -a_{-1} \frac{\mathbf{V}^{n+1}\mathbf{u}^m}{\Delta t} - \sum_{q=-1}^{p} a_q \frac{\mathbf{V}^{n-q}\mathbf{u}^{n-q}}{\Delta t} - \mathbf{q}(\mathbf{u}^m).$$

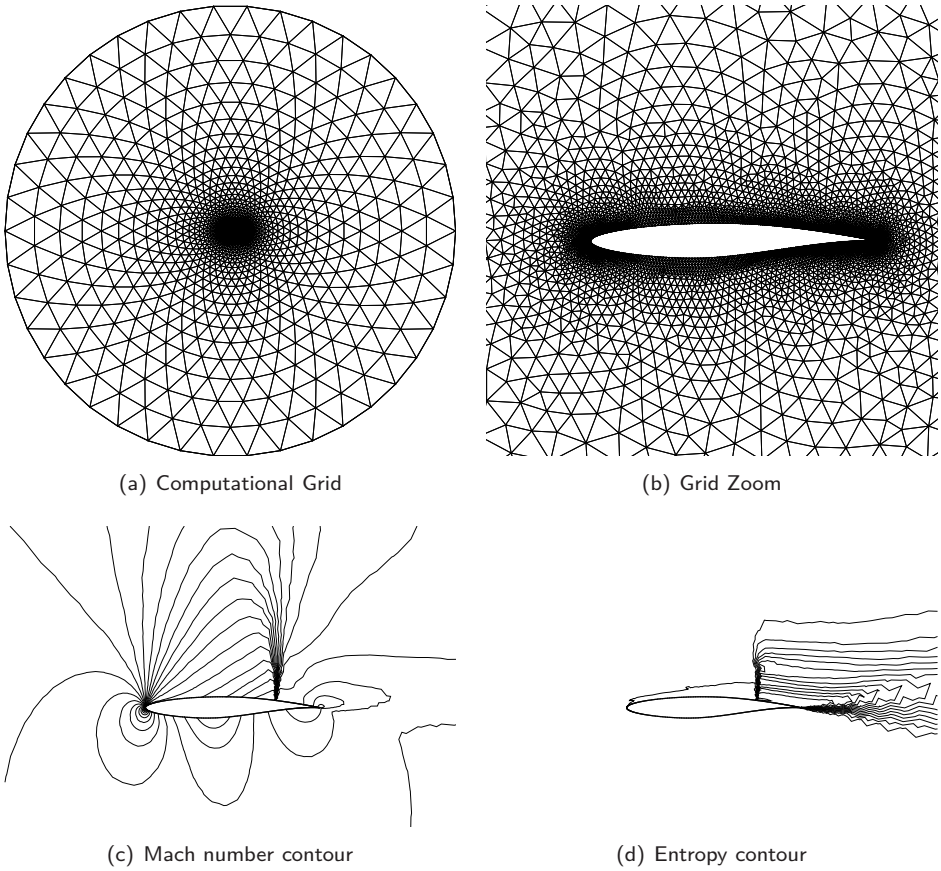
When the steady state is obtained, i.e. $\Delta \mathbf{u}^{m+1} = 0$ and $\mathbf{u}^m = \mathbf{u}^{n+1}$, the above equation reduces to

$$\sum_{q=-1}^{p} a_q \frac{\mathbf{V}^{n-q}\mathbf{u}^{n-q}}{\Delta t} + \mathbf{q}(\mathbf{u}^{n+1}) = 0,$$

which is the discrete version of Eq. (2.34). Again a local time-stepping technique is adopted to choose an appropriate value of $\Delta\tau$. Differently from the steady state computation, however, the diagonally dominance of the system matrix is already increased by the $a_{-1}/\Delta t$ term. This and the fact that, at every time step, the initial solution is relatively close to the final one allows using larger values of γ and maximum Courant number.

2.5 Test cases for steady problems

The edge-based finite volume solver is tested on standard fluid dynamic and aerodynamic test cases in order to assess the properties of the numerical scheme, i.e. steady flow past a transonic airfoil, steady low Mach number flow around a cylinder and the steady supersonic flow facing a positive slope.



(a) Computational Grid (b) Grid Zoom

(c) Mach number contour (d) Entropy contour

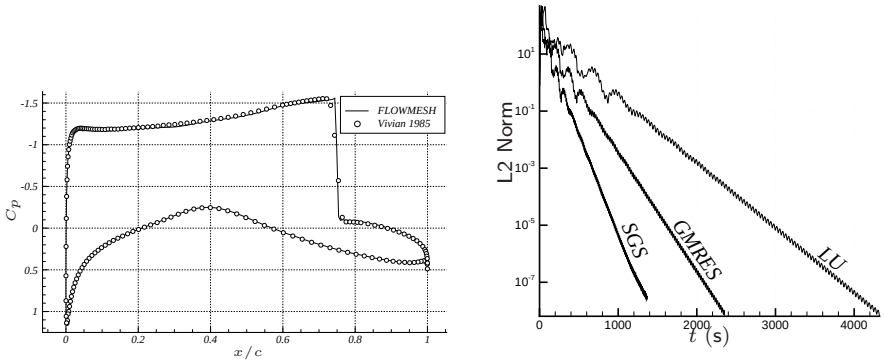
Fig. 2.6: RAE 2822 test case obtained at $M_\infty = 0.75$ and $\alpha = 3^\circ$. (a) Computational grid. (b) Computational grid close-up. (c) Mach number contour lines. (d) Entropy variation contour lines.

2.5.1 RAE 2822

The edge-based finite volume solver is tested on the popular RAE-2822 aerodynamic test case. The computational grid shown in fig. 2.6(a) and 2.6(b) is made of 11451 triangles and 5942 nodes and the external radius is located at 20 chords of distance from the nose of the airfoil. The Mach number of the free flow is $M_\infty = 0.75$ and the angle of attack is $\alpha = 3^\circ$. The transonic nature of the flow field causes the generation of a compression shock on the upper side of the airfoil as shown by the Mach number contour lines of fig. 2.6(c). The entropy contour lines of fig. 2.6(d) show the abrupt variation across the shock as well as the diffusion after the trailing edge, where the growth in the size of the elements causes the scheme to be more dissipative.

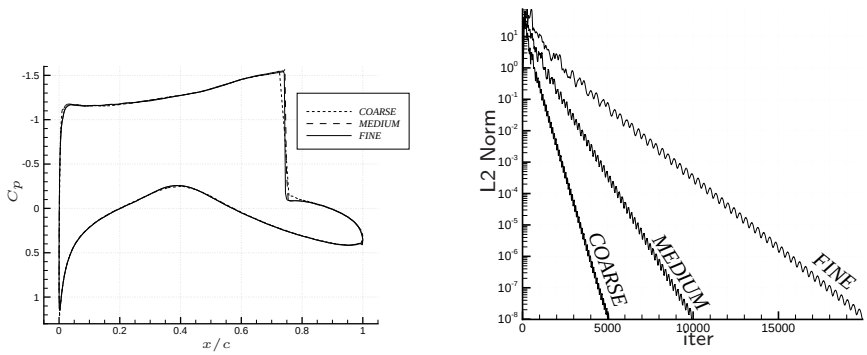
In fig. 2.7(a) the pressure coefficient, defined as

$$C_p = \frac{P - P_\infty}{\frac{1}{2}\rho_\infty v_\infty^2},$$



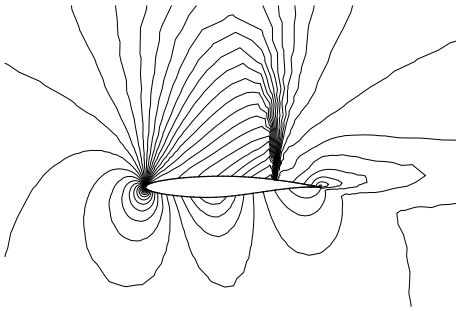
(a) Pressure coefficient (b) Convergence Curves

Fig. 2.7: (a) Comparison between the solution obtained with the SGS scheme with the reference data by Vivian [79] in terms of pressure coefficient. (b) L2 Norm of the residual as a function of the computational time for three different linear system solvers: SGS, GMRES (Sparsekit) and LU factorization (UMFPACK).

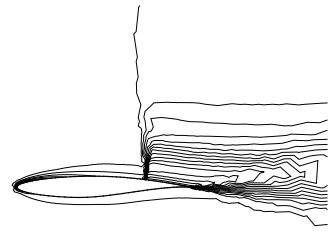


(a) Pressure coefficient (b) Convergence Curves

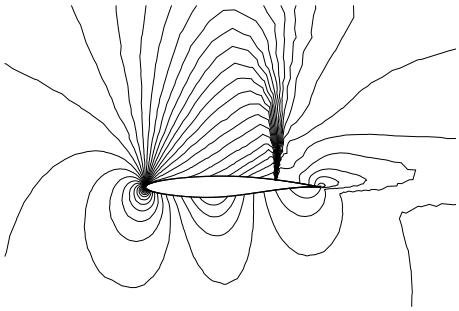
Fig. 2.8: RAE 2822 test case for different computational grids. Coarse: 2798 nodes, 5429 elements. Medium 10820 nodes, 21306 elements. Fine 42464 nodes 84260 elements. (a) Pressure coefficient on the airfoil. (b) L2 Norm of the residual as a function of the non-linear iterations.



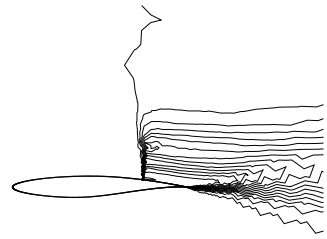
(a) Mach, coarse grid



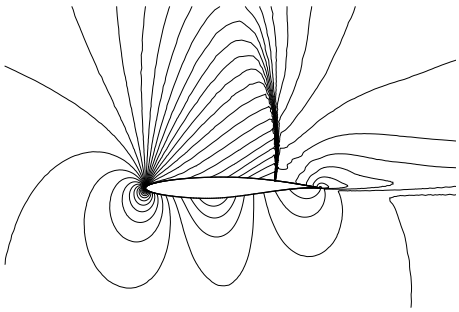
(b) Entropy, coarse grid



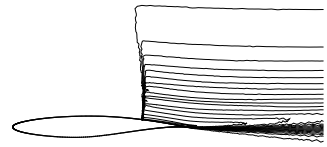
(c) Mach, medium grid



(d) Entropy, medium grid



(e) Mach, fine grid



(f) Entropy, fine grid

Fig. 2.9: Mach number and entropy contours on different grids for the RAE 2822 test case at $M_\infty = 0.75$ and $\alpha = 1^\circ$. Coarse: 2798 nodes, 5429 elements. Medium 10820 nodes, 21306 elements. Fine 42464 nodes 84260 elements.

is compared a reference solution computed with the Jameson scheme [79]. The agreement between the two is fairly good, even though the solution computed with the edge-based solver is slightly more dissipative across the shock. In fig. 2.7(b) the ratio between the L^2 norm of the residual and the one computed at the first iteration is shown as a function of the computational time. Three different linear system solver have been adopted: symmetric Gauss-Seidel, a GMRES solver (Sparsekit) and a LU factorization based solver (UMFPACK). The present implementation of the SGS scheme is approximately twice as fast as Sparsekit and four times as fast as UMFPACK[1].

In fig. 2.8(a) pressure coefficient curves for the RAE 2822 test case obtained with progressively meshes are shown. Three grids have been adopted a coarse one, with 2798 nodes and 5429 elements, an intermediate one, with 10820 nodes and 21306 elements and a fine one, with 42464 nodes 84260 elements. In fig. 2.8(b) the L2 norm of the residual at every non-linear iteration is plotted for every mesh. Doubling the number of elements doubles the iterations required to achieve convergence. The computed solutions in terms of Mach number contour lines are shown in fig. 2.9(a), 2.9(c) and 2.9(a). Increasing the grid resolution allows to better capture the shock, that appears sharper, and the airfoil wake that is caused by the non isoentropic recompression that occurs on the upper side. In fig. 2.9(b), 2.9(d) and 2.9(b) the contour lines for the entropy, s, are show. The diffusive behavior of the scheme, highlighted in the wake region, is reduced increasing the number of nodes. Moreover, in fig. 2.9(b) the presence of a numerical boundary layer is revealed by a significant increase in entropy in the region close to the nose of the airfoil. The size and intensity of the boundary layer is reduced in the finer grids and it is not visible with the given number of contour lines.

2.5.2 2D cylinder

The second test case is the incompressible steady flow around a circular cylinder section shown in fig. 2.10. The numerical solution has been computed at a free stream Mach number of 0.3, which is close to the incompressible limit. The analytical (exact) solution for the incompressible steady flow is

$$C_p^{inc} = 2\frac{a^2}{R^2}\left[\cos^2\theta - \sin^2\theta - \frac{a^2}{8R^2}\right], \tag{2.40}$$

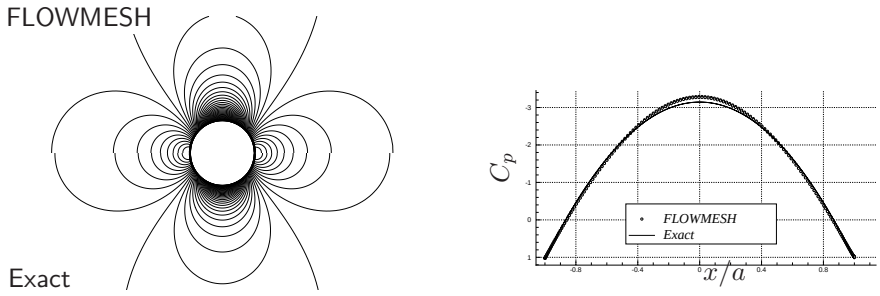
where a is the cylinder radius, $R = |\boldsymbol{x}|$ and $\theta = \tan^{-1}(y/x)$. Eq. 2.40 is valid only in the incompressible limit, therefore the pressure coefficient has been corrected with the Karman-Tsien formula, namely

$$C_p(M) = \frac{C_p^{inc}}{\sqrt{1-M^2} + \frac{C_p^{inc}}{2}\frac{M^2}{1+\sqrt{1-M^2}}},$$

which is considered to give reasonable results for values of the free-stream Mach number lower than $0.5 \div 0.6$.

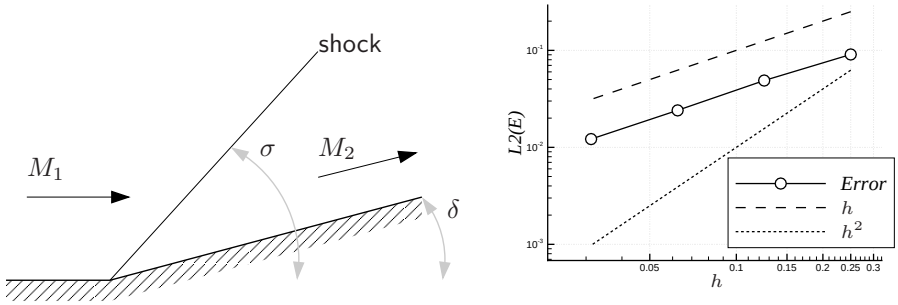
A comparison between the solution obtained with the finite volume solver and the analytical one is shown in fig. 2.10, the computed solution agrees fairly well with the reference solution. Fig. 2.10(b) shows the C_p curves along the surface of the cylinder. An error of roughly 5% is present for $x/a = 0$ and a slight non-symmetry in the solution

[1]The computations have been performed on a Intel Xeon X5650 at 2.67GHz machine.



(a) C_p contour lines (b) C_p on the cylinder

Fig. 2.10: Comparison between the solution obtained with the finite volume solver and the analytical one for the case of a quasi-incompressible flow across a 2D cylinder. The numerical solution has been computed at a free stream Mach number of 0.3 and the analytical one has been corrected using the Karman-Tsien formula.



(a) Oblique Shock (b) Convergence Curves

Fig. 2.11: Circular cylinder at

M_1	M_2	δ	σ
2	12.8	18.52°	50.2°

Fig. 2.12: Supersonic wedge test case: reference values of the analytic solution.

is visible between the lower and the upper surface caused by a small asymmetries in the computational grid.

2.5.3 Supersonic wedge

The last test case is the case of a supersonic flow encountering a positive slope schematically represented in fig. 2.11(a). An analytical (exact) solution can be computed for small values of the deflection angle δ, which correspond to an shock developing from the initial point of the slope with an angle $\sigma > \delta$. Since the shock is oblique the Mach number in the region after the shock, M_2, is still supersonic. The values for the geometry of the problem and its analytic solution are gathered in table 2.12.

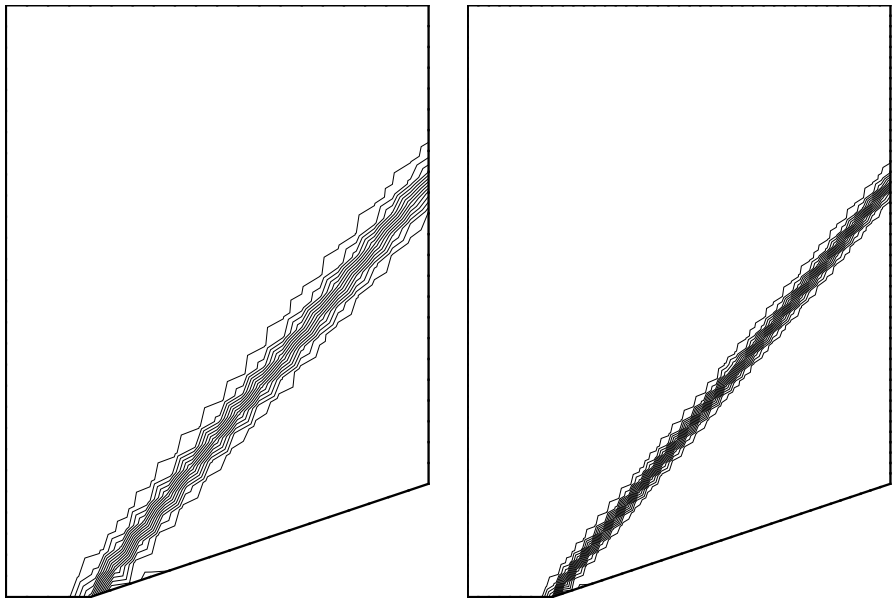
In fig. 2.13 the pressure coefficient contour lines obtained with increasingly finer meshes of uniformly distributed triangles are reported. The first, coarsest, grid has 658 nodes, 1222 elements and an average size of $h = 0.25$, where h is the smaller edge of a triangle. The second grid has 2530 nodes, 4874 elements and $h = 0.125$. The third grid is made of 9839 nodes and 19309 elements with $h = 0.0625$. The finer grid has 38827 nodes and 76919 elements, with $h = 0.03125$.

The L_2 norm of the error between the computed solution and the exact one is shown in fig. 2.11(b). The grid convergence curve show a first order behavior, which is agreement with the adopted scheme which is second order away form the discontinuities but it is only globally first-order when shocks are present.

2.6 ALE scheme for adaptive grids

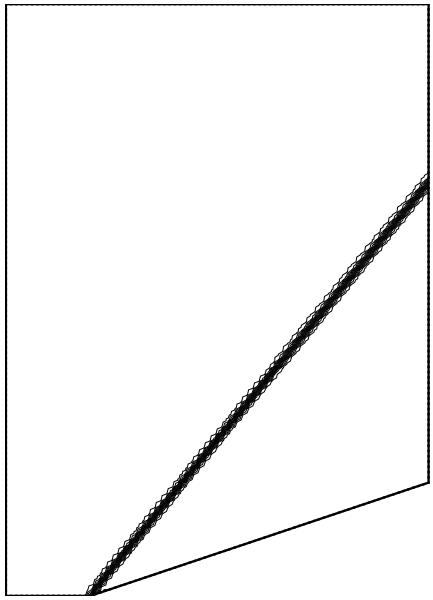
In the present section, the implicit finite-volume scheme introduced in section 2.4 is extended to the case of adaptive grids. The adaptation strategy is based only on local modifications of the topology of the mesh, that is edge-swapping, node insertion and node deletion, and displacement of nodes, e.g. mesh deformation and regularization. When no changes in the topology occurs, i.e. adaptivity is only limited to mesh movement and distortion, the change in position and shape of the finite volumes is easily taken into account by the ALE formulation as shown in section 2.4. However the modification in the shape of the volumes caused by the application of adaptation techniques, e.g. the swapping of an edge shown in fig. 2.14, have to be treated separately. If the variation of area is not correctly taken into account when projecting the solution onto the new grid, conservation of mass, momentum and total energy is to be explicitly imposed and possibly bounds on the total variation of the solutions are to be enforced [125, 120]. Moreover, if a p-step time integration scheme is adopted, such a conservative interpolation procedure must be repeated p times and thus additional difficulties may arise when dealing with multi-step high-order integration schemes. The reader is referred to references [83, 192] for a detailed discussion of this topic.

In principle, due to the modification of the grid connectivity resulting from the application of the adaptation operators, the solution at time t^{n+1} is to be interpolated over the new grid. In the present work a different strategy is proposed to avoid the introduction of any explicit interpolation step by exploiting the ALE approach, as it is commonly done when only mesh deformation is used. Admittedly, the application of ALE mapping is equivalent to an interpolation step; however, its application does not require

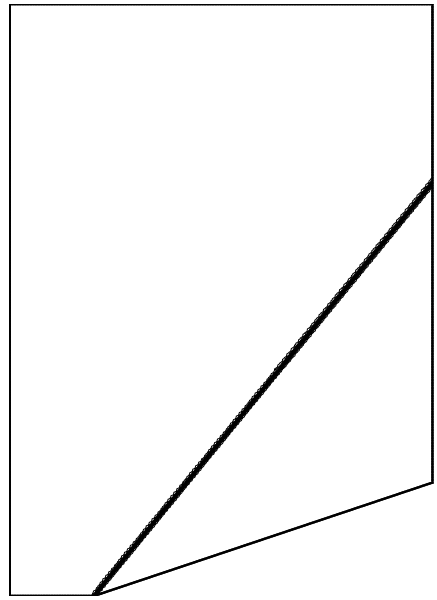


(a) Mesh R-1

(b) Mesh R-2

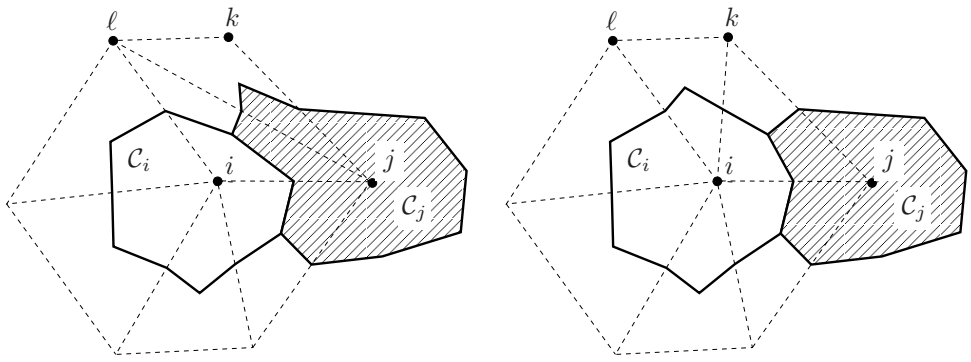


(c) Mesh R-3



(d) Mesh R-4

Fig. 2.13: Mach number contours on different grids for the oblique shock test case at $M_1 = 2$ and $\delta = 18.52°$. Mesh R-1: 658 nodes, 1222 elements, $h = 0.25$. Mesh R-2 2530 nodes, 4874 elements, $h = 0.125$. Mesh R-3 9839 nodes, 19309 elements, $h = 0.0625$. Mesh R-4 38827 nodes, 76919 elements, $h = 0.03125$.



(a) Initial configuration t^n (b) Swapped configuration t^{n+1}

Fig. 2.14: Initial and final configuration of the mesh surrounding a swapped edge. a) Initial configuration at time level t^n. b) Final configuration at time level t^{n+1}.

any special treatment to ensure appropriate accuracy, conservativeness and preservation of function signs.

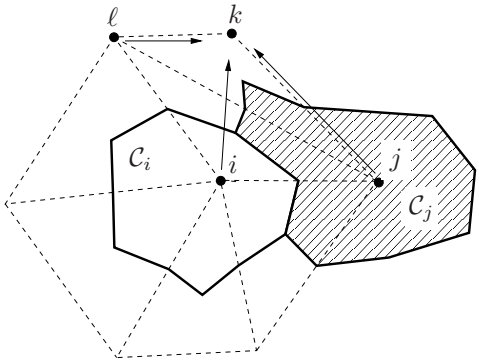
The key idea is to give an interpretation of the changes in the topology that happen in the time lapse from t^n and t^{n+1} as continuous deformation of the finite volumes performed within the time interval $[t^n, t^{n+1})$, e.g. sections 2.6.1, 2.6.3 and 2.6.5. Such approach is based on the idea of splitting the area swept by the interfaces into two separate contributions: the deformation one, namely D, arising from the continuous (in time) mesh movement and distortion and the adaptation one, namely A, which is essentially correction term. As result, the total area swept by the interface $\partial\mathcal{C}_{ik}$ during in the time interval is $\Delta V_{ik}^{n+1} = \Delta A_{ik}^{n+1} + \Delta D_{ik}^{n+1}$.

Such an interpretation reveals beneficial in many respects. First, it is no longer necessary to explicitly interpolate the old solution over the new grid, therefore the conservation of the flow variables is guaranteed by construction. Moreover, since cross-grid interpolation is avoided, the implementation of multi-step high-order schemes for time integration, e.g. BDF schemes, is straightforward and it does not require to resort to cross-grid interpolation [83, 192].

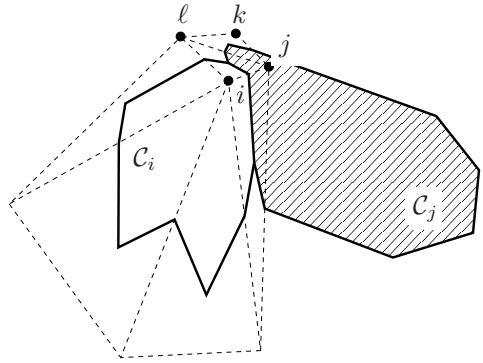
In section 2.6.1 the edge-swapping is first described, in section 2.6.2 the overall ALE scheme for grids with variable connectivity is outlined, in section 2.6.3 and 2.6.4 domain and boundary nodes insertion are depicted, section 2.6.5 and 2.6.6 illustrate the node deletion for domain and boundary nodes respectively and in section 2.6.7 the overall scheme is outlined. In the next sections the adaptation techniques are described putting the emphasis on the modifications to the ALE scheme that are necessary to avoid the grid interpolation step. Chapter 3 describes how the same techniques are put together to perform the adaptation of the grid. Although chapter 3 and the present section focus on different aspects of the grid adaptation strategy, some repetitions may still be present.

2.6.1 Edge-Swapping

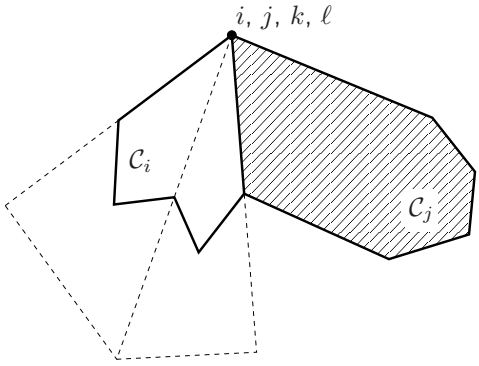
We start by commenting on fig. 2.14, which shows how the swap of edge j-ℓ into edge i-k affects the shape of the finite volumes $\mathcal{C}_i$ and $\mathcal{C}_j$. It can be observed that the modification of the node to node connectivity is reflected on a modification of the cells



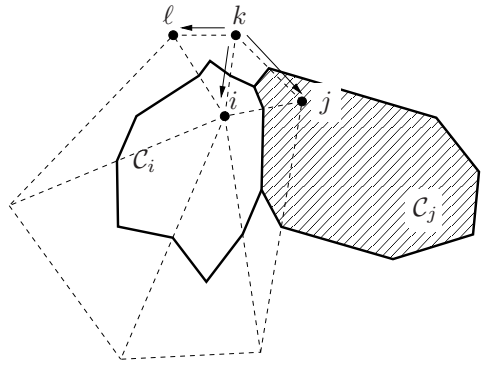
(a) $\tau = 0$



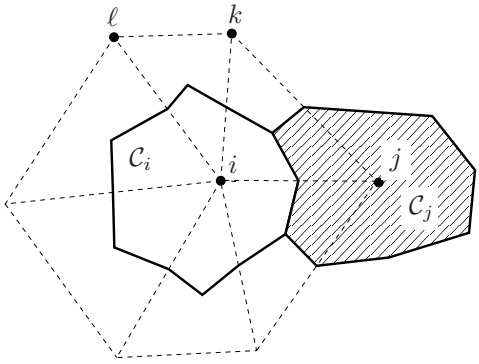
(b) $0 < \tau < \frac{1}{2}$



(c) $\tau = \frac{1}{2}$



(d) $\frac{1}{2} < \tau < 1$



(e) $\tau = 1$

Fig. 2.15: Computation of the interface velocities in case of edge-swapping. a) Initial configuration at time t^n which correspond to the non-dimensional time $\tau = 0$. b) Collapse of the quadrangle i, j, k, ℓ on the node k. c) Change of topology of the quadrangle at an non-dimensional time $\frac{1}{2}$. d) Expansion of the quadrangle from the node k. e) Final configuration at the non-dimensional time $\tau = 1$.

shape to comply with the definition of dual mesh given in section 2.3.3. Indeed due to the removal of the edge j-ℓ, the cells $\mathcal{C}_j$ and $\mathcal{C}_\ell$ no longer share the points included in $\partial\mathcal{C}_{j\ell}$ and, due to the insertion of the edge i-k, the interface $\partial\mathcal{C}_{ik}$ separates now the cell $\mathcal{C}_i$ from $\mathcal{C}_k$.

The idea is to interpret the change of area associated with the edge swap as a special case of grid deformation. Indeed, difficulties arise in the computation of areas swept by the interfaces since no change of the position of any grid node is related to the variation of cell areas caused by edge swapping, as shown by fig. 2.14. To overcome such difficulties a three-steps technique is introduced, composed by a node collapse phase, a swap phase and an expansion phase. Fig. 2.15 illustrates how the swapping of the edge $j-\ell$ can be viewed as a continuous modification of the finite volumes $\mathcal{C}_i$, $\mathcal{C}_j$, $\mathcal{C}_k$ and $\mathcal{C}_\ell$ taking place during the time interval $[t^n, t^{n+1})$, described as a function of the non-dimensional time $\tau = (t - t^n)/(t^{n+1} - t^n)$.

$\tau = 0$**:** The initial condition is sketched in fig. 2.15(a) showing the two cells $\mathcal{C}_i$ and $\mathcal{C}_j$.

$0 < \tau < \frac{1}{2}$**:** The quadrilateral i-k-j-ℓ collapses, as shown in fig. 2.15(b), giving as a result a area swept by the interfaces of the cells $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_\ell$ thus an interface velocity for the node-pairs connected with the nodes i, j and ℓ.

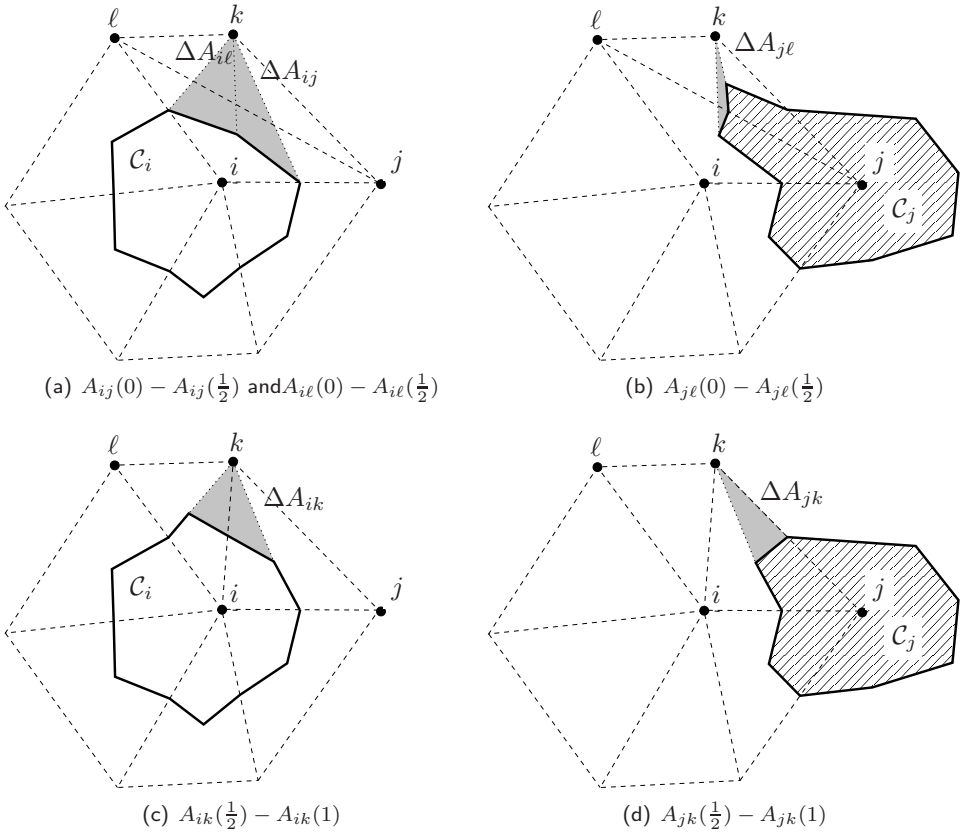
$\tau = \frac{1}{2}$**:** At this time the four grid nodes share the same position, as shown in fig. 2.15(c). The edge j-ℓ is swapped with the edge i-k. No area is swept by any interface during the change of connectivity since all the interfaces that are involved in this change of connectivity, i.e. those located inside the quadrangle i-k-j-ℓ, have null area.

$\frac{1}{2} < \tau < 1$**:** During this step, shown in fig. 2.15(d), the expansion phase takes place and the nodes i, j and ℓ are moved back to the original positions but without the edge j-ℓ. Again an area is swept by the interfaces of the cells $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_\ell$.

$\tau = 1$**:** The final configuration is reached, as shown in fig. 2.15(e).

In fig. 2.16 is shown the area swept by each cell interface ΔA. Fig. 2.16(a) shows the areas $\Delta A_{i\ell}$ and ΔA_{ij} swept in the time interval $0 < \tau < \frac{1}{2}$; the area $\Delta A_{j\ell}$ swept in the step $0 < \tau < \frac{1}{2}$ is shown in fig. 2.16(b); fig. 2.16(c) highlights the area ΔA_{ik} swept in the interval $\frac{1}{2} < \tau < 1$; the area ΔA_{jk} swept for $\frac{1}{2} < \tau < 1$ is shown in fig. 2.16(d). Both the collapse, i.e. $\tau \in (0, \frac{1}{2})$, and the expansion, i.e. $\tau \in (\frac{1}{2}, 1)$, phases are simply grid deformation steps, thus the swept areas can be easily computed following the approach already presented in section 2.4. On the contrary, the actual swap operation at time $\tau = \frac{1}{2}$ has no effects in terms of interface velocities. Moreover, since the overall area swept by each interface is given by the sum of the ΔA contributions of the collapse and the explosion steps, thus for the interfaces located outside the i-k-j-ℓ quadrangle the total swept area is null and such is the corresponding interface velocity.

The choice of the collapse point is arbitrary and for all possible choices the DGCL condition (2.33) is fulfilled. However, the farthest the collapse node is located from the swapped edge, the greater the value of the interface velocities are. Therefore, a bad choice of the collapse point location will introduce larger perturbations in the governing equations. So, even thought any combination of subsequent cell deformations is acceptable, provided that the DGCL is always satisfied, the one that minimize the swept areas should be preferred.



(a) $A_{ij}(0) - A_{ij}(\frac{1}{2})$ and$A_{i\ell}(0) - A_{i\ell}(\frac{1}{2})$ (b) $A_{j\ell}(0) - A_{j\ell}(\frac{1}{2})$

(c) $A_{ik}(\frac{1}{2}) - A_{ik}(1)$ (d) $A_{jk}(\frac{1}{2}) - A_{jk}(1)$

Fig. 2.16: Areas swept during edge swapping. a,b) Area swept by the interfaces $\partial\mathcal{C}_{ij}$, $\partial\mathcal{C}_{i\ell}$ and $\partial\mathcal{C}_{j\ell}$ during the time interval $[0, \frac{1}{2})$. c,d) Area swept by the interfaces $\partial\mathcal{C}_{ik}$, and $\partial\mathcal{C}_{jk}$ during the time interval $[\frac{1}{2}, 1)$.

Edge insertion

When an edge i-k is inserted into a mesh a new interface between two finite volumes is created to comply with the dual mesh definition given in section 2.3.3, e.g. the i-k edge of fig. 2.15. It is trivial to say that the metrics of the interface $\partial\mathcal{C}_{ik}$ for $t^m \leq t^n$ are identically equal to zero, e.g. $\eta_{ik}^m = 0$ and $\Delta V_{ik}^m = 0$, while the metrics at the time t^{n+1} are given by the current mesh configuration. Therefore the numerical fluxes for the implicit scheme are evaluated at t^{n+1} as normal, namely

$$\Phi_{ik}^{n+1} = \Phi(\mathsf{u}_i^{n+1}, \mathsf{u}_k^{n+1}, \nu_{ik}^{n+1}, \hat{\boldsymbol{\eta}}_{ik}^{n+1}, \eta_{ik}^{n+1}),$$

where ν_{ik}^{n+1} is computed according to Eq. (2.32), with ΔA_{ik}^{n+1} calculated as shown above.

Edge deletion

When an edge j-ℓ is removed from the mesh, the corresponding portion of the finite volume interface $\partial\mathcal{C}_{ik}$ is deleted to comply with the dual mesh definition given in section 2.3.3, e.g. the j-ℓ edge of fig. 2.15. Thus at the time $t^m \geq t^{n+1}$ the interface has a zero length, i.e. $\eta_{ik}^m = 0$, and the normal unit vector can be taken as $\hat{\boldsymbol{\eta}}_{j\ell}^m = \lim_{t\to t^{n+1}} \hat{\boldsymbol{\eta}}_{j\ell}(t)$ which is different from zero, since $|\hat{\boldsymbol{\eta}}_{j\ell}(t)| \neq 0, \forall t$.

Assuming for simplicity that the mesh is fixed, i.e. $D = 0$, and that no other topology changes occurs, then the integrated interface velocity is given by the correction term A only. According to Eq. (2.32), $\nu_{j\ell}^{n+1} = \alpha_{-1}\Delta A_{j\ell}^{n+1}/\Delta t$, which is generally different from zero even if $\eta_{j\ell}^{n+1} = 0$. More generally, for a $p+1$ accurate BDF scheme, the interface velocity of an edge that has been removed during the adaptation phase between the time level t^n and t^{n+1} will be different from zero for the subsequent p timesteps, namely $\nu_{j\ell}^m \neq 0$ with $m \leq n+p+1$.

Therefore the numerical flux associated to a deleted edge is

$$\Phi_{j\ell}^{n+1} = \Phi(\mathsf{u}_j^{n+1}, \mathsf{u}_\ell^{n+1}, \nu_{j\ell}^{n+1}, \hat{\boldsymbol{\eta}}_{j\ell}^{n+1}, 0),$$

Evaluating the first-order upwind fluxes of Eq. (2.21) with $\eta_{j\ell}^{n+1} = 0$ and $\hat{\boldsymbol{\eta}}_{j\ell}^{n+1} = \hat{\boldsymbol{\eta}}_{j\ell}^n$ one obtains

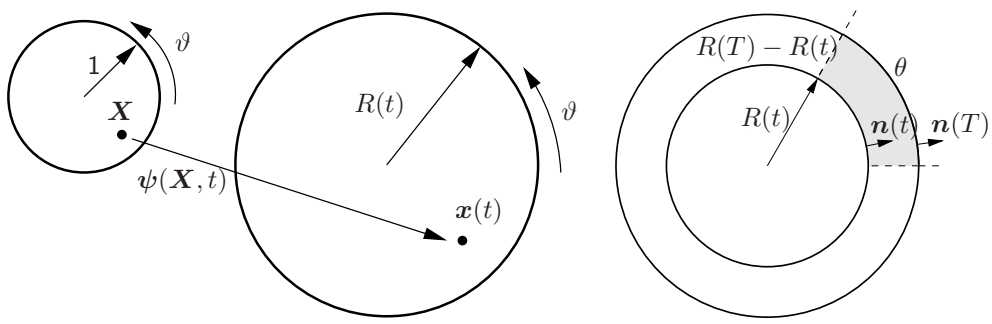
$$\Phi_{j\ell}^{\mathrm{I},n+1} = \nu_{j\ell}^{n+1}\frac{\mathsf{u}_j^{n+1} + \mathsf{u}_\ell^{n+1}}{2} - \frac{|\nu_{j\ell}^{n+1}|}{2}(\mathsf{u}_\ell^{n+1} - \mathsf{u}_j^{n+1}).$$

Indeed the average of the Euler fluxes is multiplied by $\eta_{j\ell}^{n+1}$, which is zero, and in this case A does not depend on $\widetilde{\mathsf{u}}$ and $\hat{\boldsymbol{\eta}}_{j\ell}$ since

$$\begin{aligned} \mathsf{A}(\widetilde{\mathsf{u}}, \hat{\boldsymbol{\eta}}_{j\ell}^n, 0, \nu_{j\ell}^{n+1}) &= \mathsf{R}_{j\ell}^{n+1}(\nu_{j\ell}^{n+1}\,\mathsf{I}^4)\mathsf{L}_{j\ell}^{n+1}, \\ &= \nu_{j\ell}^{n+1}\,\mathsf{I}^4. \end{aligned} \tag{2.41}$$

Similarly, when the high-resolution fluxes of Eq. (2.24) are evaluated at $\eta = 0$, the Euler contribution is identically null and the fluxes reduce to

$$\Phi_{j\ell}^{\mathrm{HR},n+1} = \nu_{j\ell}^{n+1}\frac{\mathsf{u}_j^{n+1} - \mathsf{u}_\ell^{n+1}}{2} + \frac{|\nu_{j\ell}^{n+1}|}{2}(\widetilde{\mathsf{w}} - \mathsf{u}_\ell^{n+1} - \mathsf{u}_j^{n+1}),$$



(a) Mapping (b) Area swept by the interface θ

Fig. 2.17: Mapping function between two circles of radius 1 and R. a) Mapping function. b) Area swept by the interface of arc θ from the time t to T.

where the limited jump $\widetilde{\mathrm{w}}$ is computed with the Eq. (2.25) and (2.3.1) by setting $\widetilde{\mathrm{L}} = \mathrm{I}^4$ and $\widetilde{\lambda}_{(h)} = \nu_{j\ell}^{n+1}$. It must be noted, that according to Eq. (2.41), no ad-hoc procedure is necessary for the treatment of removed edges in terms of evaluation of the numerical fluxes, which is performed with the standard routines for ALE Euler fluxes. However an alternative entropy-fix procedure is necessary, since the definition of the relative directional Mach number of Eq. 2.23 is singular for $\eta_{ik} = 0$, namely

$$\widetilde{\delta} = \begin{cases} 0.2\left(\mathrm{M}(\widetilde{\mathrm{u}}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}, \nu_{ik}) + 1\right)\eta_{ik} & \text{if} \quad \eta_{ik} > 0, \\ 0.2\,|\nu_{ik}| & \text{if} \quad \eta_{ik} = 0, \end{cases}$$

where $\widetilde{\delta}$ is the threshold of Eq. (2.22).

The numerical flux corresponding to a removed edge is therefore given by the ALE contribution to the fluxes only, which arises from the balance of the conservative variables due to the mesh movement and the topology modification. In order to preserve the conservativity of the scheme, the contribution to the fluxes associated to a removed edge can be dropped only when the corresponding interface velocity is identically null and this in turns depends on the adopted time integration scheme. For example the contribution of an edge j-ℓ removed during the adaptation step occurring in the time lapse between t^n and t^{n+1} can be dropped when $\nu_{j\ell}^m = 0$, that for a BDF scheme of order $p+1$ correspond to the condition $m > n + p + 1$.

Such remarkable result is a direct consequence of the differential nature of the GCL constraint, i.e. Eq. (2.17) and (2.18), and does not depend directly on the adopted integration scheme. Indeed it is also valid in the time-continuous framework, as it will be shown in the following example. Let us take a circular control volume $\mathcal{C}_X$ of radius 1 and a circular control volume $\mathcal{C}_x$ of radius R, as shown in fig. 2.17. The mapping between the points of the reference configuration and the current one is $\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{X}, t) = R(t)\boldsymbol{X}$, i.e. $\mathcal{C}_x$ is a obtained by shrinking/enlarging $\mathcal{C}_X$. The inverse transformation is easily obtained as $\boldsymbol{X} = \boldsymbol{\Psi}(\boldsymbol{x}, t) = \boldsymbol{x}/R(t)$. The deformation gradient is given by $\boldsymbol{F}(\boldsymbol{X}, t) = \boldsymbol{\nabla}_X \boldsymbol{\psi}(\boldsymbol{X}, t)$ and the Jacobian of the mapping is $J(\boldsymbol{X}, t) = \det \boldsymbol{F} = R^2(t)$.

The velocity of deformation of the element is therefore given by

$$v(\boldsymbol{X},t) = \frac{d\boldsymbol{\psi}}{dt} = \frac{dR}{dt}\boldsymbol{X} \qquad \text{or} \qquad v(\boldsymbol{x},t) = v(\boldsymbol{\Psi}(\boldsymbol{x},t),t) = \frac{dR}{dt}\frac{\boldsymbol{x}}{R}.$$

The integrated interface velocity for a boundary portion $\partial\mathcal{C}_{\boldsymbol{x}}(t)$ is

$$\begin{aligned}\nu(t) &= \int_{\partial\mathcal{C}_{\boldsymbol{x}}(t)} \boldsymbol{v}(\boldsymbol{x},t)\cdot\boldsymbol{n}(\boldsymbol{x},t)\,ds \\ &= \int_{\partial\mathcal{C}_{\boldsymbol{x}}(t)} \frac{1}{R}\frac{dR}{dt}(\boldsymbol{x}\cdot\boldsymbol{n})\,ds.\end{aligned} \tag{2.42}$$

For a circumference the product $\boldsymbol{x}\cdot\boldsymbol{n}$ is equal to R since $\boldsymbol{x}$ is always locally aligned with $\boldsymbol{n}$. In a polar coordinate system one obtain $ds = R\,dRd\vartheta$, therefore Eq. (2.42) becomes

$$\nu(t) = \int_0^\theta R\frac{dR}{dt}\,d\vartheta = \theta R\frac{dR}{dt} \tag{2.43}$$

where θ is the angle span of the arc corresponding to $\partial\mathcal{C}_{\boldsymbol{x}}$.

From Eq. (2.43) the value of ν depends on both the values of R and dR/dt. If, for example, the radius is progressively reduced until a null area is obtained at the time T, the value of $\nu(T)$ is not necessarily equal to zero but depends on how quickly R goes to zero with respect to dR/dt as t approaches T. If the position of the nodes varies linearly in time, i.e. $R = R_0(T-t)/T$ with R_0 the initial radius, the Eq. (2.43) becomes $\nu(t) = -\theta R_0^2/T^2\,(1-t)$ which gives $\nu(T) = 0$. However if $\nu(T)$ is directly approximated using a backward Euler scheme, as done in Eq. (2.32), one obtains $\nu = -\theta R_0^2/(2T)$, where $\theta R_0^2/2$ is the area swept by the interface in the interval $[0,T)$. Eq. (2.43) therefore becomes

$$R\frac{dR}{dt} = -\frac{R_0^2}{2T}, \qquad \text{i.e.} \qquad R(t) = R_0\sqrt{1-\frac{t}{T}}.$$

When adopting a BDF scheme, for which the backward Euler is a special case, the underlying hypothesis is that during every timestep $V_{j\ell}(t)$ depends linearly on t. The motion of the points of the interface, however, is not linear in time but it depends on the shape of the interface.

2.6.2 ALE scheme with variable connectivity

When using grids undergoing edge-swapping the local connectivity varies with the time, i.e. $\mathcal{K}_{i,\neq} = \mathcal{K}_{i,\neq}(t)$, while the total number of points $\mathcal{K}$ does not change. If the ALE interpretation of the edge-swapping described in section 2.6.1 is adopted, Eq. (2.32) can

be written for grids with variable topology and constant number of nodes, namely

$$\begin{cases} \displaystyle\sum_{q=-1}^{p} a_q V_i^{n-q}\, \mathsf{u}_i^{n-q} = \Big[\Phi^{\partial}(\mathsf{u}_i,\nu_i,\hat{\boldsymbol{\xi}}_i,\xi_i)^{n+1} + \sum_{k\in\mathcal{K}_{i,\neq}^{n+1}} \Phi(\mathsf{u}_i,\mathsf{u}_k,\nu_{ik},\hat{\boldsymbol{\eta}}_{ik},\eta_{ik})^{n+1} \\ \qquad\qquad + \displaystyle\sum_{k\in\mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i,\mathsf{u}_k,\nu_{ik},\hat{\boldsymbol{\eta}}_{ik},0)^{n+1}\Big]\Delta t, & i\in\mathcal{K} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{ik}^{n-q} = \nu_{ik}^{n+1}\Delta t, & k\in\mathcal{K}_{i,\neq}^{[n-p,n+1]} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{\ell,\partial}^{n-q} = \nu_{\ell}^{n+1}\Delta t, & \ell\in\mathcal{K}_{\partial} \end{cases}$$

where the set $\mathcal{K}_{i,\neq}^{[n-p,n+1)}$ is the set of the finite volumes that do not share any point with $\mathcal{C}_i$ at the time t^{n+1}, i.e. the edge i-k is not part of the mesh at step $n+1$, but have shared some points with $\mathcal{C}_i$ at any time in the interval $[t^{n+1-p}, t^{n+1})$. This can be formalized as

$$\mathcal{K}_{i,\neq}^{[n-p,n+1)} : \{k\in\mathcal{K}, k\notin\mathcal{K}_{i,\neq}^{n+1} \quad \text{such that} \quad \nu_{ik}^{n+1}\neq 0\}.$$

In the case of a Forward Euler scheme, for example, $\mathcal{K}_{i,\neq}^{[n,n+1)}$ identifies the edges i-k that are swapped during the adaptation occurring in the time lapse between t^n and t^{n+1} plus the edges that are created and removed during the same time interval because of possible successive swapping. In fact, the computation of the area to go from the initial to the final configuration can be easily accomplished by summing up the effects of a sequence of simple collapse/expand operations. This choice is not mandatory, but simplifies the implementation of the algorithm. The set $\mathcal{K}_{i,\neq}^{[n-p,n+1]}$ is the totality of the edges, i.e. $\mathcal{K}_{i,\neq}^{[n-p,n+1]} = \mathcal{K}_{i,\neq}^{[n-p,n+1)} \cup \mathcal{K}_{i,\neq}^{n+1}$.

The previous relations show how the governing equations take into account the edge swapping contributions in terms of the DGCL simply as a continuous (in time) deformation of the cells associated to each vertex. The computation of contributions associated with each swap step represents a simple way to compute the total cell area variation and the associated fluxes from the old grid to the new one. In this way, the dynamic problem of a moving mesh with variable topology is kept within the framework of ALE formulation.

2.6.3 Domain node insertion

The procedure used to treat in a conservative way the edge-swapping shown in sections 2.6.1 and 2.6.2 is here extended to the case of element refinement. Fig. 2.18 illustrates a possible way of interpreting the node insertion operation as a continuous deformation of the finite volumes performed in the time interval $[t^n, t^{n+1})$. A three step approach similar to the one presented above is followed and it is described in terms of non-dimensional time $\tau\in[0,1)$.

$\tau = 0$**:** the initial condition is sketched in fig. 2.18(a).

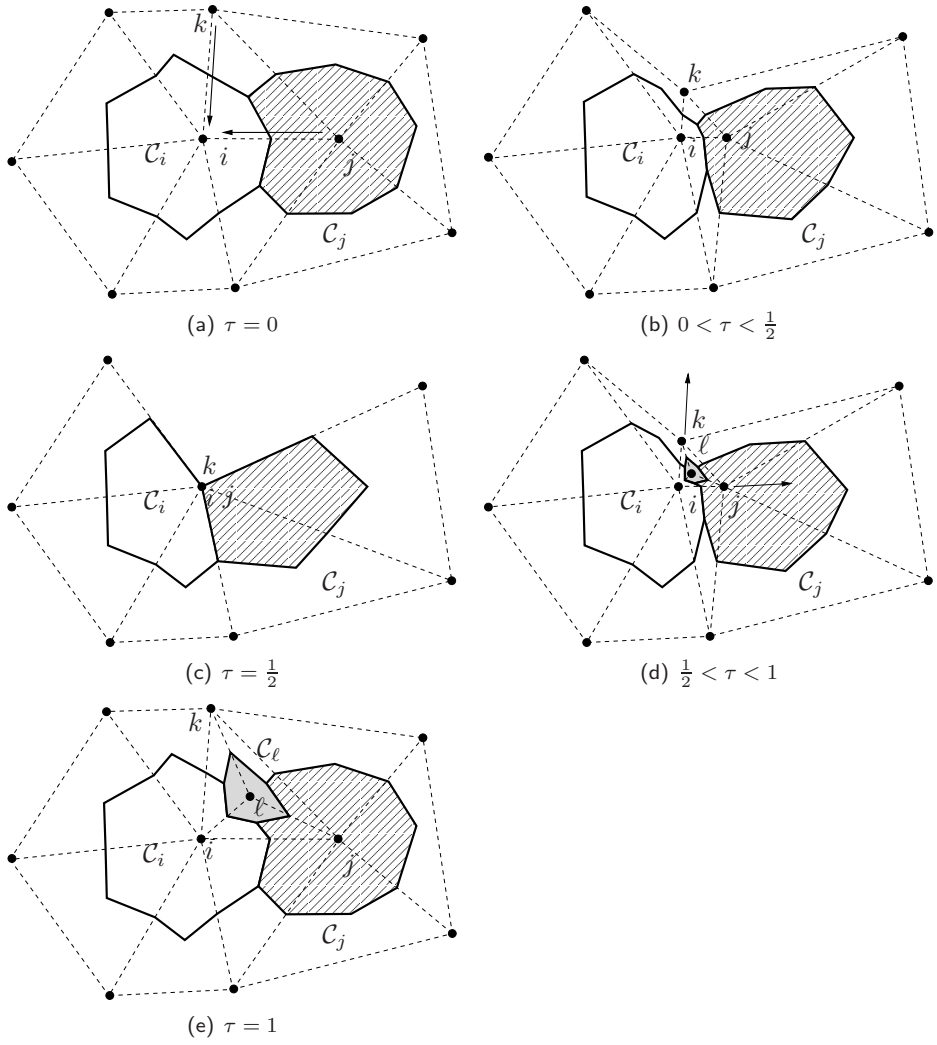


Fig. 2.18: Interpretation of the node insertion procedure as continuous finite volumes deformation in the non-dimensional time $\tau = (t - t^n)/(t^{n+1} - t^n)$. (a) Initial state. (b) The k-th and j-th nodes collapse over the i-th node, the interfaces of the cells $\mathcal{C}_j$ and $\mathcal{C}_k$ sweep a non null area. (c) Intermediate state, the element to be refined has entirely collapsed. Any change of connectivity, due to the refinement, produces no variation of area of the cells. (d) The k-th and j-th nodes return to their original locations. During the expansion step the interfaces of the cells $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_k$ sweep a non null area. (e) Final state.

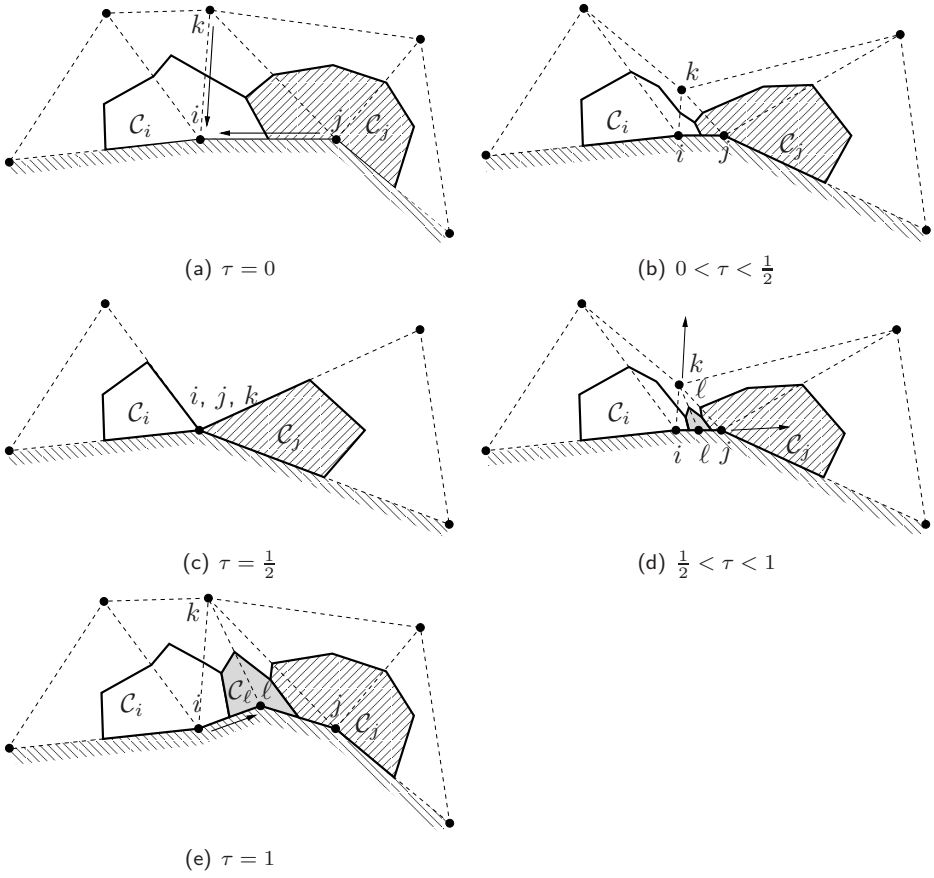


Fig. 2.19: Interpretation of the node insertion procedure along the boundary as continuous finite volumes deformation in the non-dimensional time $\tau = (t - t^n)/(t^{n+1} - t^n)$. (a) Initial state. (b) The k-th and j-th nodes collapse over the i-th node, the interfaces of the cells $\mathcal{C}_j$ and $\mathcal{C}_k$ and the boundary interfaces $\partial\mathcal{C}_i \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area. (c) Intermediate state, the element to be refined has entirely collapsed. Any change of connectivity, due to the refinement, produces no variation of area of the cells. (d) The k-th and j-th nodes return to their original locations. During the expansion step the interfaces of the cells $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_k$ and the boundary interfaces $\partial\mathcal{C}_\ell \cap \partial\Omega$, $\partial\mathcal{C}_i \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area. (e) Final state, the node $\boldsymbol{x}_\ell$ is positioned as prescribed by the continuous shape of the boundary.

$0 < \tau < 1/2$**:** the i, j, k triangle collapses over the node i. The portions of the cells that are contained inside the i, j, k triangle collapse over i. The choice of the point of collapse is arbitrary.

$\tau = 1/2$**:** the node ℓ is inserted. Since the portion of the cells contained inside the triangle i, j, k has a null area, this has no effects in terms of conservation of the areas.

$1/2 < \tau < 1$**:** the nodes i, j and k return to their original positions. The cell $\mathcal{C}_\ell$ and all the other pieces of cell contained inside the i, j, k triangle are expanded. The position of the node ℓ depends on the refinement pattern, e.g. the center off mass of the non-refined element.

$\tau = 0$**:** the final configuration is sketched in fig. 2.18(e). The finite volumes $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_\ell$ are constructed as prescribed by the dual mesh scheme of section 2.3.3.

As for the case of the edge-swapping, in the expansion and the collapse step the area swept by the interfaces is calculated as described in section 2.4.1, since the grid movement is a simple grid deformation with constant number of nodes and connectivity. The insertion step has no effects in terms of the satisfaction of the GCL since the involved cells have a null area, see fig. 2.18(c) and 2.19(c). As shown for the case of edge-swapping the overall area swept by the interfaces outside the triangle i, j, k is equal to zero and the net numerical flux is therefore null, indeed no changes of shape of those portion of cells occurs.

The governing equation of a new cell, i.e. introduced between step n and $n+1$, is

$$\begin{cases} a_{-1}\dfrac{V_i^{n+1}\,\mathsf{u}_i^{n+1}}{\Delta t} = \displaystyle\sum_{k \in \mathcal{K}_{i,\neq}^{n+1}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik})^{n+1}, & i \text{ is new} \\ \alpha_{-1}\dfrac{\Delta V_{ik}^{n+1}}{\Delta t} = \nu_{ik}^{n+1}, & k \in \mathcal{K}_{i,\neq}^{n+1} \end{cases} \tag{2.44}$$

since $V^{n-q} = 0$ and $\Delta V^{n-q} = 0$ with $q > 1$. The above equation highlights the fact that the knowledge of the solution at time levels previous than $n+1$ is not necessary, since the metrics are null, and u^{n+1} is computed by simply integrating the conservation equation for the overall system.

2.6.4 Boundary node insertion

Fig. 2.19 illustrates a possible way of interpreting the boundary node insertion operation as a continuous deformation of the finite volumes performed in the time interval $[t^n, t^{n+1})$. A three step approach similar to the one for the domain node is followed.

$\tau = 0$**:** the initial condition is sketched in fig. 2.19(a).

$0 < \tau < 1/2$**:** the i, j, k triangle collapses over the node i. The portions of the cells that are contained inside the i, j, k triangle collapse over i. The choice of the point of collapse is arbitrary.

$\tau = 1/2$**:** the node ℓ is inserted. Since the portion of the cells contained inside the triangle i, j, k has a null area, this has no effects in terms of conservation of the areas.

$1/2 < \tau < 1$**:** the nodes i, j and k return to their original positions. The cell $\mathcal{C}_\ell$ and all the other pieces of cell contained inside the i, j, k triangle are expanded. The position of the node ℓ depends on the refinement pattern, e.g. the center off mass of the non-refined element.

$\tau = 0$**:** the final configuration is sketched in fig. 2.19(e). The finite volumes $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_\ell$ are constructed as prescribed by the dual mesh scheme of section 2.3.3. The node $\boldsymbol{x}_\ell$ is positioned as prescribed by the continuous shape of the boundary, e.g. evaluating airfoil equation or a spline interpolation curve.

Again the area swept by the interfaces are calculated as shown in section 2.4.1 and the insertion step, i.e. $\tau = 1/2$, has no effects in terms of GCL satisfaction. The key difference with respect to the domain node case that the area swept by the boundary interfaces must be also taken into account. In particular $\nu^{n+1}_{\ell,\partial} = \alpha_{-1}\Delta A^{n+1}_{\ell,\partial}/\Delta t$ is the interface velocity given by the movement of the points of $\partial\mathcal{C}_\ell \cap \partial\Omega$ from the collapse point to the final configuration.

The governing equation of a new boundary cell, i.e. introduced between step n and $n+1$, is

$$\begin{cases} a_{-1}\dfrac{V_i^{n+1}\,\mathsf{u}_i^{n+1}}{\Delta t} = \displaystyle\sum_{k\in\mathcal{K}^{n+1}_{i,\neq}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik})^{n+1} \\ \qquad\qquad +\Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i)^{n+1}, & i \text{ is new} \\ \alpha_{-1}\dfrac{\Delta V^{n+1}_{ik}}{\Delta t} = \nu^{n+1}_{ik}, & k \in \mathcal{K}^{n+1}_{i,\neq} \\ \alpha_{-1}\dfrac{\Delta V^{n+1}_{\ell,\partial}}{\Delta t} = \nu^{n+1}_{\ell}, & \ell \in \mathcal{K}^{n+1}_{\partial} \end{cases}$$

where $\mathcal{K}^{n+1}_{\partial}$ is the set of nodes that discretize the boundary at the time t^{n+1}.

2.6.5 Domain node deletion

In fig. 2.20 is illustrated a possible ALE interpretation of the node deletion procedure for a domain cell. Again a three steps procedure is adopted

$\tau = 0$**:** the node i of fig. 2.20(a) is flagged for removal. The elements connecting i with the surrounding nodes, e.g. j, k and ℓ, must be deleted and topology must be reconstructed.

$0 < \tau < 1/2$**:** the bubble of nodes surrounding the node i collapses over the node $\boldsymbol{x}_i$. The choice of the collapse point is arbitrary. See fig. 2.20(b).

$\tau = 1/2$**:** the node i is removed and the connectivity is regenerated in an arbitrary fashion, fig. 2.20(c). Since the portion of the cells contained inside the bubble surrounding i has a null area, this has no effect in terms of conservation of the solution.

$1/2 < \tau < 0$**:** the nodes of the bubble return to their original positions. The choice of new connectivity is arbitrary, in the present case the nodes of the bubble are connected with an edge to the node k.

$\tau = 1$**:** the final configuration is shown in fig. 2.20(e). The optimality in terms quality of the mesh can restored performing an edge-swapping procedure.

Since the number of nodes and the connectivity is constant, the area swept by the interfaces during the expansion and the collapse is calculated as prescribed in section 2.4.1. The node deletion, with the entailed connectivity destruction and reconstruction, has no effect on the solution since the area of the involved cells is null. The swept areas so calculated allows to compute ν_{ik}^{n+1} and the related ALE fluxes for the interfaces that are removed.

The governing equation for a deleted domain cell, i.e. not considering the boundary terms, is

$$\begin{cases} \displaystyle\sum_{q=0}^{p} a_q V_i^{n-q}\, \mathsf{u}_i^{n-q} = \sum_{k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, 0)^{n+1} \Delta t, & i \text{ is removed} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{i,ik}^{n-q} = \nu_{ik}^{n+1}\, \Delta t, & k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)} \end{cases} \tag{2.45}$$

where the left hand side of the first equation does not depend on u_i^{n+1}, since $V_i^{n+1} = 0$, note that the summation starts for $q = 0$. The right hand side is given by the ALE contribution to the fluxes, e.g. second term of Eq. (2.20), of the removed edges and it does depend on the solution evaluated at t^{n+1}. As removed edges are associated to additional ALE contribution to the fluxes, removed cells are associated to additional ALE equations.

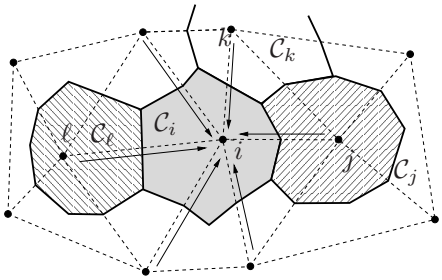
If the governing equations are integrated adopting an implicit scheme, as shown in the system (2.32) or (2.45), then the computation of the value of u on each removed cell is necessary to determine the solution on all remaining mesh nodes at the time level $n+1$ in a conservative manner. Indeed, given a removed cell $\mathcal{C}_i$ previously adjacent to the cell $\mathcal{C}_k$, the numerical fluxes across $\partial\mathcal{C}_{ik}$ are function of u_i^{n+1}, u_k^{n+1} and ν_{ik}^{n+1}, as given by Eq. (2.45). It is worth noting that, since the edges that are connected with a deleted node must be removed as well, the IVC condition of Eq. (2.45) is defined for every interface that is part of the mesh from t^{n-p} to t^{n+1} but is no longer part of the mesh at t^{n+1}.

Since the time derivative is known and $\eta_{ik}^{n+1} = 0$, Eq. (2.45) is an algebraic equation. Albeit the knowledge of the value of the solution on a removed node seems to be useless, as it is no longer part of the mesh, it is necessary indeed to balance the ALE fluxes exchanged with the surrounding cells in a conservative manner. Eq. (2.45) becomes a trivial identity only when $\nu_{ik}^{n+1} = 0,\ \forall k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)}$. Thus $p+1$ time steps after the node removal the equation is erased from the system and the solution on the removed cell is no longer computed.

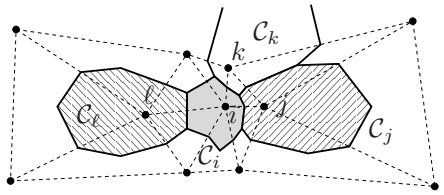
2.6.6 Boundary node deletion

In fig. 2.21 is illustrated a possible ALE interpretation of the node deletion procedure for a domain cell. Again a three steps procedure is adopted

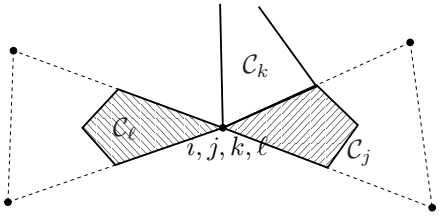
$\tau = 0$**:** the boundary node i of fig. 2.21(a) is flagged for removal. The elements connecting i with the surrounding nodes, e.g. j, k and ℓ, must be deleted and



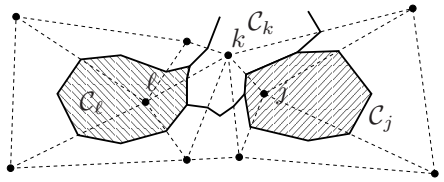
(a) $\tau = 0$



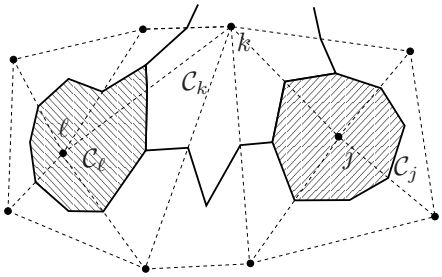
(b) $0 < \tau < \frac{1}{2}$



(c) $\tau = \frac{1}{2}$

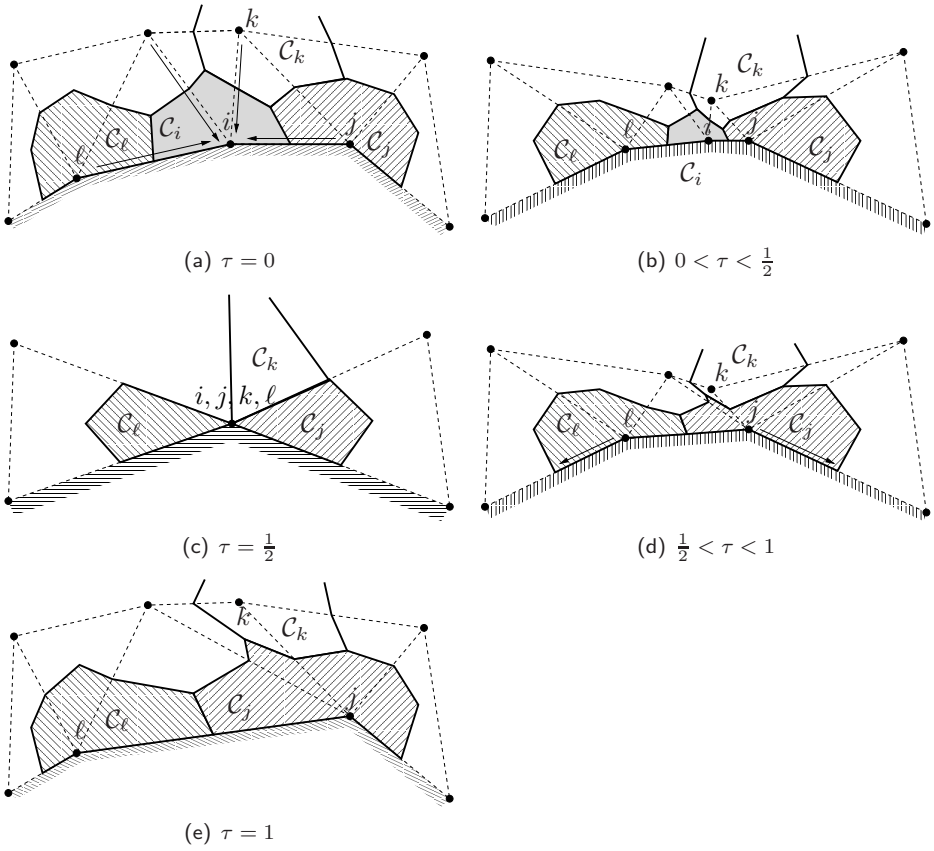


(d) $\frac{1}{2} < \tau < 1$



(e) $\tau = 1$

Fig. 2.20: Interpretation of the node deletion procedure as continuous finite volumes deformation in the non-dimensional time $\tau = (t - t^n)/(t^{n+1} - t^n)$. 2.20(a) Initial state. 2.20(b) The nodes surrounding the node to be removed, i.e. i, collapses on one location. The interfaces of the cells surrounding $\mathcal{C}_i$, e.g. $\mathcal{C}_j$, $\mathcal{C}_k$, and $\mathcal{C}_\ell$, sweep a non null area. 2.20(c) Intermediate state, bubble of nodes surrounding the i-th node has entirely collapsed. No variation in area of the cells is produced by the node deletion and the successive reconstruction of the local connectivity. 2.20(d) The nodes of the bubble return to their original locations. During the expansion step the interfaces of the cells, e.g. $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_k$ sweep a non null area. 2.20(e) Final state, the choice of the new connectivity is arbitrary.



(a) $\tau = 0$ (b) $0 < \tau < \frac{1}{2}$

(c) $\tau = \frac{1}{2}$ (d) $\frac{1}{2} < \tau < 1$

(e) $\tau = 1$

Fig. 2.21: Interpretation of the boundary node deletion procedure as continuous finite volumes deformation in the non-dimensional time $\tau = (t - t^n)/(t^{n+1} - t^n)$. 2.21(a) Initial state. 2.21(b) The nodes surrounding the node to be removed, i.e. i, collapses on one location. The interfaces of the cells surrounding $\mathcal{C}_i$, e.g. $\mathcal{C}_j$, $\mathcal{C}_k$, and $\mathcal{C}_\ell$ and the boundary interfaces $\partial\mathcal{C}_i \cap \partial\Omega$, $\partial\mathcal{C}_\ell \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area. 2.21(c) Intermediate state, bubble of nodes surrounding the i-th node has entirely collapsed. No variation in area of the cells is produced by the node deletion and the successive reconstruction of the local connectivity. 2.21(d) The nodes of the bubble return to their original locations. During the expansion step the interfaces of the cells, e.g. $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_k$ sweep a non null area. Moreover the boundary interfaces $\partial\mathcal{C}_\ell \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area. 2.21(e) Final state, the choice of the new connectivity is arbitrary.

reconstructed.

$0 < \tau < 1/2$**:** the bubble of nodes surrounding the node i collapses over the node $\boldsymbol{x}_i$. See fig. 2.21(b). The collapse point is chosen arbitrary, e.g. $\boldsymbol{x}_i(\tau = 0)$. The boundary interfaces $\partial\mathcal{C}_i \cap \partial\Omega$, $\partial\mathcal{C}_\ell \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area, i.e. $\Delta A_{i,\partial} \neq 0$, $\Delta A_{\ell,\partial} \neq 0$ and $\Delta A_{j,\partial} \neq 0$ respectively.

$\tau = 1/2$**:** the node i is removed and the connectivity is regenerated in an arbitrary fashion, fig. 2.21(c). Since the portion of the cells contained inside the bubble surrounding i has a null area, this has no effect in terms of conservation of the solution.

$1/2 < \tau < 0$**:** the nodes of the bubble return to their original positions. The choice of new connectivity is arbitrary, in the present case the nodes of the bubble are connected with an edge to the node k. The boundary interfaces $\partial\mathcal{C}_\ell \cap \partial\Omega$ and $\partial\mathcal{C}_j \cap \partial\Omega$ sweep a non null area, i.e. $\Delta A_{\ell,\partial} \neq 0$ and $\Delta A_{j,\partial} \neq 0$ respectively.

$\tau = 1$**:** the final configuration is shown in fig. 2.21(e). The optimality in terms quality of the mesh can restored performing an edge-swapping procedure.

Again the area swept by the interfaces during the expansion and the collapse is calculated as prescribed in section 2.4.1 and the node deletion, with the entailed connectivity destruction and reconstruction, has no effect in terms of ΔA. The so calculated swept area allows to compute ν_{ik}^{n+1} and the related ALE fluxes for the interfaces that are removed.

The governing equation for a deleted boundary cell is

$$\begin{cases} \displaystyle\sum_{q=-1}^{p} a_q \frac{V_i^{n-q}\,\mathsf{u}_i^{n-q}}{\Delta t} = \sum_{k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, 0)^{n+1} \\ \qquad\qquad + \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, 0) \qquad\qquad i \text{ is removed} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{i,ik}^{n-q} = \nu_{ik}^{n+1}\,\Delta t \qquad\qquad k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)} \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{\ell,\partial}^{n-q} = \nu_{\ell}^{n+1}\,\Delta t \qquad\qquad \ell \in \mathcal{K}_{\partial}^{[n-p,n+1)} \end{cases} \tag{2.46}$$

The above system is very close to the domain counterpart, i.e. system (2.45), but the boundary fluxes are also taken into account and the IVC condition is defined for the velocity of every removed boundary interface as well, i.e. ν_ℓ^{n+1}. This can be formalized as

$$\mathcal{K}_{\partial}^{[n-p,n+1)} : \{\ell \notin \mathcal{K}_{\partial}^{n+1} \quad \text{such that} \quad \nu_\ell^{n+1} \neq 0\}.$$

2.6.7 ALE scheme for grids with variable connectivity and number of nodes

Introducing $\mathcal{K}^{n+1} = \mathcal{K}(t^{n+1})$ as the set of nodes of the triangulation of Ω at the time t^{n+1} and $N_{\mathcal{K}}^{n+1} = \dim(\mathcal{K}^{n+1})$ as the total number of nodes. If N_{A} nodes are inserted

into the grid, $N_{\rm A} \times 4$ new conservation equations and $N_{\rm A}$ GCL constraints must be satisfied, i.e. system (2.44) must be included in system (2.32).

The governing equation for the deleted nodes is taken into account introducing the set of the removed cells that are associated to at least one interface with a velocity that is different from zero, i.e.

$$\mathcal{K}^{[n-p,n+1)} = \{i \notin \mathcal{K}^{n+1} : \exists k \text{ such that } \nu_{ik}^{n+1} \neq 0\}.$$

Note that, adopting a BDF scheme of order $p+1$, $\mathcal{K}^{[n-p,n+1)}$ is the set of the nodes removed not before than t^{n-p}. Moreover, introducing $\mathcal{K}^{[n-p,n+1]} = \mathcal{K}^{n+1} \cup \mathcal{K}^{[n-p,n+1)}$ as the set of all nodes, it is possible to recast system (2.32) for the case of grids with variable number of nodes as

$$\begin{cases} \displaystyle\sum_{q=-1}^{p} a_q \frac{V_i^{n-q}\,\mathsf{u}_i^{n-q}}{\Delta t} = \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i) + \sum_{k \in \mathcal{K}_{i,\neq}^{n+1}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik},)^{n+1} \\ \qquad\qquad + \displaystyle\sum_{k \in \mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, 0)^{n+1}, \qquad \left(i \in \mathcal{K}^{n+1}\right) \\ \displaystyle\sum_{q=-1}^{p} a_q \frac{V_j^{n-q}\,\mathsf{u}_j^{n-q}}{\Delta t} = \sum_{k \in \mathcal{K}_{j,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_j, \mathsf{u}_k, \nu_{jk}, \hat{\boldsymbol{\eta}}_{jk}, 0)^{n+1} \\ \qquad\qquad + \Phi^{\partial}(\mathsf{u}_j, \nu_j, \hat{\boldsymbol{\xi}}_j, 0), \qquad \left(j \in \mathcal{K}^{[n-p,n+1)}\right) \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{i,ik}^{n-q} = \nu_{ik}^{n+1}\,\Delta t, \qquad \left(k \in \mathcal{K}_{i,\neq}^{[n-p,n+1]}\right) \\ \displaystyle\sum_{q=-1}^{p-1} \alpha_q \Delta V_{\ell,\partial}^{n-q} = \nu_{\ell}^{n+1}\,\Delta t, \qquad \left(\ell \in \mathcal{K}_{\partial}^{[n-p,n+1]}\right) \end{cases} \tag{2.47}$$

Chapter 3

Mesh update strategy

In this chapter the strategies adopted to update the computational mesh are outlined. The goal is to permit the movement of the boundaries, while maintaining high level of grid quality, and change locally the distribution of the nodes to increase or decrease the resolution. To this purpose a suitable mix of mesh deformation and local topology modification techniques is implemented; all the adopted methods are very standard with few modifications. In section 3.1 the mesh deformation algorithm is presented that allows to reduce the quality losses due to the movement of the boundaries. In section 3.2 the edge-swapping technique is briefly described. In section 3.3 the refinement/derefinement procedures are outlined, with particular focus on the geometry-driven adaptation strategy (section 3.3.1) and the solution-driven adaptation strategy (section 3.3.2). The edge-swapping, node insertion and deletion techniques have been already presented in section 2.6.3, 2.6.1 and 2.6.5 in the context of the ALE formulation; in the present chapter, however, the focus will be on how these methods are used to adapt the grid.

3.1 Mesh deformation

In the present section, the mesh deformation strategy is briefly described. Mesh deformation is required for the boundary of the new (deformed) mesh to be conformal to the new boundaries of the domain. At the same time, the overall quality of the mesh elements in the inner domain must be preserved to reduce numerical errors.

The mesh movement is performed in two steps. First, the displacement of each boundary node of the fluid mesh is obtained from a given movement law; then, the position of the inner nodes is modified accordingly.

The displacement of internal nodes can be obtained using different strategies. These can be gathered in two classes: interpolation methods [30, 155], mainly used for structured meshes, and those based on some form of elastic analogy, which are more suitable for unstructured meshes. Batina [15] introduced the elastic analogy by representing each side of the grid as a spring with a nonlinear stiffness proportional to the edge length. To avoid the occurrence of invalid elements with negative areas, Degand and Farhat [45] introduced additional torsional springs at each vertex. Given its complexity, the mesh

movement step may require a non negligible computational effort to the point that it may become one of the most time-consuming tasks in the computation [45].

For the reasons described above, the envisaged deformation scheme should be designed to fulfill the following requirements:

- Robustness: the scheme must handle significant boundary displacements and be capable of producing valid grids (all elements with positive area) with an acceptable quality especially in the areas where low numerical errors are sought for, i.e. near the wall boundaries.
- Computational efficiency.
- Easy of use; the user intervention on the algorithm is to be minimal.

The grid deformation algorithm presented here extends to idea of the elastic analogy by representing each element as a deformable body and moves from the discussion presented in [18]. Differently from the spring analogy, such a choice avoids element entanglement also in the case of large deformations. To reduce the computational burden, a simple linear constitutive law is used, namely, in two spatial dimensions one has $\boldsymbol{\sigma} = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}^T$, and $\boldsymbol{\epsilon} = \{\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}\}^T$, as $\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\epsilon}$, where the $\boldsymbol{D}$ matrix is equal to

$$\boldsymbol{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix}.$$

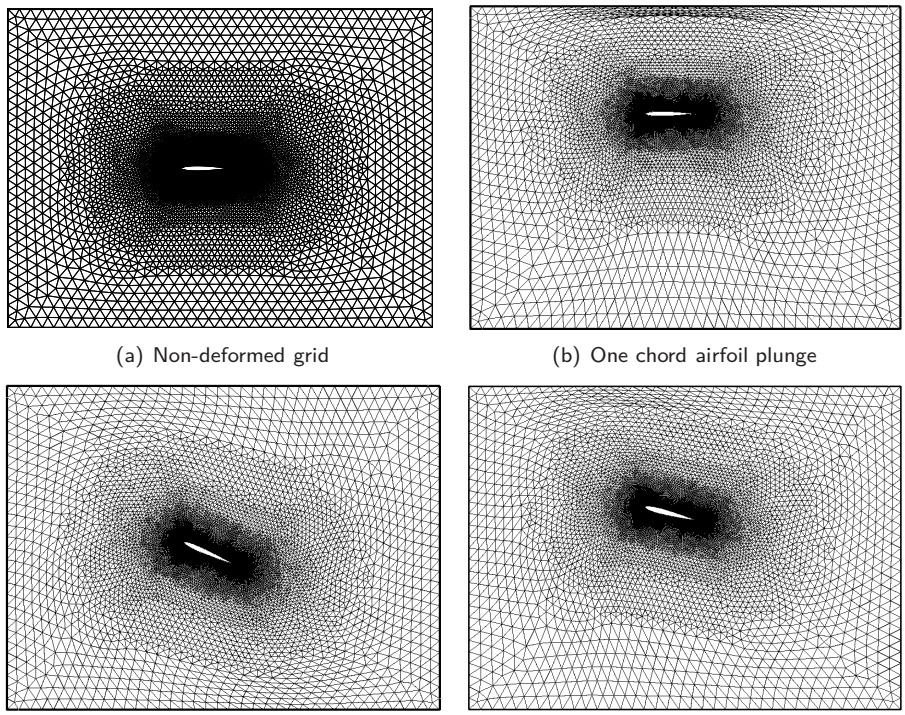
The correct grid deformation is achieved adopting a local Young modulus proportional to the minimal dimension of each element following a simple law

$$E_e = \frac{1}{\min\limits_{i,k \in \mathcal{K}_e} \|\boldsymbol{x}_i - \boldsymbol{x}_k\|^\beta}, \tag{3.1}$$

where $\mathcal{K}_e$ is the set of all nodes belonging to the e-th element. In this way the small elements close to wall boundaries are more stiff, so they tend to move rigidly with the walls, leaving the burden to absorb the global deformations on the larger elements, usually located far from the boundaries. The coefficient β can be used to control the mesh deformation behavior, increasing the stiffness ratio between small and large elements. A Poisson coefficient $\nu \in [0; 0.35]$ is chosen in order to avoid bad numerical conditioning of the problem. The mesh deformation problem is then solved by means of a standard finite element approach. The wall boundary displacements are imposed simply as Dirichlet boundary conditions for the elastic mesh problem. Further improvements can be obtained through the adoption of anisotropic continua.

The effectiveness of the proposed strategy is shown in Fig. 3.1(b), where a two-dimensional unstructured mesh around the NACA 0012 airfoil is deformed to adapt it to a one chord plunge. The variable stiffness produces an almost rigid displacement for the small triangles near the airfoil walls, with a visible distortion only near the grid external boundary, where larger numerical errors may be acceptable. A similar behavior is found for the airfoil pitch, Fig. 3.1(c).

Furthermore, the linearity of the equations describing the mesh movement problems allows for a further reduction of the computational time, since the global mesh deformation can be represented as a superposition of basic deformed grids computed in



(a) Non-deformed grid (b) One chord airfoil plunge

(c) 20 deg. airfoil pitch (d) Pitch and plunge linear combination

Fig. 3.1: Grid deformation for a NACA 0012 airfoil.

advance. An example is shown in Fig. 3.1(d), where the two movement of pitch and plunge are linearly combined. Large saving are obtained when three dimensional cases are solved, using as basic elements for the superposition the deformed meshes associated with structural normal modes [23]. Of course, the superposition approach should be applied only when small structural displacements are considered, which is usually the case when aircraft aeroelastic stability is under investigation.

3.1.1 Grid Smoothing

In order to further improve the grid quality, grid regularization is performed [178]. When barycentric smoothing is applied the new position of the i-th node is given by

$$\boldsymbol{x}_i^{n+1} = (1 - k_s)\boldsymbol{x}_i^n + \frac{k_s}{\dim(\mathcal{K}_{i,\neq})} \sum_{k \in \mathcal{K}_{i,\neq}} \boldsymbol{x}_k^n \,, \tag{3.2}$$

where $k_s = 0 \div 1$ is a relaxation parameter. A well known issue of non weighted smoothing techniques is the fact that when convergence is achieved, i.e. after several application of Eq. (3.2), the final mesh features a uniformly distributed spacing. For such reason the use of mesh smoothing should be carefully dosed. In the present work three cycles of mesh smoothing are applied and $k_R = 0.5$. In problems where the size of the elements close to the walls is critical, e.g. in the BVI problem of section 5.3.1, Eq. (3.2)

is applied only to the vertices of elements that undergo any topology a modification in the topology.

3.2 Edge-swapping

If large displacements of the boundaries occur, the application of the above mesh deformation technique may possibly result in distorted and tangled elements, leading to large numerical errors or even to complete failure if negative area elements appear. Indeed, in these cases the Jacobian J of the transformation is negative and condition (2.5) on the regularity of the ALE mapping $\boldsymbol{\psi}$ is not fulfilled. For large displacements of the mesh nodes it is therefore mandatory to alter the topology of the mesh, i.e. the element-node connectivity, or, equivalently, the definition of the mesh edges. The alteration of the mesh topology can be performed in a local or a global fashion [178]. In the present work, a well-known local topology alteration technique—the so-called edge-swap technique [64]—is adopted, in which the total number of grid nodes is preserved. The basic idea is to change the topological structure by local reconnection, without the addition or removal of vertexes. This is accomplished by altering the connectivity of a given couple of triangular elements by deleting the edge connecting the two vertexes shared by the two elements and by adding a new edge connecting the other two vertexes (see fig. 3.2). The ability of the swap operator in improving the quality of triangular or tetrahedral meshes is well assessed [64].

To decide whether an edge must be swapped it is necessary to adopt a quality measure for the elements. Following the analysis presented by Shewchuk [165], the key factor is the condition number of the stiffness matrix associated with the fluid problem, while the element size distribution is related to the need to minimize the error bound and it does not have to be fixed by the swapping procedure. Therefore, a scale-invariant quality measure Q has been chosen among those presented in [165] for two-dimensional grids, defined as

$$Q_e = \frac{V_e}{\sum_{i\in\mathcal{K}_e} \ell_i^2 + \sqrt{(\sum_{i\in\mathcal{K}_e} \ell_i{}^2)^2 - 48\, V_e^2}}, \tag{3.3}$$

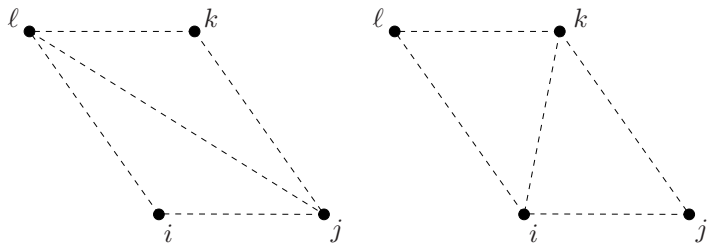
where V_e is the element are and ℓ_i the i-th edge length. For example, the element quits its maximum value for the equilateral triangle with the third vertex in $(1/2, \sqrt{3}/2)$, i.e. $Q_{\text{ideal}} = 0.1443375$. Measure Q has sign, which means that it is negative if the area is negative, so it is possible to recognize the occurrence of inverted elements.

Deformation and swapping

To obtain large displacement of the boundaries edge swapping has been used also in connection with deforming meshes [11]. With reference to figure fig. 3.3, edge-swapping is beneficial in this case in that it allows two vertexes that move in the opposite direction to disconnect to avoid excessive stretching of the element. As a result the elements seems to *flow* in the domain allowing bodies to move freely into the computational grid.

The complete mesh movement procedure is organized as described in the following pseudo-code:

```
Displacement of boundary nodes
```



(a) Reference Configuration (b) Swapped Configuration

Fig. 3.2: Application of the edge-swapping technique to the quadrangle of vertices i, j, k and ℓ.

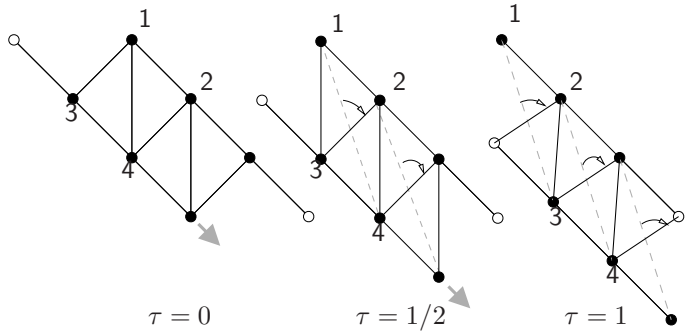


Fig. 3.3: Edge swapping applied to a deforming grid. A driver force cause the translation of the lower row of vertices; the dashed edge are flipped transforming stretched elements in more regular ones. At the end of the process the first vertex on the left of lower row is connected by a new edge with the last edge on right of the upper row.

Grid deformation

FOR each edge

Compute quality of the elements, $Q_{reference}$

Compute quality of the swapped-edge elements, $Q_{swapped}$

IF(min($Q_{swapped}$) > min($Q_{reference}$))

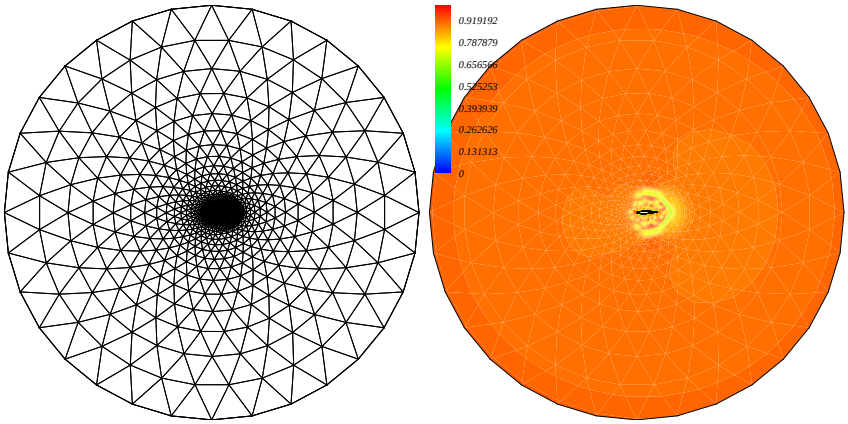
Apply swap

END

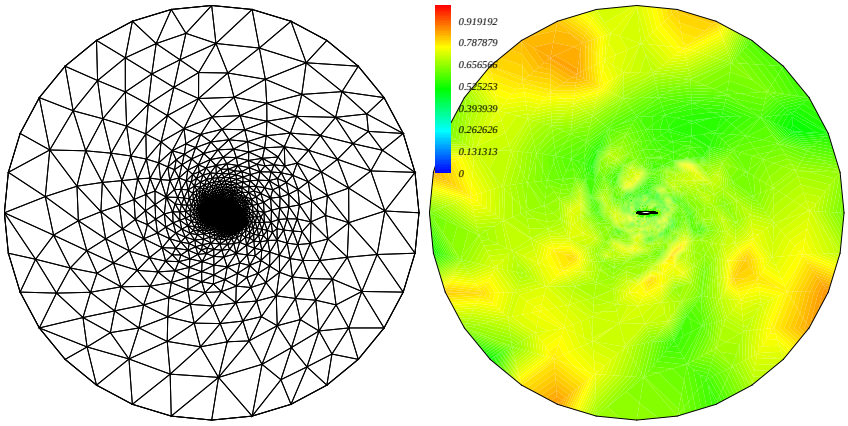
END

The cycle on grid edges can be applied a fixed number of times (one or more), or it can be repeated up to the case where no additional edge swap is performed.

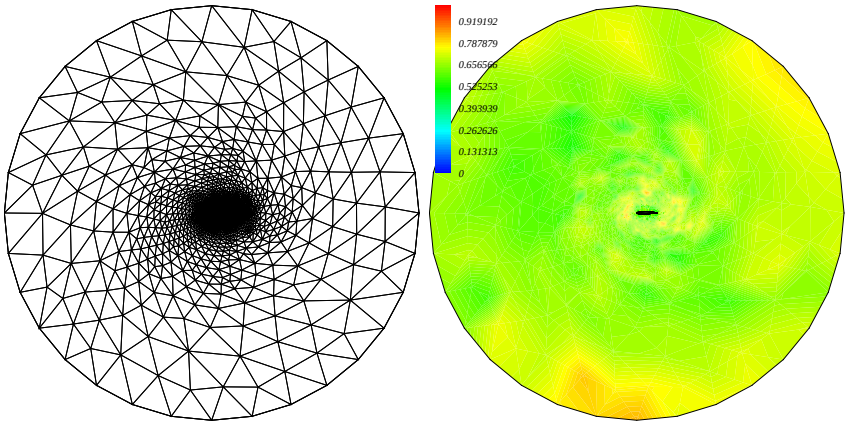
Fig. 3.4 and fig. 3.5 show the application of the present mesh update procedure to a 360° rotation and back of a RAE 2822 airfoil, located in the center of a circular domain with radius of 10 chords, about the trailing edge. The external boundary fixed. This is a typical grid arrangements to compute e.g the lift-angle-of-attack function for an oscillating airfoil. A sinusoidal law is imposed, i.e. $\theta(t) = 180°[1 - \cos(2\pi t)]$. In fig. 3.4



(a) $\theta = 0$

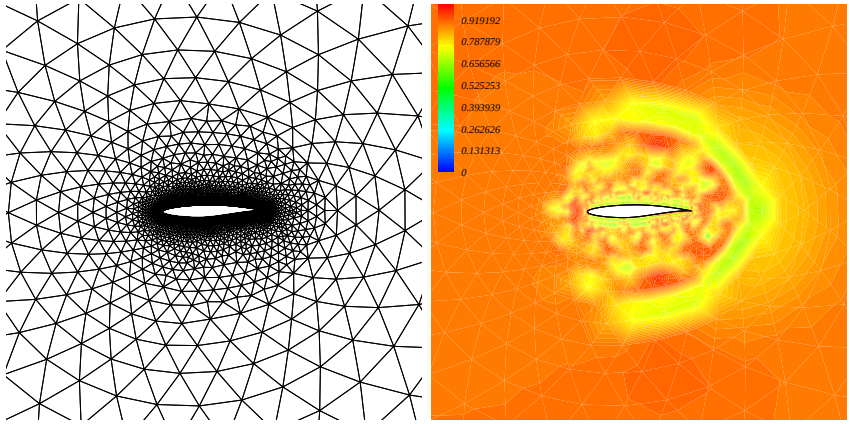


(b) $\theta = 360$

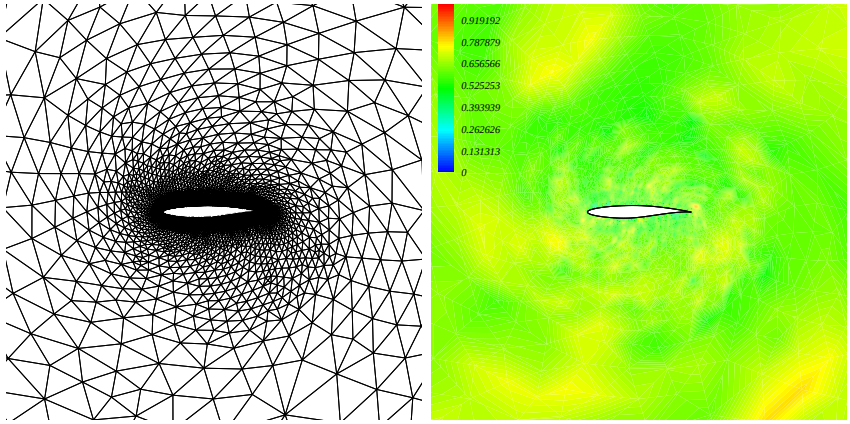


(c) $\theta = 0$

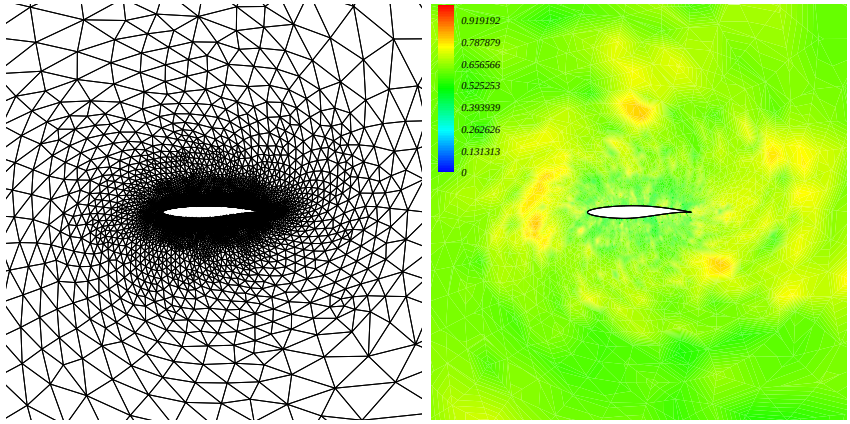
Fig. 3.4: Grid and quality contour for different values of θ angle. $N_{step} = 80$.



(a) $\theta = 0$

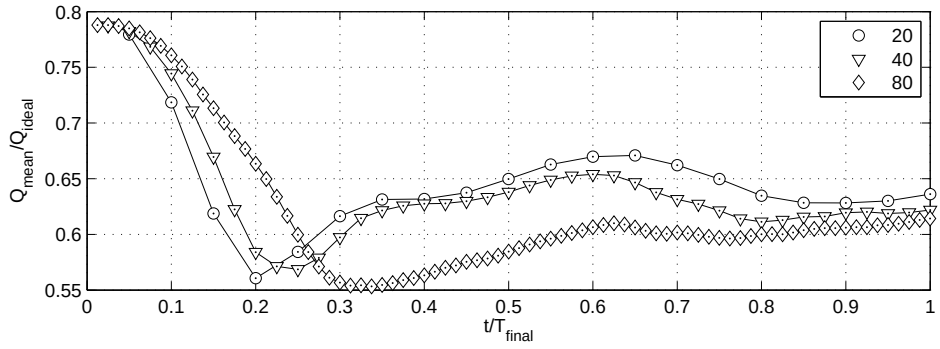


(b) $\theta = 360$

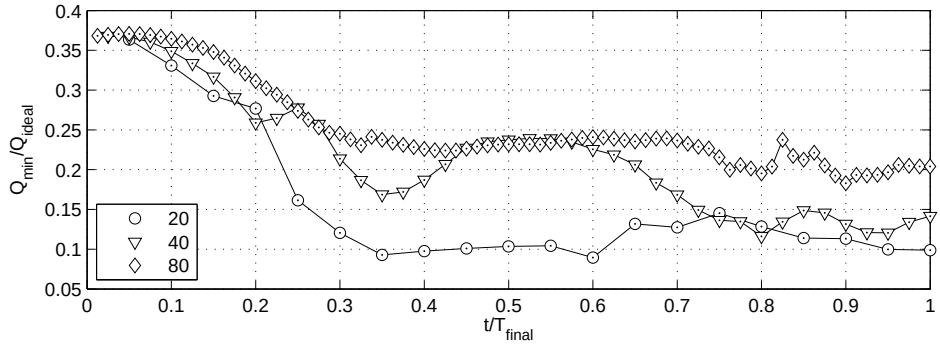


(c) $\theta = 0$

Fig. 3.5: Grid and quality contour close-up on for different values of θ angle.



(a) Mean



(b) Min

Fig. 3.6: Average and minimum quality for different numbers of steps.

and fig. 3.5 (left) the grid is plotted for three different values of the angle theta, i.e. for the initial position, the maximum rotation and the final position. The smaller and therefore more rigid elements, which are located near the airfoil, rigidly move with the body, thus preserving the mesh spacing close to the airfoil. The edge swapping allows the inner rigid core of elements to slide inside the external grid. In fig. 3.4 (right) and fig. 3.5 (right) the corresponding local normalized grid quality, i.e. Q/Q_{ideal}. The initial grid quality is in the interval $0.7 < Q/Q_{\mathrm{ideal}} < 0.9$;it is then progressively reduced to $0.5 < Q/Q_{\mathrm{ideal}} < 0.7$ and tend to be uniformly distributed among the elements. In fig. 3.6 the minimum and the average value of quality are shown as a function of the time for different values of $\Delta t = 1/N_{step}$. When the rotation is completed, the mean quality curves tend to show a similar trend, while the minimum quality increases decreasing the time step since more swapping and smoothing operations are performed over a given time interval.

3.3 Node insertion and deletion

In the present work nodes are inserted and removed from the grid to locally control the grid spacing. Several type of element refinement techniques can be found in literature,

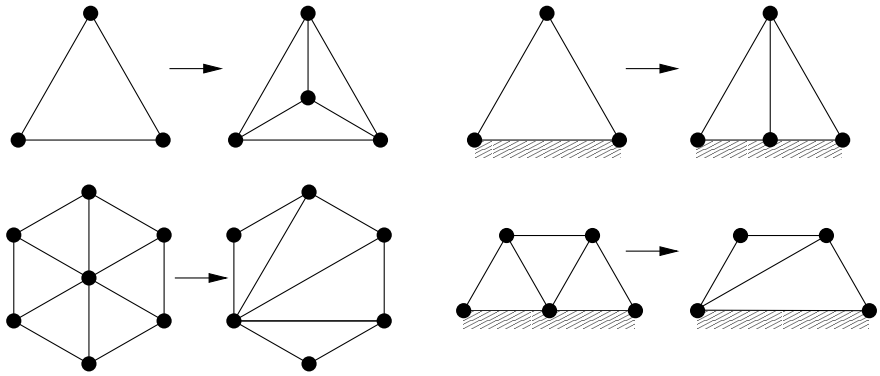


Fig. 3.7: Top: refinement pattern by node insertion in the center of mass of an existing element for a domain (top-left) element and boundary (top-right) element. Bottom: Derefinement pattern by node deletion for a domain (bottom-left) element and boundary (bottom-right) element.

even for simple triangular meshes [63]. In the present work a simple barycentric approach is followed to perform node insertion for domain nodes, see fig. 3.7 (left-top). The refinement of boundary elements is performed computing the exact position of the boundary nodes evaluating the spline curve describing the boundary, see fig. 3.7 (right-top). Domain and boundary nodes removal are performed as shown in fig. 3.7 (bottom), where the local connectivity is reconstructed in order to maximize the local quality. In order to guarantee the highest element quality through the mesh the node insertion/deletion procedures are followed by a edge-swapping one.

3.3.1 Geometry driven adaptation

The goal of mesh refinement and derefinement is to control the quality and the dimension of the elements to minimize the numerical errors. The imposed/desired mesh spacing is described by a function $\mathcal{A}(\boldsymbol{x}, t) \in \mathbb{R}$, defined over computational domain Ω for a given time t.

In the present section the size distribution function is prescribed as proportional to the distance from the boundaries, indeed in most of the cases of aerodynamic interest the smaller elements are gathered close to the solid walls. Let V_e be a suitable measure of the e-th element size, e.g. the area, and let $\mathcal{A}(\boldsymbol{x}, t)$ be a generic function that put in relation a desired triangle size and an element of Ω.

A parabolic description of the dimension distribution which is function of the distance from the b-th boundary

$$\mathcal{A}_b(\boldsymbol{x}, t) = c_{1,b} + c_{2,b} \left(\min_{\boldsymbol{x}_b \in \partial\Omega_b(t)} |\boldsymbol{x} - \boldsymbol{x}_b| \right) + c_{3,b} \left(\min_{\boldsymbol{x}_b \in \partial\Omega_b(t)} |\boldsymbol{x} - \boldsymbol{x}_b| \right)^2 \tag{3.4}$$

where $c_{1,b}$, $c_{2,b}$ and $c_{3,b}$ are coefficients chosen by the user and $\partial\Omega_b$ is the set of points that belongs to the b-th boundary. The imposed size function is therefore chosen as

$$\mathcal{A}(\boldsymbol{x}, t) = \min_{b \in \mathcal{B}} \mathcal{A}_b(\boldsymbol{x}, t)\,.$$

Thus an element is flagged for refinement if $V_e < (1-\varepsilon)\mathcal{A}(\boldsymbol{x}_e)$, where $\boldsymbol{x}_e$ is the center of mass of the e-th element and $\varepsilon = 0.1 \div 0.3$ is a suitable tolerance; similarly if

$$\frac{1}{\dim(\mathcal{E}_i)} \sum_{e \in \mathcal{E}_i} V_e > (1+\varepsilon)\mathcal{A}(\boldsymbol{x}_i)$$

the node is flagged for removal.

3.3.2 Solution driven adaptation

In this section the strategy used to adapt the grid to capture the local feature of the flow-field is described. The purpose of this technique is to locally change the grid spacing in order to distribute the numerical error in the whole computational domain according to the principle of error equidistribution. In this way, if a computational region has a numerical error higher than the average, it will be refined using a technique of node insertion; on the other hand, regions with a lower numerical error will be derefined thanks to a deletion node procedure.

The calculation of the numerical error requires the evaluation of a error estimator E on the whole domain. Depending on the applications, error estimators can be either functions of flow gradients or undivided differences [10, 117, 148, 193], or functions of the Hessian matrix $\mathcal{H}$ [37, 80, 146, 193, 200, 195] of a convenient sensor variable s which is representative of the flow features and whose choice depends on the physical problem.

Gradient

In most applications, error estimators are either functions of flow gradients or undivided differences [10, 117, 148, 193] of a convenient quantity s i.e. $E_i = V_i||\boldsymbol{\nabla} s_i||$, where $||\ \cdot\ ||$ indicates the Euclidean norm. Resorting to the nodepair based finite volume discretization of the computation domain the gradient can be computed as

$$\boldsymbol{\nabla} s_i = \frac{1}{V_i}\left[s_i \boldsymbol{\xi}_i + \sum_{k \in \mathcal{K}_{i,\neq}} \frac{s_i + s_k}{2} \boldsymbol{\eta}_{ik} \right] \tag{3.5}$$

A better approximation of the discrete gradient vector $\boldsymbol{\nabla} s$ can be computed using a finite-element approximation within the node-pair representation [163, 63]. The value of the gradient at the center of mass of an element e is given by the average of the value of the gradient at its vertices. This correspond to a piece-wise linear approximation of the gradient, which is second order accurate.

Vorticity

In many cases of aeronautical interest it is desirable to have higher resolution in the proximities of shocks or vortices. In such cases the vorticity of the solution can be taken as a sensor of the error, i.e. $E_i = |\boldsymbol{\omega}_i|$, where $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{m}/\rho$ is the vorticity vector evaluated at $\boldsymbol{x}_i$. In the nodepair based finite-volume framework the rotor is computed as

$$\boldsymbol{\omega}_i = \frac{1}{V_i}\left[\boldsymbol{\xi}_i \times \frac{\boldsymbol{m}_i}{\rho_i} + \frac{1}{2} \sum_{k \in \mathcal{K}_{i,\neq}} \boldsymbol{\eta}_{ik} \times \left(\frac{\boldsymbol{m}_i}{\rho_i} + \frac{\boldsymbol{m}_k}{\rho_k} \right) \right]. \tag{3.6}$$

The above definition of the vorticity is correct only in three dimensions, i.e. $\boldsymbol{m} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$. In 2D the vorticity is a scalar equal to the component of $\boldsymbol{\omega}$ aligned to the z-axis, computed taking $\boldsymbol{m}$, $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ as a vectors with three components and $m_z = 0$, $\eta_z = 0$ and $\xi_z = 0$ respectively.

Hessian matrix

Webster et al. show that in the one-dimensional case the root mean square of the truncation error of a finite element scheme is proportional to the second derivative of the solution and to the mesh size squared [195]. Following such principle a sensor of the error in the two-dimensional case can be built from the Hessian matrix [37, 80, 146, 193, 200, 195] as

$$E_i = V_i \sqrt{E^2(\boldsymbol{\tau}_u) + E^2(\boldsymbol{\eta}_u)}, \qquad \text{with} \qquad E(\boldsymbol{p}, s) = \frac{\boldsymbol{p}^{\mathrm{T}} \mathcal{H}(s) \boldsymbol{p}}{\boldsymbol{p}^{\mathrm{T}} \boldsymbol{\nabla} s \sqrt{V_i} + \epsilon \mu(s)} \tag{3.7}$$

$\boldsymbol{p}$ is a generic versor in $\mathbb{R}^2$, $\boldsymbol{\tau}_u$ and $\boldsymbol{\eta}_u$ are the tangent and normal versors to the local velocity vector, ϵ is a constant chosen between 0 and 1 (0.12 here) and $\mu(s)$ is the average value of s over the computational domain. The discrete Hessian matrix $\mathcal{H}(s)$ is computed as the gradient of each component of $\boldsymbol{\nabla} s$; eq. (3.5) is therefore applied twice and again a second order approximation can be computed using a finite-element scheme [163, 63].

Modified Hessian matrix

A modified version of the sensor of Eq. (3.7) is also introduced, i.e.

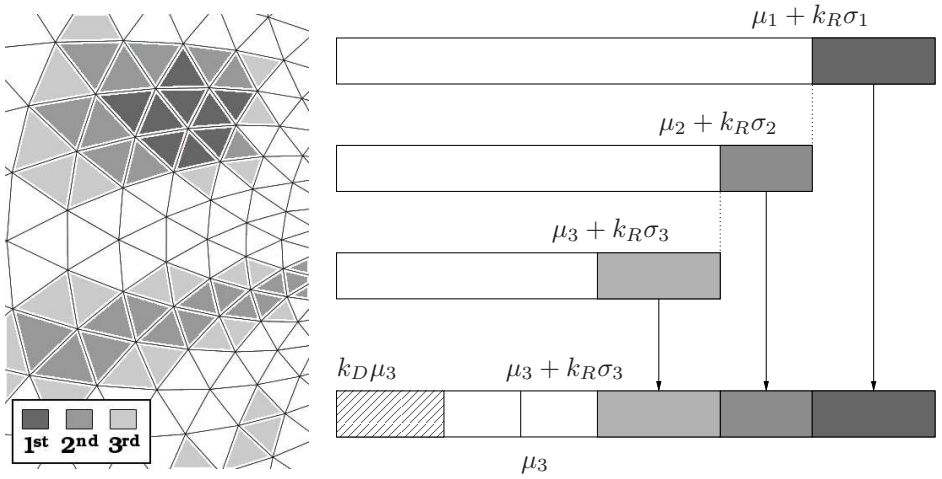
$$E(\boldsymbol{p}, s) = \frac{\boldsymbol{p}^{\mathrm{T}} \mathcal{H}(s) \boldsymbol{p}}{\boldsymbol{p}^{\mathrm{T}} \boldsymbol{\nabla} s \sqrt{V_i} + \epsilon \mu(s)} + \frac{\boldsymbol{p}^{\mathrm{T}} \boldsymbol{\nabla} s \sqrt{V_i}}{V_i \, \boldsymbol{p}^{\mathrm{T}} \mathcal{H}(s) \boldsymbol{p} + \epsilon \mu(s)}, \tag{3.8}$$

which uses the gradient of s to better capture the shocks. Indeed, due to the change in sign across the discontinuities, the second derivative goes to zero near the shocks and the grids obtained adapting on the Hessian show under resolution in such regions. The second term of Eq. (3.8), however, counterbalance such effect since it is proportional to $\boldsymbol{\nabla} s$ and inversely proportional to $\mathcal{H}(s)$.

Substantial derivative

When mesh adaptation is applied to unsteady problems, the choice of the sensor function is somewhat less straightforward. Indeed the necessity of adapting over the solution computed at every time step, could be in contrast with the variation in time of the solution itself, e.g. a normal shock wave moving in a duct. To this purpose a specific sensor for unsteady problem is the substantial derivative of a variable of interest [181], namely

$$E(s) = \frac{\partial s}{\partial t} + \frac{\boldsymbol{m}}{\rho} \cdot \boldsymbol{\nabla} s. \tag{3.9}$$



(a) Elements flagged (b) Levels of flagged elements

Fig. 3.8: Multi passage strategy for labeling procedure.

Threshold computation and multipassage strategy

The identification of subdomains concerned with significant flow features is the enabling issue for mesh adaption. Elements characterized by high error estimate suggests that some variation in the flow field may not be well resolved, while nodes with a low level of error reveal a possibly excessive resolution.

Suitable threshold values are then defined to identify low and high values of error based on statistical considerations, τ_D and τ_R respectively. Independently from the definition of the threshold values an element will be flagged for refinement if $E_e > \tau_R$, where the error on the element is computed as the average of the error on its vertices, i.e.

$$E_e = \frac{1}{\dim \mathcal{K}_e} \sum_{i \in \mathcal{K}_e} E_i.$$

During the derefinement procedure, a node i tagged for removal if $E_i < \tau_D$.

The error indicators are interpreted as a data set in which error estimates are distributed about an average value, i.e.

$$\mu(s) = \sum_{i \in \mathcal{K}} E(s_i)$$

with a certain degree of dispersion indicated by the value of the standard deviation, i.e.

$$\sigma(s) = \sqrt{\frac{1}{\dim(\mathcal{K})} \sum_{i \in \mathcal{K}} (E(s_i) - \mu)^2}.$$

In order to reduce the number of elements whose error is significantly higher than the mean value, the refinement threshold is chosen as $\tau_R = \mu + k_R\sigma$. The refinement parameter k_R has to be suitably chosen, the smaller the value the more uniformly the error will be distributed between the elements of the domain.

The derefinement threshold is simply chosen as $\tau_D = k_D\mu$, where the standard deviation is not used. Indeed in some cases the value of σ could be bigger than the mean, thus resulting in a negative derefinement threshold.

For convenience the area-weighted average error is also introduced, namely

$$\mu_w = \frac{\sum_{i\in\mathcal{K}} V_i E(s_i)}{\sum_{i\in\mathcal{K}} V_i}. \tag{3.10}$$

Standard fluid dynamics flow fields usually present features with different intensities, e.g. the gradients near the shock wave and the rarefaction fan that are computed as solution of a one dimensional Riemann problem. In order to capture flow features with lower intensities modifications to the classical error estimates have been proposed in the literature [117]. To this purpose a multiple passages technique is thus adopted to suitably modify the list of nodes that participate in the computation of the thresholds [4]. Fig. 3.8 illustrates labeling procedure in the case of refinement and derefinement. Each rectangle in the figure represents the list of elements organized by increasing error estimate from left to right. First the refinement threshold is computed considering every element of the grid (i.e. the whole rectangle) then the elements with high values of error are flagged for refinement, i.e. the dark gray colored elements of fig. 3.8. In the second passage the mean and the standard deviation, i.e. μ_2 and σ_2 respectively, are computed by considering just the elements that have not been already labeled for refinement, i.e. the white region of the rectangle on top of fig. 3.8. In this way strong features are excluded and the resulting thresholds are determined on the basis of the weaker features of the flow field. The number of passages that can be performed is arbitrary, but in practical problems no more than three levels of refinement are usually necessary. In the case of derefinement the passage the mean computed in the last refinement passage is used to identify the nodes with a low error level. No more than one passage is usually necessary, since the most important flow features have been already captured during the last level of refinement.

3.3.3 Refinement/Derefinement Strategy

In fig. 3.9 the refinement strategy based on both the solution driven adaptation and the element size strategy is shown. First the requirements on the distance-based function are met (check 1 and 2), then the equidistribution principle is enforced (check 3 and 4). An additional check on the minimum length of the edges along the boundaries is performed for triangles laying on $\partial\Omega$. Check 2 and 4 avoid the insertion of a node that has strong chances to be subsequently removed during a possible derefinement step, e.g. the area of the refined element is below the given threshold or the weighted error is below τ_D.

As shown in fig. 3.10, a similar strategy is used to perform the deletion of a node from the grid. First the requirements on the imposed dimension are checked, i.e. the average area of the elements surrounding the node must not be lower than the given threshold. The node is then removed if the average value of the error is higher than τ_D. An additional constraint on the maximum size of the edges is applied for boundary nodes.

In fig. 3.11 an example of the complete adaptation procedure, termed `AP`, is shown. First the boundaries displaced with an assigned law and the internal mesh is deformed to follow such movement, then the edges surrounded by low quality elements are swapped.

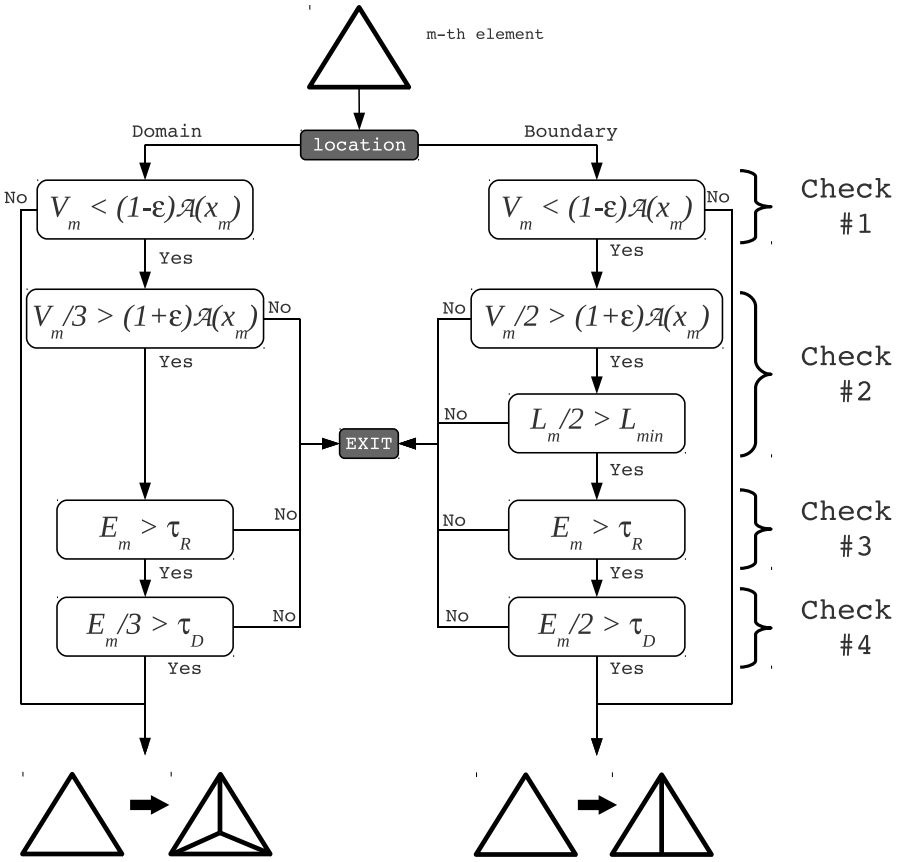


Fig. 3.9: Refinement strategy based on a combination between both the error sensor and the distance from the boundaries. The refinement is not performed if: (#1) the area of the element is lower than minimum area imposed by the boundary distance based function, (#2) the area of the refined elements is bigger than maximum area imposed by the boundary distance based function, (#3) the error of the element is smaller than the threshold τ_R, (#4) the error of the refined element is possibly larger than the threshold τ_D.

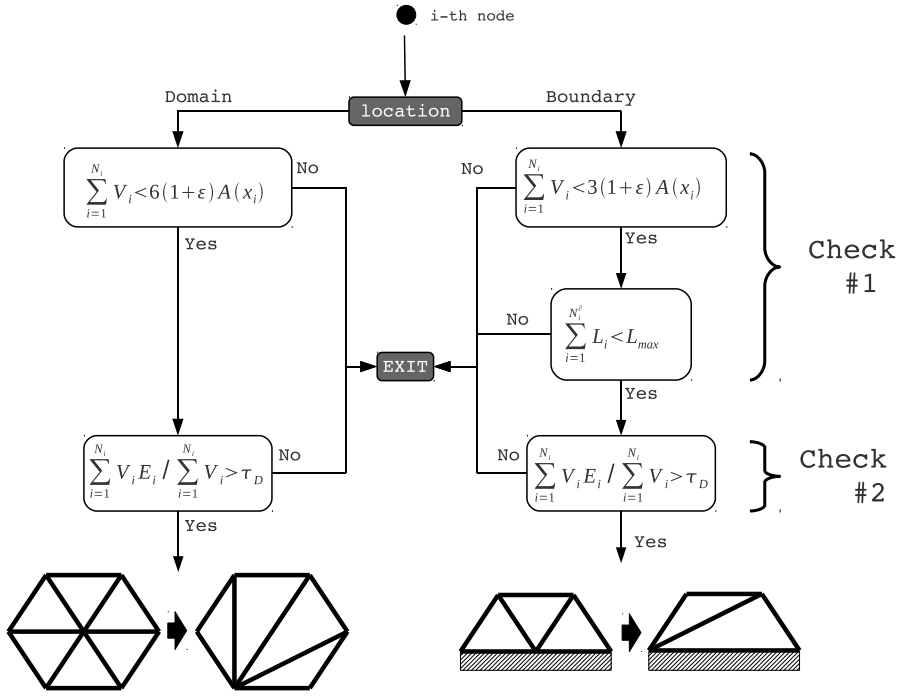


Fig. 3.10: Node removal strategy based on a combination between both the error sensor and the distance from the boundaries. The refinement is not performed if: (#1) the area of the bubble of elements surrounding the node is bigger than maximum area imposed by the boundary distance based function, (#2) the mean error on the bubble of elements surrounding the node is smaller than τ_D.

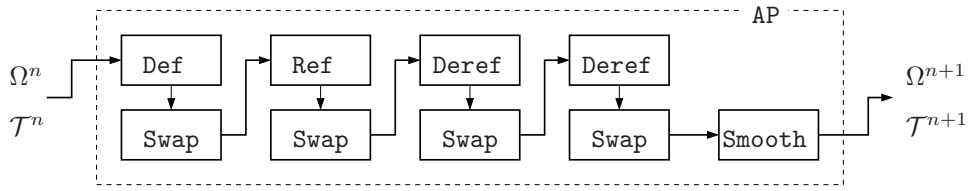


Fig. 3.11: Example of adaptation procedure. The deformation and the insertion/deletion procedures are followed by an edge-swapping cycle to restore the highest possible quality without change in the vertices position.

The second step is the application of the refinement procedure of fig. 3.9, followed by a swapping cycle to restore high quality and possibly reduce the local size of the elements. The third and fourth steps are two subsequent derefinement plus swapping procedures. The last step is the global grid regularization. The order and the number of times a single procedure is applied is arbitrary but typically mesh deformation and smoothing are the first and the last one respectively.

3.4 Grid adaptation for unsteady applications

When a steady solution is computed the implementation of an adaptation scheme is quite straightforward and it consists essentially in a three steps procedure to be repeated until convergence is achieved: steady state solution computation, error computation and grid adaptation. A solution interpolation strategy can be also used to speed-up the steady state computation step. For the application of grid adaptation to the unsteady case different approaches are possible. A classic fixed-point, here termed SAP, algorithm uses one steady adaptation cycle at each time step [136]. It can be sketched as follows:

SAP.1 The error is computed starting from the solution u^n

SAP.2 The mesh is adapted to $E(\mathrm{u}^n)$

SAP.3 If necessary, the solution u_n is interpolated over the new mesh

SAP.4 The solution is updated and the next time step is carried out

Such procedure is suitable only if very small time steps are adopted, since the grid is adapted over the "old" solution.

A different approach is the one illustrated in fig. 3.12 where the grid is adapted to the new solution by means of an iterative procedure. Such approach, referred here as SIAP, can be sketched as follows:

SIAP.1 The mesh is deformed to follow the movement (if any) of the bodies.

SIAP.2 The solution u^{n+1} over the new grid is then predicted.

SIAP.3 The error is computed based on the new predicted solution.

SIAP.4 The grid is adapted over over $E(\mathrm{u}^{n+1})$ and the old solution is interpolated, if necessary.

SIAP.5 The solution u^{n+1} is re-computed over the adapted grid in an unsteady fashion, i.e. as an unsteady step from u^n to u^{n+1}. In the present work no interpolation of u^n is necessary and the new solution is obtained simply integrating Eq. (2.47).

Step 2, 3 and 5 are repeated (looping over s) until convergence is obtained, i.e. when μ_w is lower than the threshold τ_E. While the point fixed procedure adapts the grid over u^n, the iterative one over u^{n+1}. Moreover it has to be noted that the computational burden of this iterative adaptation procedure relies on how "quickly" the error, thus the solution, varies in time. The Co number is therefore critical in the choice of a reasonable value of convergence threshold. In the present work a maximum number of 10 iterations

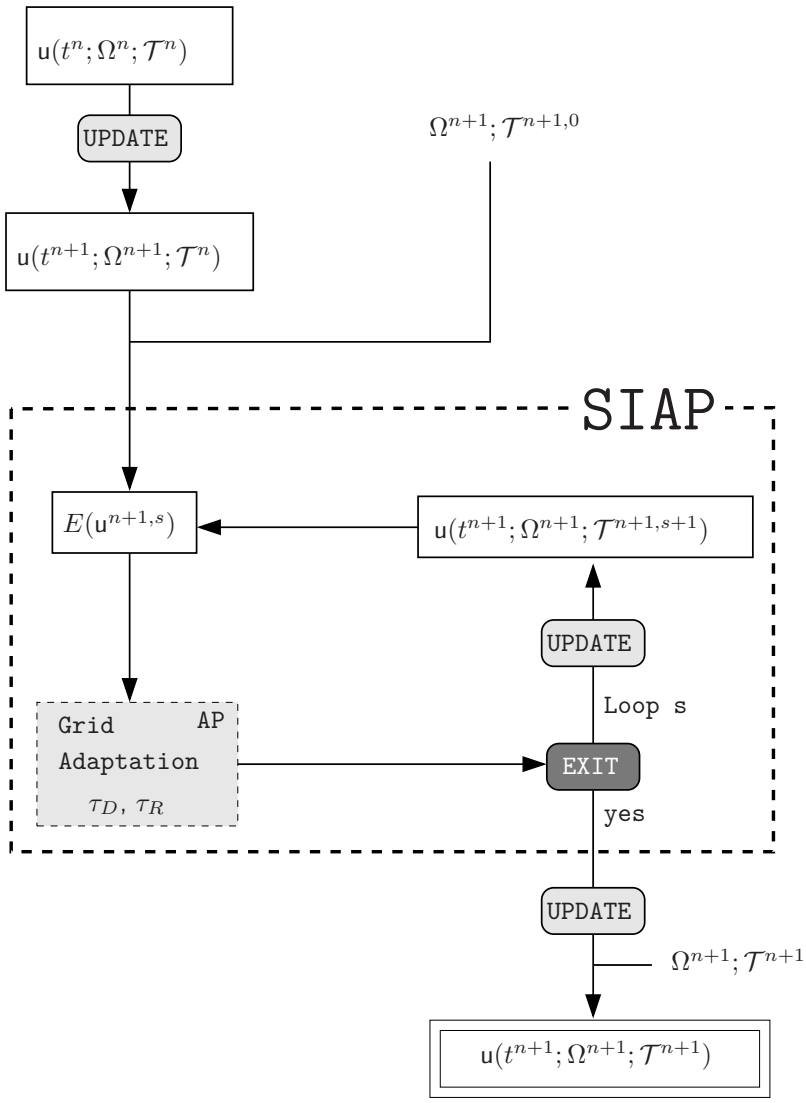


Fig. 3.12: Simple Iterative Adaptation Procedure, i.e. SIAP.

over s is found suitable in many cases if interest. For low values of Co, e.g. when high time-accuracy is required, iterations over s can be skipped and only the step from 1 to 5 described above can be carried out.

A modification of the iterative procedure SIAP has also been implemented in order to speed up the grid-solution convergence at the new time level, with particular reference to unsteady problems.

As shown in fig. 3.12 at a given adaptation step the error $E(\mathsf{u}^{n,s})$ is computed from the solution computed over the triangulation $\mathcal{T}^{n,s}$. The mesh is thus adapted over $E(\mathsf{u}^{n,s})$ and the triangulation $\mathcal{T}^{n,s+1}$ is obtained. If a further level of refinement/coarsening of the grid is necessary the sensor must be recomputed over $\mathcal{T}^{n,s+1}$

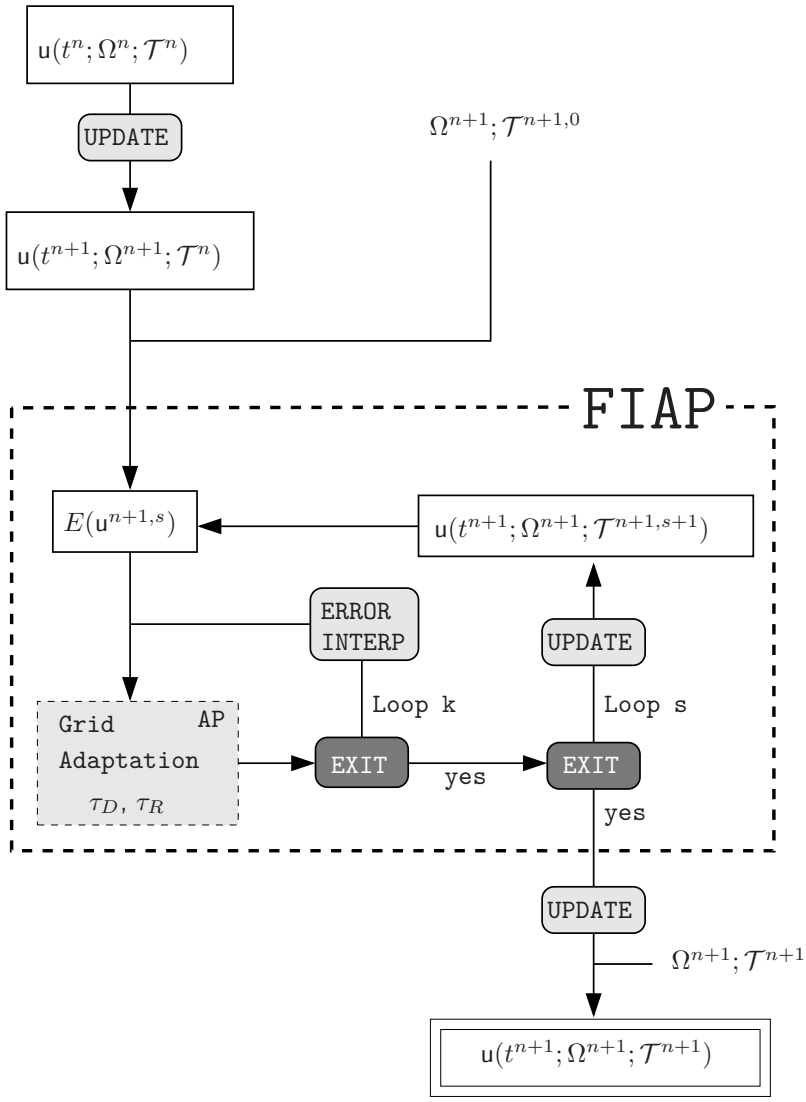


Fig. 3.13: Fast-Iterative-Adaptation-Procedure, i.e. FIAP.

and thus the solution. The burden in terms of computational time can be signficant if this operation is repeated frequently, i.e. for excessively coarse grids. To this purpose, an empirical error redistribution technique is used which project the error distribution $E(\mathsf{u}^{n,s})$ over the new adapted grid, without recomputing the solution vector. As shown in fig. 3.13 an additional loop over k has been inserted in the adaptation procedure, which correspond to the error projection step. More specifically each new/modified element inherits the area-weighted value of the error estimator of its parental element for grid refinement or neighboring elements for grid derefinement. When a node is inserted the elemental error E_m is equally distributed amongst the new elements, e.g. $E_n^{n+1,k+1} = E_m^{n+1,k}/3$ for the barycentric refinement and $E_n^{n+1,k+1} = E_m^{n+1,k}/2$ for the boundary one. Similarly when a node is removed the error is distributed amongst the nodes of the bubble surrounding the removed node, i.e.

$$E_j^{n+1,k+1} = \frac{1}{V_j^{\mathsf{n+1,k+1}}} \left(V_i E_i^{\mathsf{n+1,k}} + \sum_{k \in \mathcal{E}_i} V_k^{\mathsf{n+1,k}} E_k^{\mathsf{n+1,k}} \right),$$

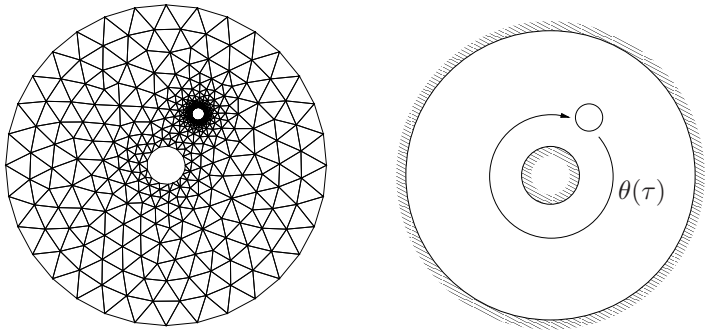
where the $k+1$ and k superscripts indicate the configuration before and after node removal respectively, e.g. fig. 2.20(a) and fig. 2.20(e).

A complete convergence of the iterative procedure at time t^{n+1} is reached when the difference between average numerical error evaluated on consecutive s-cycles is below a pre-defined threshold. After the final solution at t^{n+1} is computed on the adapted grid $\mathcal{T}^{n+1}$, the computation advances to the next time level t^{n+2}. This strategy will be called *Fast-Iterative-Adaptation-Procedure* (`FIAP`).

3.4.1 Test cases for grid movement

The adaptation scheme presented above is tested on the simple geometry of fig. 3.14. The smaller circle performs a rotation of full about the bigger one back and forth, i.e. $\theta(\tau) = \pi(1 - \cos(2\pi\tau))$, with $\tau \in [0, 1]$. The first simulation has been carried out using only a combination of mesh deformation, edge-swapping and barycentric smoothing to complete one rotation in 32 steps. In fig. 3.16 (left column) the grid obtained at different time steps is shown. The initial spacing distribution is progressively lost, although the overall element quality is fairly maintained during the movement and the smallest element remain close to the rotating circle.

The second simulation has been carried out with the distance based adaptation algorithm without solving any fluid dynamic equation. At every time step the adaptation procedure of fig. 3.11 is applied and no other iterations are performed to further adapt the mesh, e.g. neither the `SIAP` or the `FIAP` procedures have been adopted since there is no solution to adapt on. The grids obtained every 16 timesteps are shown in fig. 3.16 (right column). The initial grid spacing is very well preserved as well as the grid quality. Indeed the minimum and the average value of the quality of the elements, defined in Eq. (3.3) and shown in fig. 3.15, is higher when the complete mesh adaptation is carried out.



(a) Initial grid (b) Prescribed movement

Fig. 3.14: Rotation the smaller a circle about the larger one. (a) Initial Grid. (b) The smaller circle performs a rotation of 360° around the center back and forth.

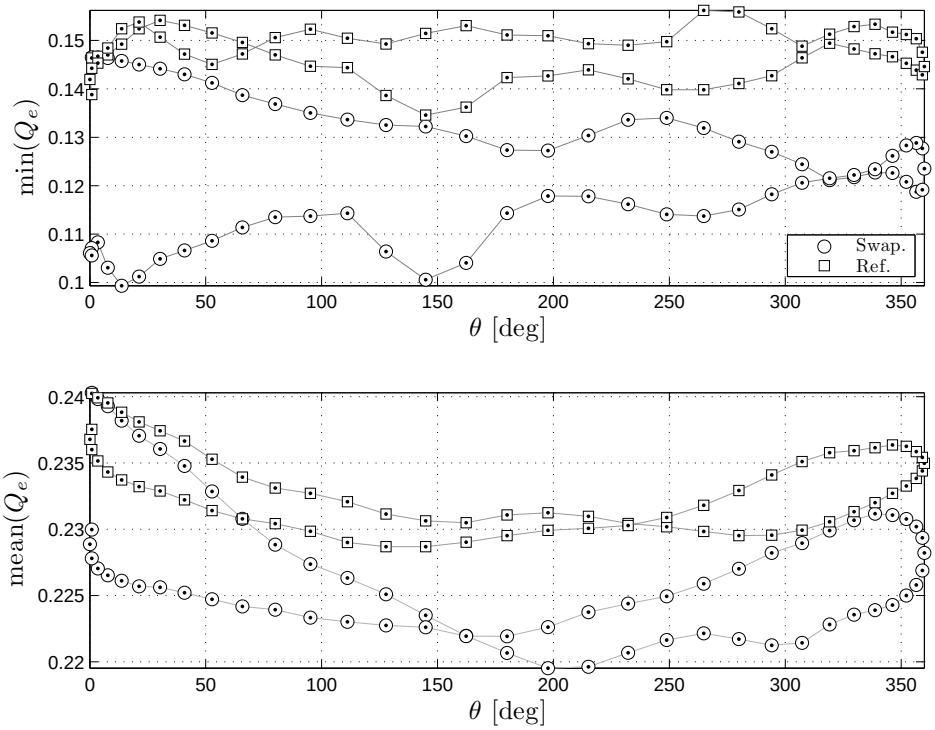


Fig. 3.15: Minimum and mean quality computed at different values of θ.

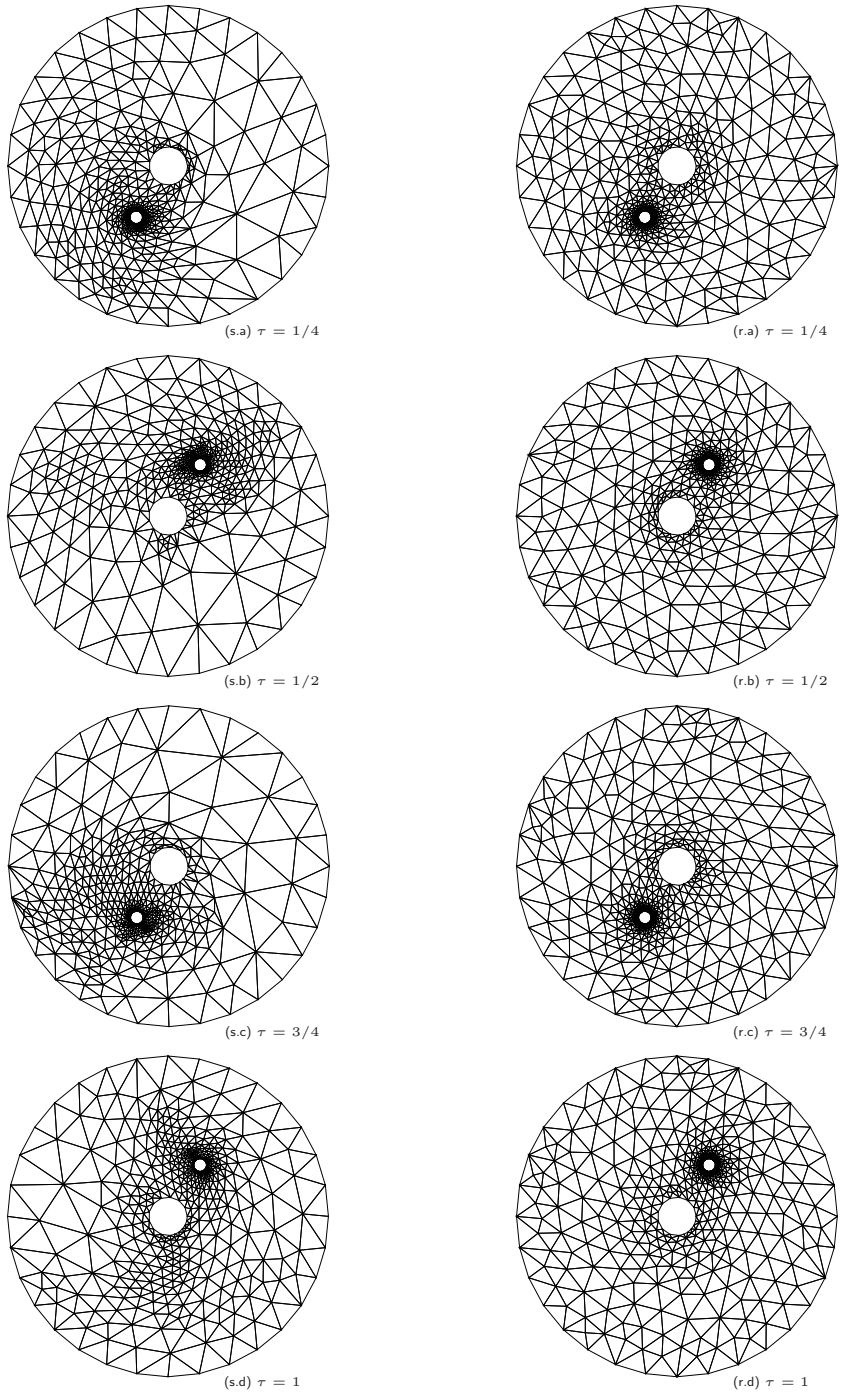


Fig. 3.16: Grid evolution for different values of the non-dimensional time τ. Left column swapping and deformation, right column complete AP refinement/coarsening based on the distance from the boundaries.

Chapter 4

Simulations of reference compressible flows

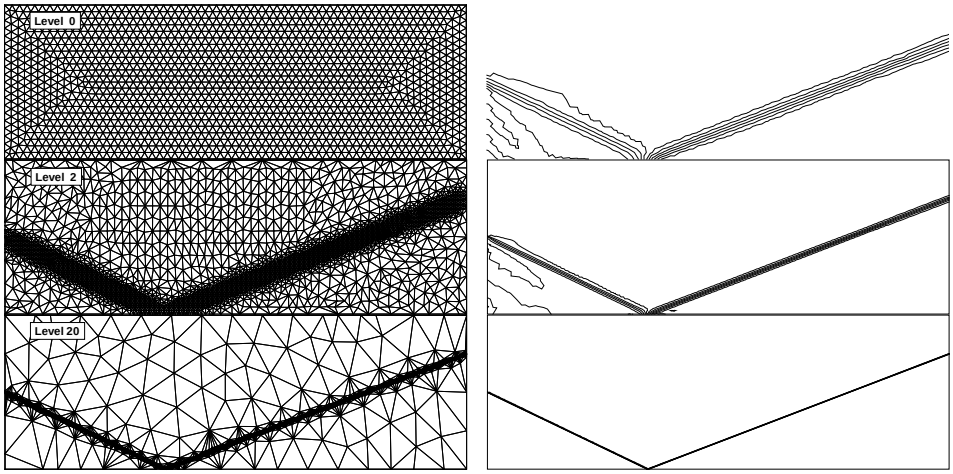
Numerical experiments are carried out to test the accuracy of the proposed solution procedure.

Simulations for both steady and unsteady test cases are carried out. First the computational efficiency of the two adaptation strategies named `SIAP` and `FIAP` discussed in section 3 is evaluated against the standard steady oblique-shock test case in section 4.1, where no complications in the domain geometry are present and the solution is simple enough that only two shocks are featured in the flow field, moreover the exact solution is available [180]. In section 4.2 and 4.3, steady flow simulations around the AGARD-02 and NACA 0012 airfoil are performed to assess the solver capabilities at capturing relevant flow features of transonic flows, including shock waves, shear lines and regions of smooth flows.

In section 4.4 the shock tube problem first proposed by Sod [167] is tackled to test the capabilities of the adaptive scheme to describe a transient solution. The Sod problem features simultaneously a shock waves, a contact discontinuity and a rarefaction fan moving inside the domain and an analytical/exact solution is available. It is thus well suited to study different error sensors in the unsteady case.

4.1 Oblique shock problem

The standard oblique-shock test problem is carried out to compare the efficiency of the two proposed adaptation strategies. On the left boundary a Mach 3 flow parallel to the wall is imposed, on the top the imposed solution correspond to a Mach 2.60243 flow with an incidence of -8.01351° and on the right boundary no conditions are imposed since it is a supersonic outflow. Due to the discontinuity in the boundary condition an oblique shock (with a -25.60° angle with respect to the wall) is generated from the top-left angle of the domain. The shock is reflected by the solid boundary with an angle of 20.935° and the flow in the last region is supersonic and parallel to the wall.



(a) Grid (b) Mach number

Fig. 4.1: Grid and Mach number in the flow field for the oblique-shock problem for SIAP. From top to bottom: original grid (1532 nodes, 2900 elements), intermediate grid at the 2-nd adaptation level (2735 nodes, 5340 elements), final grid at the 20-th adaptation level (10557 nodes, 20994 elements).

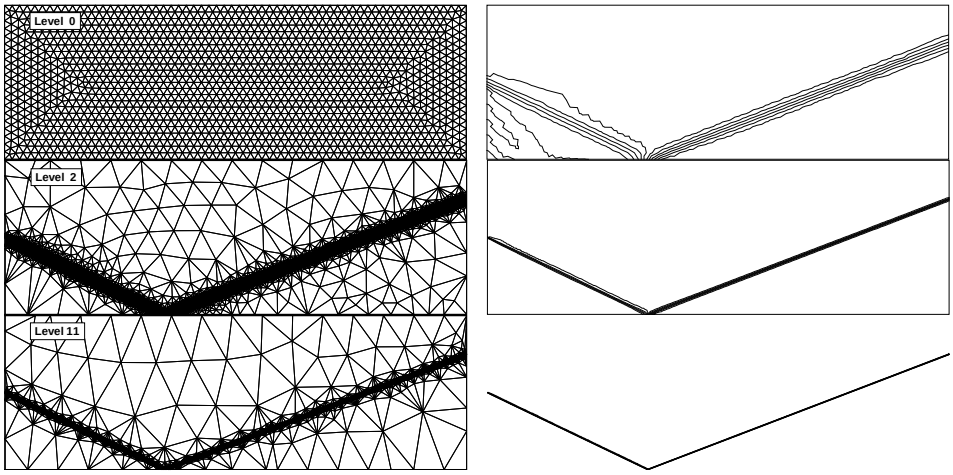
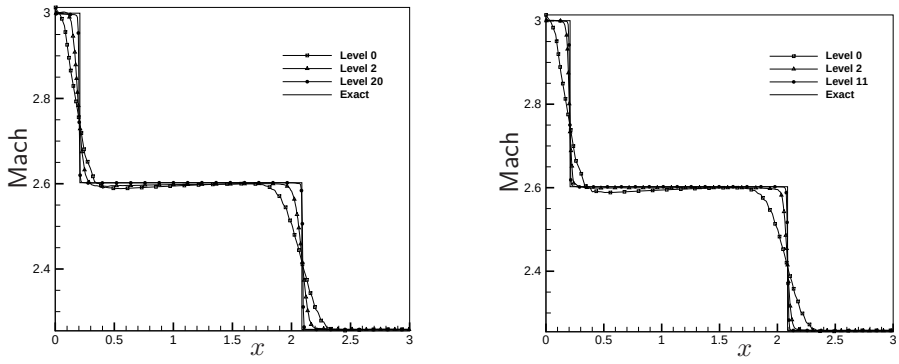


Fig. 4.2: Grid and Mach number in the flow field for the oblique-shock problem for FIAP. From top to bottom: original grid (1532 nodes, 2900 elements), intermediate grid at the 2-nd adaptation level (3787 nodes, 7460 elements), final grid at the 11-th adaptation level (11500 nodes, 22916 elements).



(a) SIAP (b) FIAP

Fig. 4.3: Comparison between analytical and numerical solution at $y = 0.4$ for the oblique-shock problem for different adaptation steps and strategies.

The initial grid is shown in fig. 4.1 (a-top) and it is made of 1532 nodes and 2900 elements. The steady adaptation procedure on the gradient of the Mach number is carried out until convergence on both grid and solution is achieved using the two proposed strategies (SIAP and FIAP). A minimum element area of 10^{-6} is imposed in both cases. The computations are interrupted when the relative difference of the area-averaged mean error μ_w of two successive adaptation cycles is below 2%, namely $\left(\mu_w^{s+1} - \mu_w^s\right)\big) / \mu_w^0 < 0.02$. Since only shocks of comparable intensity are present in the flow field the multipassage strategy has not been used.

In fig. 4.1 the evolution of the grid and of the solution in terms of Mach number at different adaptation levels for the SIAP technique is shown. The grid converges after 20 steps and the final grid is made of 10557 nodes and 20994 elements. In fig. 4.2 the grid and Mach number contour lines for the FIAP technique are shown. The adaptation procedure is stopped after 11 iteration and the adapted mesh is made of 11500 nodes and 22916 elements.

Fig. 4.3 shows the numerical solutions at $y = 0.4$ against the exact one for both the SIAP and the FIAP procedures. The FIAP technique shows better results after the second adaptation step already, indeed a higher number of nodes was inserted before the solution is re-computed.

Fig. 4.4(a) shows the error and number of points at each time steps in the SIAP case. The algebraical error μ increases, due to the redistribution of elements from low error regions (constant solution regions) to high error regions (shock waves), while the mean error scaled on areas μ_w decreases. When the minimum area threshold is reached the value of μ_w remains quite constant and the higher values of the error are located on the smaller elements. In the FIAP case, shown in fig. 4.4(a), a similar behavior is obtained but a sharper decrease in the area weighted error can be observed. The lower "readiness" of the SIAP scheme in terms of mesh adaptation is very well highlighted in fig. 4.4(b). At the fifth iteration the adapted grid is made of roughly 40000 elements, which causes the smaller elements size to be close to the minimum value. For this reason

a sharp increase in the number of elements that can not be refined is observed together with a decrease in the number of elements of the grid, that are removed from the low error regions and added in the high error ones. In the SIAP case almost 15 iterations are needed for the number of elements to be roughly constant, while in the FIAP case, shown in fig. 4.4(d), only three iterations are necessary to enforce the constraint on the minimum area.

The computational times are 170m 01s using SIAP and 53m 28s using FIAP on a single core of an Intel Xeon QuadCore 3.166GHz. Therefore, in the present case the time necessary to perform a single solution plus adaptation step two times higher when the SIAP technique is adopted instead of the FIAP, possibly because a better initial guess is computed in the FIAP case which reduces the time required to reach steady state.

4.2 AGARD 02 airfoil

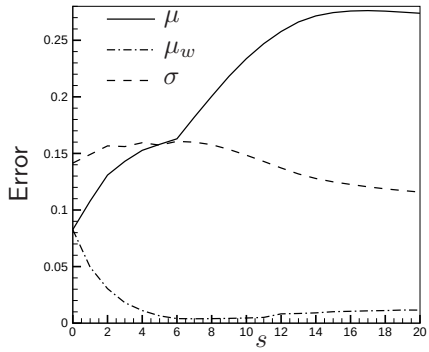
The adaptation scheme is tested on a standard transonic aerodynamic problem, namely the steady flow past a AGARD-02 airfoil [201]. The Mach number of the undisturbed flow is 0.85 with an angle of attack of 1°. As result two compression shocks are present, the most intense of the two is located near the trailing edge on the upper side of the airfoil, while the weaker one is on the lower side. A shear wake is also detaching from the trailing edge since the jump of total enthalpy and pressure is different between the upper shock and the lower one.

The initial grid, shown in fig. 4.5 (a-top), is made of 3467 nodes and 6704 elements. The FIAP scheme is used to adapt the grid to the sensor defined in Eq. (3.8) as a function of the Mach number. The multi passage strategy is used with one level of refinement. The procedure is interrupted when the variation of the area weighted mean error between two subsequent iterations is lower than 1% of the initial value of μ_w. The minimum allowed element size is set to 1×10^{-6}.

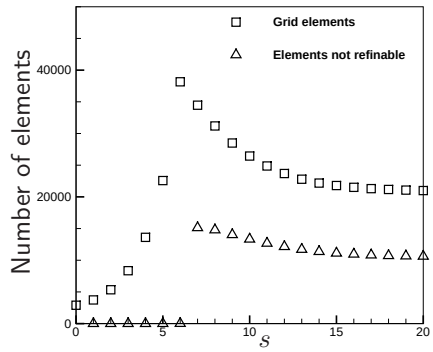
In fig. 4.5(a) the computational grids obtained at the first adaptation step (5552 nodes and 10806 elements) and at the final one (13539 nodes and 26791 elements) are shown. The adopted sensor, which is proportional to both the gradient and the Hessian of the Mach number, and the multi passage strategy allows to capture both the shocks and the rarefaction close to the nose of the airfoil. The airfoil wake, shown as a discontinuity in the Mach number profile, is only mildly captured in the region close to the trailing edge. Indeed the intensity of contact discontinuity is sensibly weaker than the one of the shocks and the rarefaction. Two or three multi passages should be sufficient to better capture this feature but a strong increase in the number of elements is to be expected near the leading edge as well.

In fig. 4.6(a) the Mach number distribution along the airfoil as a function of x/c is shown. The solution obtained with the adaptive scheme is compared with the one obtained on a fixed grid of 21383 nodes and the reference solution computed by Yoshihara and Sacher [201] on a structured grid of 20480 nodes. The adapted solution agrees fairly well with the references and the position of the shocks is predicted correctly.

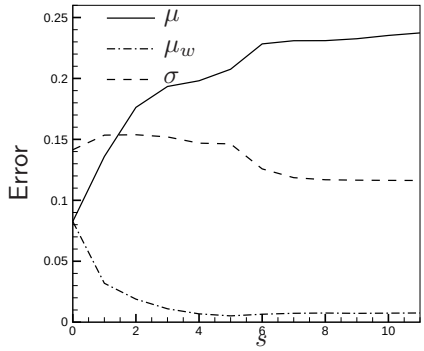
The solution obtained with the adaptation scheme is compared with the solution obtained with the same scheme but with a smaller minimum area threshold. With $V_{min} = 5 \times 10^{-7}$, the adaptation procedure converges after 11 iterations and the final grid is made of 15716 nodes, whereas, with $V_{min} = 3 \times 10^{-7}$, 13 iterations are necessary to produce a grid made of 18174 nodes.



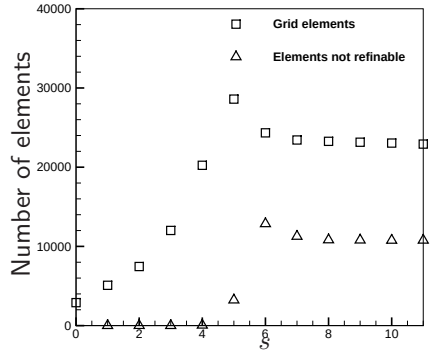
(a) Error SIAP



(b) N. of elements SIAP

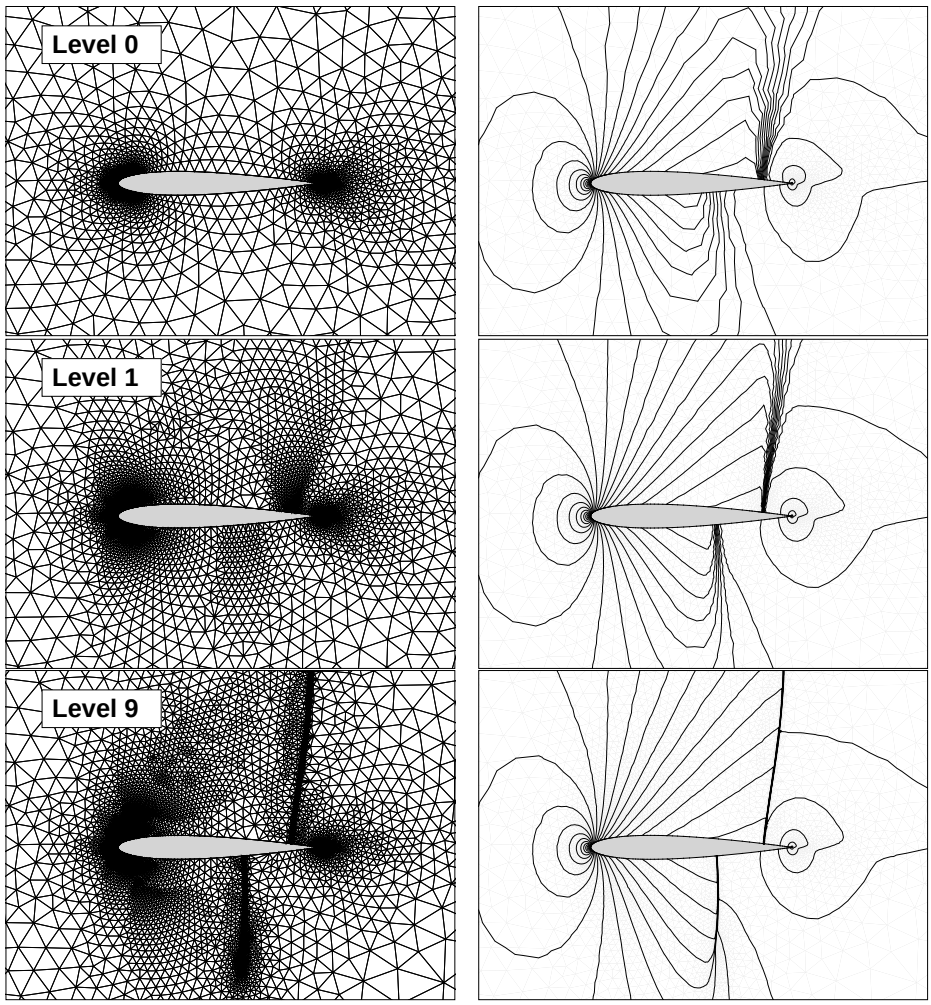


(c) Error FIAP



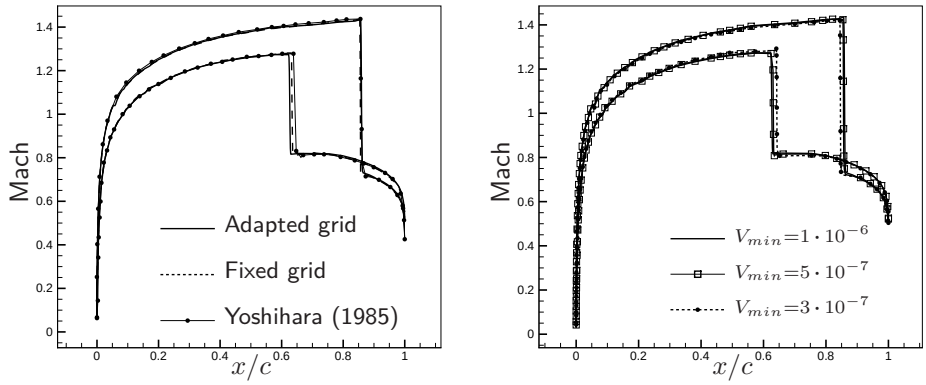
(d) N. of elements FIAP

Fig. 4.4: Adaptation technique for both SIAP and FIAP procedures. (a) and (c): algebraic mean μ, area-averaged mean μ_w and standard deviation σ as a function of the adaptation steps for the steady-state oblique-shock problem. (b) and (d): number of grid elements and of not refinable elements as a function of the adaptation step. Note that the scale of the x-axis is different for the SIAP and the FIAP plots.



(a) Computational Grid (b) Mach number

Fig. 4.5: Computational grid and Mach number contour lines, 25 levels in the range $0.07 \div 1.4$, for the AGARD 02 test case. The minimum element size is equal to $V_{min} = 1 \times 10^{-6}$. Three adaptation levels are shown: the initial grid of 3467 nodes and 6704 elements, the first adapted grid of 5552 nodes and 10806 elements and the final grid 13539 nodes and 26791 elements.



(a) Reference (b) Adaptation

Fig. 4.6: Mach number distribution along the airfoil as a function of x/c. (a) Confrontation between the solution obtained with the adaptive scheme, with $V_{m,min}3\times10^{-7}$, the one obtained on a fixed grid made of 21383 nodes and 41368 elements and the reference solution computed by Yoshihara and Sacher [201] on a structured grid of 20480 nodes. (b) Confrontation between the solution obtained using the adaptation scheme with different threshold for the smallest element: 1×10^{-6}, 5×10^{-7} and 3×10^{-7}.

In fig. 4.6(b) the Mach number distribution along the airfoil is compared for the three adaptation schemes. Reducing the minimum area to 3×10^{-7} causes the shock on the lower side to move aft, while the upper shock moves fore. If compared to the reference [201], the position of the lower shock is better predicted in the most refined case, whereas the upper shock is closer to the reference in the least refined one. The location of the shock is extremely sensible to the computational grid, indeed the variation between the finest and the coarsest adapted grids is roughly 2% of the chord and the shock computed by the solution over the fixed grid of fig. 4.6(a) is located just in between the reference and the coarsest adapted grid.

4.3 NACA 0012 airfoil

The case of the compressible inviscid flow developing past a NACA 0012 airfoil at 5° incidence and Mach $= 0.7$ is discussed. This transonic flow is characterized by the presence of a shock wave located at about 45% of the chord and by a slip (entropy) line detaching from the trailing edge.

The initial grid, made of 3179 nodes and 6030 elements, is shown in fig. 4.7 (top) together with the contour lines of the Mach number. The grid is adapted on the mixed Hessian-Gradient of the Mach number and one multi passage is used, as done in section 4.2, but this time the `SIAP` procedure is used and the minimum element area is set to 1×10^{-7}. The adaptation procedure is stopped when the relative variation of μ_w is lower than the threshold $\tau_E = 0.01$ and the obtained grid is shown in fig. 4.7 (bottom). Differently from the AGARD-02 case of section 4.2 the refinement scheme captures fairly well the shock, the rarefaction on the nose and the rotational wake. This is caused by

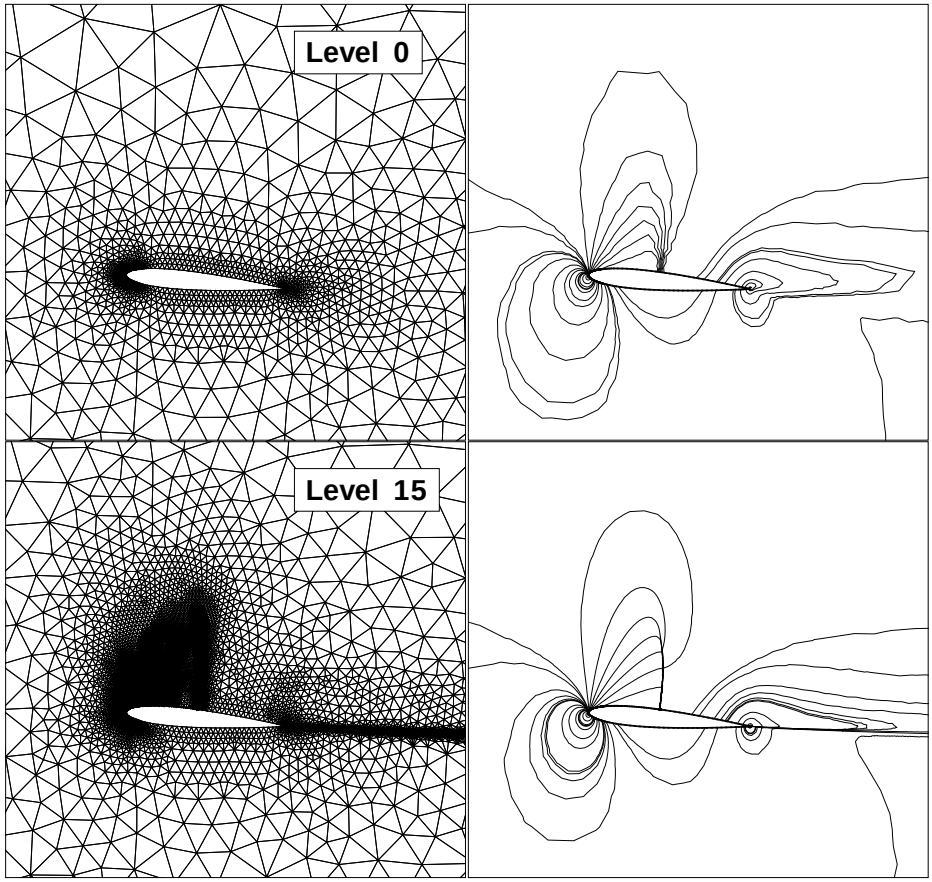


Fig. 4.7: Grid and Mach contour for NACA 0012 at Mach = 0.7. (top) original grid (3179 nodes, 6030 elements) and isolines (25 levels from 0.08 to 1.5). Second row: 15-th adaptation level (26499 nodes, 52568 elements) and isolines (25 levels from 0.08 to 1.5).

the fact that the only one shock is present in the flow field and the variation in the entropy between the upper and the lower side of the contact discontinuity is stronger.

Figure 4.8 shows the pressure coefficient over the airfoil after 0, 1 and 15 adaptation steps. Fig. 4.9 (left) shows the algebraic mean, the area-averaged mean and the standard deviation as a function of the adaptation step. Fig. 4.9 (right) shows the number of grid elements at each adaptation step and the number of elements that cannot be refined since $V_m < 10^{-7}$. During the first iterations, both μ and σ decrease, due to the insertion of additional nodes close to the airfoil. When the fifth adaptation cycle is reached these quantities increase because new nodes are inserted close to flow discontinuities (i.e. large error regions) and simultaneously elements are removed from low error regions. As new grids are produced, the number of elements that cannot be refined increases. From the tenth to the last adaptation step, no new nodes are added and only mesh coarsening occurs.

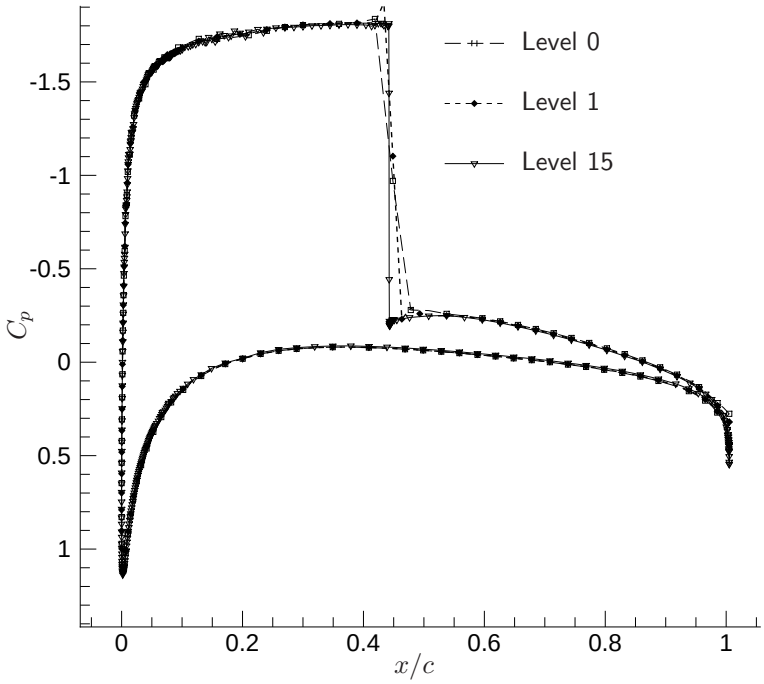


Fig. 4.8: NACA 0012 at Mach 0.7. Pressure coefficient over the airfoil for the initial grid and for adapted grids (first and last adaptation steps).

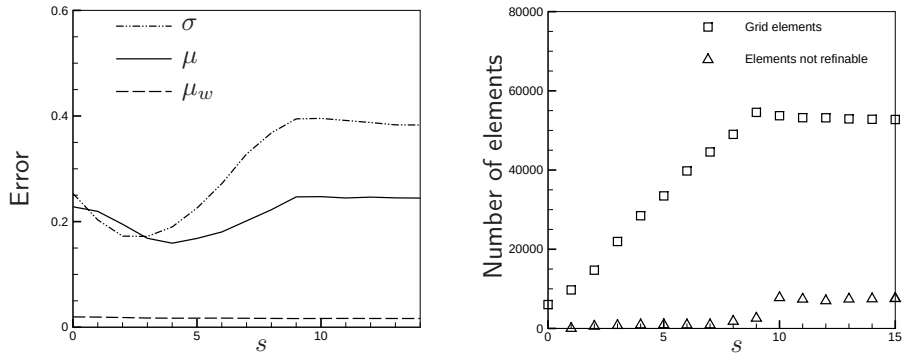


Fig. 4.9: (Left) algebraic mean μ, area-averaged mean μ_w and standard deviation σ as a function of the adaptation step for the steady-state NACA 0012 simulations. (Right) number of grid elements and of not refinable elements as a function of the adaptation step.

Grid	$V_m/10^{-7}$	$L_m/10^{-4}$	Max No. Nodes	Max No. Elements
Fixed	$3 \div 30$	$8.3 \div 26.3$	29718	58586
Initial	$3 \div 1000$	$8.3 \div 151.2$	3797	7401
Adapt. Fine	$3 \div 1000$	$8.3 \div 151.2$	13358	26421
Adapt. Coarse	$30 \div 1000$	$26.3 \div 151.2$	13358	26421

Tab. 4.2: Minimum/maximum area allowed and number of nodes/elements for the different simulations performed for the shock tube problem.

4.4 Sod problem

The first unsteady test case is the Sod shock tube problem [167]. A discontinuous initial solution is imposed such that a jump is present in the density and the pressure of the left and right state, as shown in table 4.1. The analytic solution of the corresponding one-dimensional problem consists of a three waves propagating longitudinally, i.e. along the tube: a rarefaction fan, a contact discontinuity and a shock.

A reference, fixed-grid, solution is first computed over the $5L \times L$ computational mesh of fig. 4.4 made of 29718 nodes and 58586 elements, where L is the total length of the tube. A constant time step, $\Delta t = 1.25 \times 10^{-4}$, is adopted to integrate the equations in the non-dimensional time interval $0 \div 0.25$. The area of the area of the elements is 3×10^{-7} for those located near the initial discontinuity and 3×10^{-6} elsewhere.

State	ρ	u	P
Left	1.000	0	1.0
Right	0.125	0	0.1

Tab. 4.1: Left and right state for the Sod problem. The variables are non-dimensional see section 2.1.1.

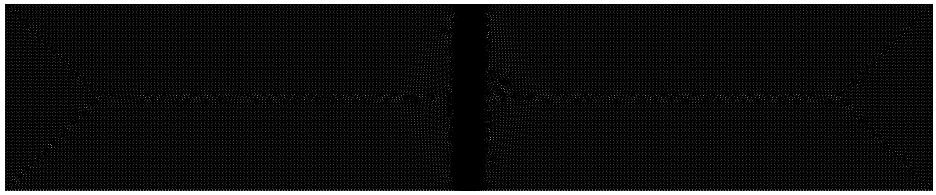
The initial grid used to carry out the adaptive computations is shown in fig. 4.10. It is made of 3797 nodes and 7401 elements, with an area of 9×10^{-7} close to the discontinuity and 9×10^{-6} elsewhere. Every four time-steps the `FIAP` procedure, with a convergence threshold $\tau_E = 0.05$ and two passages, is carried out to adapt the grid to the substantial derivative of the density.

In fig. 4.11 the adapted grids at several time levels are shown, one every 0.05 seconds. Although both the rarefaction fan and the shock wave are well captured by the substantial-derivative-based sensor, the contact discontinuity is not. This is agreement with Trivellato [181], that showed in the case of the density Eq. (3.9) reduces to

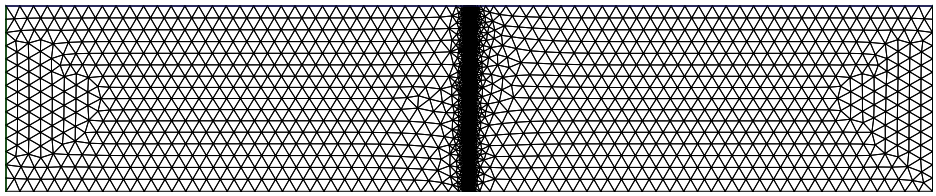
$$E(s) = \rho \left(\boldsymbol{\nabla} \cdot \frac{\boldsymbol{m}}{\rho} \right),$$

that is equal to zero since the velocity is constant across the contact discontinuity. In fig. 4.12 a comparison of the density profile obtained with the adaptive scheme and the exact solution along the center line is shown. The contact discontinuity is strongly smeared due to the under resolution of the grid.

The mixed Gradient/Hessian of the Mach number sensor is tested on the shock tube problem. Two different adaptive simulations have been carried out a fine one, with a minimum element area of 3×10^{-7}, and a coarse one, with a minimum area of 3×10^{-6}. In fig. 4.13 the computational grids for the fine case at several time levels (every 0.05



(a) Fixed grid, made of 29718 nodes and 58586 elements. $V_m \in (3 \times 10^{-7}, 3 \times 10^{-6})$.



(b) Initial grid, made of 3797 nodes and 7401 elements. $V_m \in (3 \times 10^{-7}, 1 \times 10^{-4})$.

Fig. 4.10: Initial grids used for the "fixed grid" and the adaptive simulations.

time units) are shown, while fig. 4.14 shows the grids for the coarse case. In both the simulations the sensor captures all the features of the flow field remarkably and the maximum spacing is recovered in the uniform flow regions.

In fig. 4.15 and 4.15 the density profile is plotted as a function of the non dimensional coordinate x/L along the center line of the domain, i.e. $y/L = 0.5$. The solution obtained on the fixed grid of fig. 4.4 is also shown, together with the exact solution, at several time steps. Due to its non-linear nature, once settled on its viscous profile, the normal shock propagates unmodified along the domain. The linearly degenerate contact discontinuity, however, is progressively smeared out by the numerical diffusion and, as time proceeds, the maximum value of the derivative of the solution decreases. This difference in the nature of these discontinuities is reflected on how the grid adaptation procedure. Indeed in fig. 4.13 is clearly visible that, while the mesh spacing across the shock is kept constant during the simulation, the refined zone across the contact discontinuity is progressively enlarged and unresolved with a progressive change in the intensity of the gray tone.

As shown in fig. 4.15 and 4.15, the solution computed over the fixed grid is overlaps fairly well the solution obtained with the coarse adaptive scheme. Indeed the maximum grid spacing for the fixed grid case is equal to the minimum one for the adaptive case, therefore both the discontinuities are resolved over a mesh that has roughly the same spacing.

The computations have been performed with a Intel Core 2 Duo T7500 processor at 2.20GHz. The fixed grid computation required 48 h, 57 m and 37 s ($\sim$88.1 s per iteration), the coarse adaptive one required 8 h, 49 m and 32 s ($\sim$15.9 s per iteration) and the fine adaptive simulation required 34 h, 12 m and 35 s, ($\sim$61.6 s per iteration). In the present case, therefore, the use of the adaptive scheme allowed to obtain more than a $\times 5$ speed-up with respect to the fixed grid case, that has a similar accuracy. Indeed the fixed-grid case over resolved large regions that featured a uniform flow.

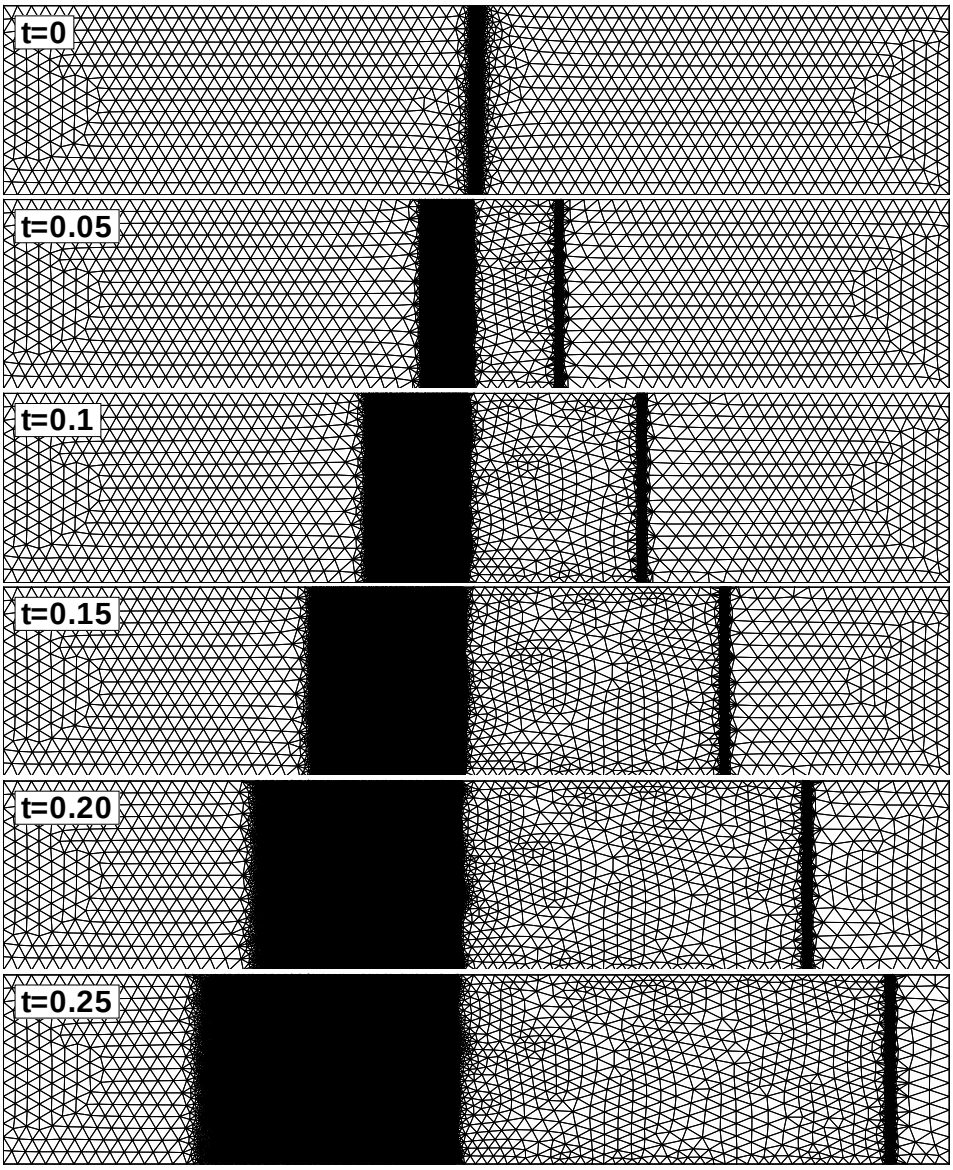
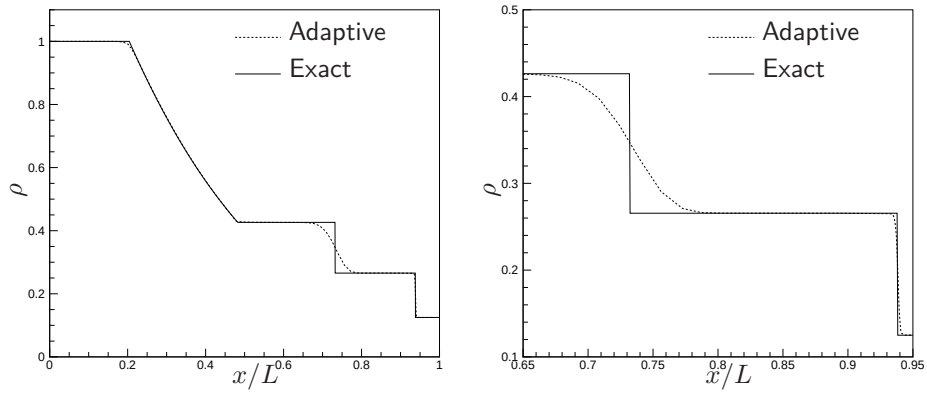


Fig. 4.11: Computational grids at various time levels, $\Delta t = 0.05$, obtained adapting on the substantial derivative of the density with a minimum allowable area of 3×10^{-7}. $t = 0$, 3797 nodes, 7401 elements; $t = 0.05$, 13035 nodes, 25765 elements; $t = 0.10$, 21542 nodes, 42708 elements; $t = 0.15$, 27171 nodes, 53905 elements; $t = 0.20$, 31007 nodes, 61535 elements; $t = 0.25$, 33899 nodes, 67280 elements.



(a) Density (b) Close-up

Fig. 4.12: Comparison between the density profile obtained adapting on the substantial derivative of the density and the exact solution, along the center line.

4.5 Forward facing step

The supersonic forward-facing step problem made famous by Woodward and Colella [199] is here presented. This test case is particularly well suited to test the solver capabilities in capturing very complex flow fields made by curved shocks that interacts with the wall and with other discontinuities. Supersonic conditions are imposed at the inlet, i.e. $M = 3$, slip conditions are imposed on the lower and upper boundaries of the duct and no conditions are imposed at the outlet. The initial solution is uniform and correspond to the one imposed at the inlet, namely $P = 1.0$, $\boldsymbol{m}/\rho = (3, 0)^{\mathrm{T}}$ and $\rho = 1.4$.

Two different computations have been carried out: a reference one with a uniform fixed grid of and an adaptive one. The fixed grid is made of 48324 points and 95618 elements, with an average element area of 2.7×10^{-6}. A second-order BDF scheme is used to integrate the equations with non-dimensional time-step of $1/300$, corresponding to a maximum Courant number of 1.67.

The adaptive procedure is performed starting from an initial grid made of 7603 nodes and 14799 elements, with an average element area of 4.33×10^{-5}. The minimum allowed element area for the refinement is 1×10^{-5}, which is roughly twice the nodes spacing of the fixed grid case. A Forward Euler scheme is adopted to integrate the equations in time and the non-dimensional time-step is again equal to $1/300$ corresponding to a maximum Courant number 2.67. The mixed Hessian/Gradient of the Mach number error estimator of equation (3.8) is adopted together with a multi-passage approach [117] with two levels. Due to the small value of the Courant number, the `FIAP` procedure is carried out at every time step without looping over the adaptation step s, i.e. using a three step procedure: solution prediction, adaptation and correction.

In fig. 4.17–4.20 the density distribution and the grid obtained the adaptive scheme are shown together with the fixed-grid and the reference solution [199] at four different time steps. In both the fixed and the adapted computations the front curved shock is very well captured and it appears to be sharper than the shock of the reference case. In the adapted case however the weaker shocks are not sufficiently highlighted by the

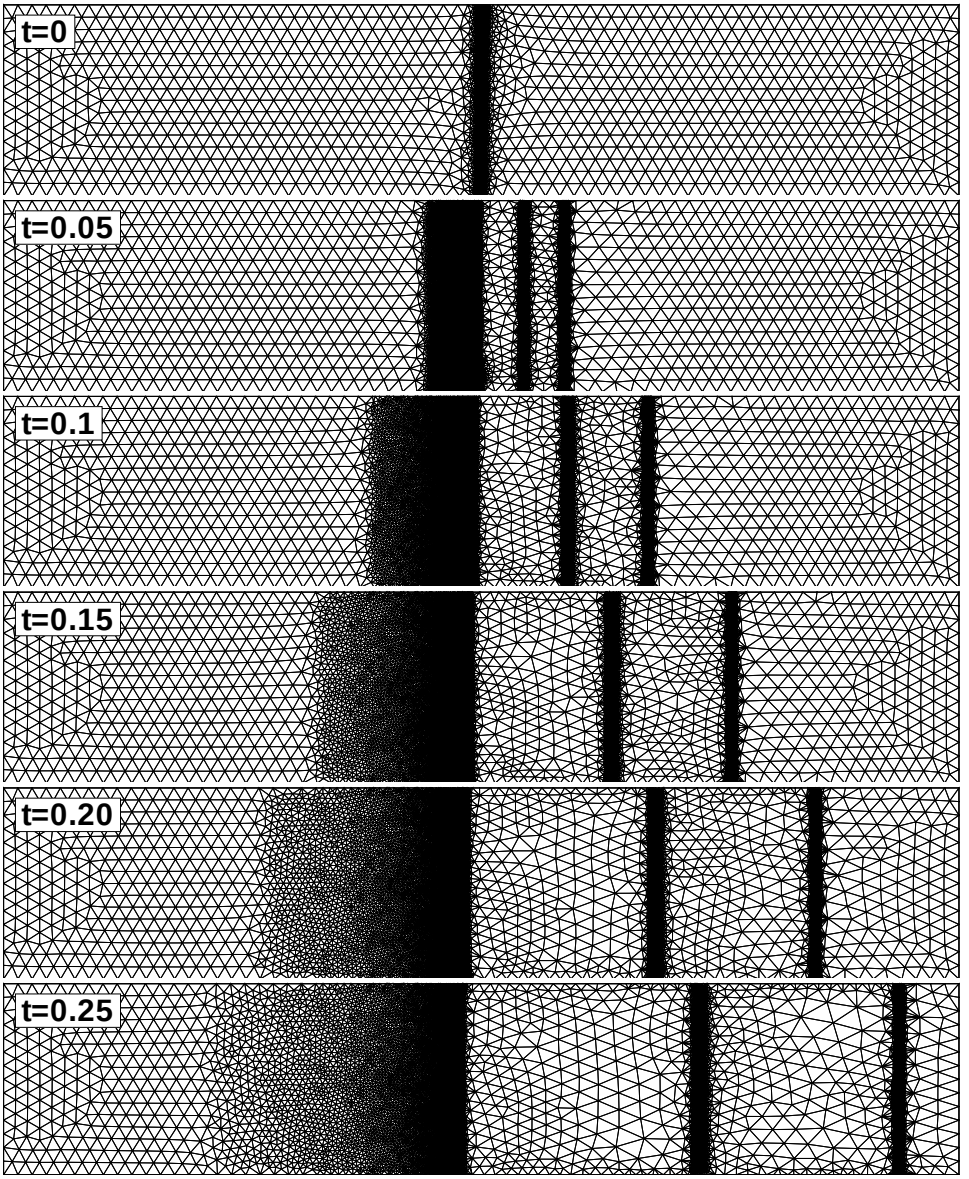


Fig. 4.13: Computational grids at various time levels, $\Delta t = 0.05$, obtained adopting the blended Gradient/Hessian of the Mach number as sensor with a minimum allowable area of 3×10^{-7}. $t = 0$, 3797 nodes, 7401 elements; $t = 0.05$, 12658 nodes, 25042 elements; $t = 0.10$ 13358 nodes, 26421 elements; $t = 0.15$, 12665 nodes, 25029 elements; $t = 0.20$, 11888 nodes, 23468 elements; $t = 0.25$, 10736 nodes, 21139 elements.

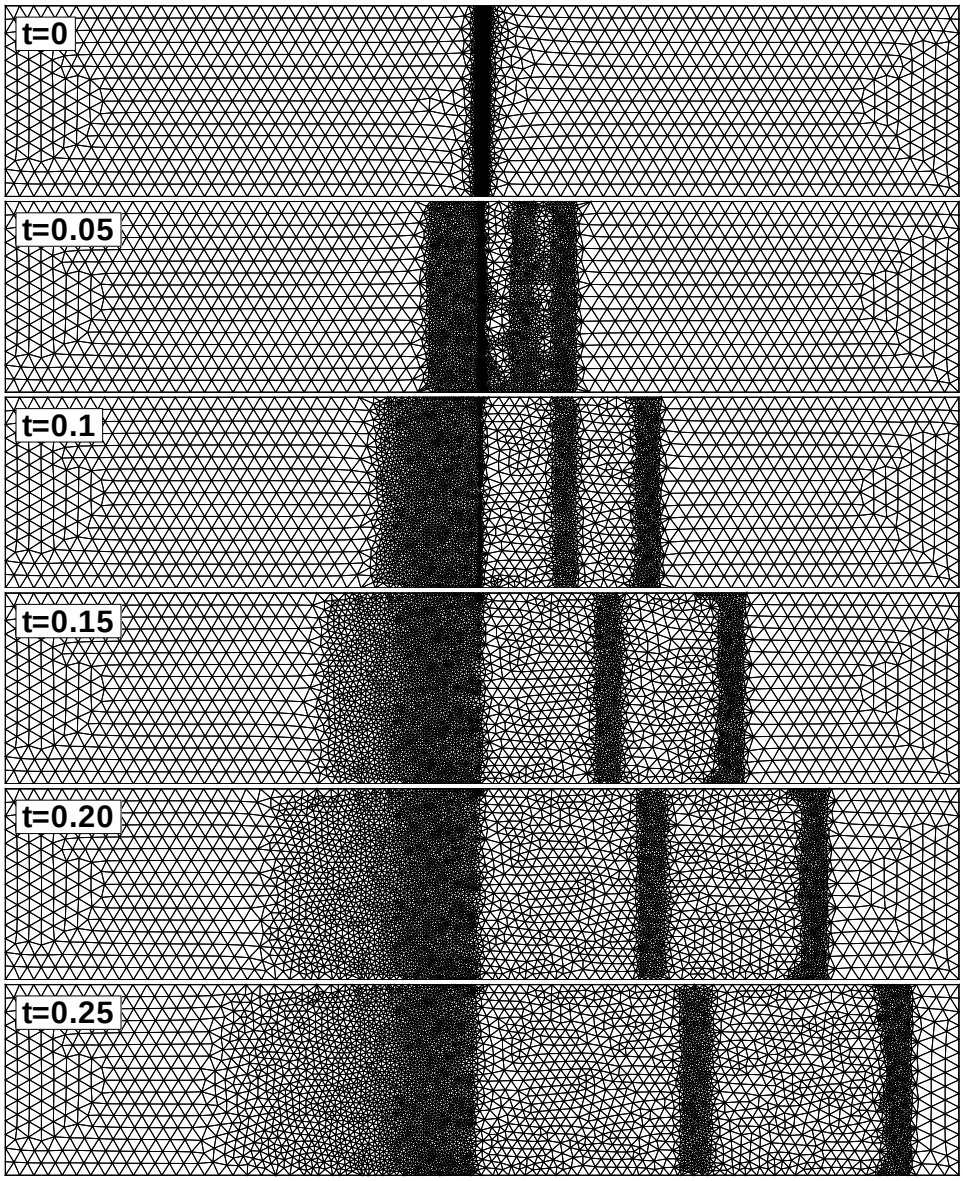
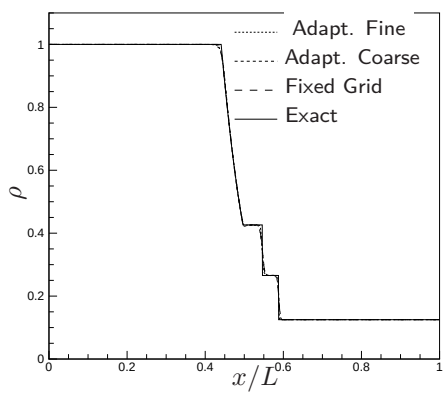
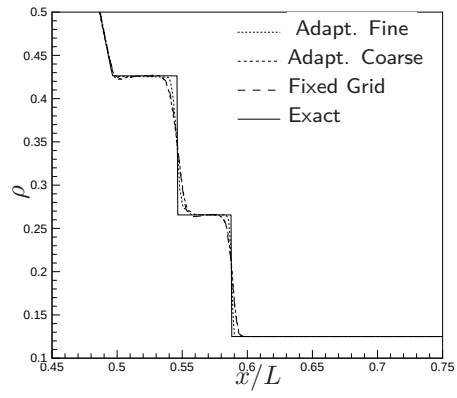


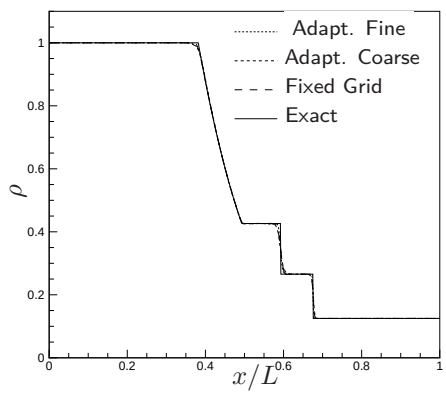
Fig. 4.14: Computational grids at various time levels, $\Delta t = 0.05$, obtained adopting the blended Gradient/Hessian of the Mach number as sensor with a minimum allowable area of 3×10^{-6}. $t = 0$, 3797 nodes, 7401 elements; $t = 0.05$, 4636 nodes, 9015 elements; $t = 0.10$, 4179 nodes, 8099 elements; $t = 0.15$, 4522 nodes, 8767 elements; $t = 0.20$, 4751 nodes, 9225 elements; $t = 0.25$, 4844 nodes, 9495 elements.



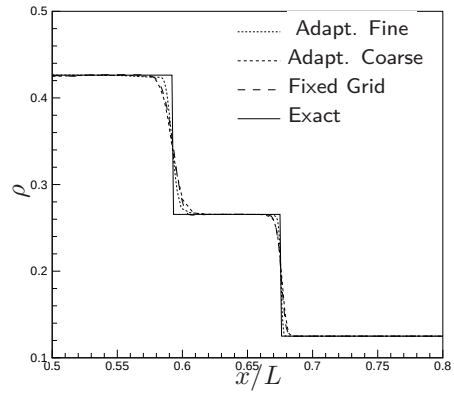
(a) $t = 0.05$



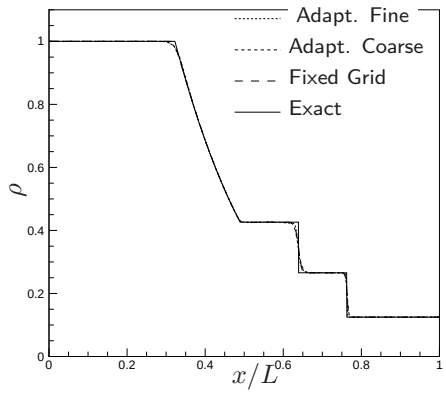
(b) $t = 0.05$ Close-up



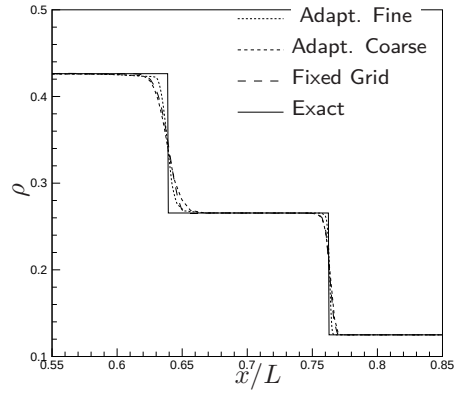
(c) $t = 0.10$



(d) $t = 0.10$ Close-up

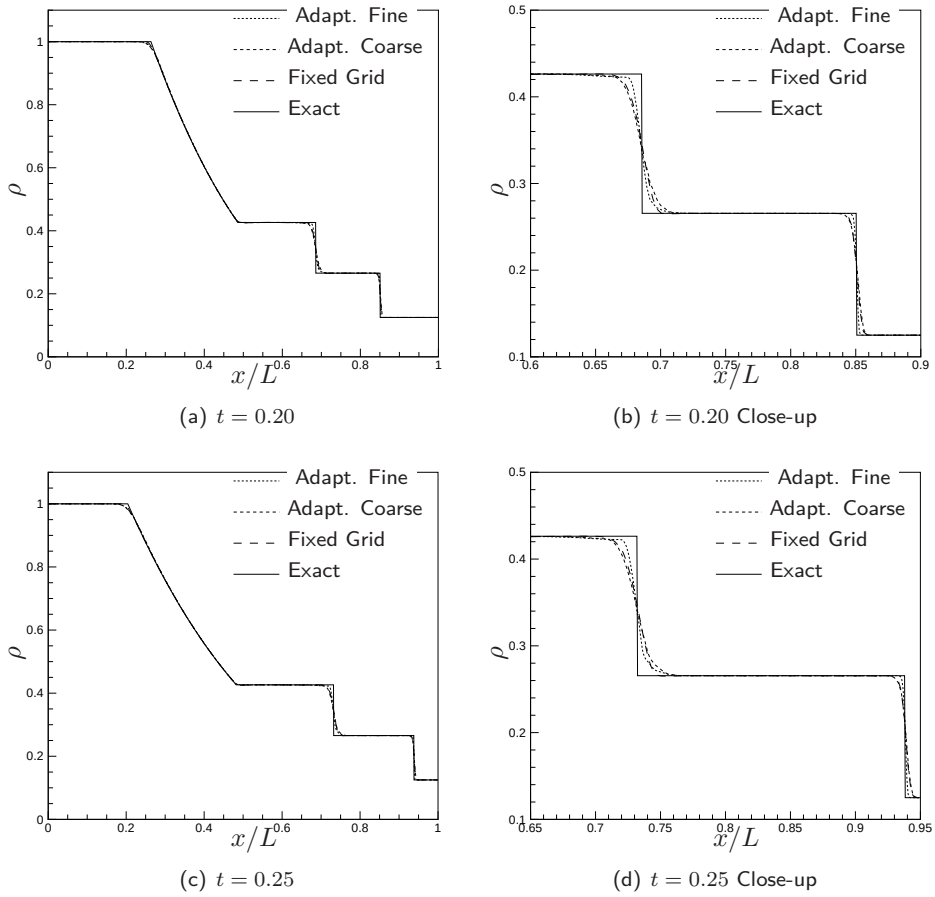


(e) $t = 0.15$



(f) $t = 0.15$ Close-up

Fig. 4.15: Comparison between the density profiles obtained adapting on the Hessian/Gradient of the Mach number, the fixed grid solution and the exact solution, along the center line.



(a) $t = 0.20$ (b) $t = 0.20$ Close-up

(c) $t = 0.25$ (d) $t = 0.25$ Close-up

Fig. 4.16: Comparison between the density profiles obtained adapting on the Hessian/Gradient of the Mach number, the fixed grid solution and the exact solution, along the center line.

sensor: for example the upper portion of the shock in fig. 4.5 and the discontinuity reflected by the lower boundary in figures 4.5 and 4.5 are significantly diffused. Even though the rarefaction fan is only slightly captured by the adaptation scheme, the overall solution does not seem to be penalized. Indeed, in all the presented cases, both the front and the reflected shocks are curved due to the interaction with the expansion fan.

As a final remark it has to be noted that the adapted grid computations resulted to be roughly ten times faster than the fixed grid ones on a single core machine. Moreover the total number of nodes required in the adaptive computation varies from 14% to 25% of the nodes used in the fixed case.

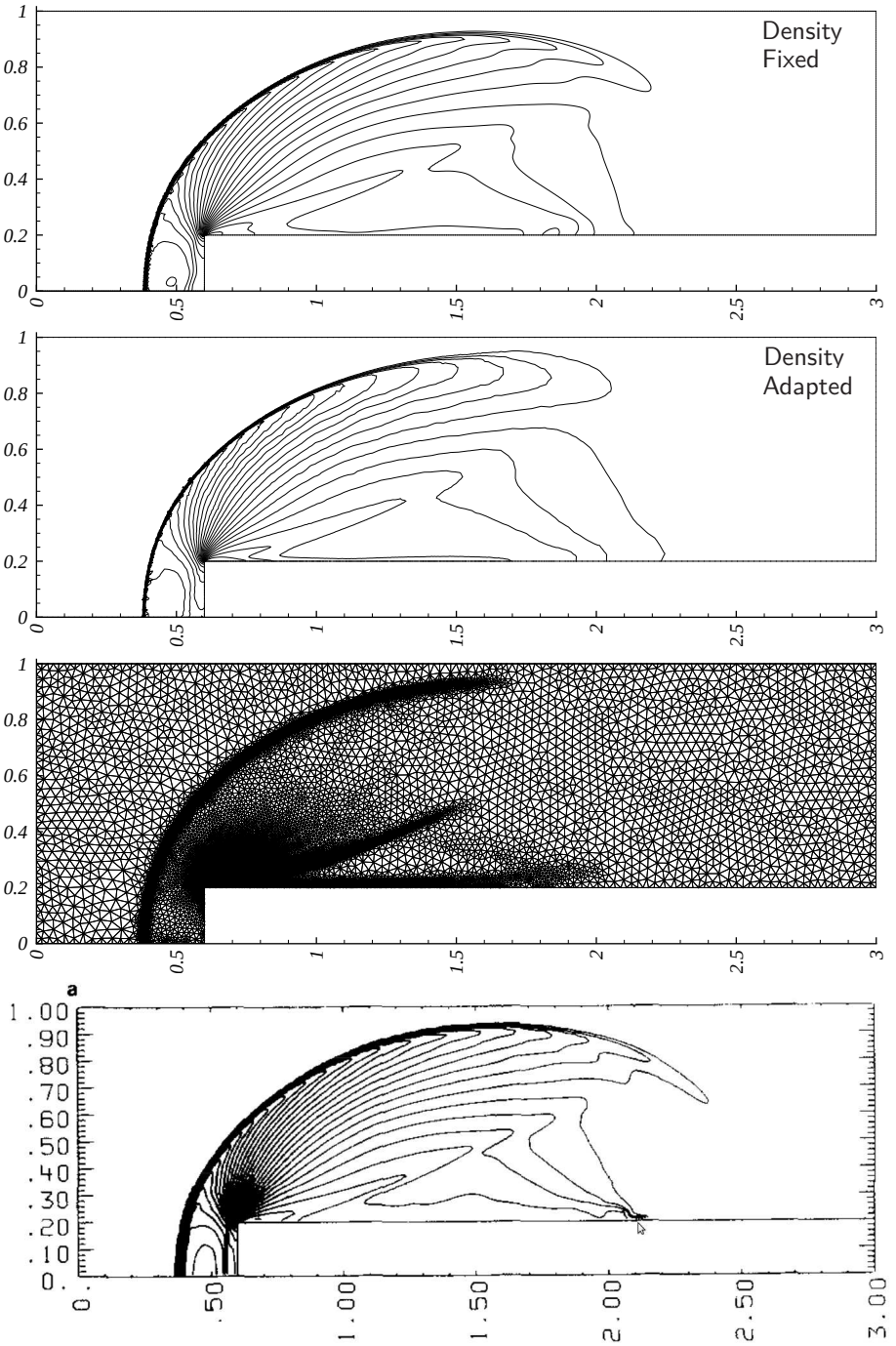


Fig. 4.17: Comparison of the density contours with the fixed grid computations and the reference [199] and computational grid at the non-dimensional time 0.500543.

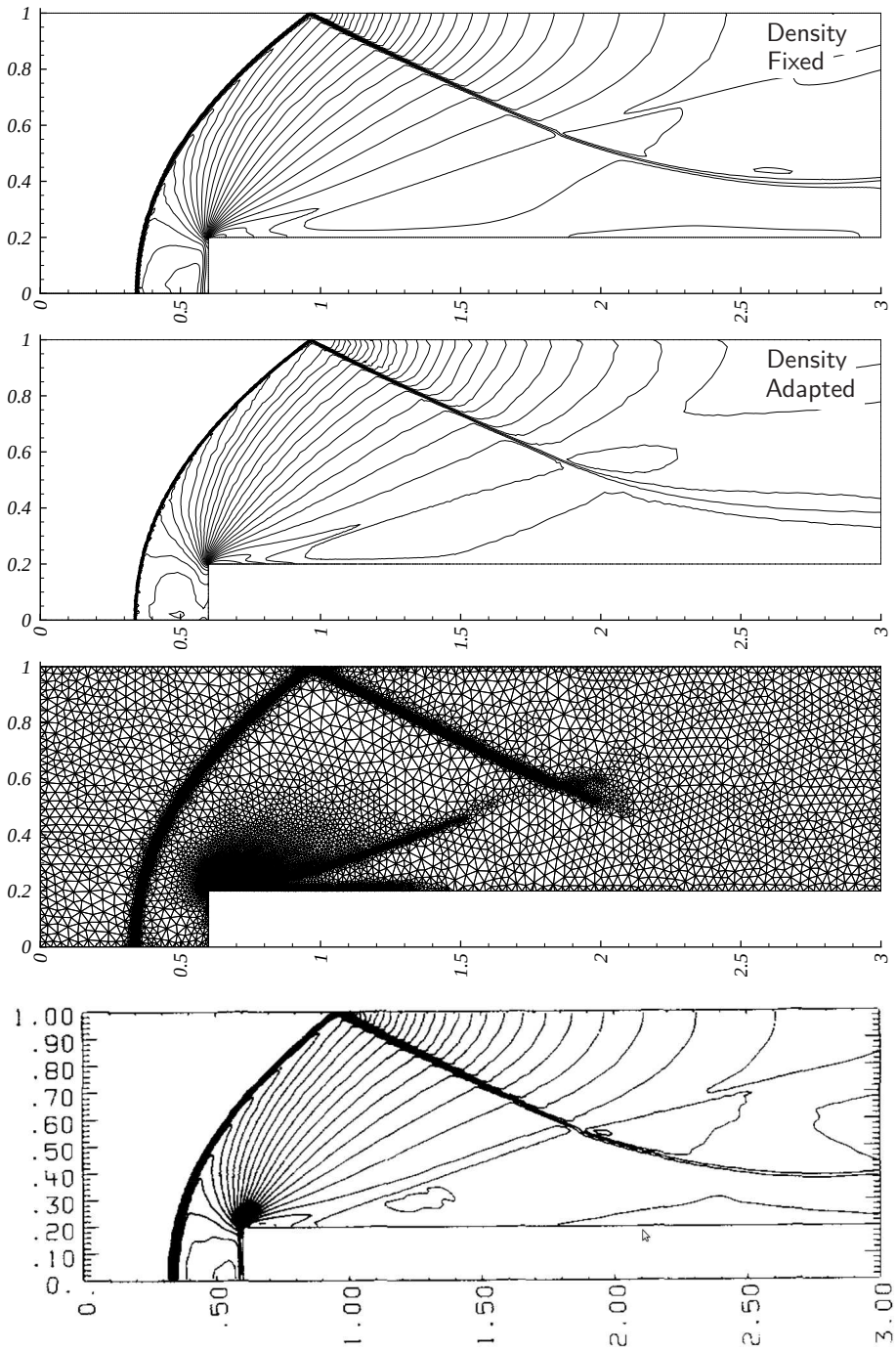


Fig. 4.18: Comparison of the density contours with the fixed grid computations and the reference [199] and computational grid at the non-dimensional time 1.00044.

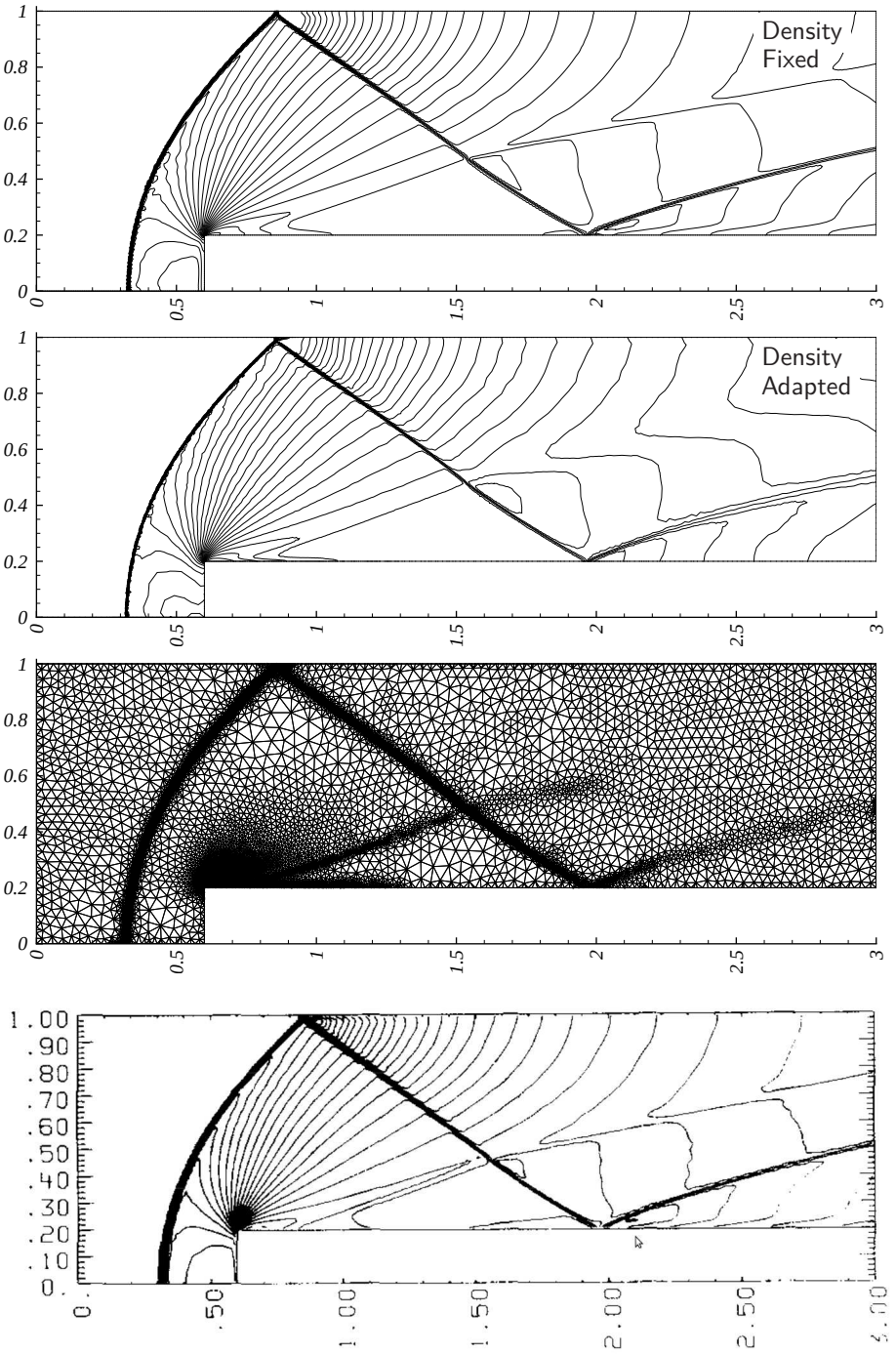


Fig. 4.19: Comparison of the density contours with the fixed grid computations and the reference [199] and computational grid at the non-dimensional time 1.50285.

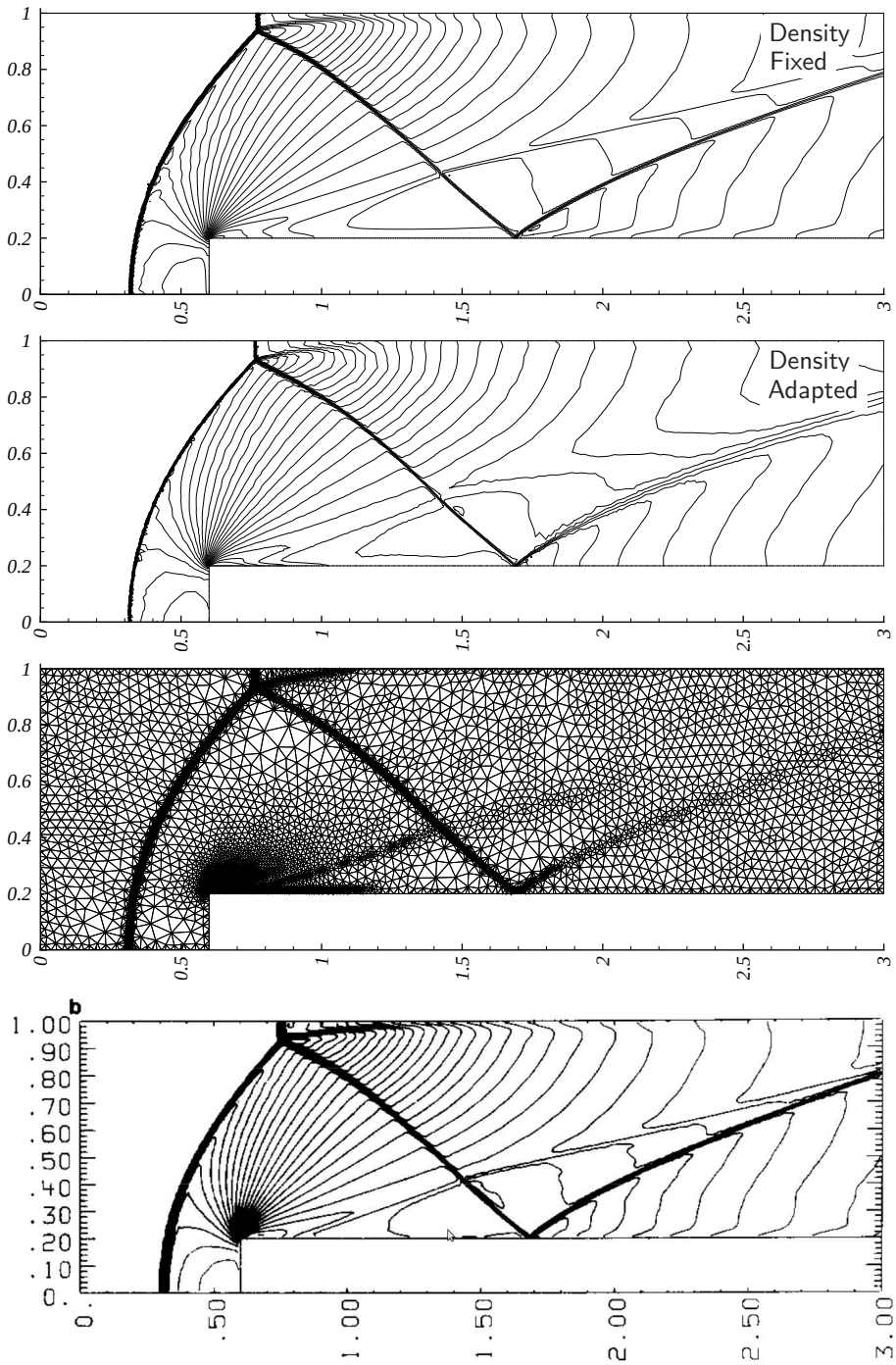


Fig. 4.20: Comparison of the density contours with the fixed grid computations and the reference [199] and computational grid at the non-dimensional time 2.00456.

Chapter 5

Simulations of compressible flows around rotorcraft blade sections

Helicopter simulation is a challenging problem due to the complexity of the flow field generated by the rotor disk, and the interaction between the vortices with the blades and fuselage.

The accurate computation of helicopter flow field in both hover and forward flights is indeed to be a complex task to tackle. Reliable prediction of helicopter performance strongly dependend on the accuracy of the prediction of the transonic flow past the blade on the advancing side of the rotor and on the resolution of blade-vortex and blade-wake interactions. Therfore a robust and accurate compressible CFD solver is essential in computing the flow around rotor blades.

An attempt to entirely simulate the main rotor system of a helicopter requires a multidisciplinary approach, involving coupling of the flow and structure models. In addition, either multi-block structured meshes or unstructured meshes are needed, and massive parallelization is a must for solving an entire helicopter including the fuselage and tail rotor. Recent comprehensive surveys of the current status of helicopter aerodynamics including both the theoretical and experimental work can be found in the article by Conlisk [42] and the book by Leishman [111] and by Friedmann [66]. While an extensive review on CFD for rotorcraft applications has been written by Caradonna [32].

Due to the increase in computational power of the last decades, simplified approaches to helicopter aerodynamics, e.g. momentum theory, blade element theory and actuator vortex theory, have been partially overtaken by CFD methods. The first rotor simulation has been carried out in 1987 by Agarwal & Deese [5], which solved the Euler equations accounting for rotor-wake effects by computing the local induced downwash with a free wake analysis method. Agarwal & Deese also performed the first RANS simulation in 1988 [6].

Srinivasan and McCroskey [171] performed Euler calculations of unsteady interaction of advancing rotor with a line vortex. The calculated results were compared to the two-bladed model helicopter rotor experiment by Caradonna and Tung [33] and consisted of

parallel and oblique shock interaction. Their results showed that subsonic parallel blade-vortex interaction was almost two-dimensional. However in the transonic regime, the 3D effects were found to be relevant. Boelens et al. [24] performed computations for one rotor blade in hover focusing on vortex wake prediction solving the arbitrary Lagrangian-Eulerian (ALE) formulation of the Euler equations over a mesh made of 823599 nodes and 726784 elements.

As briefly discussed in section 1.1, the stability of the majority of the CFD solvers is ensured by the introduction of a certain amount of numerical dissipation, which is generally proportional to the mesh size. To this purpose a method that captures the vortical structures in order to properly resolve a helicopter wake is crucial, since excessive numerical dissipation may lead to wrong preditction of the intensity of the wake or the vortices.

To this purpose the idea of Vorticity Confinement, first proposed by Steinhoff, has shown to be effective in treating concentrated vortical regions in coarse grids. This methods has also been applied to flows over airfoil [191], wings [119] and rotorcrafts [29], however it is also somewhat controversial since its anti-diffusive nature may lead to non-physical results [139]. Another approach that can be adopted consists in increase the grid resolution in the region of the domain that surrounds the vortex core. Tang and Baeder [175] used an ALE Euler solver to simulate the parallel blade-vortex interaction, where mesh a deformation algorith was used to gather mesh points close to the vortex. Oh, Kim, and Kwon [142] used adaptive grids to simulate the parallel BVI.

In the following sections typical two-dimensional problems for rotorcraft blade sections are presented. The idea is to apply the adaptive scheme outlined in chapter 3 and 2 to perform high-resolution computations involving pitching motion in section 5.1, airfoil start-up in section 5.2 and parallel blade-vortex interaction in section 5.3.

5.1 Pitching Airfoil

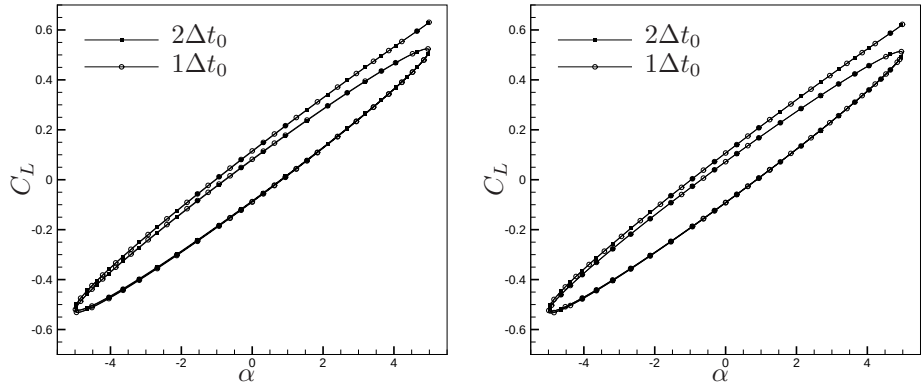
In the present section compressible inviscid subsonic and transonic flow computations around a pitching NACA 0012 airfoil are presented. The variation of the angle of attack in time is prescribed analytically as a sinusoidal function, namely

$$\alpha(t) = \alpha_0 \cos(pt), \qquad \text{with} \qquad 0 \leq t \leq \tfrac{4\pi}{p}, \tag{5.1}$$

where $\alpha(t_0) = 5^\circ$ is the initial angle of attack and $p = 0.1628$ is the dimensionless reduced frequency, obtained scaling the dimensional quantity with the asymptotic velocity U_∞ and the chord c_∞. The time variable t of Eq. (5.1) is scaled as described in section 2.1.1. The angle of attack of the airfoil with respect to the asymptotic flow is changed by rotating the airfoil around the nose point.

Simulations on both reference and adaptive grid have been performed. The reference grid is made of 29201 nodes, 57762 elements. A steady state solution is first computed at $\alpha = \alpha_0$ that is used as the initial one for the unsteady simulation during which the whole grid rotates rigidly with the airfoil, i.e. the ALE formulation is used to account for the movement but no deformation of the elements occur.

Similarly for the adaptive simulation a steady state is first obtained. During the steady state computation the `FIAP` procedure is used to adapt the grid to the mixed Hessian/Gradient of Mach number, with one level of multi-passage and $\tau_E = 0.01$. For the unsteady simulations the adaptation procedure outlined in fig. 3.11 is used, where



(a) Reference grid (b) Adaptive grid

Fig. 5.1: Lift coefficient versus angle of attack curves for the subsonic, i.e. $M_\infty = 0.3$, pitching NACA 0012 case obtained using two different time-steps.

the airfoil is moved, the grid is deformed consequently and then the grid is adapted. Moreover the convergence criterion is relaxed, setting $\tau_E = 0.05$.

For simplicity a reference non-dimensional time-step is defined as $\Delta t_0 = 0.3856$. Numerical simulations have been performed using a Forward Euler scheme and two different time-steps integration: $\Delta t = \Delta t_0$ and $\Delta t = 2\Delta t_0$, respectively 0.5% and 1% of the whole simulation time $T = 4\pi/p$. Due to the strong unsteadiness of the phenomenon, the adaptation procedure has been applied at each time-step.

In section 5.1.1 the subsonic case is presented and in section 5.1.2 the transonic one described.

5.1.1 Unsteady subsonic flow around the pitching NACA 0012 airfoil

The subsonic case, where the Mach number of the external flow is 0.3, is first discussed. The effect of the variation of the time step in terms of lift coefficient are first studied. The lift coefficient is defined as

$$C_L(t) = -\oint_{\partial\Omega,wall} C_p(s,t)\, n_y(s,t),$$

that in the finite volume representation of the equations becomes

$$C_L(t) = -\sum_{k\in\mathcal{K}_{\partial,wall}} C_p(\mathsf{u}_k,t)\, \hat{\xi}_{y,k}(t),$$

where $\hat{\xi}_{y,k}$ is the component of $\hat{\boldsymbol{\xi}}_k$ along the axis normal to the free flow. Fig. 5.1 shows the curves $C_L - \alpha$ obtained with the two different time-steps. Both the curves obtained over the reference grid, shown in fig. 5.1(a), and the ones obtained with the adaptive procedure, shown in fig. 5.1(b), suggests that the $C_L - \alpha$ curve are mildly sensible to

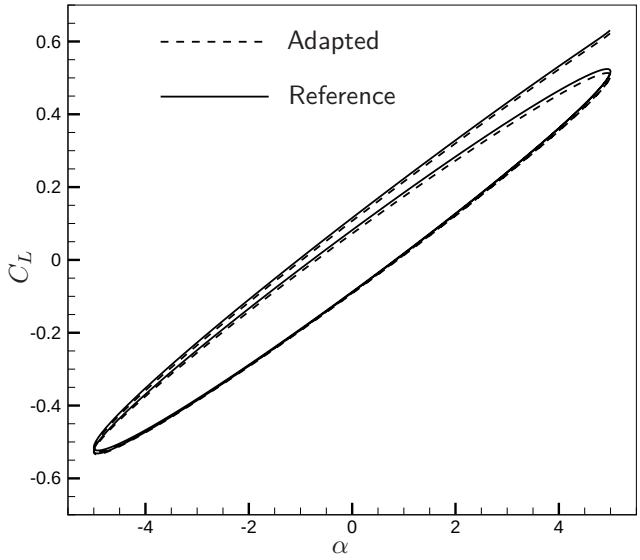


Fig. 5.2: Lift coefficient versus angle of attack curves for the subsonic, i.e. $M_\infty = 0.3$, pitching NACA 0012 case obtained using two different time-steps.

the time-step used. In fig. 5.2 the $C_L - \alpha$ curves obtained with fixed grid and adaptive procedure are plotted together; no relevant differences can be observed.

Fig. 5.3 shows the adaptive grid at different time-steps; at $t = 0$ the nodes are primarily gathered around the airfoil, and in particular near the leading edge where the Mach gradients are very strong. At successive time instants the pitching motion of the airfoil is responsible for a continuous release of vorticity from the trailing edge and the grid is adapted to refine the elements in the wake region. Indeed, estimating the interpolation error with the spatial variation of the Mach number allows also to capture the shear surfaces. In fig. 5.4 and 5.5, the Mach contour obtained using adaptive procedure and reference grid at different time-steps are shown. It can be observed how the intensity of the discontinuity across the wake is weak, compared to the rarefaction near the nose, however the use of the multipassage strategy allows to capture this feature as well.

The time required to run the simulations over a single core of an Intel Core^{TM}2 Duo T7500 processor at 2.20GHz is 14 hours, 17 minutes and 13 seconds for the computation on adaptive grid with $\Delta t = \Delta t_0$ and 80 hours, 35 minutes and 58 seconds for the computation on reference grid. A speed-up of roughly $\times 5.5$ is achieved with the adaptive strategy.

5.1.2 Unsteady transonic flow around the pitching NACA 0012 airfoil

The case of the compressible inviscid flow at $M_\infty = 0.755$ developing around a pitching NACA 0012 at is now discussed.

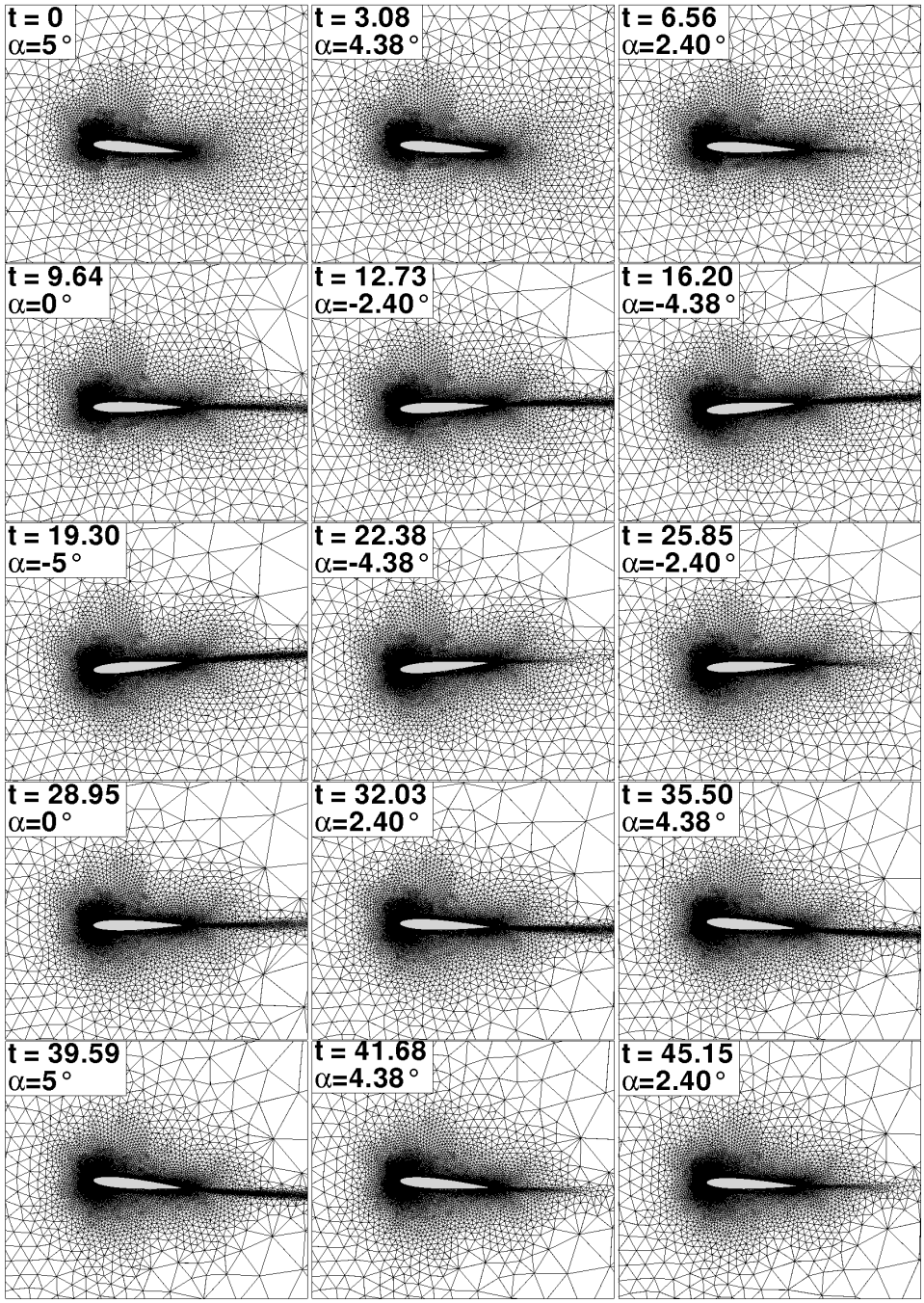


Fig. 5.3: Computational grid at several timesteps for the the subsonic, i.e. $M_\infty = 0.3$, pitching NACA 0012 case obtained with $\Delta t = \Delta t_0$.

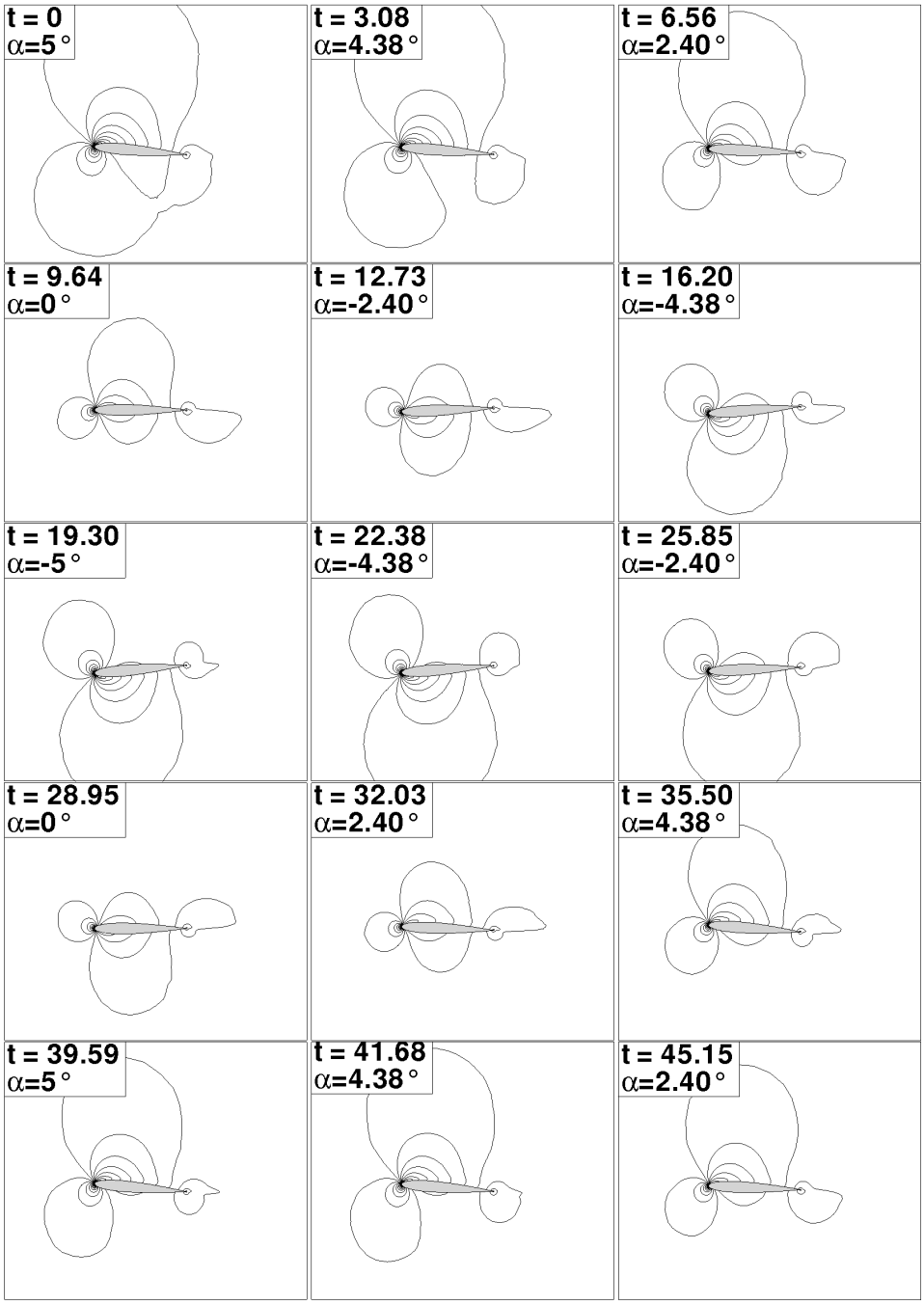


Fig. 5.4: Mach contour lines at several timesteps for the the subsonic, i.e. $M_\infty = 0.3$, pitching NACA 0012 case obtained with the adaptive scheme and $\Delta t = \Delta t_0$.

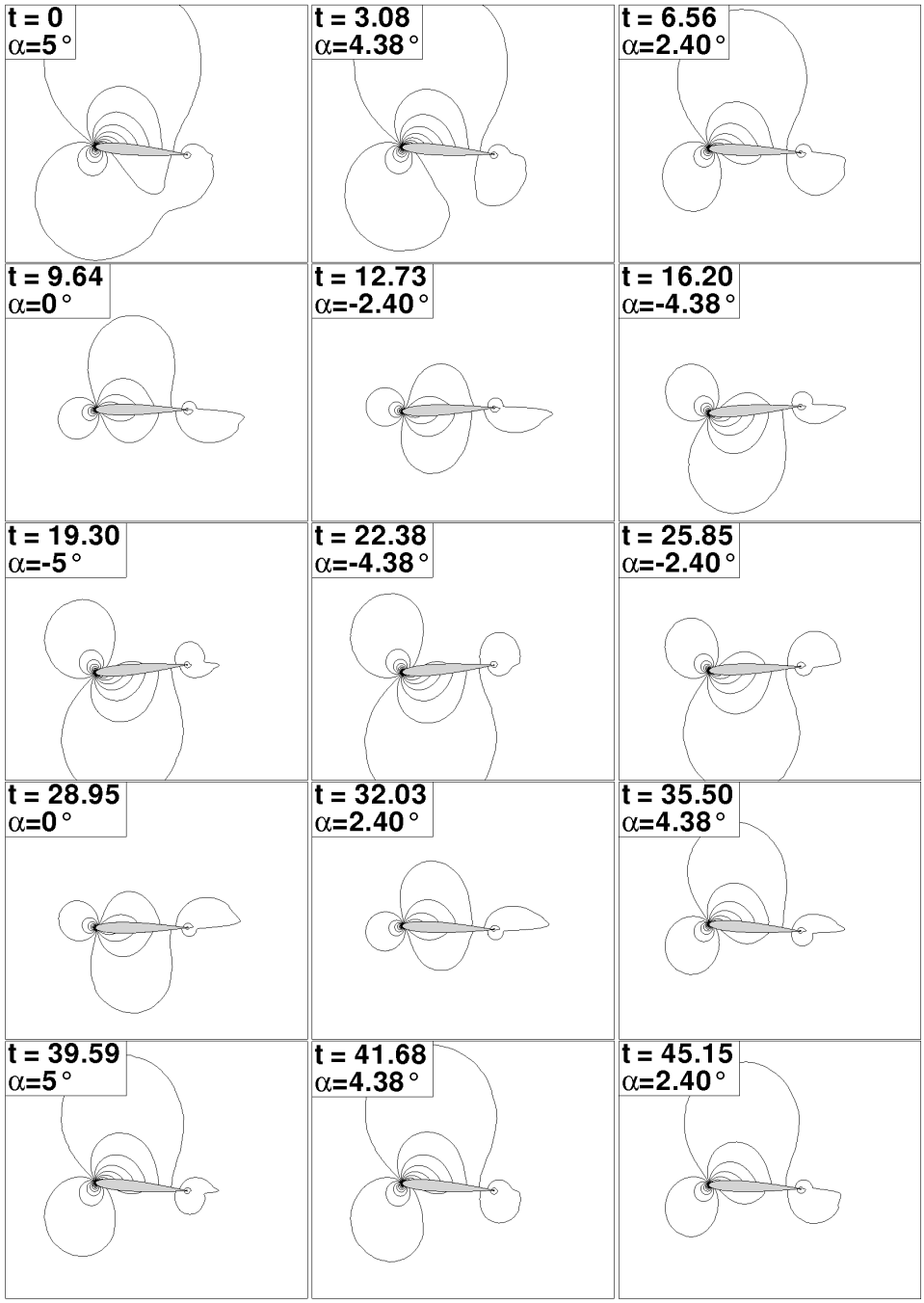
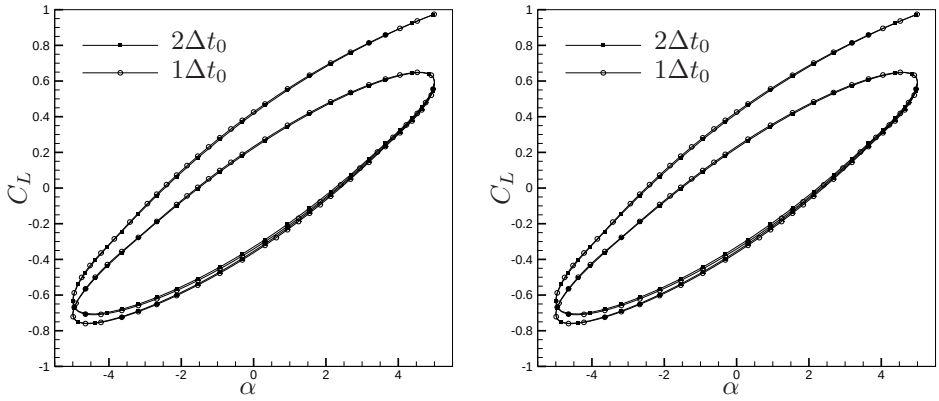


Fig. 5.5: Mach contour lines at several timesteps for the the subsonic, i.e. $M_\infty = 0.3$, pitching NACA 0012 case obtained with $\Delta t = \Delta t_0$ over the reference grid.



(a) Reference grid (b) Adapted grid

Fig. 5.6: Lift coefficient versus angle of attack curves for the transonic, i.e. $M_\infty = 0.755$, pitching NACA 0012 case obtained using two different time-steps.

As for the subsonic test case, a convergence study of the solution expressed in terms of the $C_L - \alpha$ curve has been conducted. Fig. 5.6 shows that the curves $C_L - \alpha$ are unrelated to the time-step used. Fig. 5.7 shows the curves $C_L - \alpha$ obtained with reference grid and adaptive procedure using $\Delta t = \Delta t_0$; compared to the previous cases the effects of the grid spacing on the curves are emphasized, possibly because of the presence of a shock wave in the flow-field.

Fig. 5.8 shows the adaptive grid at different time-steps t during the simulation. The grid obtain with the steady adaptation, i.e. fig. 5.8 ($t = 0$), finely captures the shock wave, the contact discontinuity starting from the trailing edge, and the leading edge rarefaction.

The unsteadiness of the phenomenon is very well captured by the scheme, indeed the reduction/growth in terms of intensity of the shock and the shear wake are followed by the grid. The `FIAP` procedure is therefore fast enough to follow the variation of the flow field with the given time step. In fig. 5.8, however, the dependency of a grid from the grid used at the previous time steps is still visible. Indeed the movement of the the shock wave from the upper to the lower side of the airfoil causes an over refinement in the region fore of the shock itself, e.g. fig. 5.8 ($t = 25.85$).

In fig. 5.9 and 5.10, the Mach contour obtained using adaptive procedure and reference grid at different time-steps are shown. Compared to the fixed-case, both the shocks and the shear wake obtained over the adaptive grid are sharper.

The CPU time required to run the simulations on a single core of an Intel Core^{TM}2 Duo T7500 at 2.20GHz with $\Delta t = \Delta t_0$ is 5 hours, 27 minutes and 47 seconds for the computation on adaptive grid, while 22 hours, 52 minutes and 47 seconds for the computation on reference grid. A speed-up of roughly $\times 4$ is achieved with the adaptive strategy.

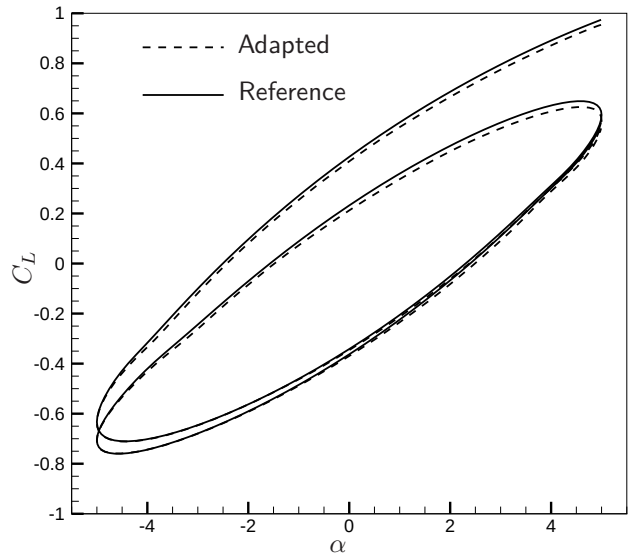


Fig. 5.7: Confrontation of the lift coefficient versus angle of attack curves for the transonic, i.e. $M_\infty = 0.755$, pitching NACA 0012 case obtained using time-step $\Delta t = \Delta t_0$ over the reference and the adaptive grid.

5.2 Start-up vortex from the NACA 0012 airfoil

The proposed methodology is now applied to the computation of the 2D unsteady flow past the NACA 0012 airfoil impulsively set into motion. Since the generation of the vortical structure is not due to the viscous effects but to the singularity in the geometry of the body, Euler equations can be used to correctly represent the space-time evolution of the start-up vortex in the computational domain.

As a consequence of lift generation over the airfoil, a trailing-edge vortex is produced. This vortex is named start-up vortex and its dynamics strongly influence the time history of the force coefficient over the airfoil. Since a clockwise circulation around the airfoil is generated, the starting vortex is associated to a counterclockwise circulation. As the distance between the airfoil and the start-up vortex increases due to their relative motion, the influence of the latter on the aerodynamic coefficients vanishes. Eventually, the steady-state value of lift coefficient $C_L(t)$ and circulation is attained.

The influence of the start-up vortex dynamics on the airfoil is very difficult to investigate numerically. Since both the flow field around the airfoil and the start-up vortex need to be captured accurately at the same time, indeed the circulation around the airfoil and the intensity of the start-up vortex balances instantaneously, due to the Kelvin's circulation theorem. It is therefore mandatory to adapt the computational grid in an unsteady fashion to follow the vortex dynamics and, at the same time, accurately compute the flow field close to the airfoil.

In the numerical tests the free stream velocity is a step function that starts at zero and reaches its asymptotic value V_∞ at the time $t > 0$, i.e. a uniform flow at V_∞ is used

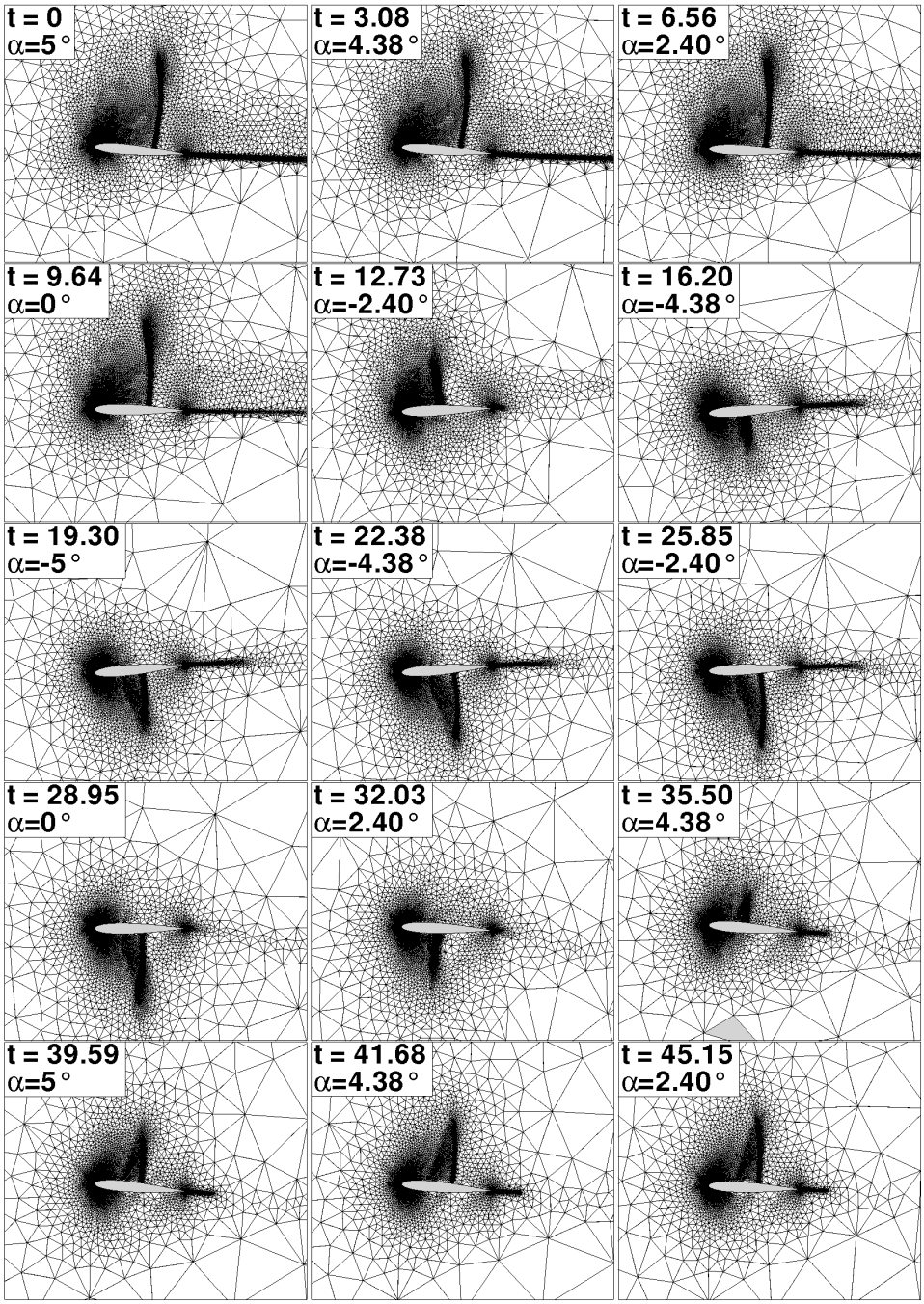


Fig. 5.8: Computational grid at several timesteps for the the transonic, i.e. $M_\infty = 0.755$, pitching NACA 0012 case obtained with $\Delta t = \Delta t_0$.

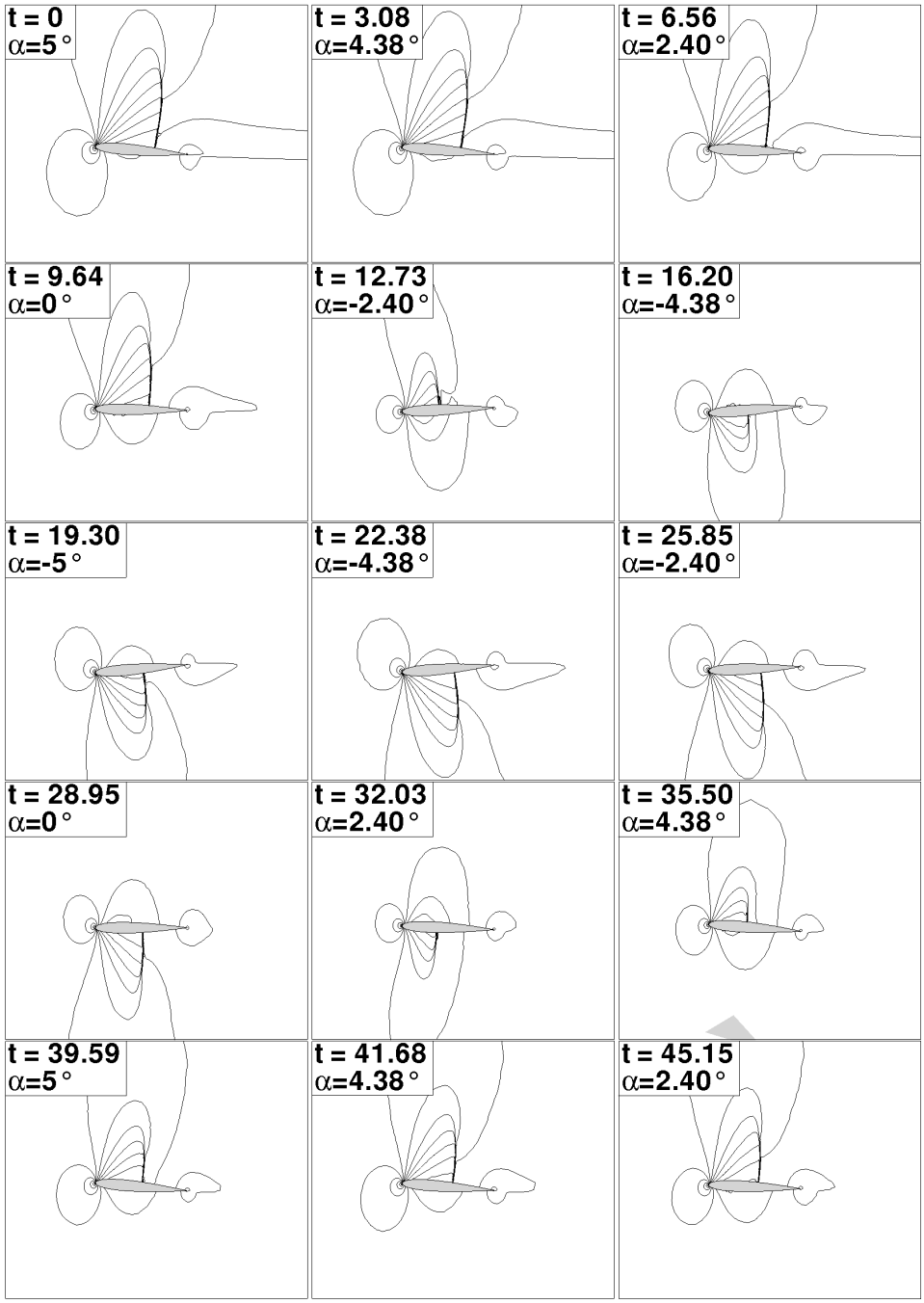


Fig. 5.9: Mach contour lines at several timesteps for the the transonic, i.e. $M_\infty = 0.755$, pitching NACA 0012 case obtained with the adaptive scheme and $\Delta t = \Delta t_0$.

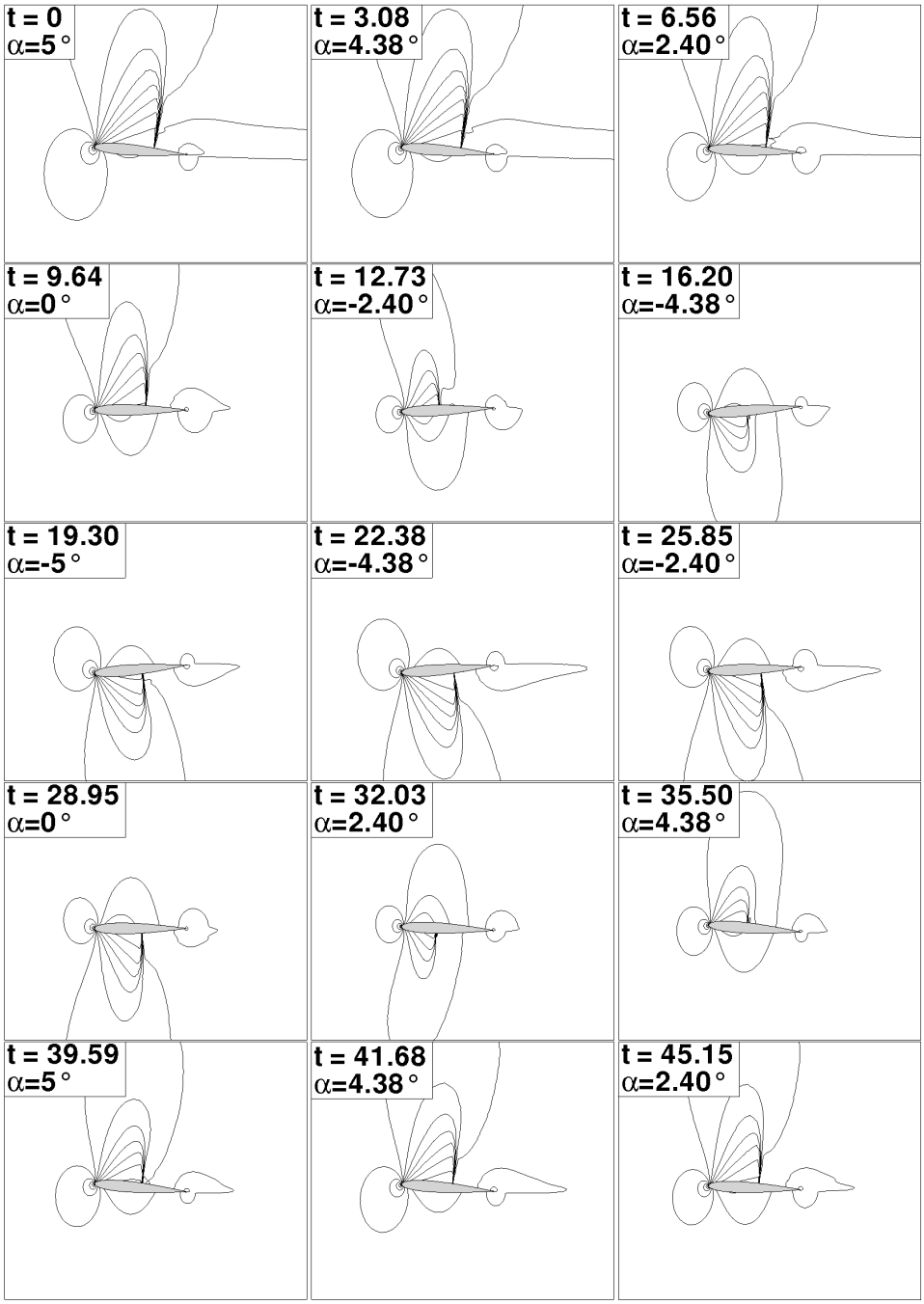
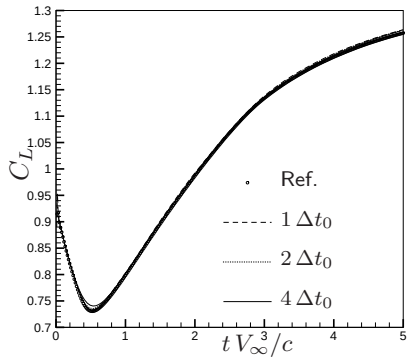
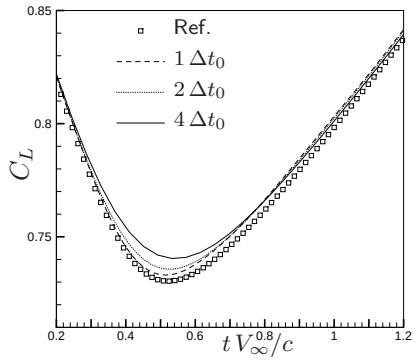


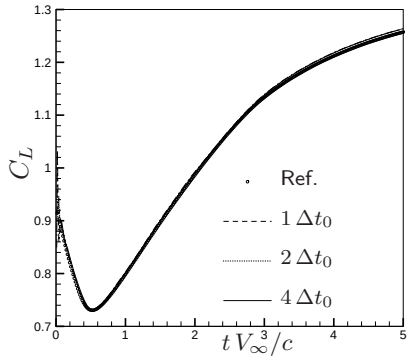
Fig. 5.10: Mach contour lines at several timesteps for the the transonic, i.e. $M_\infty = 0.755$, pitching NACA 0012 case obtained with $\Delta t = \Delta t_0$ over the reference grid.



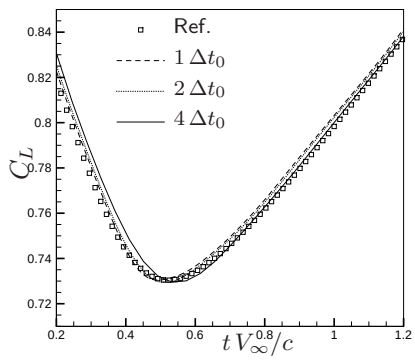
(a) BDF1 Scheme



(b) BDF1 Scheme (close-up)



(c) BDF2 Scheme

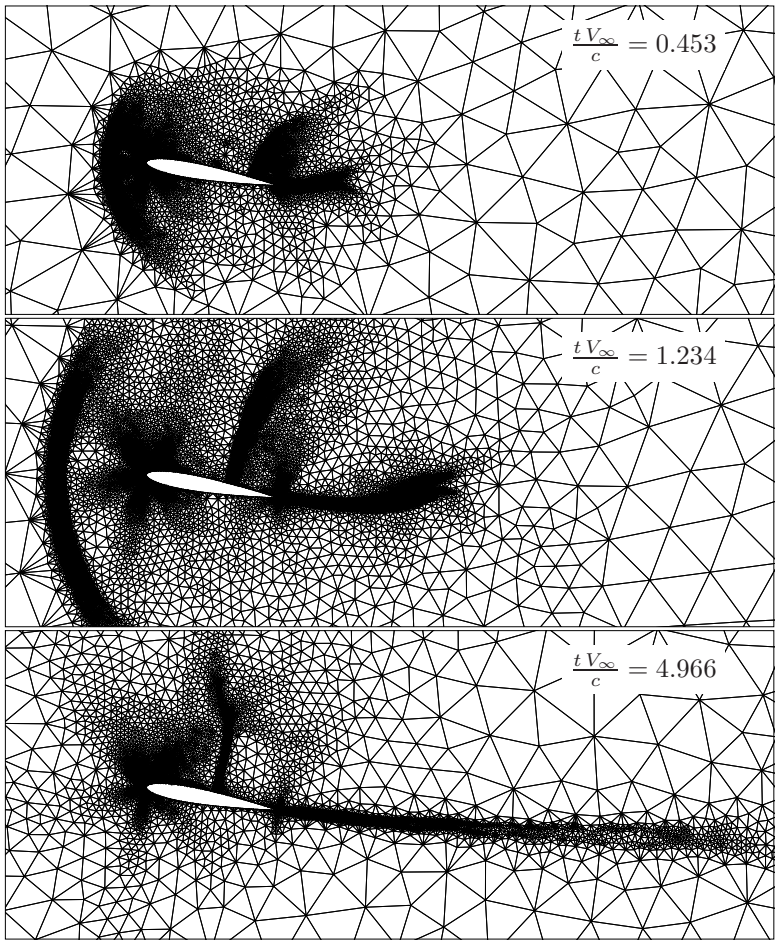


(d) BDF2 Scheme (close-up)

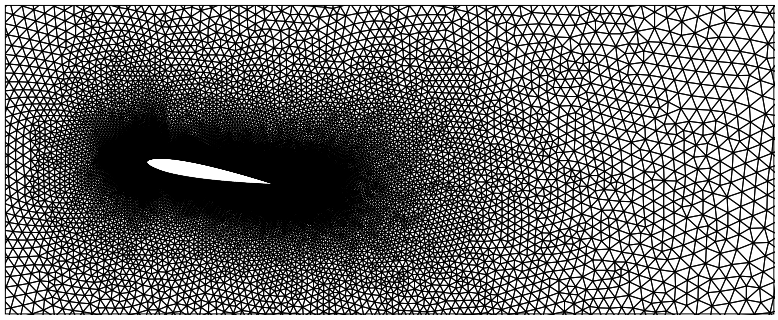
Fig. 5.11: Impulsively-started NACA 0012. Mach $= 0.7$, $\alpha = 10°$: lift coefficient time history for the reference grid against solution over the initial coarse grid for different values of the time step using the one-step BDF1 (top) and the two-step BDF2 (bottom) schemes (c). In (b) and (d) an enlarged view of the time interval across the minimum C_L is shown.

as initial condition. In the computations, the time variable t is made dimensionless by the free-stream velocity V_∞ divided by the airfoil chord c. The considered dimensionless time interval is $\frac{t\,V_\infty}{c} \in [0, 5]$.

Tests were performed for a free-stream Mach number $M_\infty = 0.7$ and for $\alpha = 10°$ incidence. Simulations on both adapted and fixed grid were carried out. The fixed grid used to compute the reference solution is made of 29367 nodes and 58094 elements. The adaptation process starts from an initial grid made of 4989 nodes and 9627 elements and the `FIAP` procedure is performed at every time-step until a the variation of μ_w is lower than the 5% of the initial value. Two computations with different thresholds for the minimum area of the elements, namely, 3×10^{-5} and 3×10^{-4} have been carried out. The error is evaluated as the Gradient/Hessian of the local Mach number.



(a) Adapted grid



(b) Reference grid

Fig. 5.12: Impulsively-started NACA 0012 for Mach = 0.7, $\alpha = 10^\circ$: close-up of the computational grids at three different time levels (a) and reference grid (b).

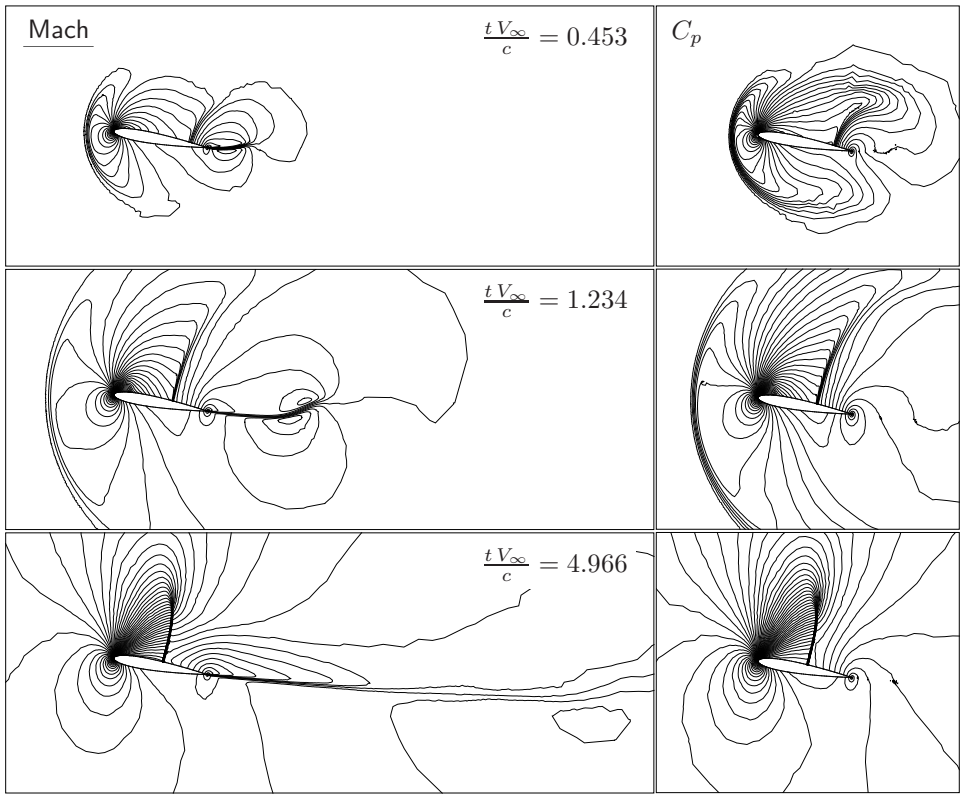


Fig. 5.13: Impulsively-started NACA 0012 for Mach = 0.7, $\alpha = 10^\circ$: contour line of the Mach number and of the pressure coefficient C_P at different time levels computed with the adaptive scheme.

First, to assess time convergence, a comparison between the solution obtained over the reference grid and the initial one, without the adaptive scheme, have been carried out. Time integration over the reference grid were performed using the BDF1 scheme with $\Delta t = 0.0164$, while both the first and second order BDF schemes were used over the initial grid for $\Delta t = \Delta t_0 = 0.0103$, $\Delta t = 2\Delta t_0$, $\Delta t = 4\Delta t_0$. Results for the lift coefficient C_L as a function of the non dimensional time are reported in fig. 5.11 and demonstrate time convergence to the reference solution for the smallest time step for both the BDF1 and BDF2 schemes. For BDF2, the solution is almost indistinguishable from the reference also for $\Delta t = 2\Delta t_0$. Therefore, the simulations were run using the `FIAP` adaptation procedure with $\Delta t = \Delta t_0$ and the BDF1 time integration scheme.

Fig. 5.12 shows the adapted grids for the simulation with the minimum element area equal to 3×10^{-5} at three different dimensionless time levels. These correspond to the minimum of the lift coefficient ($t\,V_\infty/c = 0.453$, 9213 nodes and 18196 elements), a condition where the influence of the start-up vortex begins to be negligible ($t\,V_\infty/c = 1.234$, 8360 nodes and 16494 elements) and the final configuration ($t\,V_\infty/c = 4.966$, 16986 nodes and 33733 elements), respectively. Fig. 5.13 shows the Mach number and the pressure coefficient C_P in the flow field at the three time levels. The $C_P - t$ profile

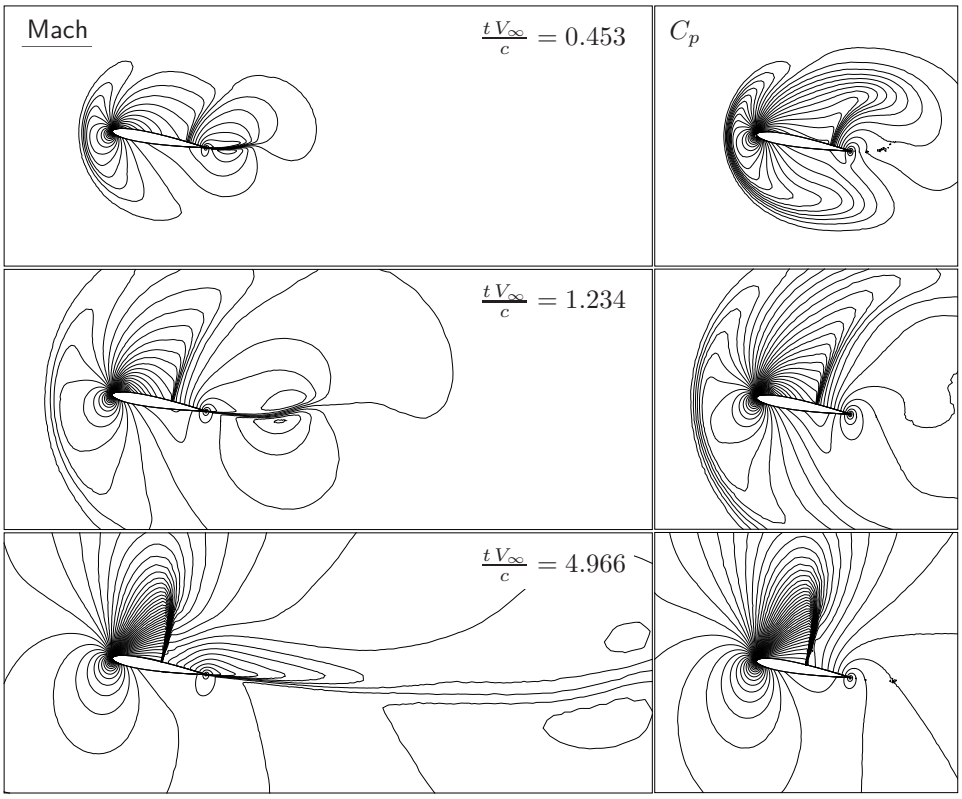


Fig. 5.14: Impulsively-started NACA 0012 for Mach $= 0.7$, $\alpha = 10°$: contour line of the Mach number and of the pressure coefficient C_P at different time levels over the reference grid.

over the airfoil is depicted in fig. 5.15 for the three considered time values. The airfoil lift coefficient is plotted as a function of the dimensionless time in fig. 5.16.

A shock wave is formed on the top of the airfoils and terminate a supersonic region. The shock wave evolution is nicely captured by the adapted grid scheme, whereas in the reference dense grid computations the shock is smeared far from the airfoil. The start-up vortex is generated at $tV_\infty/c = 0^+$; its evolution is well captured by the adaptive scheme, which is also capable of refining the grid close to the slip line from the trailing edge. With reference to fig. 5.12, in adapted grid simulations a large number of nodes is added to capture the pressure wave in front of the airfoil which is clearly visible at time $tV_\infty/c = 1.234$); this accounts for the higher computational time. The propagation of this perturbation does not affect significantly the overall value of the lift coefficient, which is used here for comparisons. The above can be appreciated in fig. 5.17, where the total number of grid nodes at each adaptation step is shown. In the finer grid case, a large number of nodes is inserted during adaptation cycles between 100 and 200 to capture the pressure wave dynamics. The CPU time required to run each simulation is 9 hours, 19 minutes and 51 s for the reference simulations, which corresponds to an average of 110s for time step (305 time steps total), 3 hours, 21 minutes and 29 s (25 s

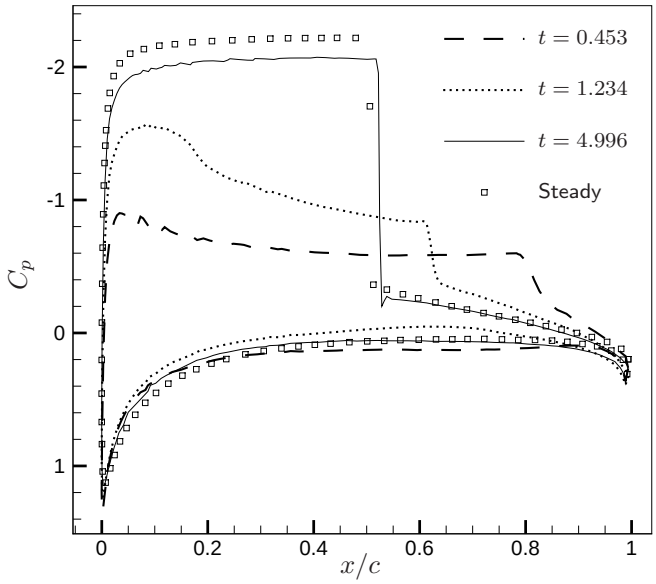


Fig. 5.15: Impulsively-started NACA 0012. Mach = 0.7, $\alpha = 10°$: pressure coefficient over the airfoil at different time level for adapted grids.

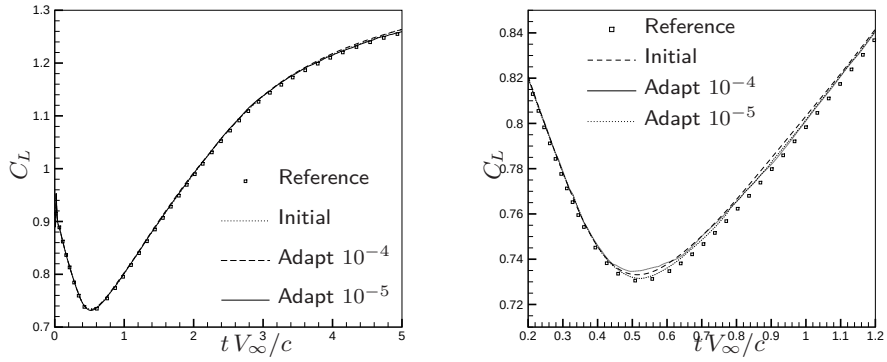
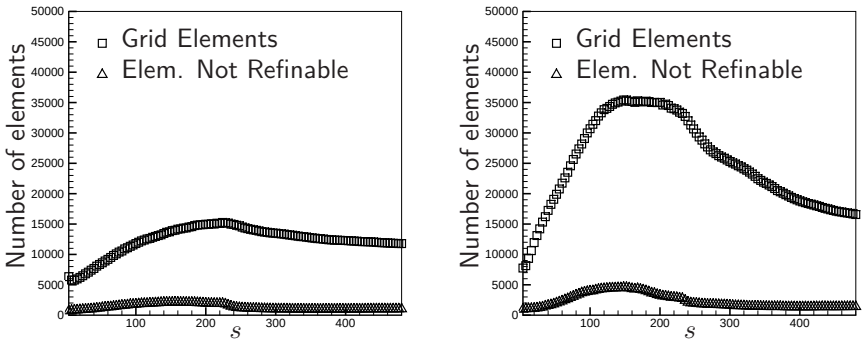


Fig. 5.16: Impulsively-started NACA 0012. Mach = 0.7, $\alpha = 10°$: lift coefficient time history for the reference grid against adapted grids with minimum reference length of 3×10^{-5}.



(a) $h_{\min} = 10^{-4}$ (b) $h_{\min} = 10^{-5}$

Fig. 5.17: Mach = 0.7, $\alpha = 10°$: number of grid elements and of not refinable elements as a function of the adaptation step for element size limit of 10^{-4} (left) and 10^{-5} (right).

$\times$ 486) for the adapted-grid simulation with minimum element area of 3×10^{-4} and 14 hours, 18 minutes and 44 s (106 s $\times$ 486) for a minimum area of 3×10^{-5} on a single core of an Intel Xeon QuadCore 3.166GHz. The adoption of the adaptation procedure is found not to influence the (average) computational effort per time step.

5.3 Parallel blade-vortex interaction

In the present section the parallel blade-vortex interaction problem is tackled. The BVI phenomenon occurs when a rotor blade passes within a close proximity of the shed tip vortices from a previous blade. This causes a rapid, impulsive change in the pressure distribution along the blade resulting in the generation of highly directional impulsive loading noise. It has been shown that the main parameters governing the strength of a BVI are the distance between the blade and the vortex, termed miss-distance, the vortex strength at the time of the interaction, and how parallel or oblique the interaction is [81, 123]. The parallel BVI is the most critical configuration and occurs when the axis of the filament-like vortex is aligned with the axis of the blade.

Due to the high aspect-ratio of conventional blades a parallel BVI problem can generally be reduced to a two-dimensional airfoil-vortex interaction (AVI). This approach is ideal to study the underlying physical mechanisms involved in the interaction as it removes many of the complications of a three-dimensional BVI simulation and is computationally less expensive. One of the major challenges faced when simulating an AVI is to preserve the vortex structure accurately as it convects through the solution and minimize the numerical dissipation that is inherent in CFD simulations. The work of Oh et al. [142] addressed this problem by the use of adaptive unstructured meshes to simulate a two-dimensional AVI. This method dynamically concentrates mesh points in region of large flow gradients, providing high resolution in the region of any vortices and other important flow features. Excellent results were achieved in this study and a similar approach has been adopted here.

To better study the capabilities of the adaptive flow solver the simulation vortices

transport within the flow field, the free-vortex advection problem is first tackled in section 5.3.1 on both fixed and adaptive grids. The airfoil-vortex interaction problem is presented in section 5.3.2.

5.3.1 Free vortex advection

The case of the advection of a vortex in an horizontal flow is presented. A two dimensional vortex is represented by the Bagai-Lieshman compressible vortex [9]. The so called $n = 1$ Scully [161] model is used for the velocity field, namely

$$\frac{m_\theta(\hat{r})}{\rho(\hat{r})} = \frac{2\,\hat{r}}{(1+\hat{r}^2)} M_c\, c_\infty, \tag{5.2}$$

where $\hat{r} = |\boldsymbol{x}|/r_c$, r_c is the vortex core radius, M_c is a reference value for the vortex core Mach number and c_∞ is the value the speed of sound for $\hat{r}$ that goes to infinity. As it is commonly done in the literature the vortex core Mach number can be expressed in terms of the vortex intensity Γ_c and radius r_c, namely $M_c = \Gamma_c/(4\pi\, r_c\, c_\infty)$.

Following Bagai and Lieshman [9], the density and pressure field are computed from the radial momentum component of the compressible Navier-Stokes equations for an isoentropic flow and an ideal gas, namely

$$\rho(\hat{r}) = \rho_\infty \left(1 - 2\frac{\gamma-1}{1+\hat{r}^2} M_c^2\right)^{\frac{1}{\gamma-1}}, \tag{5.3}$$

and

$$P(\hat{r}) = \frac{c_\infty^2 \rho_\infty}{\gamma} \left(\frac{\rho(\hat{r})}{\rho_\infty}\right)^\gamma \tag{5.4}$$

where ρ_∞ is the density value far away from the vortex.

The compressible vortex defined above is then inserted in a uniform horizontal flow which is completely defined by the Mach number M_∞, the density ρ_∞ and the momentum modulus m_∞. Indeed, the non-dimensional speed of sound of Eq. (5.2) is therefore given by $c_\infty = \frac{m_\infty}{M_\infty \rho_\infty}$. In the present work a unit value has been chosen for both the free-flow density and momentum, thus only the free-flow and vortex Mach number are used to completely define the flow field.

Fixed grid computations

The finite-volume scheme is first tested over the compressible vortex advection case. The free flow Mach number is 0.8, the vortex Mach number is 0.2 and $r_c = 0.1$ grid units. The lower half of the fixed computational grid is shown in fig. 5.18 together with the upper half of the density contour lines. The grid dimensions are $240\, r_c \times 120\, r_c$ and it is made of 61015 nodes and 121722 elements. The far field boundary conditions described in section 2.3.2 are imposed on every side of the rectangular domain, where the far field state u_∞ is taken as the exact solution to the problem, i.e. the rigid displacement of the vortex along the horizontal axis with velocity $M_\infty c_\infty$.

To test the time-convergence properties of the scheme unsteady computations have been carried out for different values of the Courant number and with different time

schemes, i.e. BDF scheme of order 1, 2 and 3. The global Courant number is computed as

$$\mathrm{Co} = \frac{\boldsymbol{m}_\infty}{\rho_\infty}\frac{\Delta t}{h_{min}},$$

where $h_{min} = 0.005$ is the smallest edge of the grid, and ranges between 0.1 and 20. The computations are interrupted when the vortex has been displaced of 100 core radius, i.e. $t = 10$.

In fig. 5.19 the final solution obtained adopting different schemes and time-steps is plotted in terms of tangential component of the velocity computed along the symmetry plane, i.e. $y = 0$. For values of the Courant number lower than 1 the numerical error introduced by the time scheme is very small, indeed the curves obtained with the three schemes are overlapped and difference with respect to the exact solution is given by the error in space. Increasing the Courant number to 1 highlights the differences between the first order scheme and the more accurate ones. In the $\mathrm{Co} = 20$ case, shown in fig. 5.19(c), the difference between the exact solution and the numerical one is increased and the behavior of the three schemes differs. The curve obtained with the first order BDF is strongly smeared but still monotone. The solution obtained with the second order scheme is less dissipated but shows an error in phase that is not present in the other cases. The curve computed with the third-order scheme shows a similar delay in phase but, differently form the BDF2 scheme, does not show a monotone behavior. This result is in agreement with the fact that the high-order (in time) extension of first order TVD scheme does not necessarily share the total variation diminishing property. Indeed Fernanez [55] showed that in the 1D Sod problem the implicit BDF2-Roe scheme is not monotone for $\mathrm{Co} = 5$, while Ruuth at al. [159, 88] set a maximum Courant to ensure monotonicity of a BDF2 scheme in the one-dimensional case at 0.5 times the maximum Courant of the corresponding first-order explicit scheme.

In fig. 5.20 the iso-vorticity lines at $t = 10$ are shown for the exact solution, while the one obtained with the tested numerical schemes are presented in fig. 5.21. The number of contour lines and the spacing is the same adopted in fig. 5.20.

For $\mathrm{Co} = 0.1$ the numerical solutions are almost distinguishable amongst each other, while for $\mathrm{Co} = 1$ only the vorticity computed with the first order BDF appear to be smeared and the effects of the entropy fix are visible, i.e. the different amount of introduced numerical dissipation between the upper and the lower side of the vortex also causes an error in phase. For $\mathrm{Co} = 20$ the solution obtained with the first order scheme is almost completely dissipated, while the one obtained with the high-order schemes features an error in both phase and amplitude. The non-monotone behavior of the scheme that has been shown in fig. 5.19(c) is here not visible due to the close-up view, but it is nonetheless present in the vorticity as well.

Therefore to ensure the monotonicity of the solution the first-order Forward Euler scheme is adopted, indeed the bound in terms of Courant number is such that the error introduced by the first order and the high-order schemes is comparable, as shown in fig. 5.19(a) and 5.21(a).

Adaptive grid computations

As shown in the previous section the artificial dissipation introduced by the scheme is responsible of the destruction of the vortex core. This can be avoided reducing the time

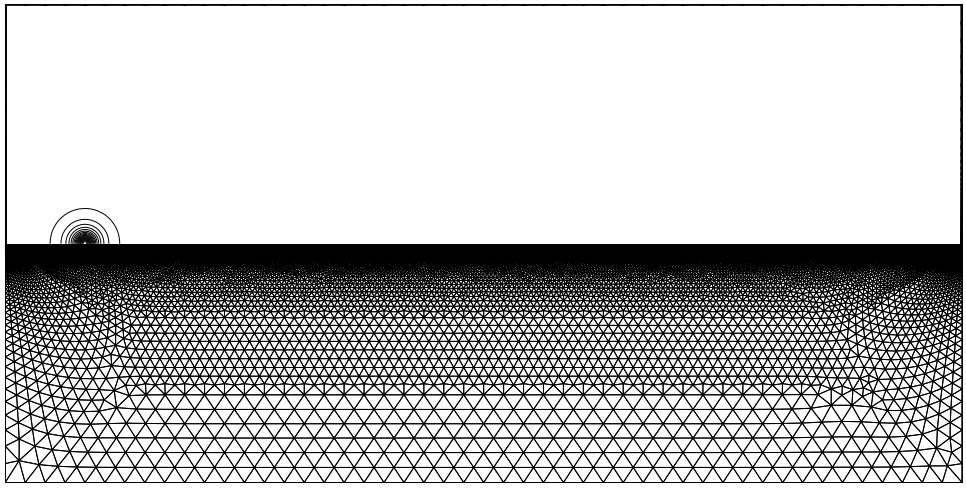
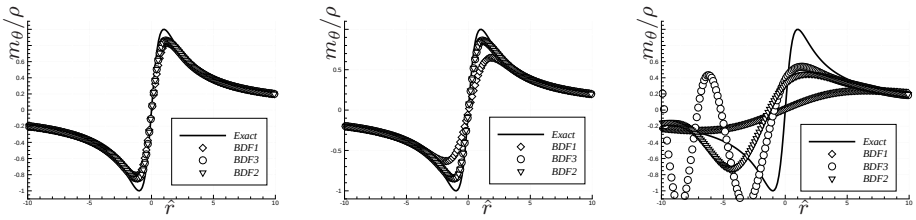


Fig. 5.18: Lower half of the fixed mesh and upper half of the initial density contour lines. The size of the mesh is $240\,r_c \times 120\,r_c$, with 61015 nodes and 121722 elements. The minimum node spacing is roughly of 0.005 grid units.



(a) $\mathrm{Co} = 0.1$ (b) $\mathrm{Co} = 1$ (c) $\mathrm{Co} = 20$

Fig. 5.19: Tangential component of the velocity along the symmetry line as a function of the radial coordinate at the non-dimensional time $t = 10$. The vortex displacement is equal to $100\,r_c$.

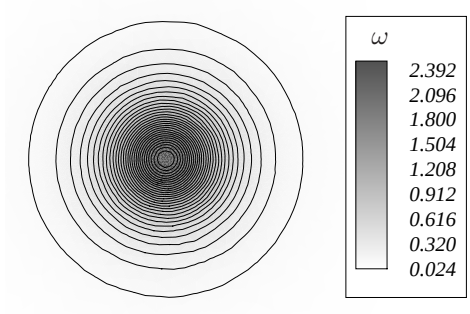
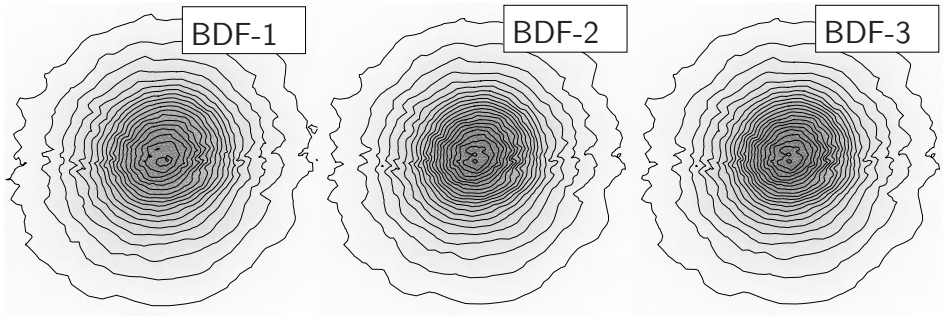
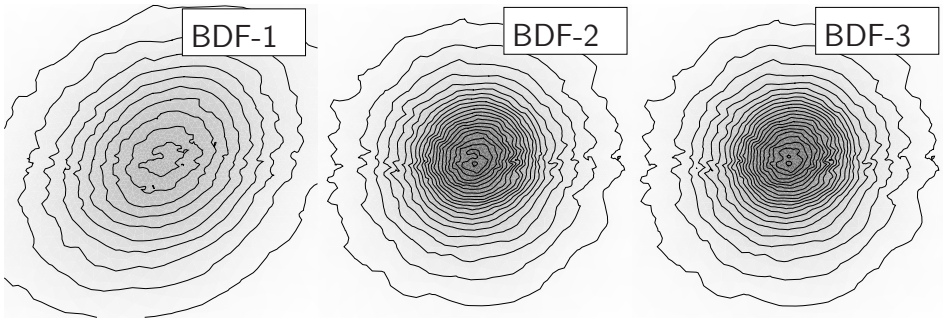


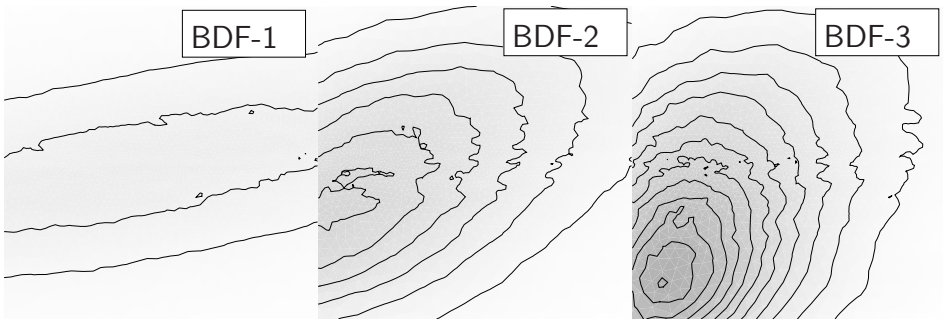
Fig. 5.20: Contour lines of the magnitude of the vorticity vector of the initial solution.



(a) $\mathrm{Co} = 0.1$



(b) $\mathrm{Co} = 1$



(c) $\mathrm{Co} = 20$

Fig. 5.21: Contour lines of the magnitude of the vorticity vector obtained at $t = 10$ with different time-steps and numerical schemes. The number of contour lines and the scale is the same adopted in fig. 5.20

Test Case	Adapted	Rigid	Rigid Adapted	Baeder 8th order	Baeder 5th order	Kwon
$\frac{\Delta v_{\mathsf{v}}(T)}{\Delta v_{\mathsf{v}}(0)} - 1$	-0.58	-0.051	-0.05	+0.032	-0.15	-0.02

Tab. 5.1: Variation of the velocity on the edge of the vortex core with respect to the initial value for the vortex advection problem.

step and the local grid spacing. To this hand the grid adaptation tools described in chapter 3 are here applied to the vortex transport problem.

The free-flow Mach number is 0.8, vortex core Mach number is 0.255 and vortex radius is 0.05. The initial grid and the vorticity field are shown in fig. 5.22. The grid is $480\, r_{\mathsf{v}} \times 80\, r_{\mathsf{v}}$ and is made of 33017 nodes and 65368 elements. Away from the vortex the grid spacing is 0.08, i.e. $V_{m,max} = 3 \times 10^{-3}$, and around the vortex core a refined region is created with elements of area 3×10^{-3}, i.e. $h_{min} = 0.002$. The geometry-driven adaptation outlined in section 3.3.1 is used to generate a constant area region inside a circumference of radius $2r_{\mathsf{v}}$ centered on the vortex core. Moreover the element size decreases linearly and at $r \simeq 4r_{\mathsf{v}}$ the maximum area is recovered, as shown in fig. 5.22(d).

The contour lines for the magnitude of the vorticity vector are shown in fig. 5.22(c). Although the prescribed solution of Eq. (5.2) is smooth, the vorticity computed with Eq. (3.6) is slightly non monotone possibly due to the effect of the variable grid spacing.

Following [142], in the unsteady computations the position of the vortex core $\boldsymbol{x}_{\mathsf{v}}^{n+1}$ is taken as the grid vertex featuring the minimum/maximum of ω within the circle of radius $0.5\, r_{\mathsf{v}}$ centered in $\boldsymbol{x}_{\mathsf{v}}^{n}$. Following [175] to measure how well the initial solution is preserved the variation of the velocity along the vortex radius is introduced, i.e.

$$\Delta v_{\mathsf{v}}(t) = \max_{i \in \mathcal{K}(t)} \left| \frac{\boldsymbol{m}_i(t)}{\rho_i(t)} - \frac{\boldsymbol{m}_\infty}{\rho_\infty} \right| - \min_{i \in \mathcal{K}(t)} \left| \frac{\boldsymbol{m}_i(t)}{\rho_i(t)} - \frac{\boldsymbol{m}_\infty}{\rho_\infty} \right|.$$

At a given time the error indicator is thus taken as $\Delta v_{\mathsf{v}}(t)/\Delta v_{\mathsf{v}}(0) - 1$.

The `FIAP` procedure is carried out, without looping over s, to adapt the solution over a sensor made by the sum of the magnitude of vorticity and the magnitude of the gradient of ρ and to satisfy the geometric constraints sketched above, i.e. the element size decreases linearly with the distance from the vortex core but the extrema are bounded by the smallest and the largest elements present in the the initial mesh, i.e.

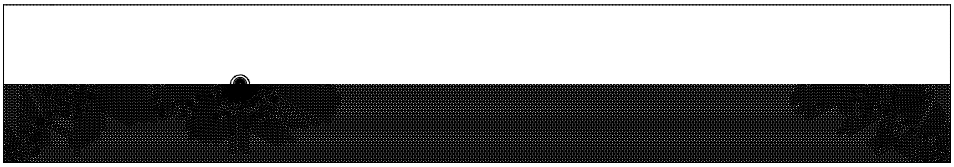
$$\mathcal{A}(\boldsymbol{x}, t) = (V_{m,max} - V_{m,min}) \mathcal{A}_{\mathsf{v}}(\boldsymbol{x}, t) + V_{m,min}, \tag{5.5}$$

where

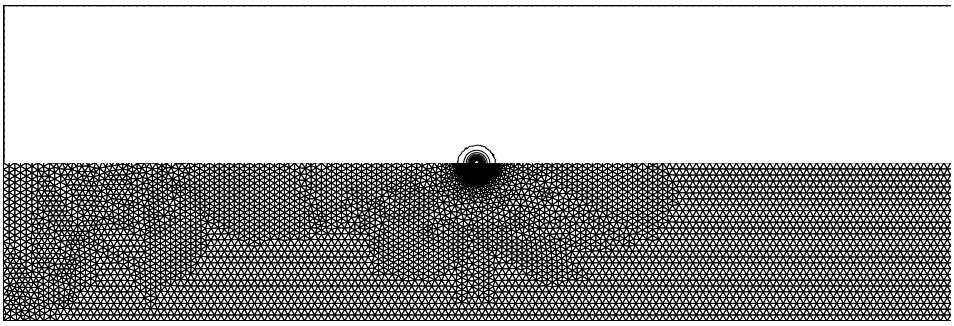
$$\mathcal{A}_{\mathsf{v}}(\boldsymbol{x}, t) = \frac{1}{3} \min \left(\max \left(\frac{|\boldsymbol{x} - \boldsymbol{x}_{\mathsf{v}}(t)|}{r_{\mathsf{v}}}, 3 \right), 0 \right)$$

is the normalized distance from the core and $V_{m,min}$ and $V_{m,max}$ are the minimum and maximum element area of the domain, respectively.

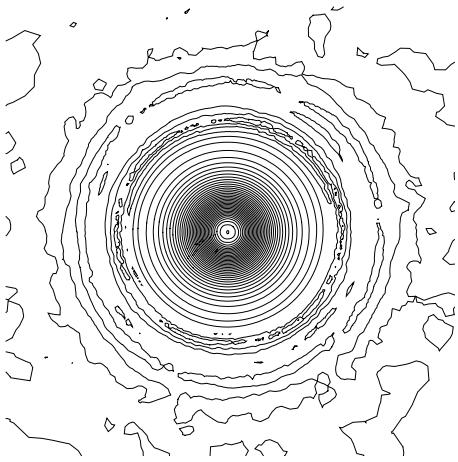
The computations are carried out with a non-dimensional time step of 5×10^{-4}, which correspond to a Courant number of 0.1, and are interrupted at $t = 4$, i.e. when the total distance traveled by the vortex core is $80 r_{\mathsf{v}}$. The final solution and grid are plotted in fig. 5.23. The overall gird-quality is unsatisfactory and this is indeed reflected



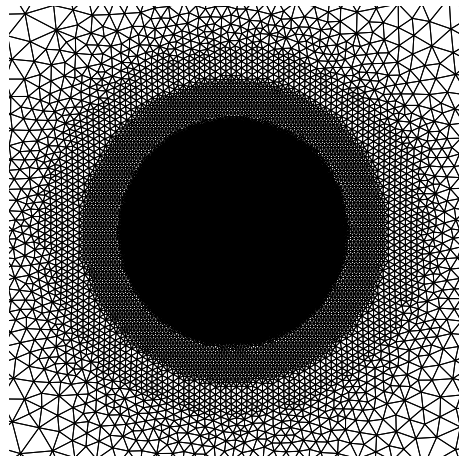
(a) Initial grid and vorticity contour



(b) Left half of the initial with vorticity contour



(c) Vorticity contour close-up



(d) Initial grid close-up

Fig. 5.22: Initial grid and solution vorticity magnitude for the vortex advection problem with adaptive the scheme. Grid made of 33017 nodes and 65368 elements, $h_{max} = 0.08$ and $h_{min} = 0.002$, with $V_{m,max} = 3 \times 10^{-3}$ and $V_{m,min} = 1.863 \times 10^{-6}$.

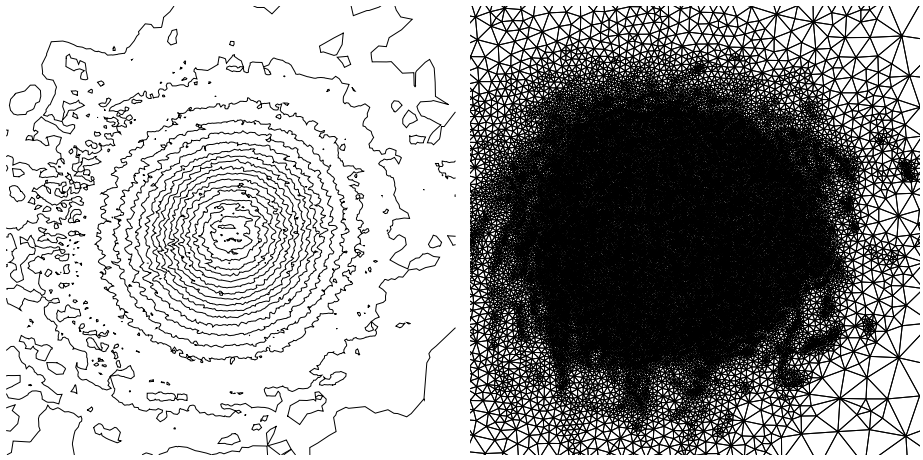


Fig. 5.23: Final grid and solution vorticity magnitude for the vortex advection problem with adaptive the scheme and no mesh deformation, 40687 nodes and 80623 elements.

over the iso-vorticity lines of fig. 5.23, which appear to be excessively irregular. Overall the solution obtained adapting over ω and $\boldsymbol{\nabla}\rho$ is severely smeared resulting in a 58% error, as shown in tab. 5.1. This result is unsatisfactory if compared to other adaptive mesh approaches to the vortex advection problem [142, 175].

As shown in fig. 5.23 the applied adaptive scheme is strongly dissipative. The numerical dissipation introduced by the Roe scheme of Eq. (2.21) is proportional to the eigenvalues of the Jacobian matrix, i.e. $\boldsymbol{m}/\rho$, c and ν, and to the cell size, i.e. η. As detailed in section 2.6, the grid velocity term is split into two different contribution the deformation one and the correction to take into account the mesh adaptation, namely

$$\nu^{n+1} = \frac{\Delta D^{n+1}}{\Delta t} + \frac{\Delta A^{n+1}}{\Delta t},$$

where a Forward Euler scheme has been used. The deformation contribution to the area swept by the interface during the time-step depends from Δt and, for smooth grid movement laws, $\lim_{\Delta t \to 0} \Delta D^{n+1}/\Delta t$ is bounded and correspond to $\nu(t)$. The adaptation correction however does not depend on the time step and $\lim_{\Delta t \to 0} \Delta A^{n+1}/\Delta t$ does not exists. This result does not depend on the type of time-integration scheme adopted and it is therefore valid for any BDF method as well. From the governing equations point of view this means that the more the time step is reduced, and the more frequently the grid is changed, the more numerical dissipation will be introduced by the scheme.

In the adaptation case of fig. 5.23 the grid around core is continuously changing due to the effect of the vortex displacement and of the solution smearing as well. To overcome such issue a different approach has been studied that limits the amount of topology modifications performed to move the vortex core. The mesh deformation algorithm presented in section 3.1 is modified to displace in rigid-like fashion the elements around the vortex core, $\boldsymbol{x}_{\mathrm{v}}$, and a predictor/corrector-like scheme is set up as follows

1. First the position of the vertex representing the vortex center point at the new

time step is predicted as

$$\hat{\boldsymbol{x}}_{\mathsf{v}}^{n+1} = \boldsymbol{x}_{\mathsf{v}}^{n} + \frac{\Delta t}{2}\left(\frac{\boldsymbol{m}_{\mathsf{v}}^{n}}{\rho_{\mathsf{v}}^{n}} + \frac{\boldsymbol{x}_{\mathsf{v}}^{n} - \boldsymbol{x}_{\mathsf{v}}^{n-1}}{\Delta t}\right),$$

where $\boldsymbol{m}_{\mathsf{v}}$ and ρ_{v} are the values of momentum and density on the core node, respectively.

2. The position of the rest of the grid nodes $\boldsymbol{x}_i^{n+1}$ is computed with the mesh deformation scheme based on the continuum analogy. The difference in size between the elements close to the vortex core and the elements located in the rest of the domain allows for the vortex core to be displaced almost rigidly while most of the mesh deformation is absorbed by the elements located away from the core. To better enforce the rigid displacement of the vortex the element young modulus appearing in Eq. (3.1) is modified as

$$\hat{E}_e = \mathcal{A}_{\mathsf{v}}(\boldsymbol{x}, t)(E_e - E_{\mathsf{v}}) + E_{\mathsf{v}},$$

where $\mathcal{A}_{\mathsf{v}}$ is defined in Eq. (5.5) and E_{v} is the stiffness factor for the elements of the vortex. In the present work E_{v} is equal to the minimum Young modulus over the mesh elements, defined by Eq. (3.1), multiplied by a factor 10^{-3}.

3. Following the `FIAP` procedure, the solution at the new time step is predicted and then the position $\boldsymbol{x}_{\mathsf{v}}^{n+1}$ is updated locating the minimum/maximum of the vorticity, as sketched above. A common issue with the application of the barycentric grid regularization technique described in section 3.1.1 is the fact that it produces meshes with uniformly distributed nodes and stretched elements on the boundaries. In the present case the application of the smoothing technique inside the vortex core is avoided, since it would a negative impact on the grid spacing, thus on the solution. The relaxation parameter of Eq. (3.2) is modified as

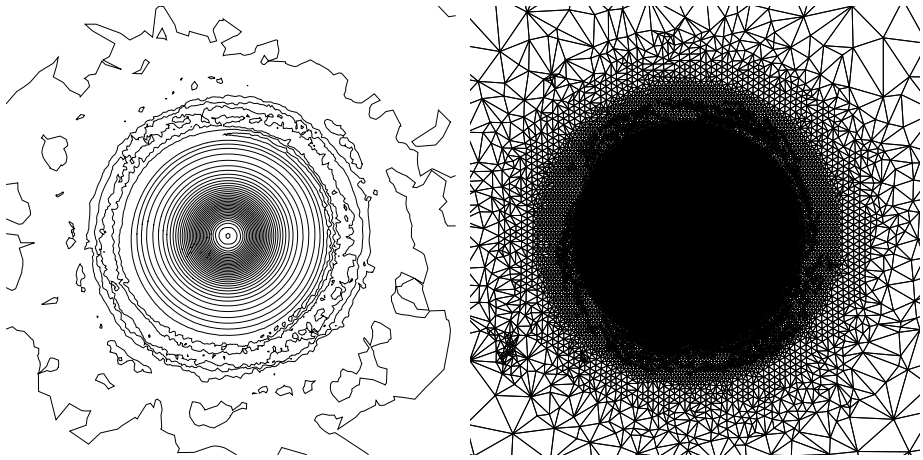
$$\hat{k}_s = k_s\, \mathcal{A}_{\mathsf{v}}(\boldsymbol{x}, t),$$

such that the nodes located inside the core are node moved.

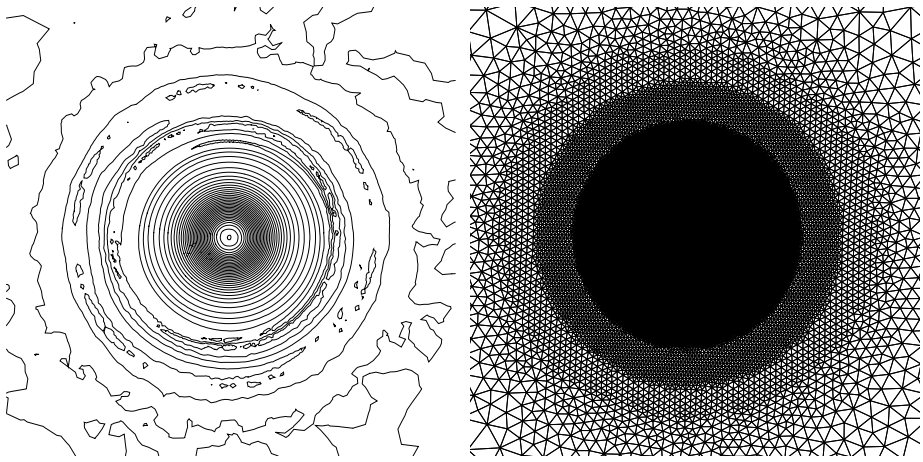
4. The de/refinement scheme outlined in fig. 3.9 and 3.10 is carried out based on the corrected position for the vortex core and the predicted solution the grid. This allows to impose simultaneously the constraints based on the error equidistribution theory and the geometric ones.

5. The solution u^{n+1} is updated with the ALE scheme over the adapted grid.

The solution computed with the scheme outlined above are shown in fig. 5.24(a). The initial grid quality and spacing are very well preserved, and the vorticity field is very close to the exact one, indeed, as shown in tab. 5.1 a 5% loss is achieved that is comparable with the 8-th order scheme from [175]. A similar result can be obtained if no nodes are inserted or deleted and the movement is carried out only with mesh deformation and swapping, as shown in fig. 5.24(b). Indeed, since the grid inside the vortex core is translated almost rigidly, no changes in topology occurs inside this region and the vortex is thus translating "together" with a high quality/resolutions mesh.



(a) Adaptive, 36999 nodes and 73198 elements



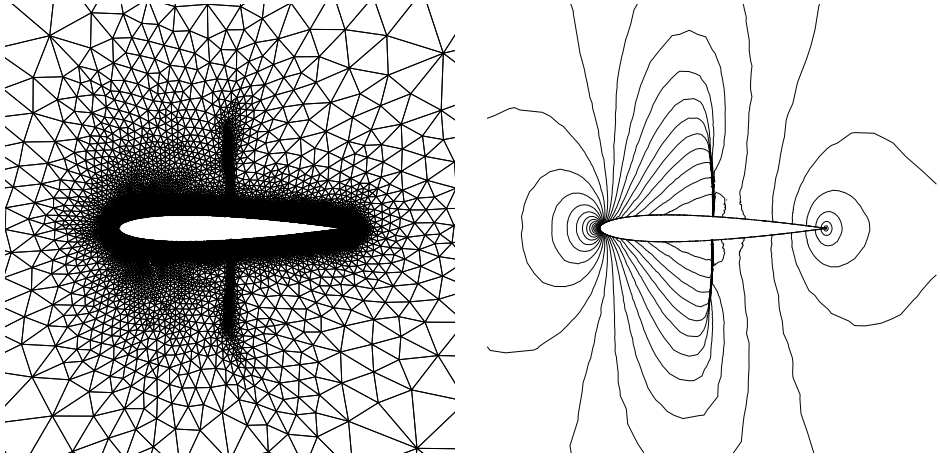
(b) No-adaptation, 33017 nodes and 65368 elements.

Fig. 5.24: Final grid and solution vorticity magnitude for the vortex advection problem with the adaptive scheme. (a) Mesh deformation and FIAP adaptation with geometric constraints on the element dimension. (b) Mesh deformation and swapping only.

5.3.2 Interaction with a NACA 0012 airfoil

The interaction between a NACA 0012 airfoil and a vortex is here presented. The flow field Mach number is 0.8 and the vortex reference Mach number is 0.259154, which correspond to $\Gamma = -0.2$, i.e. rotating clockwise. The airfoil has a unit chord value, i.e. $c = 1$, and the vortex core has a $0.05c$ radius and the initial position is $x_v(0) = -5c$ and $y_v(0) = -0.26c$.

As shown by [172, 173, 142], when the distance between the vortex and the lower side of airfoil is sufficiently small an increase in the value of the local velocity on the wall is observed, while the flow field on the upper side is only slightly affected. This causes a pressure wave to be released by the nose of the airfoil, that propagates upstream, and



(a) Grid (b) Mach

Fig. 5.25: Initial adapted grid and solution for a $M_\infty = 0.8$ flow past a NACA 0012 at zero angle of attack. Initial grid is made of 50867 nodes and 100595 elements, i.e. $V_{m,min} = 4 \times 10^{-7}$ and $V_{m,max} = 1$.

x_v/c	-0.6	-0.2	0.2	0.6	1.0	1.4
$\frac{\Delta v_\mathrm{v}(t)}{\Delta v_\mathrm{v}(0)} - 1$	0.1333	-0.1084	-0.3791	-0.3857	-0.4621	-0.5097

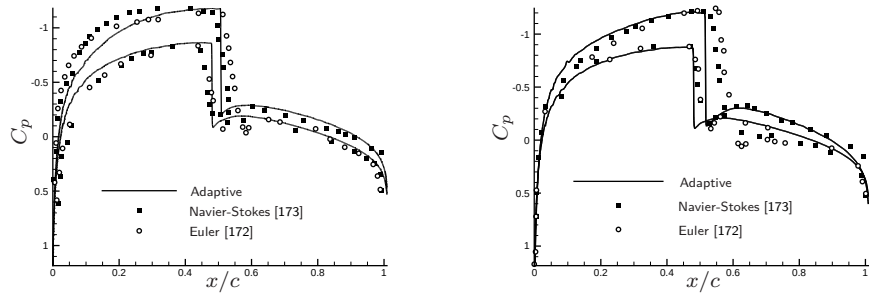
Tab. 5.2: Reduction in vortex intensity measured in terms of tangential velocity across the vortex core.

a aft movement of the shock wave on the lower side due to the increase in streamwise velocity. When the vortex reaches the trailing edge the shock wave on the lower side moves fore, due to the reduction of the jump of velocity/pressure, and as the time proceeds the original, steady, state is recovered.

First steady computations are carried out with adopting the `FIAP` adaptive scheme for the NACA 0012 airfoil test case at zero angle of attack without vortex. The solution features two strong shocks on the upper and lower side of the airfoil with equal intensity, indeed no shear surface is present.

The `FIAP` scheme for steady applications is adopted, i.e. without looping over s, to adapt the solution to the mixed Gradient/Hessian of the Mach number until a 5% convergence is obtained for the relative variation of μ_w. The distance-based adaptation of section 3.3.1 is also carried out imposing that the size of the elements decreases with the distance from the airfoil, i.e. evaluating Eq. (3.4) with $c_{1,\mathrm{wall}} = 4 \times 10^{-6}$, $c_{2,\mathrm{wall}} = 3.897 \times 10^{-3}$ and $c_{3,\mathrm{wall}} = 3.897 \times 10^{-2}$, and with the distance from the vortex located in $\boldsymbol{x}_\mathrm{v}(0)$, i.e. evaluating Eq. (5.5) with $V_{m,max} = 1$ and $V_{m,min} = 4 \times 10^{-7}$. Therefore the area of the elements located inside the core is one order of magnitude smaller than the elements located on the boundary of the airfoil.

The obtained grid is shown in fig. 5.25(a) which is made of 50867 nodes and 100595 elements. The simultaneous use of the solution-based and the geometry-based adaptation strategies allows to obtain a grid that is very well refined near close to the shocks



(a) $x_v/c = 0$ (b) $x_v/c = 0.5$

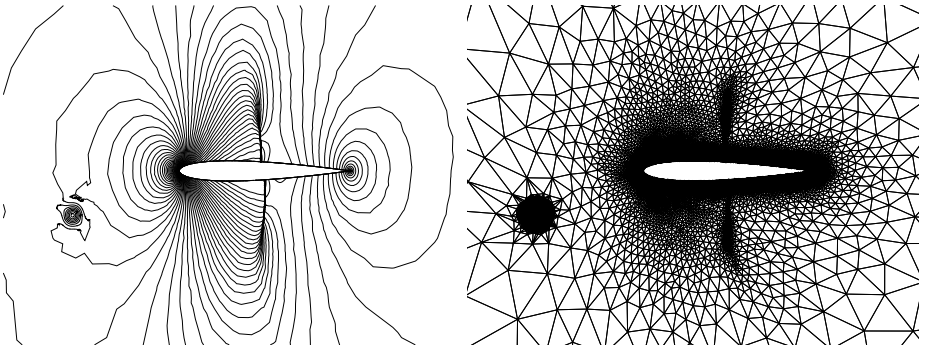
Fig. 5.26: Comparison between the pressure coefficient computed with the adaptive scheme for the Parallel BVI problem and the Euler [172] and Navier-Stokes [173] reference solutions.

and, on the other hand, is not under refined where the error sensor is small. Indeed the grid obtained with the solution-driven scheme only feature very large elements in the region between the nose of the airfoil and the shock wave, the use of the geometry-driven scheme avoid such behavior. This "conservative" approach, which limits the grid derefinement, is considered a better choice when performing unsteady computations during which the mesh undergoes significant modifications. In fig. 5.25(b) the contour lines for the Mach number are shown: the solution is overall symmetric, the shock wave are very well resolved and no shear wake is present.

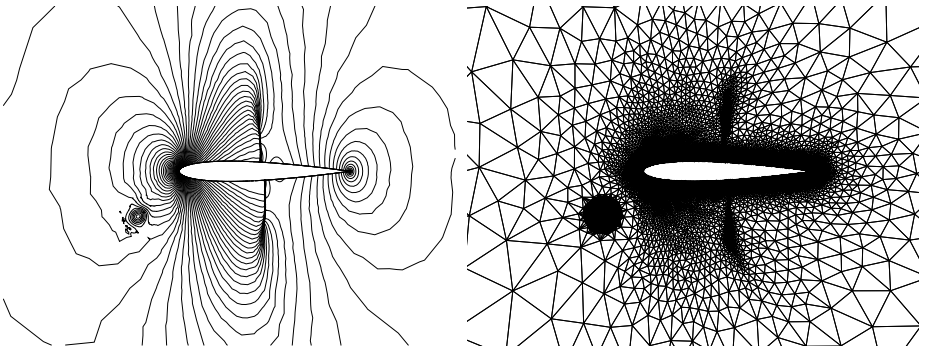
To perform unsteady computations where the vortex travels very close to the airfoil, i.e. with very a small miss-distance, a vortex is "inserted" in the flow field. To this purpose the initial solution is calculated superimposing the solution obtained with the adaptive steady computations, i.e. the one of fig. 5.25, and the solution computed evaluating Eq. (5.2), (5.3) and (5.4) for the vortex with described above. The `FIAP` scheme is then carried out without looping over s, i.e. performing one adaptation procedure per time instant, using a Forward Euler scheme with a non dimensional time-step of 0.08, corresponding to a maximum Courant number of 80.

The computational grid is shown in fig. 5.27 and 5.28 together with the pressure contour lines. The grid around the core follows closely the vortex, that is convected inside the domain and passes at small distance from the airfoil. The vortex, highlighted as a minimum in the pressure field, looses most of its intensity after the interaction with the airfoil with a 52% loss in terms of Δv_v when the core is located at $1.4c$, i.e. fig. 5.28(c). Fig. 5.27(b) and 5.27(c) shows that no pressure wave detaches from the leading edge as reported by [142], this could be caused by a reduction of the vortex intensity, as shown in tab. 5.2. The fore movement of the shock wave is also only mildly captured, to this end the reduction of almost 50% of the vortex intensity is a key factor together with the decrease of mesh quality that is caused by the close interaction of the vortex and the shock wave shown in fig. 5.28(a).

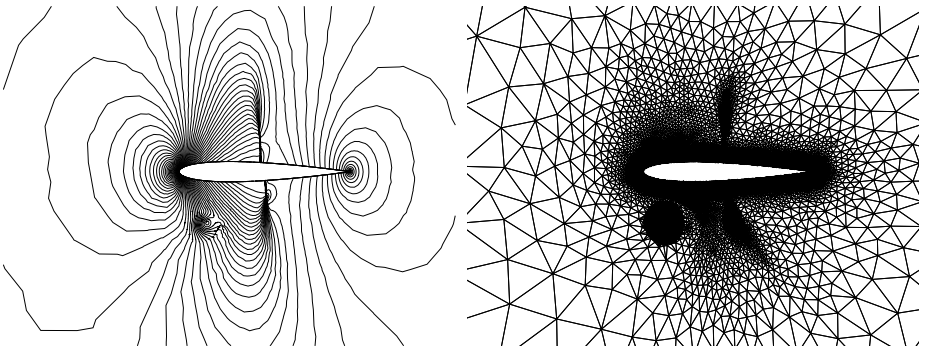
Fig. 5.26 shows the comparison between the distribution of the pressure coefficient along the airfoil computed with the adaptive scheme and the reference solutions obtained with an Euler solver [172] and a Navier-Stokes solver [173]. For $x_v = 0$ the curves on



(a) $x_\mathrm{v}/c = -0.6$, 48928 nodes and 96500 elements

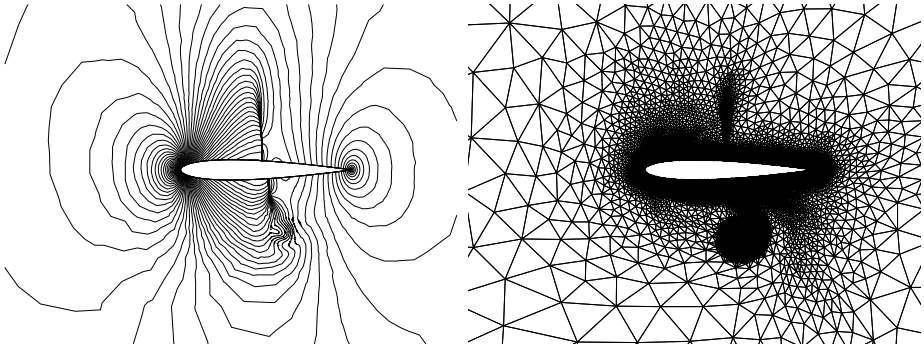


(b) $x_\mathrm{v}/c = -0.2$, 48937 nodes and 96516 elements

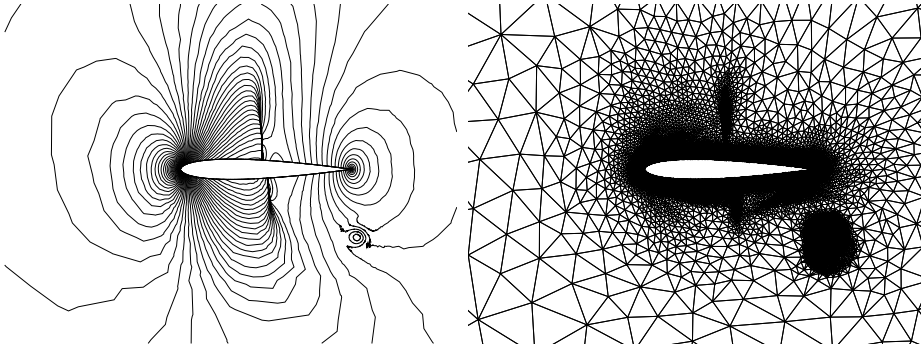


(c) $x_\mathrm{v}/c = 0.2$, 48375 nodes and 95388 elements

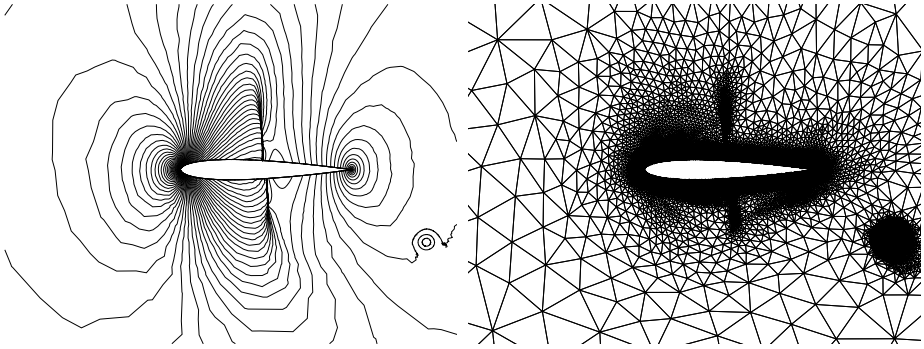
Fig. 5.27: Computational grid and pressure contour for the parallel BVI NACA 0012 for $M_\infty = 0.8$, $M_\mathrm{v} = 0.259154$ and $\mathrm{Co} = 80$.



(a) $x_v/c = 0.6$, 48254 nodes and 95145 elements



(b) $x_v/c = 1$, 50565 nodes and 99761 elements



(c) $x_v/c = 1.4$, 50178 nodes and 98987 elements

Fig. 5.28: Computational grid and pressure contour for the parallel BVI NACA 0012 for $M_\infty = 0.8$, $M_v = 0.259154$ and $\mathrm{Co} = 80$.

the upper side of the airfoil overlap fairly well, while the value of C_p on the lower side is higher than expected. This is in agreement with the fact that no compression wave is detached from the nose and could be caused by the reduction of the vortex intensity, as discussed above. The position of the lower-side shock and its intensity agree with the references. For $x_{\mathrm{v}} = 0.5c$ the solution shows a better agreement in terms of pressure coefficient close to the nose, but the predicted aft movement of the upper shock is significantly underpredicted together with the increase in intensity. As before this is deemed to be caused by the strong reduction in the vortex intensity shown in tab. 5.2.

Conclusion and future development

A novel method for compressible gasdynamics for two-dimensional rotorcrafts applications has been presented. As shown in chapter 2 the ALE Euler equations are discretized resorting to a standard finite-volume method over adaptive grids. The interpolation step between the old grid and the adapted one is skipped by exploiting the ALE formulation of the governing equations and simply integrating them to obtain the solution over the new mesh. To do so, the modifications to the topology of the grid are locally interpreted in terms of continuous deformation of the finite volumes built around the nodes, as shown in section 2.6.

In section 2.5 the convergence properties of the finite-volume scheme has been tested over a fixed grid for reference compressible flows. The scheme has proven to be first order accurate in space when discontinuities in the solution are present. The present implementation of symmetric Gauss-Seidel scheme showed a faster convergence rate than standard GMRES and LU schemes, when applied to the transonic flow past an airfoil.

As presented in chapter 3, grid adaptation is performed by resorting to a suitable mix of very popular and standard techniques: i.e. mesh deformation, edge-swapping, node insertion and removal. The grid spacing is controlled by both a geometry-driven and a solution-driven criteria. The capabilities of the former to maintain the initial grid spacing and quality when large displacements of the boundaries occurs has been tested in section 3.4.1. Two different procedures have been presented to adapt the grid to the solution in the steady case: the `SIAP`, that consists in simple fixed-point iterations, and the `FIAP`, in which the error is interpolated amongst refined elements. Both the procedures have been tested in the oblique shock steady problem, in the transonic AGARD 02 case and in the transonic NACA 0012 problem. The use of the `FIAP` procedure has proven to be extremely beneficial reducing both the number of iterations required and the overall computational time.

In the unsteady case a fixed-point approach to mesh adaptation has been used, by iterating until both grid and solution have converged for each timestep. Adaptive unsteady simulations have been carried out for the Sod problem and the forward facing step problem, that demonstrate the correctness of the adopted approach to capture multidimensional unsteady flow features with different intensities.

The adaptive scheme outlined above has been used to tackle typical two-dimensional problems for rotorcraft blade sections, where mesh adaptation is of primary importance to perform efficient unsteady computations while highlighting relevant flow features,

such as shocks, wakes or vortices. High-resolution computations over three selected problems of interest for rotorcraft aerodynamics have been carried out: an oscillating airfoil, an impulsively started airfoil and parallel blade-vortex interaction.

In both the cases of the pitching airfoil of the airfoil impulsively set in motion, presented in section 5.1 and 5.2 respectively, the adaptive grid captures remarkably well the most important flow features. Moreover the confrontation with reference solutions shows a good agreement. The parallel blade-vortex interaction problem has been discussed in section 5.3. Preliminary computations had shown that high-order time integration schemes do not preserve the monotonicity of the solution in the vortex advection test case and an error in phase is observed for high values of the Courant number. To reduce the numerical dissipation introduced by the scheme in the vortex region an *ad-hoc* procedure had been implemented that preserve the grid spacing inside the vortex core by rigidly displacing the mesh in the area surrounding the core. Such approach has been also used to perform the unsteady interaction between an airfoil and a free-vortex. Preliminary computations had shown that the vortex intensity is reduced of almost 50% by the interaction with the airfoil, this is reflected in a strong underprediction of the fore movement of the lower shock wave. Further investigations of the vortex transport problem are required in order to reduce the introduced numerical dissipation.

The extension of the present approach to the three dimensional case is under study and preliminary results for simulations of compressible flows past a wing are presented in section A.2. An ALE Euler solver for deforming grid is already available and preliminary results for a pitching wing seems promising. The extension of the ALE approach to mesh adaptation in three dimension, however, poses several difficulties. The extension of the ALE-based method, as it has been conceived and presented in this thesis, to grids made of tetrahedra is fairly straightforward. Indeed when a local modification of the topology occurs, e.g. face-swapping or node insertion, the three-step interpretation of the continuous deformation of the volumes, i.e. collapse, topology modification and expansion, can still be applied. The computations of the correction terms for the interface velocities appear, therefore, to be fairly straightforward.

Major difficulties, however, arise when the above scheme is to be implemented within the framework of 3D adaptive meshes. Indeed a naïve approach to mesh adaptation in three dimensions would probably reveal disastrous in terms of computational efficiency and mesh quality. To this purpose very promising results with 3D adaptive meshes have been obtained by some researh groups: e.g. the open source program of the ITAPS project, by the open source Stellar program of the Unversity of Berkeley or by the open source `mmg3d` program by the team BACCHUS of INRIA. Even with such powerful tools available, the implementation of an ALE-based solver for 3D adaptive meshes appears still to be a challenging task. Indeed the need of storing additional contributions to the fluxes and additional governing equations imposes to somewhat keep tracks of the grid history and nformation must go from the solver to the grid and *viceversa*. Nonetheless "smart" approaches to the problem are possible which, for example, envisage the use dynamic data structure such as lists. Moreover the use of multi-stage schemes, such as Runge-Kutta, would sensibly reduce the implementation complexities, since it would be necessary to store the information form only two grids, i.e. the old one and the adapted one.

Appendix A

Appendix

In this chapter additional results obtained with the adaptive ALE scheme outlined in the thesis are presented. The transonic two dimensional aileron buzz problem is described in section A.1. In section A.2 the ALE scheme described in section 2.2 is applied to the 3D case of an oscillating airfoil/wing.

A.1 Aileron buzz

The investigation of aeroelastic stability boundaries by means of Fluid Structure Interaction (FSI) analysis is becoming very popular for the preliminary and verification phases of new aircraft design [20, 160].

To obtain reliable results, appropriate meshes of the fluid domain must be used. In the present section this problem of the dependency from the grid of the numerically evaluated stability boundary of a the transonic aileron buzz is analyzed. This is an instability involving the interaction of a single structural degree of freedom, associated to the aileron rotation about its hinge, with unsteady aerodynamic forces caused by strong shock waves dwelling close to the hinge axis. The instability may evolve into self-sustained Limit Cycle Oscillations [107, 108].

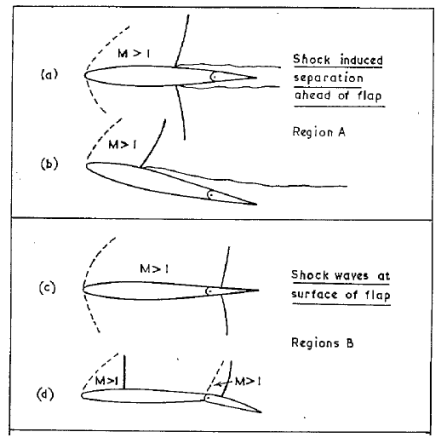


Fig. A.1: Aerodynamic conditions for which buzz may occur [108].

Following the classification proposed by Lambourne [107], the so called non-classical aileron buzz problem is tackled, which results from the interaction of the shock-waves with the aileron movement without significant intervention of the boundary layer. For numerical analysis of Type A phenomena see [22] and references therein.

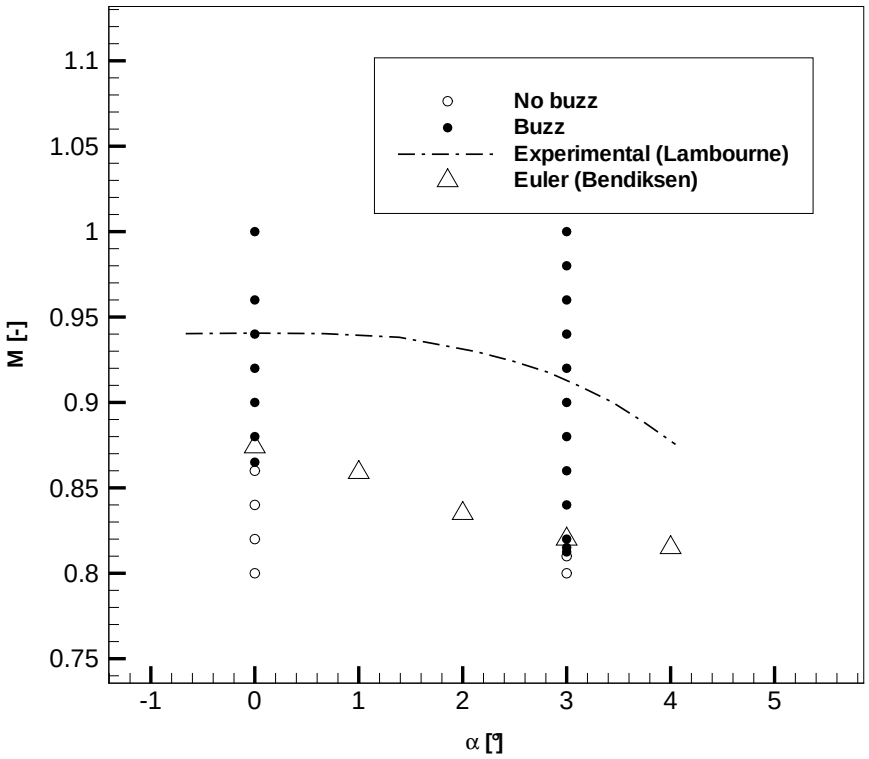


Fig. A.2: Comparison of experimental and numerical stability boundaries of the Mach-angle-of attack plane.

The structural model used to describe the aileron motion is a simple one degree of freedom equation expressing equilibrium of moments about the aileron hinge, namely $I\ddot{\beta} = H$, where I is the inertia moment of the flap around the hinge and H is the aerodynamic moment about the hinge line. Given the value of the flap rotation β at a given time t^{n+1}, the position of the grid nodes belonging to the boundary is updated and the inner nodes are displaced resorting to the mesh deformation algorithm described in section 3.1.

The direct time integration of the fluid-structure interaction problem is tackled using a partitioned loosely coupled algorithm. Both aerodynamic and structural systems are integrated using an implicit scheme, thus achieving linear stability for any value of time-step Δt. A second order accurate BDF scheme is used for the flow equation, while a predictor-corrector method derived from Crank-Nicholson [71] has been adopted for the structural subsystem. The latter scheme is here briefly outlined.

Structure predictor: The known values of flap angle, flap angular velocity and aerodynamic loads at the time n are used to prediction of the structural state at time $n+1$,

i.e.

$$\begin{cases} \beta_p^{n+1} = \beta^n + \Delta t\, \dot{\beta}^n + \dfrac{\Delta t^2}{2I} H^n, \\ \dot{\beta}_p^{n+1} = \dot{\beta}^n + \dfrac{\Delta t}{I} H^n. \end{cases}$$

CFD computation: The predicted predicted structural state, β_p^{n+1} and $\dot{\beta}_p^{n+1}$ is used to compute the new mesh and subsequently the aerodynamic loads, $H^{n+1} = H(\beta_p^{n+1}, \dot{\beta}_p^{n+1})$.

Structure corrector: The predicted loads are used to correct the value of the structural state at t^{n+1}, i.e.

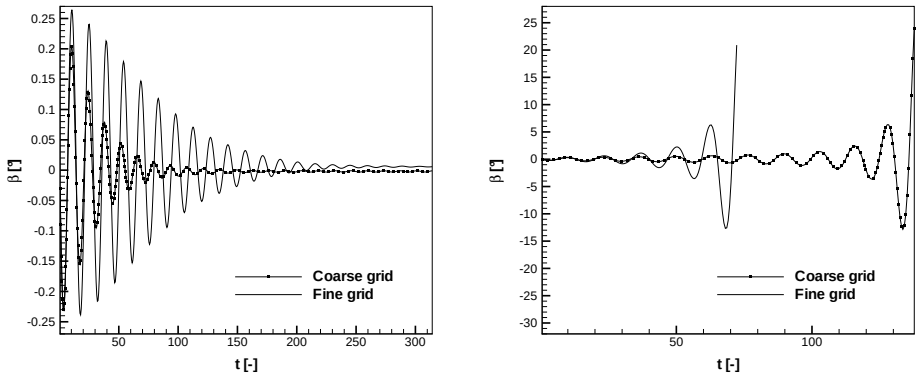
$$\begin{cases} \beta^{n+1} = \beta^n + \Delta t\, \dot{\beta}^n + \dfrac{\Delta t^2}{2I} \dfrac{H^{n+1} + H^n}{2}, \\ \dot{\beta}^{n+1} = \dot{\beta}^n + \dfrac{\Delta t}{I} \dfrac{H^{n+1} + H^n}{2}. \end{cases}$$

Aileron buzz is now examined to study the suitability of the proposed approach to investigate aeroelastic phenomena in two dimensional cases. The prerogative is to assess non-classical aileron buzz over a range of transonic Mach numbers. Tests are conducted on a RAE 102 typical section model clamped on its mass center to avoid pitch or plunge movements and to allow only flap rotation around its hinge. Flap-chord/chord ratio is 25% and non-dimensional frequency parameter is $f = 0.063$ approximately (which corresponds to a reduced frequency of 0.1). A circular domain with a radius of 20 chords is chosen to avoid far-field boundary conditions interferences on the unsteady phenomenon. The result obtained is the stability boundary in the Mach-angle-of-attack plane. Since RAE 102 is a symmetrical airfoil, tests at $\alpha = 0^\circ$ are conducted imposing a non-dimensional initial angular velocity of flap around the hinge different from zero, i.e. $\dot{\beta}(0) = -10^{-3}$. Differently, computations with an angle of attack $\alpha = 3^\circ$ do not need an initial perturbation thanks to a non-null hinge moment.

To define numerical boundary stability, mumerical computations have been conducted at different Mach numbers ranging from 0.8 to 1, with two different angles of attack, i.e. $\alpha = 0^\circ$ and $\alpha = 3^\circ$. Moreover two different grids have been used to test the sensitivity of the phenomenon with respect to the discretization spacing: a coarse grid with 10396 nodes and 19745 elements and a fine one with 20845 nodes and 40482 elements. All the computations have been carried out resorting to a second-order BDF scheme with a time step of 0.5235.

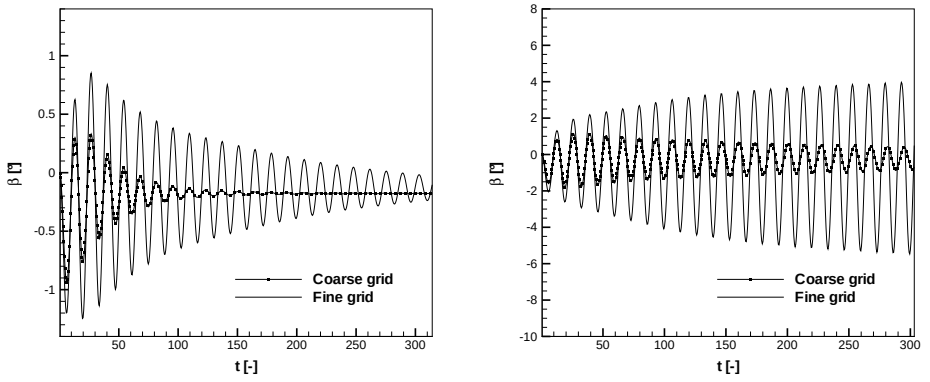
Fig. A.2 shows the comparison of the computed stability boundary on the Mach-angle-of-attack plane with the experimental one. The Euler equations model produces a result far away from the experimental model instability [107] due to the neglection of the viscous effects. Indeed the presence of the boundary layer is reflected in an increase of the airfoil thickness for the inviscid flow point of view, providing an higher stability boundary in terms of Mach number. The numerical results are indeed in fairly good agreement with the inviscid computations by Bendiksen [19].

In fig. A.3 is shown the response in terms of flap rotation β for $\alpha = 0^\circ$ using two different grid spacings and two different values of Mach number across the stability boundary, respectively $M_\infty = 0.86375$ and $M_\infty = 0.865$. A reduced grid spacing determines a more negative aerodynamic damping effect, both for stable and unstable



(a) M=0.86375 (b) M=0.865

Fig. A.3: Comparison of flap rotation transient at $\alpha = 0°$ on different grids : (a) stable responses; (b) unstable responses.



(a) M=0.81 (b) M=0.8125

Fig. A.4: Comparison of flap rotation transient at $\alpha = 3°$ on different grids : (a) stable response; (b) unstable/stable responses.

cases, due to numerical viscosity increase with volume cell. The flap rotation responses for $\alpha = 3°$ are shown in fig. A.4. In this case numerical stability boundary has been evaluated about $M_\infty = 0.8125$. Indeed the fine grid gives an unstable response while the coarse grid a stable one, this is in agreement with the increase of aerodynamic damping effect with cell volume.

Adaptive computation has been performed to improve the capturing of shocks movements on airfoil and flap surface. Mesh adaptation has been driven by an error indicator based on the mixed gradient/Hessian of the Mach number over the whole computation domain. The initial grid is obtained adapting the mesh to the steady (freezed hinge) case and is made of 21625 nodes and 42747 elements. Fig. A.5 and fig. A.6 show the

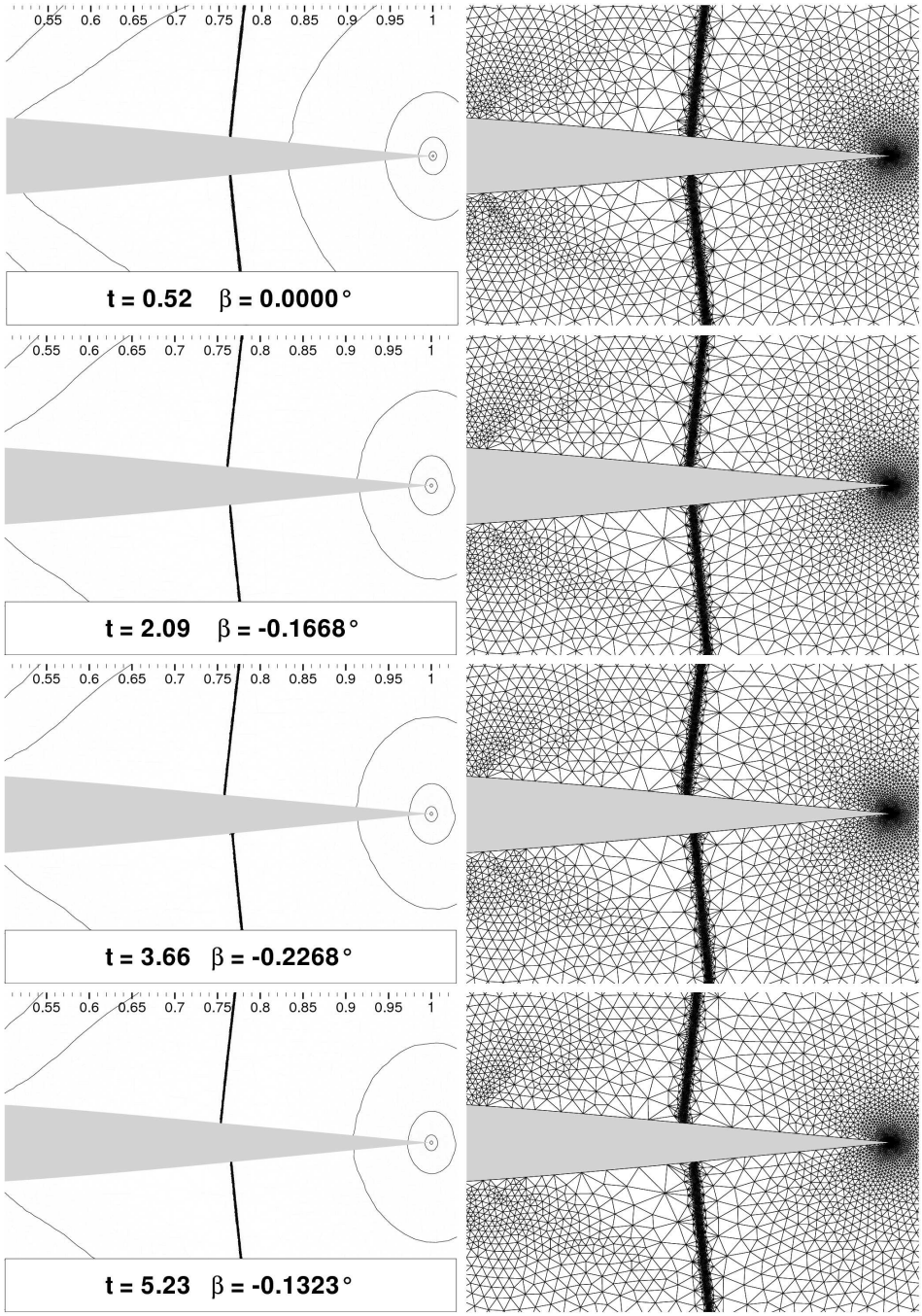


Fig. A.5: Snapshots of adapted grids and Mach number contour during a cycle of an unstable aileron buzz case (M=0.865,$\alpha = 0°$).

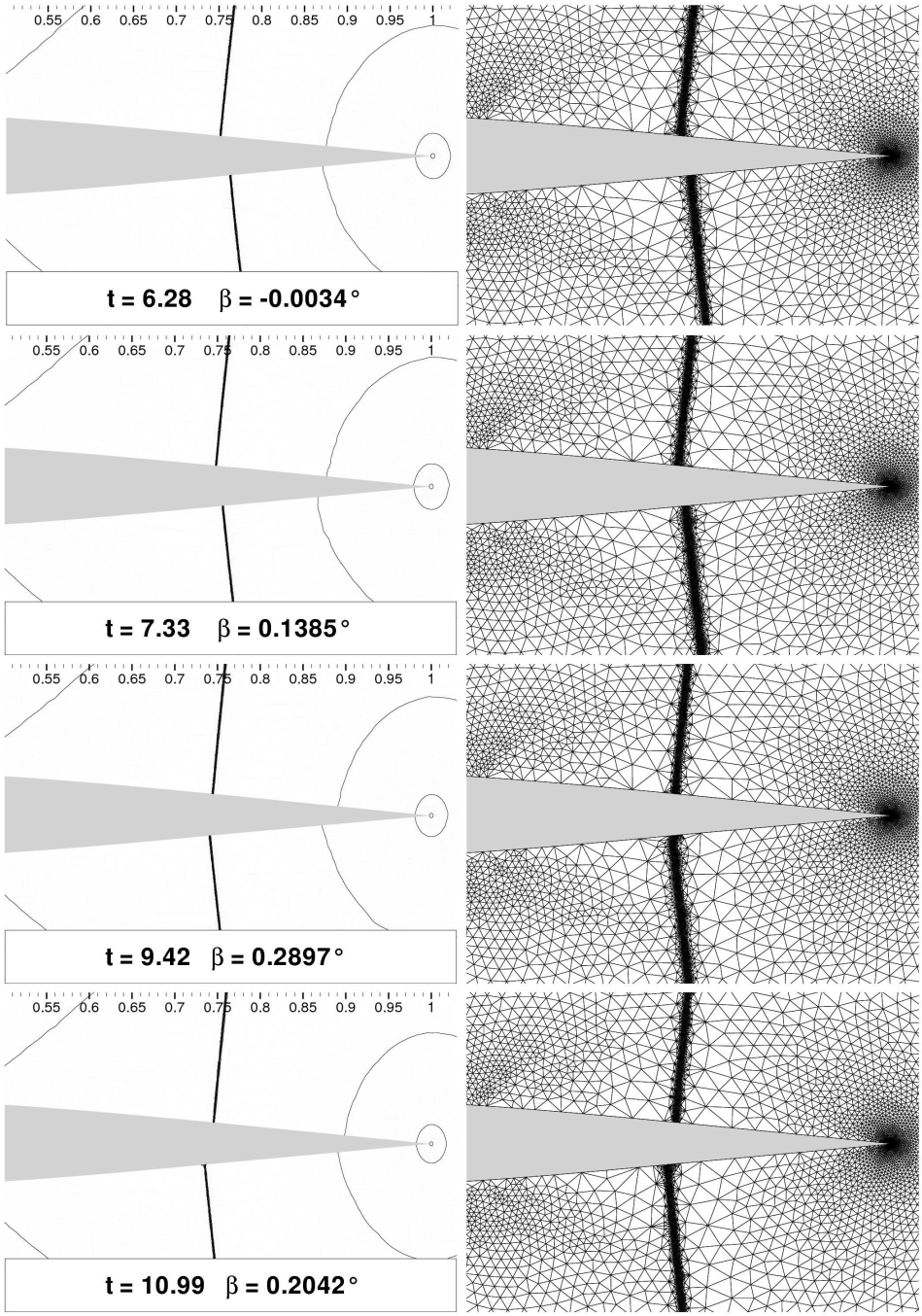


Fig. A.6: Snapshots of adapted grids and Mach number contour during a cycle of an unstable aileron buzz case (M=0.865,$\alpha = 0°$).

details of some subsequent flow-fields around the flap during an unstable response at $M_\infty = 0.865$ and $\alpha = 0°$. When the flap starts moving upward the upper shock moves forward and lower its strength while the lower shock does the opposite.

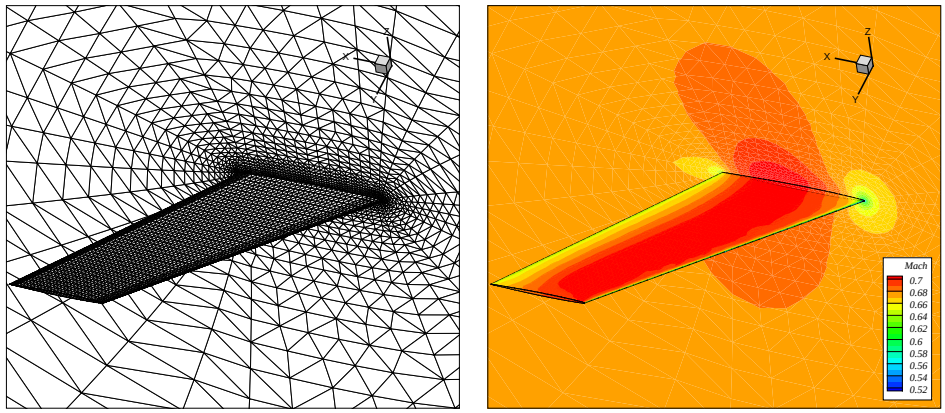
A.2 Preliminary simulations of 3D compressible flows

In the present section the ALE scheme described in section 2.2, i.e. without mesh adaptation, is applied to the 3D problems.

Preliminary results for the case of the the AGARD 445.6 are shown in fig. A.7. The free stream Mach number is 0.6 and a zero angle of attack is imposed. The computational mesh is made of 22014 nodes and 118480 tetrahedra. The far field boundary is located at distance of 15 root chords from the wing and symmetry boundary conditions are imposed at the wing root. In fig. A.7(a) a close-up of the computational mesh over the wing surface is shown, while in fig. A.7(b) the Mach distribution over the surface is illustrated.

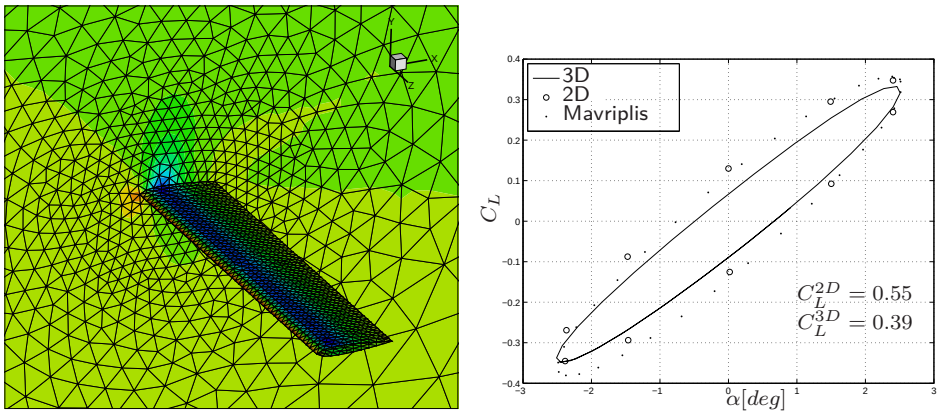
Another preliminary test case is carried out to test the ALE scheme in the three dimensional setting. A wing is generated by extrusion of a NACA 0012 airfoil and the computational mesh of 6732 nodes and 31434 tetrahedra is then generated around it, as shown in fig. A.8(a). A free flow is imposed with a Mach number of 0.755 and an angle of attack of 0.016 degrees, and the steady solution is shown in fig. A.8(a) in terms of pressure coefficient distribution. A sinusoidal pitching motion is thus imposed with a variation in the angle of attack of 2.51 degrees and a reduced frequency of 0.0816. A second order BDF scheme is used to integrate the equations in time.

The solution in terms of lift coefficient versus angle of attack is shown in fig. A.8(b). Comparison are performed with the results obtained solving the same case with a 2D grid and with the data from Mavriplis [126]. The error between the 3D solution and the 2D one in terms of C_L is comparable with the values of lift coefficient calculated at the beginning of the computations. Indeed the computational grid for the 3D case



(a) Mesh (b) Mach

Fig. A.7: Computational mesh and Mach number distribution for the AGARD 445.6 test case. $M_\infty = 0.6$, $\alpha = 0$, with a mesh made of 22014 nodes and 118480 elements.



(a) Grid and C_p (b) C_L-α

Fig. A.8: Computational mesh, pressure coefficient and lift coefficient for the NACA 0012 wing case. $M_\infty = 0.755$, $\Delta\alpha = 2.51$, $\alpha_0 = 0.016$, $k = 0.0816$, 6732 nodes and 31434 elements.

is excessively coarse. Further simulations obtained with finer grids and different time-integrators are currently under investigation.

Bibliography

[1] R. Abgrall. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys.*, 214:773–808, May 2006.

[2] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of high-order non upwind distribution schemes. In N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sorensen, editors, *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 107–128. Springer Berlin / Heidelberg, 2010.

[3] R. Abgrall, A. Larat, M. Ricchiuto, and C. Tavé. A simple construction of very high order non-oscillatory compact schemes on unstructured meshes. *Comp. Fluids*, 38(7):1314 – 1323, 2009. Special Issue Dedicated to Professor Alain Lerat on the Occasion of his 60th Birthday.

[4] M. Aftosmis. A second-order TVD method for the solution of the 3-D Euler and Navier-Stokes equations on adaptively refined meshes. In *Proc. 13th Int. Conf. Num. Meth. Fluid Dyn.*, 1992.

[5] R. Agarwal and J. Deese. Euler calculation for flowfield of a helicopter rotor in hover. *J. Aircr.*, 24(4):231–237, April 1987.

[6] R. Agarwal and J. Deese. Numerical simulation of vortical flows using vorticity confinement coupled with unstructured grid. In *Proceedings of the* 26^{th} *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1988. AIAA Paper 1988-0106.

[7] F. Alauzet, P. L. George, B. Mohammadi, P. Frey, and H. Borouchaki. Transient fixed point-based unstructured mesh adaptation. *Int. J. Numer. Meth. Fluids.*, 43(6-7):729–745, 2003.

[8] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, May 2001.

[9] A. Bagai and J. G. Leishman. Flow visualization of compressible vortex structures using density gradient techniques. *Exp. Fluids*, 15(6):431–442, 1993.

[10] T. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elem. Anal. Des.*, 25:243–273, 1997.

[11] T. J. Baker. Mesh deformation and modification for time dependent problems. *Int. J. Numer. Meth. Fluids.*, 43(6-7):747–768, 2003.

[12] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and navier-stokes equations. AGARD AR 787-6 on Unstructured Grid Methods for Advection Dominated Flows, NATO, 1992.

[13] F. Bassi, L. Botti, A. Colombo, A. Crivellini, N. Franchina, A. Ghidoni, and S. Rebay. Very high-order accurate discontinuous galerkin computation of transonic turbulent flows on aeronautical configurations. In N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sorensen, editors, *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 25–38. Springer Berlin / Heidelberg, 2010.

[14] F. Bassi, A. Colombo, N. Franchina, A. Ghidoni, and S. Rebay. Robust and efficient implementation of very high-order discontinuous galerkin methods in cfd. In N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sorensen, editors, *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 287–299. Springer Berlin / Heidelberg, 2010.

[15] J. Batina. Unsteady Euler airfoil solution using unstructured dynamic meshes. *AIAA J.*, 28:1381–1388, 1990.

[16] R. Becker and R. Rannacher. Weighted a posteriori error control in fe methods. Technical report, Universitat Heidelberg, 1995.

[17] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Num.*, 10:1–102, 2001.

[18] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley &; Sons, Chichester, UK, 2000.

[19] O. O. Bendiksen. Nonclassical aileron buzz in transonic flow. In *34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, number AIAA Paper 93–1479, April 1993.

[20] R. M. Bennet and J. W. Edwards. An overview of recent developments in computational aeroelasticity. In *Proceedings of the* 29^{th} *AIAA Fluid Dynamics Conference*, Albuquerque, NM, June 15-18 1998.

[21] M. J. Berger and A. Jameson. Automatic adaptive grid refinement for the Euler equations. *AIAA J.*, 23:561–568, 1985.

[22] E. Biganzoli and G. Quaranta. Nonlinear reduced order models for aileron buzz. In *International Forum on Aeroelasticity and Structural Dynamics, IFASD*, Seattle, WA, USA, 21–25 June 2009.

[23] R. L. Bisplinghoff and H. Ashley. *Principles of Aeroelasticity*. Wiley & Sons, New York, 1962.

[24] O. J. Boelens, H. van der Ven, B. Oskam, and A. A. Hassan. Accurate and efficient vortex-capturing for a helicopter rotor in hover. (NLR-TP-2000-420), 2000.

[25] H. Borouchaki, P. George, F. Hecht, P. Laug, and E. Saltel. Mailleur bidimensionnel de delaunay gouverné par une carte de metriques. partie i: Algorithmes. Research report 274, INRIA-Rocquencourt, France, 1995.

[26] J. U. Brackbill. An adaptive grid with directional control. *J. Comput. Phys.*, 108:38–50, September 1993.

[27] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46(3):342 – 368, 1982.

[28] M. O. Bristeau, R. Glowinski, J. Périaux, P. Perrier, O. Pironneau, and G. Poirier. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods ii. application to transonic flow simulations. *Comp. Meth. Appl. Mech. Engng.*, 51, 1985.

[29] N. Butsuntorn and A. Jameson. Time spectral method for rotorcraft flow with vorticity confinement. In *26th AIAA Applied Aerodynamics Conference*, Honolulu, Hawaii, August 18–21 2008.

[30] C. Byun and G. Guruswamy. A parallel, multi-block, moving grid method for aeroelastic applications on full aircraft. AIAA Paper 98-4782, September 1998.

[31] W. Cao, W. Huang, and R. D. Russell. An r-adaptive finite element method based upon moving mesh pdes. *J. Comput. Phys.*, 149(2):221 – 244, 1999.

[32] F. X. Caradonna. Developements and challenges in rotorcraft aerodynamics. In *Proceedings of the* 38^{th} *Aerospace and Science Meeting and Exhibit*, Reno, NV, 10–13 January 2000.

[33] F. X. Caradonna and C. Tung. Experimental and analytical studies of a model helicopter rotor in hover. Technical Report NASA Technical Memorandum 81232, National Aeronautics and Space Administration, 1981.

[34] E. Caramana and M. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of lagrangian subzonal masses and pressures. *J. Comput. Phys.*, 142(2):521 – 561, 1998.

[35] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, October 1998.

[36] G. Carpentieri. *An Adjoint-Based Shape-Optimization Method for Aerodynamic Design*. PhD thesis, Technische Universiteit Delft, Netherlands, September 2009.

[37] M. Castro-Diaz, F. Hecht, B. Mohammad, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *Int. J. Numer. Meth. Fluids.*, 25:475–491, 1997.

[38] M. Castro-Diaz, F. Hecht, and B. Mohammadi. New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. Research report RR-2671, INRIA-Rocquencourt, France, 1995.

[39] H. D. Ceniceros and T. Y. Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys*, 172:609–639, 2001.

[40] P. Chadwick. *Continuum mechanics: concise theory and problems*. Dover books on physics. Dover Publications, 1999.

[41] R. F. Chen and Z. Wang. An improved LU-SGS scheme with faster convergence for unstructured grids of arbitrary topology. *AIAA Paper*, 99-16759, Jan 1999.

[42] A. Conlisk. Modern helicopters aerodynamics. *Ann. Rev. Fluid Mech.*, 29:515–567, 1997.

[43] J. F. Dannenhoffer III and J. R. Baron. Aadaptive procedure for steady state solution of hyperbolic equations. *AIAA Paper*, 84-0005:1–102, 1989.

[44] S. F. Davis and J. E. Flaherty. An adaptive finite element method for initial-boundary value problems for partial differential equations. *SIAM J. Sci. Statist. Comput.*, 3:6–27, 1982.

[45] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Comp. Struct.*, 80:305–316, 2002.

[46] A. Dervieux, D. Leservoisier, P. George, and Y. Coudiere. About theoretical and practical impact of mesh adaptation on approximation of functions and solutions of PDE. In *ECCOMAS CFD conference, Swansea, UK*, 2001.

[47] J. A. Desideri and A. Dervieux. Compressible flow solvers using unstructured grids. Technical Report Lecture Series 1988-05, Von Karman Institute for Fluid Dynamics, 1988.

[48] V. Dolejsi. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Comp. Vis. Sci.*, 1:165–178, 1998.

[49] J. Donea. Arbitrary Lagrangian–Eulerian finite element methods. In T. Belytschko and T. J. Hughes, editors, *Computational Methods for Transient Analysis*, chapter 10, pages 474–516. Elsevier Science Publisher, Amsterdam, The Netherlands, 1983.

[50] A. S. Dvinsky. Adaptive grid generation from harmonic maps on riemannian manifolds. *J. Comput. Phys.*, 95(2):450 – 476, 1991.

[51] S. Etienne, A. Garon, and D. Pelletier. Perspective on the geometric conservation law and finite element methods for ale simulations of incompressible flow. *J. Comput. Phys.*, 228(7):2313 – 2333, 2009.

[52] T. Fanion, M. A. Fernández, and P. Le Tallec. Deriving adequate formulations for fluid-structure interaction problems: from ALE to transpiration. *Rev. Européenne Élém. Finis*, 9(6–7):681–708, 2000.

[53] C. Farhat and P. Geuzaine. Design and analysis of robust ale time-integrators for the solution of unsteady flow problems on moving grids. *Comp. Meth. Appl. Mech. Engng.*, 193:4073–4095, October 2004.

[54] C. Farhat, P. Geuzaine, and C. Crandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *J. Comput. Phys.*, 174:669–694, 2001.

[55] G. Fernandez. Implicit Conservative Upwind Schemes for Strongly Transient Flows. Unite de Recherche, INRIA-Sophia Antipolis 873, INRIA, 1988.

[56] G. Forestieri, A. Guardone, D. Isola, F. Marulli, and G. Quaranta. Numerical simulation of the vortical wake from an oscillating airfoil at subsonic and transonic speeds. In *5^{th} International Conference on Vortex Flows and Vortex Models*, November 7–10 2010.

[57] G. Forestieri, A. Guardone, D. Isola, F. Marulli, and G. Quaranta. A novel adaptive ale scheme for accurate simulations of compressible unsteady fluid dynamics. In *15^{th} International Forum on Aeroelasticity and Structural Dynamics, IFASD*, June 26–30 2011.

[58] G. Forestieri, A. Guardone, D. Isola, F. Marulli, and G. Quaranta. Numerical simulation of aileron buzz using an adaptive-grid compressible flow solver for dynamic meshes. In *Proceedings of the IV Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*. CIMNE, Barcelona, 2011.

[59] G. Forestieri, D. Isola, F. Marulli, G. Quaranta, and A. Guardone. Numerical simulation of compressible vortical flows using a conservative unstructured-grid adaptive scheme. In *2nd European Seminar on Coupled Problems*, June 28^{th}–July 2^{nd} 2010.

[60] G. Forestieri, D. Isola, F. Marulli, G. Quaranta, and A. Guardone. Numerical simulation of compressible vortical flows using a conservative unstructured-grid adaptive scheme. *Commun. Comput. Phys.*, 12:866–884, March 2012.

[61] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaption with application to CFD problems. In *Proceedings of WCCMV, Fifth World Congress on Computational Mechanics*, Wien, Austria, 2002.

[62] L. Formaggia and F. Nobile. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West J. Num. Math.*, 7:105–132, 1999.

[63] M. Fossati, A. Guardone, and L. Vigevano. A node-pair finite element / finite volume mesh adaptation technique for compressible flows. In *$40th$ Fluid Dynamics Conference and Exhibit*, 2010.

[64] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Meth. Eng.*, 40(21):3979–4002, 1997.

[65] P. J. Frey and P. L. George. *Mesh Generation: Application to Finite Elements*. Hermes Science, 2000.

[66] P. P. Friedmann. *Rotary Wing Aeroelasticity*. John Wiley & Sons, Ltd, 2010.

[67] R. Garimella, M. Kucharik, and M. Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Comp. Fluids*, 36(2):224 – 237, 2007.

[68] P. L. George, H. Borouchaki, and P. Laug. An efficient algorithm for 3D adaptive meshing. *Adv. Eng. Soft.*, 33(7):377–387, 2002.

[69] P. Geuzaine, C. Grandmont, and C. Farhat. Design and analysis of ale schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *J. Comput. Phys.*, 191:206–227, October 2003.

[70] M. Giles, M. G. Larson, J. M. Levenstam, and E. Suli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. Technical report, SIAM, 1997.

[71] M. B. Giles. Stability analysis of a Galerkin/runge–Kutta navier–stokes discretization on unstructured tetrahedral grids. *J. Comput. Phys.*, 132:201–214, 1997.

[72] E. Godlewski and P. A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*. Springer-Verlag, New York, 1994.

[73] S. K. Godunov. A difference method for the numerical computation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb.*, 47(3):271–306, 1959.

[74] A. Guardone, D. Isola, and G. Quaranta. An ale scheme without interpolation for moving domain with adaptive grids. *J. Aerosp. Sci. Tech. Sys.*, 88:71–81, 2009.

[75] A. Guardone, D. Isola, and G. Quaranta. Time-accurate conservative adaptive schemes for compressible flow simulations. In *10^{th} International Meeting on Applied Scientific Computing and Tools*, October 20–22 2010.

[76] A. Guardone, D. Isola, and G. Quaranta. Arbitrary lagrangian eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping. *J. Comput. Phys.*, 230(20):7706–7722, 2011.

[77] A. Guardone, D. Isola, and G. Quaranta. Transient spoiler aerodynamics using a conservative adaptive-grid ale scheme. In *CEAS 2011, XXI AIDAA Congress*, October 24–28 2011.

[78] A. Guardone and L. Quartapelle. Spatially factorized Galerkin and Taylor-Galerkin schemes for multidimensional conservation laws. Scientific Report DIA-SR 00-18, Politecnico di Milano, Italy, 2000.

[79] V. H. Numerical solution of two-dimensional reference test cases. AGARD AR-211-6, NATO, 1985.

[80] W. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M. Vallet. Anisotropic mesh adaptation: towards user-indipendent, mesh-indipendent and solver-indipendent CFD. Part I: General principles. *Int. J. Numer. Meth. Fluids*, 32:725–744, 2000.

[81] N. G. Hardin and S. M. Lamkin. Concepts for reduction of blade/vortex interaction noise. *J. Aircr.*, 24(2), 1987.

[82] A. Haselbacher, J. J. McGuirk, and G. J. Page. Finite volume discretization aspects for viscous flows on mixed unstructured grid. *AIAA J.*, 37:177–184, Feb 1999.

[83] O. Hassan, K. Morgan, and N. P. Weatherill. Unstructured mesh methods for the solution of the unsteady compressible flow equations with moving boundary components. *Phil. Trans. R. Soc. Lond.*, A 365:2531–2552, 2007'.

[84] P. Hemker and S. Spekreijse. Multiple grid and osher's scheme for the efficient solution of the steady euler equations. *App. Num. Math.*, 2(6), 1986.

[85] J. Hess and A. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aeronautical Sciences*, 8, 1967.

[86] C. Hirsch. *Numerical computations of internal and external flows*. Wiley, 1988.

[87] C. Hirt, A. Amsden, and J. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14(3):227 – 253, 1974.

[88] W. Hundsdorfer, S. J. Ruuth, and R. J. Spiteri. Monotonicity-preserving linear multistep methods. *SIAM J. Num. Anal.*, 41(2):605–623, Apr. 2003.

[89] D. Isola. An ALE scheme without interpolation for moving domains with adaptive grids. In *5th Pegasus-AIAA student conference*, Toulose, France, March 12$^{\mathrm{th}}$ 2009.

[90] D. Isola, A. Guardone, and G. Quaranta. An Edge-Based ALE Scheme for Adaptive Dynamic Meshes. In *20th AIDAA Congress*, Milano, Italy, June 29$^{\mathrm{th}}$-July 3$^{\mathrm{rd}}$ 2009.

[91] D. Isola, A. Guardone, and G. Quaranta. Arbitrary lagrangian eulerian formulation for grids with variable topology. In *Proceedings of the III Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*. CIMNE, Barcelona, 2009.

[92] D. Isola, A. Guardone, and G. Quaranta. Arbitrary langrangian eulerian formulation for grids with variable topology. In *Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, Ischia, Italy, 2009.

[93] D. Isola, A. Guardone, and G. Quaranta. An ale scheme without interpolation for moving domain with adaptive grids. In *40th Fluid Dynamics Conference and Exhibit, AIAA*, 2010.

[94] A. Jameson. Iterative solution of transonic flows over airfoils and wings, including flow at Mach 1. *Comm. Pure Appl. Math.*, 17, 1974.

[95] A. Jameson. Cfd for aerodynamic design and optimization: Its evolution over the last three decades. *AIAA Paper*, 2003-3438, 2003.

[96] A. Jameson, T. Baker, and N. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. *AIAA Paper*, 86-0103, 1986.

[97] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. In *14th AIAA Fluid and Plasma Dynamic Conference*, pages 1–17, 1981. AIAA Paper 81-1259.

[98] A. Jameson and S. Yoon. Lower-upper implicit schemes with multiple grids for the euler equations. *AIAA J.*, 25:929–935, 1987.

[99] F. T. Johnson, E. N. Tinoco, and N. J. Yu. Thirty years of development and application of cfd at boeing commercial airplanes, seattle. *Comp. Fluids*, 34:1115–1151, December 2005.

[100] J. Katz and J. Plotkin. *Low Speed Aerodynamics*. Cambridge University Press, 2001. second edition.

[101] B. Koren. Defect correction and multigrid for an efficient and accurate computation of airfoil flows. *J. Comput. Phys.*, 77(1):183–206, 1988.

[102] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. Ven, and K. Sorensen. *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009*. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2010.

[103] M. Kucharik. *Arbitrary Lagrangian-Eulerian (ALE) methods in plasma physics*. PhD thesis, Czech Technical University in Prague, 2006.

[104] M. Kucharik, R. Liska, P. Váchal, and M. Shashkov. Arbitrary lagrangian-eulerian (ale) methods in compressible fluid dynamics. In *Programs and Algorithms of Numerical Mathematics 13*, pages 178–183, 2006.

[105] M. Kucharik, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping method. *J. Comput. Phys.*, 188:462–471, July 2003.

[106] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: the Initial Value Problem*. John Wiley & Sons, Chichester, UK, 1991.

[107] N. Lambourne. Flutter in one degree of freedom. In *Manual on Aeroelasticity, Part V*, chapter 5. AGARD, 1961.

[108] N. Lambourne. Control–surface buzz. Reports and Memoranda 3364, Aeronautical Research Council, London, UK, May 1964.

[109] C. B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.

[110] M. Larson and T. Barth. A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems. Technical report NAS-99-010, NASA, 1999.

[111] G. Leishman. *Principles of Helicopter Aerodynamics (Cambridge Aerospace Series)*. Cambridge University Press, 2 edition, April 2006.

[112] R. J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser, Basel, 1992.

[113] R. J. LeVeque. *Finite volume methods for conservation laws and hyperbolic systems*. Cambridge University Press, 2002.

[114] R. Li, T. Tang, and P. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys*, 170:562–588, 2000.

[115] R. Li, T. Tang, and P. Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *J. Comput. Phys.*, 177(2):365 – 393, 2002.

[116] S. Li and L. Petzold. Moving mesh methods with upwinding schemes for time-dependent pdes. *J. Comput. Phys*, 131:368–377, 1997.

[117] R. Lohner. Mesh adaptation in fluid mechanics. *Eng. Fract. Mech.*, 50:819–847, 1995.

[118] R. Lohner and J. D. Baum. Adaptive h-refinement on 3D unstructured grids for transient problems. *Int. J. Numer. Meth. Fluids.*, 14:1407–1419, June 1992.

[119] R. Lohner and C. Yang. Vorticity confinement on unstructured grids. In *40^{th} AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2002. AIAA Paper 2002-0137.

[120] R. Loubère, P. H. Maire, M. Shashkov, J. Breil, and S. Galera. Reale: A reconnection-based arbitrary-lagrangian-eulerian method. *J. Comput. Phys.*, 229:4724–4761, 2010.

[121] R. L. H. Luo and J. D. Baum. A fast, matrix-free implicit method for compressible flows on unstructured grids. *J. Comput. Phys.*, 146:664–690, 1998.

[122] L. Machiels, J. Peraire, and A. Patera. A posteriori finite element output bounds for the incompressible Navier-Stokes equations: application to a natural convection problem. *J. Comput. Phys.*, 172:401–425, 2001.

[123] F. Malovrh and P. O. Gandhi. Sensitivity of helicopter blade-vortex-interaction noise and vibration to interaction parameters. *J. Aircr.*, 42(3), May–June 2005.

[124] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

[125] L. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J. Comput. Phys.*, 184:266–298, 2003'.

[126] D. Mavriplis. On convergence acceleration techniques for unstructured meshes. *AIAA Paper*, 98-2966, June 1998.

[127] D. Mavriplis. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *Int. J. Numer. Meth. Fluids*, 34:93–111, 2000.

[128] D. Mavriplis and Z. Yang. Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. *J. Comput. Phys.*, 213(3):557–573, April 2006. 2006.

[129] D. J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report Lecture Series VKI-LS 1995-02, Von Karman Institute for Fluid Dynamics, 1995.

[130] D. J. Mavriplis. Unstructured grid techniques. *Ann. Rev. Fluid Mech.*, 29, 1997.

[131] D. J. Mavriplis. Unstructured and adaptive mesh generation for high-reynolds number viscous flows. Report 187534, NACA ICASE, 2005.

[132] D. J. Mavriplis and Z. Yang. Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. *J. Comput. Phys.*, 213:557–573, 2006.

[133] D. S. McRae. r-refinement grid adaptation algorithms and issues. *Comp. Meth. Appl. Mech. Engng.*, 189(4):1161 – 1182, 2000.

[134] C. Michler, H. D. Sterck, and H. Deconinck. An arbitrary lagrangian eulerian formulation for residual distribution schemes on moving grids. *Comp. Fluids*, 32(1):59 – 71, 2003.

[135] K. Miller and R. N. Miller. Moving finite elements. i. *SIAM J. Num. Anal.*, 18(6):pp. 1019–1032, 1981.

[136] B. Mohammadi and F. Hecht. Mesh adaptation for time dependent simulation, optimization and control. *Rev. Europ. Elem. Fin.*, 10(5):575–593, 2001.

[137] L. Morino and C. Kuo. Subsonic potential aerodynamics for complex configurations: a general theory. *AIAA J.*, 12(2), 1974.

[138] W. A. Mulder. Multigrid relaxation for the euler equations. *J. Comput. Phys.*, 60(2), 1985.

[139] M. Murayama, K. Nakahashi, and S. Obayashi. Numerical simulation of vortical flows using vorticity confinement coupled with unstructured grid. In *39^{th} AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 08–11 2001.

[140] E. Murman and J. Cole. Calculation of plane steady transonic flows. *AIAA J.*, 9, 1971.

[141] F. Nobile. *Numerical Approximation of Fluid–Structure Interaction Problems with Application to Hemodynamics*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, September 2001.

[142] W. Oh, J. Kim, and O. Kwon. Numerical simulation of two-dimensional blade-vortex interactions using unstructured adaptive meshes. *AIAA J.*, 40(3):474–480, 2002.

[143] S. Osher and S. Chakravarthy. Upwind schemes and boundary conditions with applications to euler equations in general geometries. *J. Comput. Phys.*, 50(3), June 1983.

[144] C. C. Pain, A. P. Umpleby, C. R. E. de Oliveira, and A. J. H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Comp. Meth. Appl. Mech. Engng.*, 190(29–30):3771 – 3796, 2001.

[145] J. Peraire and A. Patera. Asymptotic a posteriori finite element bounds for the outputs of non-coercive problems: the Helmholtz and Burgers equations. *Comput. Meth. Appl. Eng.*, 171:77–86, 1999.

[146] J. Peraire, M. Vadhati, K. Morgan, and O. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J.Comput. Phys.*, 72:449–466, 1987.

[147] S. Piperno and C. Farhat. Partitioned procedure for the transient solution of coupled aeroelastic problems. Part II: Energy transfer analysis and three dimensional applications. *Comp. Meth. Appl. Mech. Engng.*, 190:3147–3170, 2001.

[148] S. Pirzadeh. Unstructured viscous grid genereration by the advancing layers method. *AIAA J.*, 32(8):1735–1737, 1994.

[149] S. Pirzadeh. An adaptive unstructured grid method by grid subdivision, local remeshing and grid movement. In *14th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 99-3255, Norfolk, Virginia, 1999.

[150] D. Pisano, F. R. Van der Linden, and F. H. W. Winter. *Chuck Yeager and the Bell X-1: Breaking the Sound Barrier*. Washington, D.C.: Smithsonian National Air and Space Museum, 2006.

[151] G. Quaranta, D. Isola, and A. Guardone. Numerical simulation of the opening of aerodynamic control surfaces with two-dimensional unstructured adaptive meshes. In *5^{th} European Conference on Computational Fluid Dynamics, ECCOMAS CFD*, June 14–17 2010.

[152] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *J. Comput. Appl. Math.*, 128:205–233, 2001.

[153] R. D. Rausch, J. T. Batina, and H. T. Y. Yang. *Spatial adaptation procedures on tetrahedral meshes for unsteady aerodynamic flow calculations*. NASA, Langley Research Center; National Technical Information Service, distributor, Hampton, Va, 1993.

[154] S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and Bowyer-watson algorithm. *J. Comput. Phys.*, 106:125–138, 1993.

[155] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamics shape optimization of complex aircraft configurations via an adjoint formulation. AIAA Paper 96-0094, AIAA, January 1996.

[156] M. Ricchiuto. *Construction and Analysis of Compact Residual Discretizations for Conservation Laws on Unstructured Meshes*. PhD thesis, von Karman Institute for Fluid Dynamics, 2005.

[157] P. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43(2), 1981.

[158] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[159] S. J. Ruuth and W. Hundsdorfer. High-order linear multistep methods with general monotonicity and boundedness properties. *J. Comput. Phys.*, 209:226–248, October 2005.

[160] D. M. Schuster, D. D. Liu, and L. J. Huttsell. Computational aeroelasticity: Success, progress, challenge. *J. Aircr.*, 40(5):843–856, 2003.

[161] M. P. Scully. Computation of helicopter wake geometry and its influence of rotor harmonic airloads. Report ASR TR178-1, MIT, 1975.

[162] V. Selmin. The node-centred finite volume approach: bridge between finite differences and finite elements. *Comp. Meth. Appl. Mech. Engng.*, 102(1):107–138, 1993.

[163] V. Selmin. The node-centred finite volume approach: bridge between finite differences and finite elements. *Comp. Meth. Appl. Mech. Engng.*, 102:107–138, 1993.

[164] V. Selmin and L. Formaggia. Unified construction of finite element and finite volume discretizations for compressible flows. *Int. J. Numer. Meth. Eng.*, 39:1–32, 1996.

[165] J. Shewchuk. What is a good linear element? Interpolation, conditioning, and quality measures. In 11^{th} *International Meshing Roundtable*, pages 115–126, Ithaca, NY, September 15–18 2002.

[166] W. Shyy, R. W. Smith, H. S. Udaykumar, and M. M. Rao. *Computational Fluid Dynamics with Moving Boundaries*. Taylor & Francis, Inc., Bristol, PA, USA, 1996.

[167] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic consevation laws. *J. Comput. Phys.*, 27(1), 1978.

[168] T. Sonar. Strong and weak norm refinement indicators based on the finite element residual for compressible flow computation. *IMPACT Comp. Sci. Eng.*, 5:111–127, 1993.

[169] T. Sonar, V. Hannemann, and D. Hempel. Dynamic adaptivity and residual control in unsteady compressible flow computation. *Math. Comput. Model.*, 20:201, 1994.

[170] S. Spekreijse. Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws. *Math. Comput.*, 49(179), 1987.

[171] G. R. Srinivasan and W. J. McCroskey. Euler Calculations of Unsteady Interaction of Advancing Rotor with a Line Vortex. *AIAA J.*, 31:1659–1666, Sep 1993.

[172] G. R. Srinivasan, W. J. McCroskey, and J. D. Baeder. Aerodynamics of two-dimensional blade-vortex interaction. *AIAA J.*, 24(10), 1986.

[173] G. R. Srinivasan, W. J. McCroskey, and J. D. Baeder. Numerical simulations of unsteady airfoil-vortex interactions. *Vertica*, 11(1), 1987.

[174] J. M. Stockie, J. A. Mackenzie, and R. D. Russell. A moving mesh method for one-dimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 22:1791–1813, May 2000.

[175] L. Tang and J. D. Baeder. Adaptive Euler simulations of airfoil-vortex interaction. *Int. J. Numer. Meth. Fluids.*, 53(5):777–792, 2007.

[176] T. Theodorsen and I. E. Garrick. General potential theory of arbitrary wing sections. Technical Report 452, NACA Report, 1933.

[177] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA J.*, 17:1030–1037, 1979.

[178] J. F. Thompson. *Numerical grid generation: foundations and applications*. Elsevier Science Publishing, 1985.

[179] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications*. North-Holland, Elsevier, 1985.

[180] P. A. Thompson. *Compressilbe Fluid Dynamics*. McGraw-Hill, 1988.

[181] N. Trivellato. *Un approccio adattivo per la soluzione delle equazioni di Eulero su griglie non strutturate*. PhD thesis, Politecnico di Milano, 1995.

[182] C. Truesdel. *A first course in rational continuum mechanics: General concepts*. Academic Press, 1991.

[183] P. Váchal, R. V. Garimella, and M. J. Shashkov. Untangling of 2D meshes in ale simulations. *J. Comput. Phys*, 196:627–644, 2004.

[184] P. Váchal and R. Liska. Sequential flux-corrected remapping for ale methods. In *Numerical Mathematics and Advanced Applications*, pages 671–679. Springer Berlin Heidelberg, 2006.

[185] G. van Albada, B. van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astron. Astrophys.*, 108(1), 1982.

[186] B. van Leer. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361–370, 1974.

[187] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two dimensional viscous flows. *J. Comput. Phys.*, 187:22–46, 2003.

[188] V. Venkatakrishnan. Perspective on unstructured grid flow solvers. *AIAA J.*, 34(3), 1996.

[189] V. Venkatakrishnan and D. J. Mavriplis. Implicit solvers for unstructured meshes. *J. Comput. Phys.*, 105:83–91, 1993.

[190] V. Venkatakrishnan and D. J. Mavriplis. Implicit method for the computation of unsteady flows on unstructured grids. *J. Comput. Phys.*, 127:380–397, 1996.

[191] C. M. Wang, J. S. Steinhoff, and Y. Wenren. Numerical vorticity confinement for vortex-solid body interaction problems. *AIAA J.*, 33:1447–1453, aug 1995.

[192] R. Wang, P. Keast, and P. Muir. A comparison of adaptive software for 1D parabolic PDEs. *J. Comput. Appl. Math.*, 169(1):127–150, 2004.

[193] G. P. Warren, W. K. Anderson, J. T. Thomas, and S. L. Krist. Grid convergence for adaptive methods. In *AIAA 10^{th} Computational Fluid Dynamics Conference*, 1991. AIAA Paper 91-1592.

[194] N. P. Weatherill, O. Hassan, M. Marchant, and D. Marcum. Adaptive inviscid solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. In *AIAA Paper 93-3390-CP*. 11th AIAA Computational Fluid Dynamics Conference, Orlando, FL, 1993.

[195] B. Webster, M. Shepard, Z. Rhusak, and J. Flaherty. Automated adaptive time discontinuous finite element method for unsteady compressible airfoil. *AIAA J.*, 32:748–757, 1994.

[196] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, 2001.

[197] A. Winslow. Equipotential zoning of two-dimensional meshes. Technical Report UCRL-7312, California Univ., Livermore (USA). Lawrence Livermore Lab., 1963.

[198] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.*, 1(2):149 – 172, 1966.

[199] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:114–173, 1984.

[200] G. Xia, D. Li, and C. L. Merkle. Anisotropic grid adaptation on unstructured meshes. In *39^{th} Aerospace Sciences Meeting and Exhibit*, 2001. AIAA Paper 2001-0443.

[201] H. Yoshihara and P. Sacher. Test cases for inviscid flow field methods. AGARD AR-211-6, NATO, 1985.

[202] L. P. Zhang, X. H. Chang, X. P. Duan, Z. Y. Wang, and H. X. Zhang. A block LU-SGS implicit unsteady incompressible flow solver on hybrid dynamic grids for 2D external bio-fluid simulations. *Comp. Fluids*, 38(2):290 – 308, 2009.

[203] O. Zienkiewicz, R. Taylor, and P. Nithiarasu. *The Finite Element Method For Fluid Dynamics*. Butterworth-Heinemann, 2005. sixth edition.

Acknowledgments

A dear friend of mine once told me that ninety nine percent of the Ph.D. thesis consists in "*moving bones from one grave to another*". This one probably makes no exception.

As most of the students already know, getting through the years of a Ph.D. program is never easy. It has nothing to do with the long hours spent in front of non-responsive computer program, mostly from 11pm to 3am, while spasmodically surfing the Internet between one compilation and the other. It is not related to the frustration caused by the collected data that always manage to be inconsistent with each other, and sometimes with themselves, nor to the constant fear of a `segmentation fault`. After a while you get used to everything and you learn that what you can not deal with, it is probably better left alone. The hardest part is learning to live with the idea that you are procrastinating something that eventually will come at you: job, moving out, moving in, earning big, life. Whether you made plans or not, you can not help but feeling like a sitting duck. This is why having people around you, supporting you, is so desperately important.

I want to express my gratitude to my supervisor, Prof. Guardone, for the support and the guidance provided along the way. The first time I met him I was a very young student who could not even write down the compressible Navier-Stokes equations, after five years I probably forgot how to do that but I learned much more from him. I sincerely enjoyed the endless hours spent talking and arguing in his office about data structure, numerical dissipation, equation scaling or Galilean invariance. I am proud to be the first Ph.D. student that he have had. Thanks to Prof. Quaranta, for showing me more practical approaches to the problems and to Prof. Masarati and Prof. Vigevano for the suggestions provided along the years. I want to thank Prof. Baeder, Prof. Celi and the rest of the crew from University of Maryland, where I spent my time abroad. I say thank you to Prof. Abgrall, who offered to review this work and provided very valuable suggestions. I also want to acknowledge the priceless help of Giuseppe Forestieri and Filippo Marulli, I really enjoyed working with them.

Thanks to my parents and my sister for supporting me and being there for me in their own weird way. Thanks to my aunt Marita for loving me like I am still five years old, it keeps me young. Thanks to my uncle Francesco for patiently putting up with my rude manners. Also thanks to all of my closest friends for helping me getting though this. To Alberto for always having my back, for listening to my doubts and frustrations and for the confidence he always had in me. To Gaetano which is amongst the persons I look up to the most. To Pietro for giving me food and a shelter every other day. To Simone for being the ideal neighbor that everybody wishes to have. To Roberto who provided the cover page design.

At last Lorenza. No matter what, you were there for me. If I ever thought that I had a hard time, you probably had it worse. We got through the year I spent in Maryland and we are not done yet. There is actually nothing I could say but thank you and I am sorry. I promise I will make it up to you.