UNIVERSITÀ DEGLI STUDI DI BERGAMO
Facoltà di Ingegneria
Dipartimento di Ingegneria Industriale

DOTTORATO DI RICERCA
IN
TECNOLOGIE PER L'ENERGIA E L'AMBIENTE

XXIII ciclo

anno 2011

AN AGGLOMERATION-BASED DISCONTINUOUS GALERKIN METHOD FOR COMPRESSIBLE FLOWS

Tesi di Dottorato di:
Alessandro Colombo

UNIVERSITÀ DEGLI STUDI DI BERGAMO
Facoltà di Ingegneria
Dipartimento di Ingegneria Industriale

DOTTORATO DI RICERCA
IN
TECNOLOGIE PER L'ENERGIA E L'AMBIENTE

XXIII ciclo

anno 2011

AN AGGLOMERATION-BASED DISCONTINUOUS GALERKIN METHOD FOR COMPRESSIBLE FLOWS

Tesi di Dottorato di:
Alessandro Colombo

Supervisore:
Prof. Francesco Bassi

Bergamo, Dipartimento di Ingegneria Industriale -
Università degli studi di Bergamo, 2011
ISBN: 978-88-97413-01-1

Abstract

This thesis investigates the flexibility associated to Discontinuous Galerkin (DG) discretization on very general meshes obtained by means of agglomeration techniques. The work begins with a brief overview of the main tools that have been extended or specifically developed to deal with arbitrarily shaped elements in the DG context. Then two different implementations of the BRMPS scheme introduced by Bassi, Rebay, Mariotti, Pedinotti and Savini in [16] for the DG discretization of the Laplace operator on arbitrarily shaped elements have been presented. The validation of the scheme on a Poisson problem shows that the discrete polynomial space preserves optimal convergence properties.

The discretization of the second order differential operator has been directly extended to the Navier-Stokes equations and the Reynolds Averaged Navier-Stokes (RANS) equations coupled with the k-ω turbulence model of Wilcox [54]. In this regard, an implicit time integration strategy has been considered and assessed on classical validation test cases for the compressible fluid dynamics.

Then a simple alternative approach to high-order mesh generation is presented. Indeed, once a standard fine grid able to provide an accurate domain discretization has been produced by means of standard low-order grid generation tools, a computational mesh suitable for the desired accuracy and computationally affordable can be obtained via agglomeration while keeping the boundary resolution of the fine grid. The effectiveness of this approach in representing the geometry of the domain is numerically assessed both on a Poisson model problem and on challenging inviscid and viscous test cases.

Finally, the freedom in simply defining the topology of agglomerated meshes leads to a nonstandard approach to h-adaptivity that exploits adaptive agglomeration coarsening of a properly fine underlying grid. The effectiveness of this approach has been assessed on test cases involving both error-based and flow feature-based simple estimators.

Keywords: Discontinuous Galerkin methods, Laplacian discretization, Polymorphic elements, Orthonormal hierarchical basis functions, h-adaptivity, Compressible Euler and Navier-Stokes equations, RANS+k-ω equations, Curved boundaries
PACS: 02.70.Dh, 02.60.Lj, 47.10.ad, 47.11.Fg

Acknowledgments

It is a pleasure to thank the people who made this work possible.

First and foremost, I would like to thank my supervisor Prof. Francesco Bassi for his guidance, patience and great efforts to explain things clearly and simply.

I am very grateful to Lorenzo Botti, his collaboration and contagious enthusiasm has been of great importance for the realization of this work.

I am thankful to Nicoletta Franchina for the collaboration and helpfulness during these years.

Finally, I would like to thank Alice for her infinite patience and comprehension and my family for the support during these years.

Contents

Chapter 1

Introduction

1.1 Motivation

Computational Fluids Dynamics (CFD) is applied to ever more numerous problems arising in disparate research fields and industrial applications. This trend has been fostered by the concurrent development of hardware capabilities and of accurate, robust and flexible numerical methods, which are applied to reliably solve ever more complex physical models. As a result new higher-order methods combining the above features are expected to be implemented in a new generation of CFD codes in the near future.

In recent years, the ability to deal with general unstructured polymorphic grids has received growing interest in the CFD community. Several Finite Volume (FV) schemes have been extended in this direction in order to improve grid quality and accuracy of numerical solutions for a given computational cost. Unlike standard FV methods, Discontinuous Galerkin (DG) methods are pretty insensitive to the shape of mesh elements, and apparently there would be no special reason to extend the method to element shapes other that those usually employed in unstructured grids (triangles and quadrilaterals in 2D, tetrahedra, prisms, pyramids and hexahedra in 3D). Instead, just the flexibility offered by the high-order DG discretiza-

tion suggests and makes it straightforward to implement the method on elements of any shape in order to fully exploit numerical strategies that already proved successful, and new ones. The geometric h-multigrid method is an example of the former while an agglomeration-based h-adaptivity and an alternative approach to high-order mesh generation are examples of the latter.

In this context, the first objective of this thesis is to propose a DG method suitable for the robust high-order solution of compressible flows on grids consisting of arbitrarily shaped elements. The DG discretization is performed on meshes obtained by agglomerating elements belonging to an underlying fine grid, following the approach devised by Tesini, [52], in the framework of h-multigrid. This strategy relies on the use of discrete polynomial spaces defined in the physical frame as the key ingredient to handle high-order approximation on general meshes.

The main building block of a DG discretization on arbitrarily shaped elements allows to achieve two valuable objectives that naturally fit in this framework. The first one is the ability to preserve the domain discretization of the underlying fine grid on the coarse agglomerated grid, thus leading to an alternative approach to high-order mesh generation techniques. In fact, this strategy allows to obtain a faithful representation of the geometry as a result of the collection of low-order (*e.g.*, linear or quadratic) faces of the underlying grid forming the bounding faces of an agglomerated element. The second objective is to set up a nonstandard approach to Adaptive Mesh Refinement (AMR) based on mesh agglomeration possibly driven by means of suitable *a posteriori* estimators. Of course, this peculiar approach to h-adaptivity maintains the geometrical representation guaranteed by the underlying fine mesh at each adaptation step.

1.2 Thesis outline

The thesis is organized as follows. Chapter 2 presents the key-tools required to build discrete DG spaces on very general meshes resulting from agglomeration coarsening of standard grids. The definition of hierarchical

orthonormal shape functions in the real space and the numerical integration on polymorphic elements are here briefly introduced.

Chapter 3 is devoted to the DG discretization of the Laplace operator. We we discuss how to extend the BRMPS scheme introduced by Bassi, Rebay, Mariotti, Pedinotti and Savini in [16] to arbitrarily shaped elements. For this purpose two different implementations of the method and the related stability coefficients are given. The method validation is performed by means of numerical tests on a Poisson problem and on L^2-orthogonal projection. The major result of this chapter is to show that optimal convergence properties are maintained for the discrete polynomial spaces.

In Chapter 4 an implicit discontinuous finite element method for compressible fluid dynamics is presented and assessed with several test cases. Both inviscid and viscous transonic flows are considered and the behaviour on coarse polymorphic grids of the shock-capturing approach devised by Bassi, Botti, Colombo, Crivellini, Franchina, Ghidoni and Rebay in [5] is evaluated. Moreover, some early stage results obtained on turbulent flows employing the Reynolds averaged Navier-Stokes (RANS) equations coupled with the k-ω model of Wilcox [54] are shown.

Chapter 5 is devoted to a simple alternative approach to high-order mesh generation. Indeed, once a standard fine grid able to provide an accurate domain discretization has been produced by means of standard low-order grid generation tools, a computational mesh suitable for the desired accuracy and computationally affordable can be obtained via agglomeration while keeping the boundary resolution of the fine grid. The effectiveness of this approach in representing the geometry of the domain is numerically assessed both on a Poisson model problem and on challenging inviscid and viscous test cases.

Finally, Chapter 6 presents a novel approach to adaptive mesh refinement where the agglomeration of a very fine grid is driven by a suitable estimator. Test cases involving both error-based and flow feature-based estimators are presented.

Chapter 2

Basic concepts for DG methods on arbitrarily shaped elements

2.1 Introduction

The purpose of this chapter is to briefly introduce some tools that can be extended or have been specifically developed to deal with arbitrarily shaped elements in the DG context. This chapter has to be considered as an overture to this work of thesis.

2.2 Extension of DG method to arbitrary grids

When working with elements of arbitrary and unpredictable shape the definition of reference element fails to exist. For this reason the classical approach based on the definition of polynomial basis functions in the reference frame cannot be applied. Therefore, a different standpoint has to be adopted by defining the discrete polynomial spaces in the physical frame.

2.2.1 Discrete DG space and basis functions definition on arbitrarily shaped elements

Let $\mathcal{T}_h$ denote a (possibly non conforming) mesh of the domain $\Omega \quad \mathbb{R}^d, d$ $2, 3$ such that the computational approximation of the domain, Ω_h is partitioned into a finite set of non-overlapping elements of general shape T

$$\overline{\Omega}_h = \bigcup_{T \in \mathcal{T}_h} \overline{T}. \tag{2.1}$$

Following the idea to define discrete polynomial spaces in physical coordinates, see *e.g.*, [16, 15, 9, 10, 22, 25], let us consider DG approximation based on the space

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \Big\{ v_h \quad L^2(\Omega) \quad v_{h|T} \quad \mathbb{P}_d^k(T), \quad T \quad \mathcal{T}_h \Big\}, \tag{2.2}$$

where k is a non-negative integer and $\mathbb{P}_d^k(T)$ is the restriction to T of the polynomials functions of d variables and global degree at most k.

For a given polynomial of degree $k \geq 0$, let us assume $V_h \stackrel{\text{def}}{=} \mathbb{P}_d^k(\mathcal{T}_h)$, a basis for this space is given by

$$\Phi^k \stackrel{\text{def}}{=} \quad \Phi_T^k \quad _{T \in \mathcal{T}_h}, \tag{2.3}$$

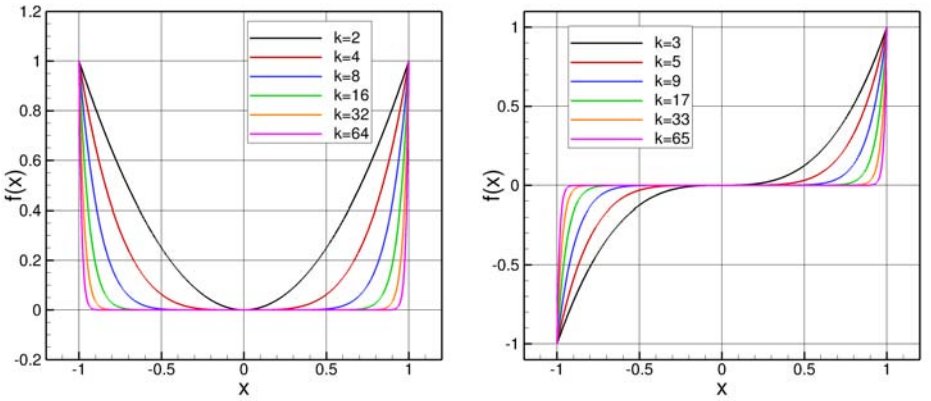
where

$$\Phi_T^k = \quad \varphi_i^T \quad _{i \in \{1, \ldots, N_{\text{dof}}^T\}}, \tag{2.4}$$

is a basis for $\mathbb{P}_d^k(T)$ and N_{dof}^T its cardinality.

A simple choice is to take Φ_T^k equal to the set of monomials in the physical space coordinates $\mathbf{x} = (x_i)_{1 \leq i \leq d}$. In this case, it is convenient to define $\mathbf{x}$ with respect to an element-specific physical frame of reference whose axes are aligned with the principal axes of inertia of T and origin coincides with the barycenter of T, see [52, 6]

$$\Phi_T^k = \quad \mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x}) \quad _{\boldsymbol{\alpha} \in \mathbb{N}^d, \, \|\boldsymbol{\alpha}\|_1 \leq k}, \qquad \mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} x_i^{\alpha_i}. \tag{2.5}$$



(a) Monomials, even modes

(b) Monomials, odd modes

Figure 2.1: Monomial basis on a 1D linear elements.

It is well known that monomial basis functions present severe drawbacks when high-degree polynomials and/or stretched curved elements are considered. Indeed, their approximation proprieties degenerate as depicted in Figure 2.1 where for higher orders the different modes in the middle of the element are almost indistinguishable. This behavior reflects in a degradation of the condition number of the elementary mass matrices. To avoid this disadvantage, a satisfactory basis for the space (2.2) can be obtained by means of the procedure derived by Tesini [52], see also Di Pietro and Ern [21, 6], allowing to obtain orthonormal and hierarchical basis functions through the modified Gram-Schmidt (MGS) algorithm [28]. The MGS procedure needs a suitable initial guess, a simple and effective choice is the set of monomials, see [52, 6]. The sole requirement to apply this algorithm is the ability to compute integrals of polynomial functions on each arbitrarily shaped element $T \quad \mathcal{T}_h$, see § 2.2.3.

For the sake of completeness the modified Gram-Schmidt (MGS) algorithm with re-orthogonalization is reported below, where

$$\widehat{\Phi}_T^k = \quad \widehat{\varphi}_i^T \quad _{i\in\{1,\dots,N_{\mathrm{dof}}^T\}}, \tag{2.6}$$

is the initial set of linearly independent basis functions.
For all $T \quad \mathcal{T}_h$,

1: **for** $i = 1$ **to** N_{dof}^T **do**
2: **for** $j = 1$ **to** $i - 1$ **do**
3: $r_{ij}^T \quad \left(\widehat{\varphi}_i^T, \varphi_j^T\right)_T$
4: $\widehat{\varphi}_i^T \quad \widehat{\varphi}_i^T - r_{ij}^T \varphi_j^T$ Remove the projection of $\widehat{\varphi}_i^T$ onto φ_j^T
5: **end for**
6: $r_{ii}^T \quad \sqrt{\left(\widehat{\varphi}_i^T, \widehat{\varphi}_i^T\right)_T}$
7: $\widehat{\varphi}_i^T \quad \widehat{\varphi}_i^T / r_{ii}^T$ Normalize
8: $\varphi_i^T \quad \widehat{\varphi}_i^T$
9: **end for**

where $(\cdot, \cdot)_T$ denotes an inner product on T.

In practice, the orthonormalization procedure can be applied more than once to get rid of numerical round-off errors. Repeating the procedure twice is generally sufficient to obtain a basis which is orthonormal up to machine precision. This result is in accordance with what reported in [28]. An important remark is that once the coefficients r_{ii}^T and r_{ij}^T, have been calculated they can be employed to evaluate the basis functions and their derivatives at any point of element $T \quad \mathcal{T}_h$. Indeed, it is a simple matter to check that the spatial derivatives $\partial_k \varphi_i^T$, $k \quad 1, 3$, can be evaluated as follows: For all $T \quad \mathcal{T}_h$,

1: **for** $i = 1$ **to** N_{dof}^T **do**
2: **for** $j = 1$ **to** $i - 1$ **do**
3: $\partial_k \widehat{\varphi}_i^T \quad \partial_k \widehat{\varphi}_i^T - r_{ij}^T \partial_k \varphi_j^T$
4: **end for**
5: $\partial_k \widehat{\varphi}_i^T \quad \partial_k \widehat{\varphi}_i^T / r_{ii}^T$
6: $\partial_k \varphi_i^T \quad \partial_k \widehat{\varphi}_i^T$
7: **end for**

As already mentioned, besides being orthonormal, the bases obtained by

means of this procedure are also hierarchical, *i.e.*,

$$\Phi^{k-1} \subset \Phi^k, \qquad k \geq 1. \tag{2.7}$$

This feature is very relevant for methods where the polynomial degree can be locally varied according to a suitable *a-posteriori* estimator, *i.e.*, *k*-adaptivity. In order to appreciate the geometrical flexibility of basis functions defined in the physical space, the components of Φ^4_T are present in Figure 2.2 for an L-shaped polygonal element $T = (-1,1)^2 \quad (-0.2,1) \times (0,1)$.

For completeness, let us remark that other approaches to deal with non-standard meshes exist and deserve to be mentioned. A standard (continuous) Galerkin discretization on arbitrary polyhedral elements has been proposed by Rashid and Selimotic [49] in the context of elastostatics and elastic-plastic problems of solid mechanics. In [49], the definition of the Lagrangian basis functions in the physical frame is strictly related to the number of nodes defining the geometry of the element. The approach has been extended to DG methods by Gassner, Lörcher, Munz and Hesthaven [26] in order to obtain a quadrature free strategy.

2.2.2 Arbitrary mesh generation

Up to this point no assumptions have been made on how the computational grids are obtained. In this thesis the possibility to obtain general meshes starting from standard grids and relying on agglomeration algorithms originally developed in the framework of geometric multigrid has been explored. In particular we rely on the widely used multilevel agglomeration library MGridGen [42] developed by Moulitsas and Karypis. The coarse grid creation is divided in two conceptually distinct phases: first, a sequence of coarse graphs is created; second, starting from the coarse graphs, the refinement process takes place in order to optimize the quality of the final mesh. The library provides the possibility to drive the cardinality of the agglomerated grid through a user-defined parameter that specifies an upper bound for the number of sub-elements inside an aggregated element, in this work this parameter will be referred to as *agglomeration factor*, $\mathcal{A}_c$.

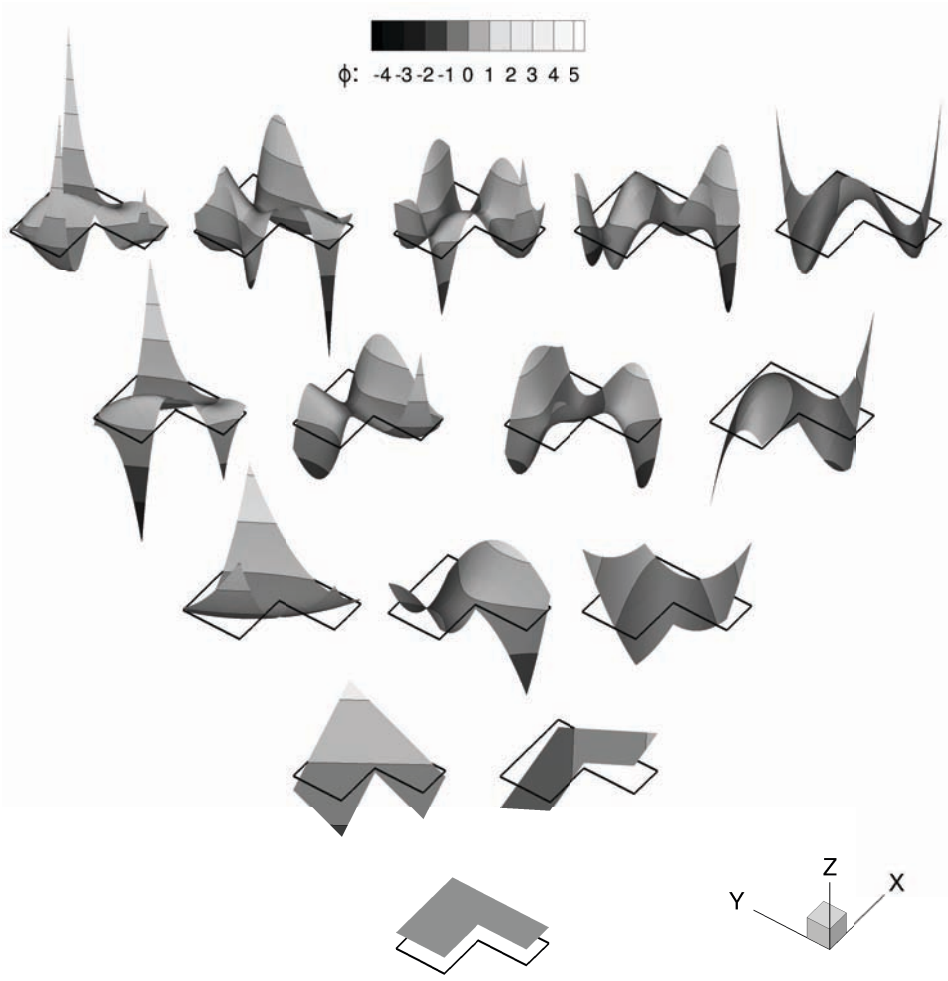
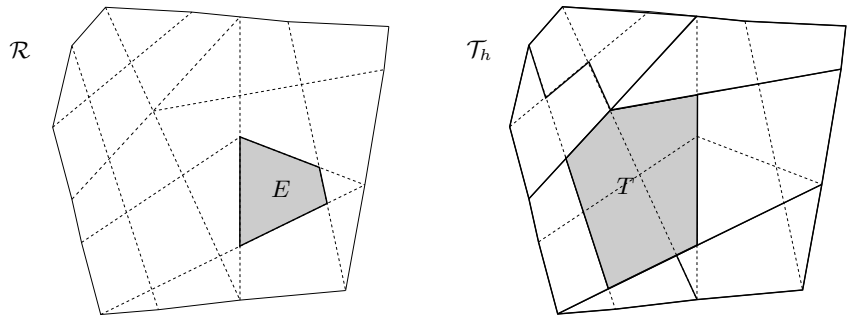


Figure 2.2: Elements of the orthonormal basis Φ_T^4 on a L-shaped polygonal element, $T = (-1,1)^2 \quad (-0.2,1) \times (0,1)$.



(a) Fine grid $\mathcal{R}$ composed of standard elements

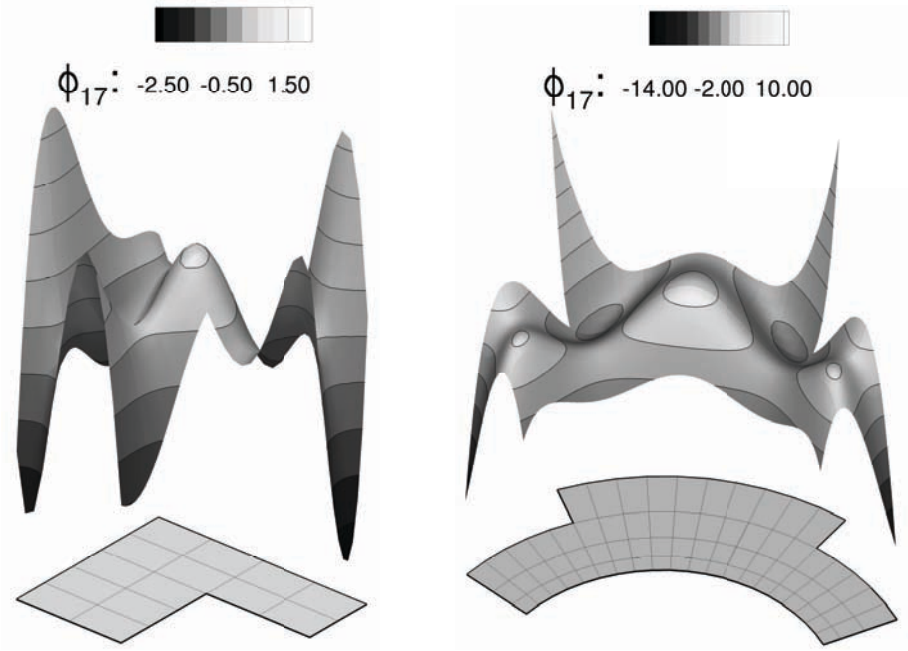
(b) Agglomerated grid $\mathcal{T}_h$ built on top of $\mathcal{R}$

Figure 2.3: *Left and right,* example of a standard element $E \quad \mathcal{R}$ and an agglomerated element $T \quad \mathcal{T}_h$, such that card$(\mathcal{R}_T)= 4$.

Several optimization strategies are provided to the user [42]. In all the numerical tests the default settings of the library have been used, that is, the coarse graphs are obtained by means of a globular agglomeration algorithm, while the uncoarsening/refinement phase minimizes a dual objective function. As described in [43], the weighted sum of aspect ratios and the maximum aspect ratio are considered in order to optimize the quality and the cardinality of the resulting mesh.

In this context some notation has to be introduced. Let us denote by $\mathcal{R}$ a fine mesh of the domain $\Omega \subset \mathbb{R}^d$, $d = \ 2, 3\ $, and by $\mathcal{T}_h$ a coarsening of $\mathcal{R}$ obtained by agglomeration, see Figure 2.3. Most often, the fine mesh $\mathcal{R}$ can be generated via standard meshing tools and is possibly composed of curved elements. Moreover, grid $\mathcal{R}$ is composed of elements E such that for any $E \quad \mathcal{R}$ there exists a reference element E^{ref} and a polynomial mapping $\Psi_E : E^{\text{ref}} \quad E$ such that $\Psi \quad \mathbb{P}_d^m(E^{\text{ref}})$ for some $m \geq 1$ and $E = \Psi_E(E^{\text{ref}})$. Let us assume that quadrature rules of arbitrary order are available on every E^{ref}. Finally, for every agglomerated element $T \quad \mathcal{T}_h$ let us define a subset $\mathcal{R}_T \subset \mathcal{R}$ such that

$$\overline{T} = \bigcup_{E \in \mathcal{R}_T} \overline{E}. \tag{2.8}$$



(a) Affine fine mesh $\mathcal{R}$ (b) Curved fine mesh $\mathcal{R}$

Figure 2.4: Examples of general elements T and of basis functions obtained from the method outlined in §2.2.1. The elements of $\mathcal{R}_T$ are represented in thin line.

Examples of aggregated elements T, obtained on top of a straight and curved side 2D mesh $\mathcal{R}$ are shown in Figure 2.4.

2.2.3 Numerical integration

When targeting to a DG formulation on general grids a technique for evaluating volume and surface integrals on arbitrarily and unpredictably shaped elements is required. Furthermore, as mentioned earlier, even defining basis function in accordance with §2.2.1 entails the ability to perform numerical

integration on general cells.

A straightforward approach is to exploit the possibility to decompose any agglomerated element T as the union of standard elements, $E \quad \mathcal{R}_T$, that can be integrated by means of standard quadrature rules

$$\begin{aligned} \int_T v(\mathbf{x})\ \mathrm{d}\mathbf{x} &= \sum_{E\in\mathcal{R}_T} \int_E v(\mathbf{x})\ \mathrm{d}\mathbf{x} \\ &= \sum_{E\in\mathcal{R}_T,\, E=\Psi_E(E^{\mathrm{ref}})} \int_{E^{\mathrm{ref}}} (v\circ\Psi_E)(\boldsymbol{\xi})\ J_{\Psi_E}(\boldsymbol{\xi})\ \ \mathrm{d}\boldsymbol{\xi}, \end{aligned} \tag{2.9}$$

where $\mathbf{x}$ and $\boldsymbol{\xi}$ are physical and reference space coordinates respectively, and J_{Ψ_E} is the Jacobian of the mapping function Ψ_E, which is here assumed to be a polynomial of degree m.

The polynomial degree q of the integrand in (2.9) results from the product of the polynomial degrees of v and Ψ_E plus the polynomial degree j of J_{Ψ_E} , as follows

$$q = km + j. \tag{2.10}$$

Let us remark that q rapidly increases when considering high-order polynomials on curved element and so does the number of quadrature nodes required to compute the integral exactly. When an aggregated element contains a large number of underlying cells the above approach can turn out to be very expensive even for low-order approximations. To partially mitigate this issue an error-based adaptive criterion to estimate the degree of exactness of quadrature rules has been introduced in [52, 6]. However, this technique can be regarded as just a partial solution for integration issues when dealing with a very large number of sub-elements and for this reason future work will be devoted to investigate strategies capable to overcome this drawback. In this regard, new attractive approaches for the construction of efficient quadratures on arbitrary polygons proposed by Mousavi, Xiao and Sukumar [44] and Xiao and Gimbutas [55] deserve attention. As in this thesis the approach to an efficient quadrature has been left to a very preliminary stage, no results in terms of CPU time will be reported.

2.3 Conclusions

In this chapter all the building blocks needed to employ a DG discretization on grids that consist of polymorphic elements have been briefly introduced. The main drawback of DG implementations that rely on arbitrarily shaped elements can be ascribed to the cost of quadrature rules. Indeed, when integrals on polymorphic elements are computed as the sum of the integrals on standard sub-elements, the computational effort due to quadrature can easily become very high. This is true especially when the polynomial degree of solution approximation is raised, curved geometries are employed or an aggregated element contains a large number of underlying cells. To this end, a more efficient procedure to derive numerical quadrature on elements of arbitrary shape should lead to large computational savings and hence will be subject of future work.

Chapter 3

DG discretization of the Laplace operator on arbitrarily shaped elements

3.1 Introduction

In this chapter the DG discretization of Laplace operator on grids that consist of arbitrarily shaped elements is treated. Two different implementations of the BRMPS method introduced by Bassi, Rebay, Mariotti, Pedinotti and Savini in [16] are considered and validated by means of an h-convergence test on an exact solution of Poisson problem. The discretization of the Laplacian has to be considered as a crucial building block when targeting to more complex problems such as the compressible Navier-Stokes equations.

3.2 Generalization of the BRMPS method to arbitrarily shaped elements

For the sake of simplicity and to ease of presentation the following model problem is considered

$$\begin{cases} -\quad u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{3.1}$$

For an integer $k \geq 1$, we look for a discrete solution in the discrete space

$$V_h = \mathbb{P}_d^k(\mathcal{T}_h).$$

Two different implementations of the BRMPS method are here considered, corresponding to different definitions of faces for agglomerated meshes. In both cases the resulting method is coercive, but the amount of stabilization varies as shown by Bassi, Botti, Colombo, Di Pietro and Tesini in [6].

On the face definitions for aggregated elements

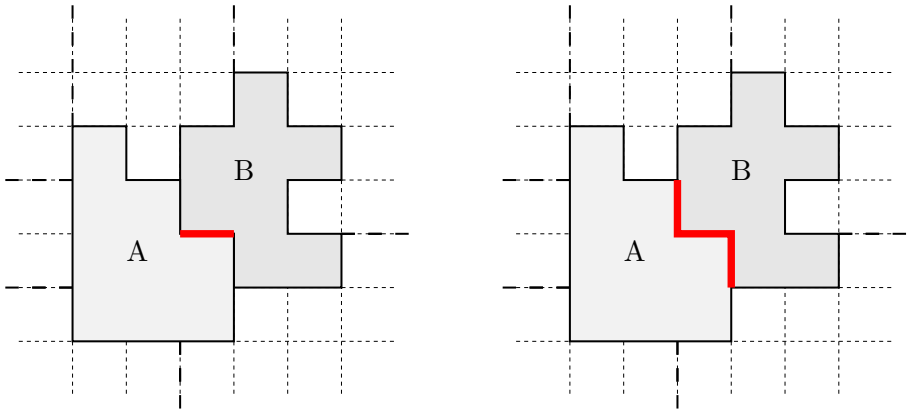
When dealing with agglomerated elements two possible definition of face can be introduced, see [6].

Definition 1 (Facet)**.** *Let define* facet σ *of an agglomerate element* $T \quad \mathcal{T}_h$ *a portion of* ∂T *such that there exist* $E \quad \mathcal{R}_T$ *and a (hyperplanar) face* σ^{ref} *of the corresponding reference element* E^{ref} *such that* σ *is the image of* σ^{ref} *through the mapping* Ψ_E*, see Figure 3.1(a).*
Facets are collected in the set $\mathcal{F}_h^\sigma$*.*

Definition 2 (Mesh face)**.** *Let define a* mesh face Σ *of an agglomerate element* $T \quad \mathcal{T}_h$ *as a portion of* ∂T *such that either* $\Sigma = \partial T \setminus \partial\Omega$ *or there exists* $T' \quad \mathcal{T}_h$*,* $T' = T$*, such that* $\Sigma = \partial T \setminus \partial T'$*, see Figure 3.1(b).*
Mesh faces are collected in the set $\mathcal{F}_h^\Sigma$*.*

For every face Σ the set $\sigma_h^\Sigma \subset \mathcal{F}_h^\sigma$ collecting the facets partitioning Σ can be defined, *i.e.*,

$$\overline{\Sigma} = \bigcup_{\sigma \in \sigma_h^\Sigma} \overline{\sigma}.$$



(a) Facet, σ (b) Mesh face, Σ

Figure 3.1: Facet and mesh face definition for a generic agglomerated grid. Sub-elements sides are depicted by dashed line.

Example 1 (Mesh faces and facets). Consider Figure 3.2, the number of mesh faces of the elements A and B is 6 and 4 while the number of facets is 14 and 16, respectively. It is clear that for the same cell the number of mesh faces Σ and facets σ can be very different.

Hereinafter, the two different implementations of the BRMPS method employing facets or mesh faces will be presented. For the sake of simplicity let us denote by F a generic element of $\mathcal{F}_h$, where $\mathcal{F}_h$ it can be either $\mathcal{F}_h^\sigma$ or $\mathcal{F}_h^\Sigma$. The set $\mathcal{F}_h$ can be partitioned into $\mathcal{F}_h \stackrel{\text{def}}{=} \mathcal{F}_h^{\text{i}} \quad \mathcal{F}_h^{\text{b}}$ where $\mathcal{F}_h^{\text{b}}$ collects the faces located on the boundary of Ω_h and for any $F \quad \mathcal{F}_h^{\text{i}}$ there exist two elements $T^+, T^- \quad \mathcal{T}_h$ such that $F \subseteq \partial T^+ \setminus \partial T^-$. For all $F \quad \mathcal{F}_h^{\text{b}}$, $\mathbf{n}_F$ denotes the unit outward normal to Ω_h, whereas, for all $F \quad \mathcal{F}_h^{\text{i}}$, $\mathbf{n}_F^-$ and $\mathbf{n}_F^+$ are defined as the unit vectors normal to F pointing exterior to T^- and T^+, respectively (the order of the elements sharing F is arbitrary but fixed), see Figure 3.3.

Having no global continuity requirement for V_h on Ω_h, functions are double-valued on $\mathcal{F}_h^{\text{i}}$ and single valued on $\mathcal{F}_h^{\text{b}}$. Therefore, following Brezzi,

	Facets, σ	Mesh faces, Σ
A	14	6
B	16	4

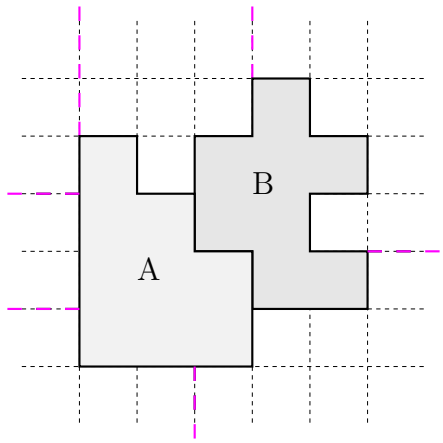


Figure 3.2: Example of agglomerated polygonal mesh with facets and mesh faces defined according to Definition (1) and (2) respectively.

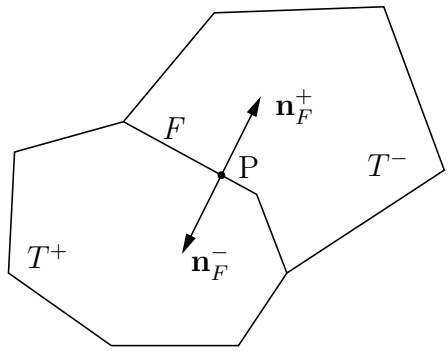


Figure 3.3: Normal at point P on face F.

Manzini, Marini, Pietra and Russo [17], it is useful to introduce the following trace operators, *i.e.*, *average* and *jump*

$$v_h \overset{\text{def}}{=} \frac{v_{h|T^+} + v_{h|T^-}}{2}, \qquad [\![v_h]\!] \overset{\text{def}}{=} v_{h|T^+}\mathbf{n}_F^+ + v_{h|T^-}\mathbf{n}_F^-. \tag{3.2}$$

When trace operators are applied to vector functions, the jump and the average act componentwise. These definitions can be suitably extended to faces F $\mathcal{F}_h^{\mathrm{b}}$ accounting for the weak imposition of boundary conditions.

The lifting operators, which are functions that "extend" to the elements the effects of functions defined at interfaces, are introduced below. For all F $\mathcal{F}_h$, the local lifting operator $\mathbf{r}_F : \left[L^2(F)\right]^d$ $[\mathbb{P}_d^k(\mathcal{T}_h)]^d$ is defined so that, for all $\mathbf{v}$ $\left[L^2(F)\right]^d$,

$$\int_\Omega \boldsymbol{\tau}_h \cdot \mathbf{r}_F(\mathbf{v}) \, \mathrm{d}\mathbf{x} = -\int_F \boldsymbol{\tau}_h \cdot \mathbf{v} \, \mathrm{d}\sigma, \qquad \boldsymbol{\tau}_h \quad [\mathbb{P}_d^k(\mathcal{T}_h)]^d, \tag{3.3}$$

and the global lifting operator $\mathbf{R}$ is related to $\mathbf{r}_F$ by the equation

$$\mathbf{R}(\mathbf{v}) \overset{\text{def}}{=} \sum_{F\in\mathcal{F}_h} \mathbf{r}_F(\mathbf{v}). \tag{3.4}$$

The DG discretization of the Laplace operator by means of the BRMPS method can now be written as the following bilinear form

$$\begin{aligned} a_h(u_h, v_h) \overset{\text{def}}{=} & \int_\Omega [\ {}_h u_h + \mathbf{R}([\![u_h]\!])] \cdot [\ {}_h v_h + \mathbf{R}([\![v_h]\!])] \, \mathrm{d}\mathbf{x} \\ & - \int_\Omega \mathbf{R}([\![u_h]\!]) \cdot \mathbf{R}([\![v_h]\!]) \, \mathrm{d}\mathbf{x} + s_h(u_h, v_h), \end{aligned} \tag{3.5}$$

where s_h is stabilization term

$$s_h(u_h, v_h) \overset{\text{def}}{=} \sum_{F\in\mathcal{F}_h} \eta_F \int_\Omega \mathbf{r}_F([\![u_h]\!]) \cdot \mathbf{r}_F([\![v_h]\!]) \, \mathrm{d}\mathbf{x}. \tag{3.6}$$

Finally, the discrete solution of the Poisson problem in Eq. (3.1) can be written as

$$\text{find } u_h \quad V_h \quad \text{s.t.} \quad a_h(u_h, v_h) = \int_\Omega f v_h \quad \text{for all} \quad v_h \quad V_h. \tag{3.7}$$

Let us now consider the differences that arise in bilinear form (3.5) when facets or mesh faces are employed. The first two terms in Eq. (3.5) are independent of the face definition, since the global lifting $\mathbf{R}$ does not change

$$\mathbf{R}(\llbracket v_h \rrbracket) = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \mathbf{r}_\Sigma(\llbracket v_h \rrbracket) = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \sum_{\sigma \in \sigma_h^\Sigma} \mathbf{r}_\sigma(\llbracket v_h \rrbracket) = \sum_{\sigma \in \mathcal{F}_h^\sigma} \mathbf{r}_\sigma(\llbracket v_h \rrbracket). \quad (3.8)$$

On the contrary, dealing with facets rather than mesh faces leads to different stabilization terms s_h that require two distinct values of the stabilization parameter η_F. It is important to accurately tune the stabilization parameter since over-penalization may result in ill-conditioned linear systems, spoiling the efficiency and the accuracy of the numerical solution as shown by Bassi, Crivellini, Di Pietro and Rebay in [10].

3.2.1 Extended definition of BRMPS stabilization parameter

In this section, we provide values of the stabilization parameter η_F, suitable for the two different implementations described above, which generalize the result of Arnold, Brezzi, Cockburn and Marini [2]. Some preliminary convenient notation is introduced below. For all $F \quad \mathcal{F}_h$ let us define the set of elements in whose boundary F is contained, *i.e.*,

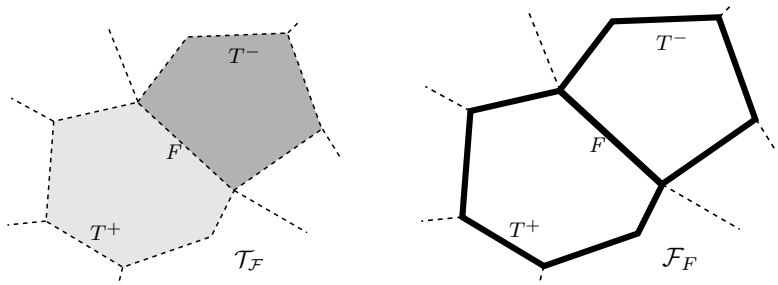
$$\mathcal{T}_F \stackrel{\text{def}}{=} \quad T \quad \mathcal{T}_h \quad F \subset \partial T \quad , \quad (3.9)$$

which contains exactly two elements if $F \quad \mathcal{F}_h^{\text{i}}$ and exactly one if $F \quad \mathcal{F}_h^{\text{b}}$, see Figure 3.4(a). Symmetrically, for all $T \quad \mathcal{T}_h$, the set

$$\mathcal{F}_T \stackrel{\text{def}}{=} \quad F \quad \mathcal{F}_h \quad F \subset \partial T \quad , \quad (3.10)$$

collects the faces composing the boundary of T. Finally, for all $F \quad \mathcal{F}_h$, we define

$$\mathcal{F}_F \stackrel{\text{def}}{=} \bigcup_{T \in \mathcal{T}_F} \mathcal{F}_T, \quad (3.11)$$



(a) (b)

Figure 3.4: 3.4(a): Set of elements $\mathcal{T}_F$ in whose F $\mathcal{F}_h^{\mathrm{i}}$ is contained. 3.4(b): Ensemble $\mathcal{F}_T$ collecting faces of the elements whose boundary contains F.

that is to say, the set of faces of the elements whose boundary contains F, see Figure 3.4(b).

A suitable estimate for the stability parameters comes from the following theorem. For readers interested in more details and proof refer to Bassi, Botti, Colombo, Di Pietro and Tesini [6] and Di Pietro and Ern [21].

Theorem 1 (Coercivity on general meshes)**.** *Assume that*

$$F \quad \mathcal{F}_h, \quad \eta_F > 1 + \frac{1}{2}\,\mathrm{card}(\mathcal{F}_F \quad F \ ). \tag{3.12}$$

Then, there exists $C > 0$ independent of the meshsize such that

$$v_h, \qquad a_h(v_h, v_h) \geq C \ \ v_h \ \ ^2,$$

with a_h defined by (3.5)*, and*

$$v_h \ \ ^2 \stackrel{\mathrm{def}}{=} \quad {}_h v_h \ \ ^2_{[L^2(\Omega)]^d} + \sum_{F \in \mathcal{F}_h} \ \mathbf{r}_F([\![v_h]\!]) \ \ ^2_{[L^2(\Omega)]^d}. \tag{3.13}$$

Put another way, closer to code implementation:
Whether working with mesh faces or facets the coercivity of bilinear form (3.5)

with respect to $\cdot$ *-norm is attained provided, for all* $F \quad \mathcal{F}_h$, η_F *is larger than the average number of mesh faces/facets of the elements sharing* F. Figure 3.5 displays the prescribed values for the lower bound of the stability parameters if polymorphic grids are handled with mesh faces rather than facets.

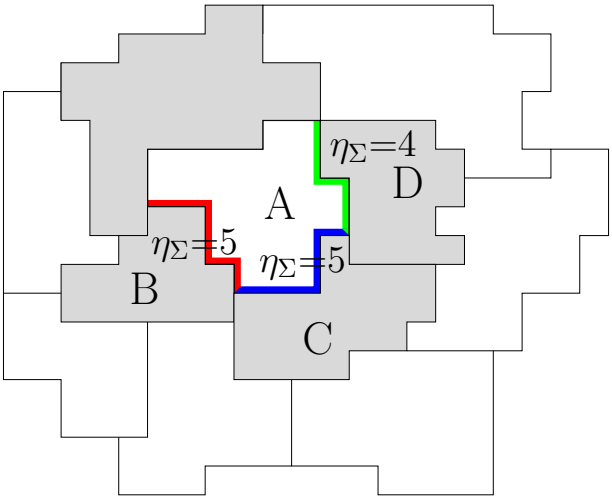
A remarkable difference between the two approaches catches the eye. On the one hand, using facets the stabilization parameter results in a larger value but on the other hand, the integral that appears in the stabilization term has a value smaller than using mesh faces [6]. Applying the definition of lifting operator in Eq. (3.3) the stabilization term for facets reads

$$\begin{aligned} s_h(u_h, v_h) &= \sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\sigma \int_\Omega \mathbf{r}_\sigma(\llbracket u_h \rrbracket) \cdot \mathbf{r}_\sigma(\llbracket v_h \rrbracket) \, \mathrm{d}\mathbf{x} \\ &= -\sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\sigma \int_\sigma \mathbf{r}_\sigma(\llbracket u_h \rrbracket) \cdot \llbracket v_h \rrbracket \, \mathrm{d}\sigma, \end{aligned} \tag{3.14}$$

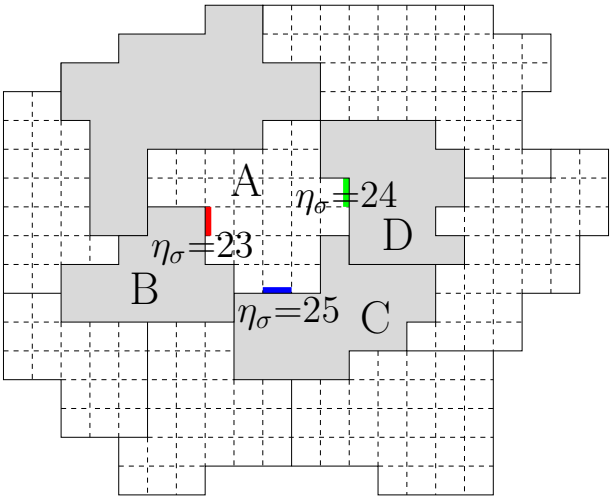
while working with mesh faces

$$\begin{aligned} s_h(u_h, v_h) &= \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Omega \mathbf{r}_\Sigma(\llbracket u_h \rrbracket) \cdot \mathbf{r}_\Sigma(\llbracket v_h \rrbracket) \, \mathrm{d}\mathbf{x} \\ &= -\sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Sigma \mathbf{r}_\Sigma(\llbracket u_h \rrbracket) \cdot \llbracket v_h \rrbracket \, \mathrm{d}\sigma \\ &= -\sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Sigma \left\{ \sum_{\sigma \in \sigma_h^\Sigma} \mathbf{r}_\sigma(\llbracket u_h \rrbracket) \right\} \cdot \llbracket v_h \rrbracket \, \mathrm{d}\sigma \\ &= -\sum_{\Sigma \in \mathcal{F}_h^\Sigma} \sum_{\sigma \in \sigma_h^\Sigma} \eta_\Sigma \int_\Sigma \mathbf{r}_\sigma(\llbracket u_h \rrbracket) \cdot \llbracket v_h \rrbracket \, \mathrm{d}\sigma \\ &= -\sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\Sigma \int_\Sigma \mathbf{r}_\sigma(\llbracket u_h \rrbracket) \cdot \llbracket v_h \rrbracket \, \mathrm{d}\sigma. \end{aligned} \tag{3.15}$$

Let us remark that integrals in the right-hand side of expressions (3.14) and (3.15) have the same integrand but different domain of integration. In



(a) Mesh faces and values of η_Σ



(b) Facets and values of η_σ

Figure 3.5: Examples of stability parameters prescribed if a polymorphic mesh is handled with mesh faces or facets, the neighbours of the element A are shaded.

particular, the integral that appears inside the sum over facets is over a facet σ in the former case and over a mesh face Σ in the latter. As a result, a larger stabilization parameter is justified when facets are employed since it compensates for the smaller contribution to the stabilization term of the integral over facets with respect to that over mesh faces.

3.3 Numerical examples

To assess the accuracy of the presented implementations a comparison for the L^2-orthogonal projection and for the BRMPS DG discretization on grids that consist of arbitrarily shaped elements and uniform grids of quadrilaterals has been carried out. In all the tests here performed orthonormal physical space basis functions have been obtained according to § 2.2.1.

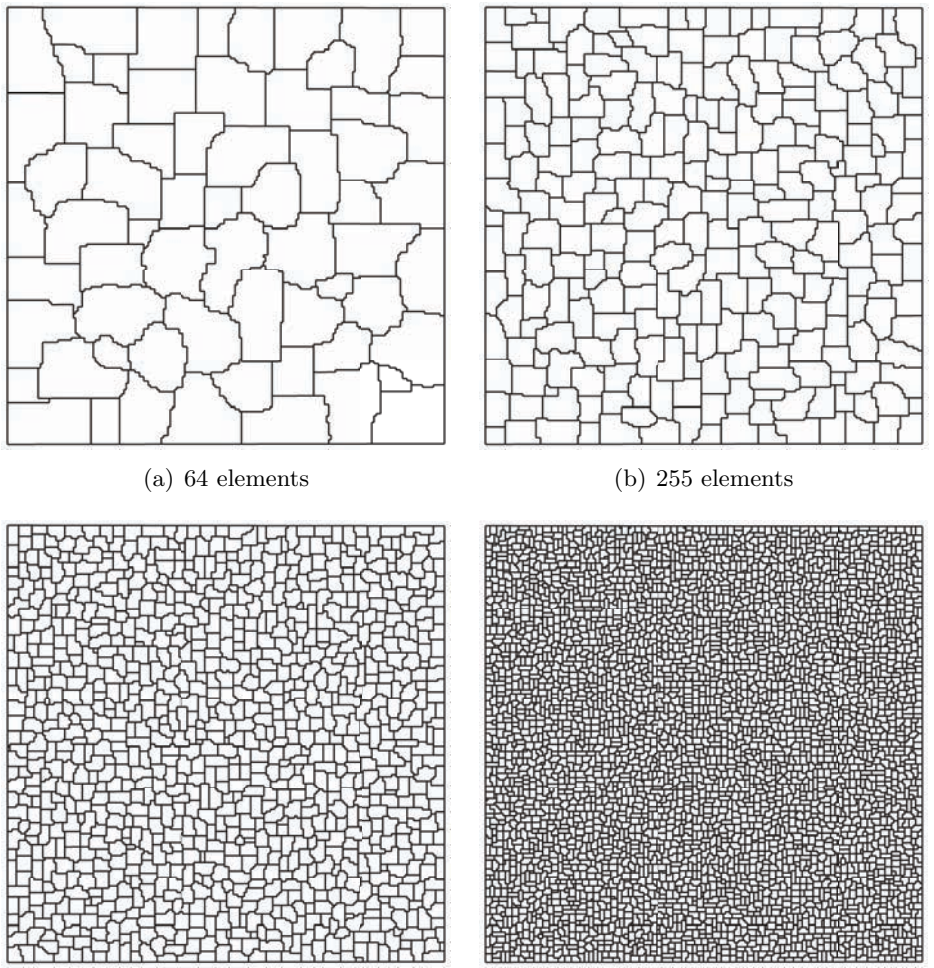
The polygonal grids have been created by means of an agglomeration process performed on a very fine 200×200 uniform quadrilateral grid of a square $[-1, 1]^2$. The library MGridGen [42] has been used to generate a sequence of h-refined grids of 64, 255, 1028, and 4122 polygonal elements, see Figure 3.6. The benchmark results have been obtained on a set of uniform nested grids of 64, 256, 1028 and 4096 quadrilateral elements, respectively. The maximum and average mesh size for aggregated and quadrilateral grids are depicted in Figure 3.8(b).

Validation tests on the BRMPS DG discretization have been performed on a Poisson problem (3.1), proposed by Karniadakis and Sherwin [38], that admits exact solution

$$u = e^{-2.5[(x-1)^2+(y-1)^2]}. \tag{3.16}$$

This analytical solution has been also used to investigate the L^2-orthogonal projection convergence.

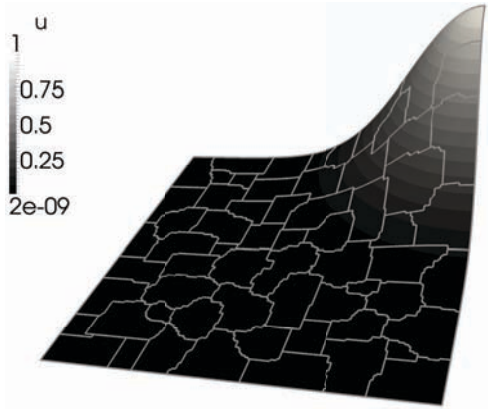
The h-convergence rates of the L^2-projection and of the two implementations of the BRMPS method proposed in § 3.1 have been displayed in Figures 3.8(a), 3.8(c) and 3.8(d). Although the L^2-norm of the error computed on uniform quadrilateral grids is almost invariably smaller than the error computed in the case of polygonal grids, the theoretical convergence



(a) 64 elements (b) 255 elements

(c) 1028 elements (d) 4122 elements

Figure 3.6: Polygonal meshes obtained by agglomeration of a 200×200 uniform quadrilateral grid.



(a) $\mathbb{P}^6$, 64 elements

(b) Error distribution, 4096 elements Cartesian grid

Figure 3.7: BRMPS solution for the test case of Karniadakis and Sherwin [38].

rates are achieved. However, the better approximation properties of uniform grids are well known, and this result was somehow to be expected. Let us remark that numerical results shown in Figure 3.8(d) suggest that an implementation of the BRMPS scheme that relies on mesh faces rather than facets provides similar results and thus both forms of the stabilization term can be used with confidence. Choosing one of the two forms of the stabilization parameter is just a matter of implementation convenience.

3.4 Conclusions

In this chapter the feasibility of an accurate DG solution for diffusion problems on meshes that consist of arbitrarily shaped cells has been demonstrated. Two different implementations for grids of aggregated polymorphic elements have been presented and validated by means of an exact solution of a Poisson problem. The two strategies rely on a different defi-

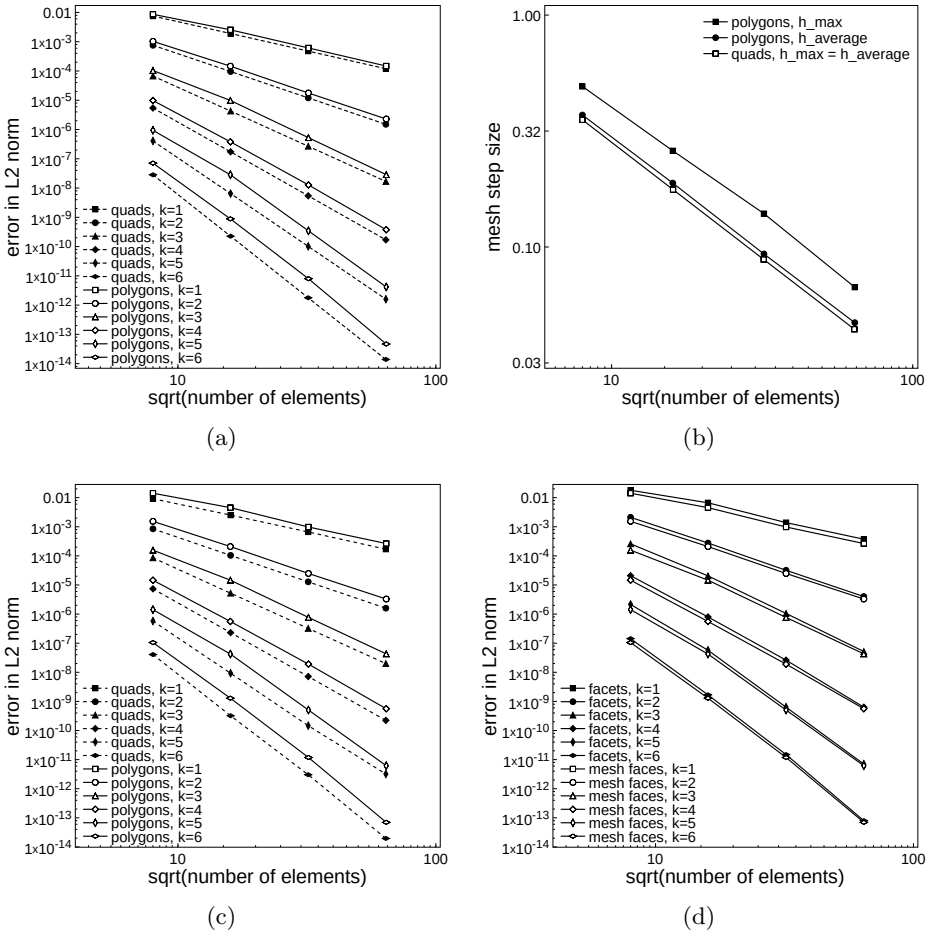


Figure 3.8: *Top row.* 3.8(a): Convergence of the L^2-orthogonal projection on uniform quadrilateral grids (dashed lines) and polygonal grids obtained by agglomeration (solid lines), $\mathbb{P}^{1\to 6}$. 3.8(b): Maximum and average mesh size h on uniform quadrilateral grids and polygonal grids. *Bottom row.* BRMPS solution for the test case of Karniadakis and Sherwin [38]. 3.8(c): Convergence rates on uniform quadrilateral grids (dashes lines) and polygonal grids obtained by agglomeration (solid lines). 3.8(d) Convergence rates on polygonal grids obtained by agglomeration when mesh faces or, alternatively, facets are used for stabilization.

nition of element face, but both achieve the theoretical convergence rates. Therefore, the choice between the two implementations may be dictated by programming needs or preferences. This results have to be considered as a main ingredient when targeting to a robust and accurate solution of more complex problems, such as the equations of compressible fluid dynamics, on general meshes.

Chapter 4

Application of DG discretization on polymorphic meshes to inviscid and viscous flows

4.1 Introduction

This chapter deals with the accurate DG solution of the compressible Navier-Stokes and RANS+k-$\widetilde{\omega}$ equations on meshes that consist of arbitrarily shaped elements. Implicit time integration is applied to the Navier–Stokes as well as the fully coupled RANS+k-$\widetilde{\omega}$ equations. The performance and the robustness of the proposed method is illustrated by computing some classical inviscid and viscous validation test cases.

4.2 DG formulation of Navier–Stokes equations

4.2.1 Governing equations

The Navier–Stokes equations are a system of partial differential equations that describe the motion of fluids. Let us consider the two-dimensional

Navier-Stokes equations written in compact conservation form

$$\frac{\partial \mathbf{u}}{\partial t} \; + \; \boldsymbol{\nabla} \cdot \mathbf{F}_c(\mathbf{u}) \; + \; \boldsymbol{\nabla} \cdot \mathbf{F}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) = \mathbf{0}, \tag{4.1}$$

equipped with suitable initial-boundary conditions, where $\mathbf{u}$ is the vector of the M conservative variables, $\mathbf{F}_c, \mathbf{F}_v \quad \mathbb{R}^M \otimes \mathbb{R}^2$ are defined as the inviscid and viscous flux functions, respectively, and d is the space dimension.

The conservative variables $\mathbf{u}$ and the Cartesian components $\mathbf{f}_c(\mathbf{u})$ and $\mathbf{g}_c(\mathbf{u})$ of the inviscid $\mathbf{F}_c(\mathbf{u})$ flux function, equal

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho e_0 \\ \rho u \\ \rho v \end{bmatrix}, \quad \mathbf{f}_c(\mathbf{u}) \; = \; \begin{bmatrix} \rho u \\ \rho h_0 u \\ \rho u u + p \\ \rho v u \end{bmatrix}, \quad \mathbf{g}_c(\mathbf{u}) \; = \; \begin{bmatrix} \rho v \\ \rho h_0 v \\ \rho u v \\ \rho v v + p \end{bmatrix}, \tag{4.2}$$

where ρ is the fluid density, p is the pressure, u and v are the velocity components, e_0 and h_0 are the total energy and enthalpy per unit mass, respectively. The total enthalpy per unit mass is defined as $h_0 = e_0 + p/\rho$. By assuming that the fluid obeys to the perfect gas state equation, p can be computed as $p = (\gamma - 1)\,\rho\left[e_0 - \left(u^2 + v^2\right)/2\right]$ where γ indicates the ratio between the specific heats of the fluid.

The Cartesian components $\mathbf{f}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u})$ and $\mathbf{g}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u})$ of the viscous flux function $\mathbf{F}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u})$, are given by

$$\mathbf{f}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) = -\mu \begin{bmatrix} 0 \\ u\left[2\partial_x u + \lambda\left(\partial_x u + \partial_y v\right)\right] + v\left(\partial_x v + \partial_y u\right) + \left(\gamma/\mathrm{Pr}\right)\partial_x e \\ 2\partial_x u + \lambda\left(\partial_x u + \partial_y v\right) \\ \partial_x v + \partial_y u \end{bmatrix}, \tag{4.3}$$

$$\mathbf{g}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) = -\mu \begin{bmatrix} 0 \\ u\left(\partial_x v + \partial_y u\right) + v\left[2\partial_y v + \lambda\left(\partial_x u + \partial_y v\right)\right] + \left(\gamma/\mathrm{Pr}\right)\partial_y e \\ \partial_x v + \partial_y u \\ 2\partial_x v + \lambda\left(\partial_x u + \partial_y v\right) \end{bmatrix}, \tag{4.4}$$

where ∂_x and ∂_y refer to partial derivatives with respect to variables x and y, μ is the dynamic viscosity coefficient, Pr is the Prandtl number, and, using the Stokes hypothesis, $\lambda = -\frac{2}{3}$.

4.2.2 DG discretization of the compressible Navier–Stokes equations

A weak formulation of the Navier–Stokes equations is obtained multiplying each scalar conservation law in (4.1) by an arbitrary smooth test function $v_j \quad \mathbf{v}$, $1 \leq j \leq M$, and integrating by parts, that is

$$\int_\Omega v_j \frac{\partial u_j}{\partial t} \, \mathrm{d}\mathbf{x} - \int_\Omega \boldsymbol{\nabla} v_j \cdot \mathbf{F}_j(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} v_j \mathbf{F}_j(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}\sigma = 0, \tag{4.5}$$

where $\mathbf{F}$ is the sum of the inviscid and viscous fluxes.

To discretize (4.5) we replace the analytical solution $\mathbf{u}$ and the test function $\mathbf{v}$ with a finite element approximation $\mathbf{u}_h$ and a discrete test function $\mathbf{v}_h$ respectively, where $\mathbf{u}_h$ and $\mathbf{v}_h$ belong to the space $V_h \stackrel{\text{def}}{=} [\mathbb{P}_2^k(\mathcal{T}_h)]^M$, and

$$\mathbb{P}_2^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{ v_h \quad L^2(\Omega) \quad v_{h|T} \quad \mathbb{P}_2^k(T), \quad T \quad \mathcal{T}_h \right\}. \tag{4.6}$$

The DG discretization is defined on a triangulation $\mathcal{T}_h = \quad T \quad$ of an approximation Ω_h of Ω consisting of a set of non-overlapping arbitrarily shaped elements T. Let denote by F a generic element of $\mathcal{F}_h$, where $\mathcal{F}_h$ it can be either $\mathcal{F}_h^\sigma$ or $\mathcal{F}_h^\Sigma$ according to Definition 1 or Definition 2, cfr. §3.2. The set $\mathcal{F}_h$ can be partitioned into $\mathcal{F}_h \stackrel{\text{def}}{=} \mathcal{F}_h^{\mathrm{i}} \quad \mathcal{F}_h^{\mathrm{b}}$ where $\mathcal{F}_h^{\mathrm{b}}$ collect the faces located on the boundary of Ω_h and for any $F \quad \mathcal{F}_h^{\mathrm{i}}$ there exist two elements $T^+, T^- \quad \mathcal{T}_h$ such that $F \subseteq \partial T^+ \setminus \partial T^-$. For all $F \quad \mathcal{F}_h^{\mathrm{b}}$, $\mathbf{n}_F$ denotes the unit outward normal to Ω_h, whereas, for all $F \quad \mathcal{F}_h^{\mathrm{i}}$, $\mathbf{n}_F^-$ and $\mathbf{n}_F^+$ are defined as the unit vectors normal to F pointing exterior to T^- and T^+, respectively (the order of the elements sharing F is arbitrary but fixed).

By splitting integrals over the entire domain that appear in Eq. (4.5),

into sums of integrals over elements, $T \quad \mathcal{T}_h$

$$\sum_{T\in\mathcal{T}_h}\int_T v_{h,j}\frac{\partial u_{h,j}}{\partial t}\,\mathrm{d}\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \boldsymbol{\nabla}_h v_{h,j}\cdot\mathbf{F}_j(\mathbf{u}_h,\boldsymbol{\nabla}_h\mathbf{u}_h)\,\mathrm{d}\mathbf{x} + \sum_{T\in\mathcal{T}_h}\int_{\partial T} v_{h,j}\mathbf{F}_j(\mathbf{u}_h,\boldsymbol{\nabla}_h\mathbf{u}_h)\cdot\mathbf{n}\,\mathrm{d}\sigma = 0 \qquad \mathbf{v}_h \quad V_h. \tag{4.7}$$

The discontinuous approximation of the numerical solution requires introducing a special treatment of the inviscid interface flux and of the viscous flux. For the former it is common practice to substitute the flux $\mathbf{F}_c(\mathbf{u}_{h\ T})$ with a suitably defined numerical flux function $\widehat{\mathbf{F}}_c(\mathbf{u}_h^{\pm})$ which ensure conservation and account for wave propagation. In this work it is usually employed the Godunov flux computed with exact Riemann solver or, alternatively, the van Leer-Hänel [30] flux-splitting scheme. It is the *upwind* nature of these fluxes that guarantees the stability of the convective part of equations.

Concerning the latter many techniques are available for the DG space discretizations of diffusive terms and a detailed review can be found in [2]. In this work the BRMPS scheme, devised by Bassi, Rebay, Mariotti, Pedinotti and Savini in [16] and theoretically analyzed by Brezzi, Manzini, Marini, Pietra and Russo in [17], is employed to obtain a consistent, stable and accurate discretization of the viscous flux. An implementation based on the so-called facets σ is here adopted according with §3.1.

The viscous numerical flux can hence be defined in a "centered" way as

$$\widehat{\mathbf{F}}_v\left(\mathbf{u}_h^{\pm},\mathbf{z}_{h\ \sigma}^{\ \pm}\right) \stackrel{\text{def}}{=} \mathbf{F}_v\left(\mathbf{u}_h,\mathbf{z}_{h\ \sigma}\right) \quad , \tag{4.8}$$

where discrete gradients are given by

$$\mathbf{z}_h = \boldsymbol{\nabla}_h\mathbf{u}_h + \mathbf{R}(\llbracket\mathbf{u}_h\rrbracket), \qquad \mathbf{z}_{h\ \sigma} = \boldsymbol{\nabla}_h\mathbf{u}_h + \eta_\sigma\mathbf{r}_\sigma(\llbracket\mathbf{u}_h\rrbracket), \tag{4.9}$$

and $\mathbf{R}$ and $\mathbf{r}_\sigma$ are the global and local lifting operators defined in Eq. (3.3). The trace operators, *jump* $\llbracket\cdot\rrbracket$ and *average* $\cdot$ are those defined in Eq. (3.2).

In order to ensure the stability of the diffusive part, the penalty parameter η_σ is set larger than the average number of facets of the elements sharing σ, see §3.2.1.

Accounting for these aspects, the DG formulation for the compressible Navier-Stokes equations consists in seeking $\mathbf{u}_h \quad V_h$ such that

$$\sum_{T\in\mathcal{T}_h}\int_T v_{h,j}\frac{\partial u_{h,j}}{\partial t}\,\mathrm{d}\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \boldsymbol{\nabla}_h v_{h,j}\cdot\mathbf{F}_j\left(\mathbf{u}_h,\mathbf{z}_h\right)\,\mathrm{d}\mathbf{x} + \sum_{\sigma\in\mathcal{F}_h^\sigma}\int_\sigma [\![v_{h,j}]\!]\cdot\widehat{\mathbf{F}}_j\left(\mathbf{u}_h^{\pm},\mathbf{z}_{h\ \sigma}^{\ \pm}\right)\,\mathrm{d}\sigma = 0 \qquad \mathbf{v}_h \quad V_h. \tag{4.10}$$

The suitable boundary conditions are weakly imposed by properly defining boundary values for the variables which, together with their values inside the domain at the boundary interface, allow to compute the numerical fluxes and the lifting operators on $\mathcal{F}_h^{\mathrm{b}}$, see [16].

The actual implementation of Eq. (4.10) requires specifying the test and trial functions within each element $T \quad \mathcal{T}_h$, this work relies on a set of hierarchical and orthonormal basis functions obtained by means of a modified Grahm-Schmidt (MGS) algorithm and using monomials as starting set, according to the procedure reported in §2.2.1.

Since in this work the grids are obtained by means of an agglomeration process as referred in §2.2.2, the volume and surface integrals appearing in Eq. (4.10) can be computed as sum of integrals on the underlying sub-elements. This approach can turn out to be expensive when high-order solution approximations and/or curved elements are used. To mitigate this problem, an error-based adaptive criterion has been employed to estimate the degree of exactness of quadrature rules on sub-elements, as proposed in [52, 6].

The "ingredients" described above provide a discretization as compact as possible because, for each element T, it only couples the nearest neighbour elements. This feature is very attractive for the implicit implementation of the method.

4.3 Time integration

Within each element $T \quad \mathcal{T}_h$, the solution $\mathbf{u}_h$ can be expressed as

$$\mathbf{u}_h^T = \sum_{j=1}^{N_{\text{dof}}^T} \mathbf{U}_j^T \varphi_j^T, \tag{4.11}$$

where

$$\varphi_i^T \quad _{i\in\{1,\dots,N_{\text{dof}}^T\}}, \tag{4.12}$$

is a basis for the local polynomial space $\mathbb{P}_2^k(T)$, such that $N_{\text{dof}}^T = \dim\left[\mathbb{P}_2^k(T)\right]$ and the expansion coefficients $\mathbf{U}^T$ are the degrees of freedom of the numerical solution in T.

By assembling together all the elemental contributions the system of ordinary differential equations governing the evolution in time of the discrete solution can be written as

$$\mathbf{M}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \mathbf{R}\left(\mathbf{U}\right) = \mathbf{0}, \tag{4.13}$$

where $\mathbf{U}$ is the global vector of unknown degrees of freedom, $\mathbf{M}$ is a global block diagonal matrix and $\mathbf{R}\left(\mathbf{U}\right)$ is the vector of "residuals", *i.e.*, the vector of nonlinear functions of $\mathbf{U}$ resulting from the integrals of the DG discretized space differential operators in Eq. (4.10). The matrix $\mathbf{M}$ represents the global block diagonal mass matrix, which, using orthonormal basis functions, reduces to the identity matrix.

Backward Euler scheme

For the time discretization of the semi-discrete problem in Eq. (4.13) the classical backward Euler scheme is employed in case of steady computations. This is obtained by linearizing Eq. (4.13) at time level n

$$\underbrace{\left[\frac{\mathbf{M}}{\Delta t} + \frac{\partial \mathbf{R}\left(\mathbf{U}^n\right)}{\partial \mathbf{U}}\right]}_{\mathbf{A}} \left(\mathbf{U}^{n+1} - \mathbf{U}^n\right) = -\mathbf{R}\left(\mathbf{U}^n\right), \tag{4.14}$$

where $\frac{\partial \mathbf{R}(\mathbf{U}^n)}{\partial \mathbf{U}}$ is the Jacobian matrix of the DG space discretization that has been derived analytically taking full account of the dependence of the residual on the unknown vector and on its derivatives, including the implicit treatment of the lifting operators. Let us denote with $\mathbf{A}$ the linear system matrix, we remark that the compact DG discretization here employed minimizes the number of non-zero blocks for each (block) row of $\mathbf{A}$. To solve Eq. (4.14) at each time step the matrix-explicit or the matrix-free GMRES algorithm can be used, for this purpose linear algebra and parallelization are handled through PETSc library [4]. In both cases system preconditioning is required to make the convergence of the GMRES solver acceptable in problems of practical interest.

For steady solutions the choice of the time step can significantly affect the efficiency and the robustness of the method. The pseudo-transient continuation strategy proposed in [5] in the context of turbulent problems is here employed. The local time step is hence given by

$$\Delta t_T = CFL \frac{h_T}{v_c + v_d}, \tag{4.15}$$

where

$$v_c = \overline{\mathbf{v}} + a, \qquad v_d = 2\frac{\mu_e + \lambda_e}{h_T}, \qquad h_K = d\frac{\Omega_T}{S_T}, \tag{4.16}$$

define convective and diffusive velocities and the reference dimension of the generic element T, respectively. The coefficients μ_e and λ_e are the effective dynamic viscosity and conductivity, $\overline{\mathbf{v}}$ is the velocity vector, while Ω_T and S_T denote the volume and the surface of T. All quantities depending on $\mathbf{u}_h$ in the above relations are computed from mean values of $\mathbf{u}_h$.

The employed CFL number evolution law depends upon both the L^∞ and the L^2 norms of the residual. Denoting with y the CFL number, the rule can be summarized as follows

$$\begin{cases} y = \frac{y_0}{x^\alpha} & \text{if } x \leq 1, \\ y = y_e + (y_0 - y_e) e^{\alpha \frac{y_0}{y_0 - y_e}(1-x)} & \text{if } x > 1, \end{cases} \tag{4.17}$$

denoting by $x_{L^2} = \max(\, R_{i\ L^2} / \, R_{i0\ L^2})$ and $x_{L^\infty} = \max(\, R_{i\ L^\infty} / \, R_{i0\ L^\infty})$

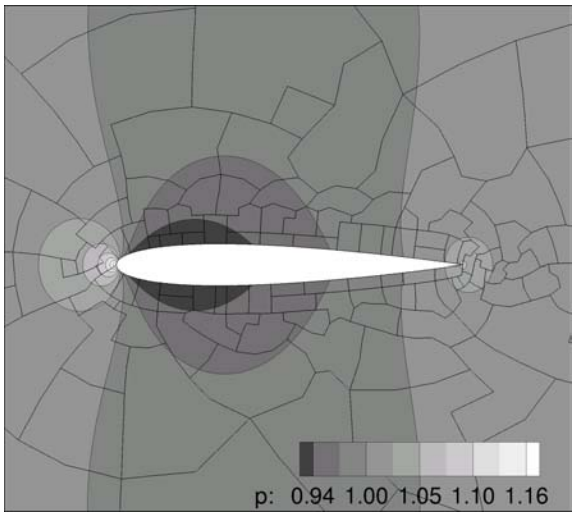
for $i = 1, \ldots, M$, we define

$$\begin{cases} x = \min\left(x_{L^2}, 1\right) & \text{if } x_{L^\infty} \leq 1, \\ x = x_{L^\infty} & \text{if } x_{L^\infty} > 1, \end{cases} \tag{4.18}$$

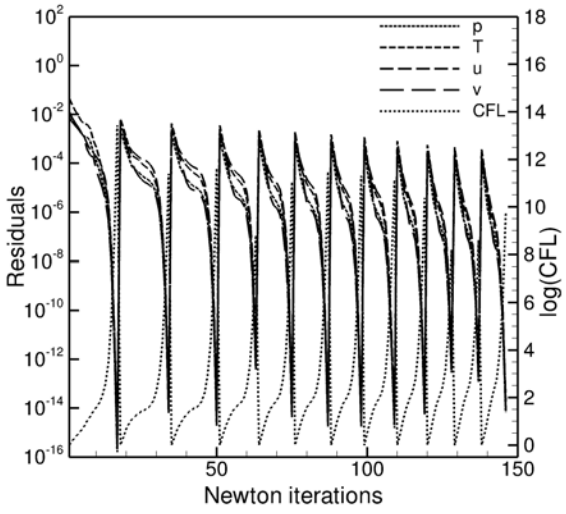
and $y_0 = \mathrm{CFL}_{min}$, $y_e = \mathrm{CFL}_{exp}$ and α are the user-supplied minimum CFL number, the maximum allowable CFL number of explicit schemes and the exponent (usually ≤ 1) governing the growth rate of the CFL number, respectively. For simple steady test cases implicit time integration combined with this CFL number evolution strategy provides quadratic Newton convergence to machine accuracy as displayed in Figure 4.1 for an inviscid flow around the NACA0012 airfoil.

4.4 Shock-capturing approach

When dealing with convection-dominated problems high-order DG methods can display oscillations around flow discontinuities which can lead to spoiled solutions or even to breakdown of computations. Several stabilization techniques have been proposed in the recent past and the number of new approaches is still growing. Part of the stabilization techniques for DG methods can be traced back to well-proven approaches used in the FV and (continuous)-finite elements (FE) contexts. In fact, this is the case for the (generalized) limiting procedure introduced by Cockburn, Shu and coworkers in a series of papers, [20, 19, 18], and for the stabilization technique, based on the modification of the test functions and the addition of an artificial viscosity depending on the local residual, borrowed in the DG method from SD (or SUPG) FE methods and analyzed by Jaffre, Johnson and Szepessy in [37]. The approach of introducing limiting procedures to control the oscillations of DG solutions around discontinuities has been revisited by Qiu and Shu in the framework of WENO (Weighted Essentially Non-Oscillatory), [48], and HWENO (Hermite WENO), [46, 47], schemes. The idea is to first identify "troubled cells", namely those cells where limiting might be needed, then to abandon all moments in those cells



(a)



(b)

Figure 4.1: NACA0012 airfoil, $M_\infty = 0.5$, $\alpha_\infty = 0°$. 4.1(a): Pressure contours of $\mathbb{P}^{12}$ solution. 4.1(b): Residuals convergence history of $\mathbb{P}^{1\to12}$ solutions. Aggregated mesh of 178 elements built on top of a standard hybrid grid of 1197 13-node quadrilateral and 10-node triangular elements.

except the cell averages and reconstruct those moments from the information of neighboring cells using first WENO and then HWENO methodologies. On the side of "artificial viscosity" techniques, new strategies inspired by the early artificial viscosity methods have been presented by Persson and Peraire, [45]. The old idea of adding an explicit viscous term to the governing equations has been revisited in the context of high-order DG discretization. The basic idea of explicit artificial viscosity methods is to spread discontinuities over a length scale that can be resolved within the space of approximating functions, by adding enough viscosity to the original equations. In this thesis an approach for the control of oscillations of DG methods, similar in the spirit to the SUPG shock-capturing technique analyzed in [37] has been employed. As such, the method explicitly adds to the governing equations (4.10) an artificial viscosity term that aims at controlling the high-order modes within elements while preserving the resolution of discontinuities, *i.e.*, it aims at achieving sub-cell resolution of discontinuities. Indeed, a good robustness and a crisp representation of shocks profiles are the two essential features of a reliable shock-capturing approach. The artificial viscosity term here employed is that proposed by Bassi, Botti, Colombo, Crivellini, Franchina, Ghidoni and Rebay in [5]

$$\sum_{T\in\mathcal{T}_h}\int_T \epsilon_p(\mathbf{u}_h^{\pm},\mathbf{u}_h)\,(\boldsymbol{\nabla}_h v_h\cdot\mathbf{b})\,(\boldsymbol{\nabla}_h\mathbf{u}_h\cdot\mathbf{b})\;\mathrm{d}\mathbf{x}. \tag{4.19}$$

First forms of this term were presented already in [12, 13] and a similar technique, with a different choice for the residual-based viscosity coefficient, was also used by Hartmann and Houston in [33]. Unlike other shock-capturing techniques used in DG methods, this approach does not rely on troubled cell indicator to detect elements where artificial viscosity is needed. In fact, the above shock-capturing term is active in every element, but artificial viscosity, ϵ_p depends on the (inviscid) residual of the DG space discretization and thus it is everywhere negligible except than at location of flow discontinuities. The amount of artificial diffusion is proportional to the value of the inviscid flux divergence inside elements and to the value of an element-wise function $\mathbf{s}$, which is actually the lifting of the interface jump

in normal direction between the numerical and internal inviscid flux components. Furthermore, the viscous term is only applied in the direction of the pressure gradient **b**. For a detailed description of the shock-capturing term and its dependencies refer to [5].

4.5 Numerical examples

In this section the solution of the of Euler and Navier–Stokes equations on agglomerated grids obtained by means of MGridGen library [42] is reported. Classical inviscid transonic and viscous test cases are used to demonstrate the accuracy and robustness of DG discretization on very general meshes.

Smooth inviscid flow configuration

The Ringleb flow

The Ringleb problem represents a transonic inviscid flow which turns around a symmetric obstacle. The flowfield is irrotational and isentropic, and the streamlines are symmetric with respect to the axis of symmetry of the obstacle. For this case the analytical solution can be obtained using the "hodograph transformation", see, *e.g.*, [1]

$$x(q,k) = \pm\frac{1}{k\,q\,\rho}\left(1-\frac{q^2}{k^2}\right)^{\frac{1}{2}}, \tag{4.20}$$

$$y(q,k) = \frac{1}{2\rho}\left(\frac{1}{q^2}-\frac{2}{k^2}\right)+\frac{1}{2}J, \tag{4.21}$$

$$J = \frac{1}{c}+\frac{1}{3c^3}+\frac{1}{5c^5}-\frac{1}{2}\ln\left(\frac{1+c}{1-c}\right). \tag{4.22}$$

Let us consider the following computational domain

$$\Omega = \quad (x,y) : k \quad (0.7,1.2) \quad \text{et} \quad q \quad (0.5,k) \quad , \tag{4.23}$$

where k is a stream-function, *i.e.*, it is constant along the streamlines, and can be expressed as

$$k = \frac{q}{\sin\theta}\,, \tag{4.24}$$

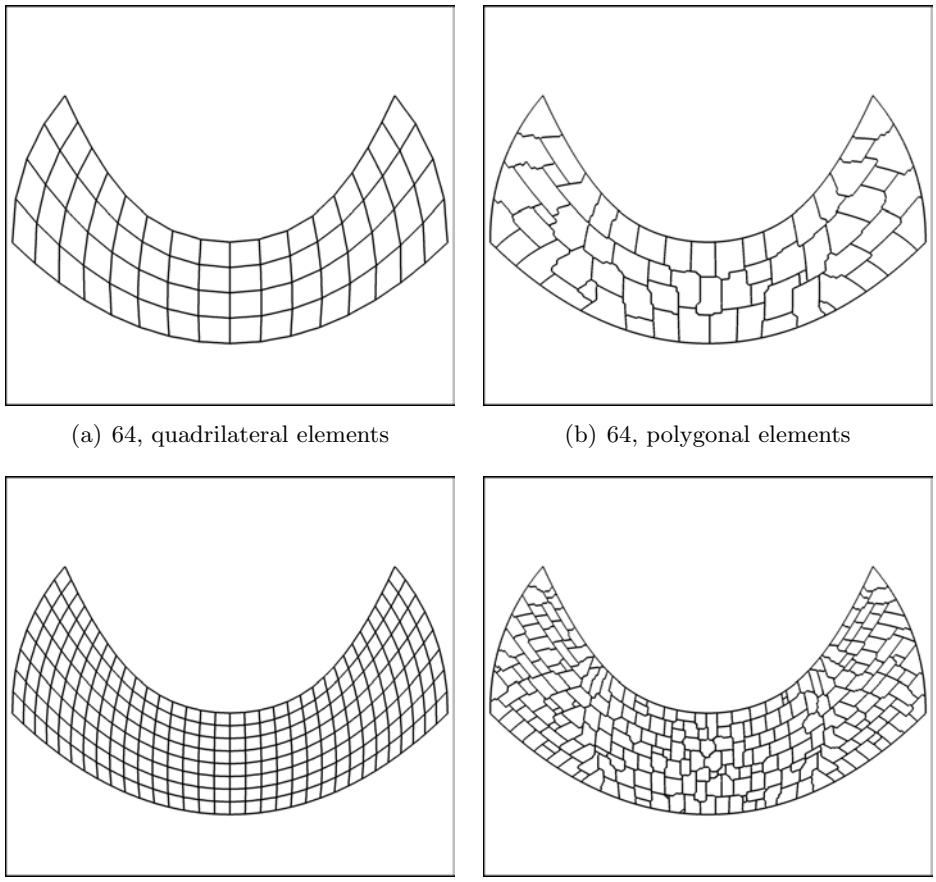
where θ defines the angle between the velocity vector and the vertical direction. Moreover q, c and ρ are dimensionless quantities, corresponding to the ratio between local velocity magnitude and the speed of sound at the total condition, c_0, the local speed of sound divided by c_0 and the ratio between local and total fluid densities, respectively.

To assess the accuracy of the proposed implementation a comparison between meshes that consist of arbitrarily shaped elements and uniform grids of quadrilaterals has been carried out. The polygonal grids have been obtained by means of an agglomeration process on top of a 32768 3-node triangular elements grid. The library MGridGen [42] generated a sequence of h-refined grids of 64, 257, 1019, and 4119 polygonal elements, see Figure 4.2. The benchmark results are obtained on a set of uniform grids of 64, 256, 1028 and 4096 4-node quadrilateral elements, respectively. Since low-order approximations of boundaries reduce the order of convergence, the analytical solution has been set on the whole domain boundary $\partial\Omega$. Although the strong unevenness of polygonal grids seems to affect the behaviour of error in L^2-norm for the y-component of the velocity at low polynomial approximations, in general the theoretical convergence rates can be considered achieved, as displayed in Figure 4.3.

Inviscid transonic flow

NACA0012 $\mathbf{M_\infty = 0.8}$, $\alpha_\infty = \mathbf{1.25^\circ}$

This test case aims at demonstrating the capabilities of the employed shock-capturing approach on polymorphic meshes. The problem involves the inviscid transonic flow around the NACA0012 airfoil at an angle of attack $\alpha_\infty = 1.25^\circ$ and freestream Mach number of $\mathrm{M}_\infty = 0.8$. The flow configuration bears to a steady solution with a strong shock on the suction side and a weak one on the pressure side. This test has been used as benchmark solution between different high-order methods during the FP6 EU ADIGMA



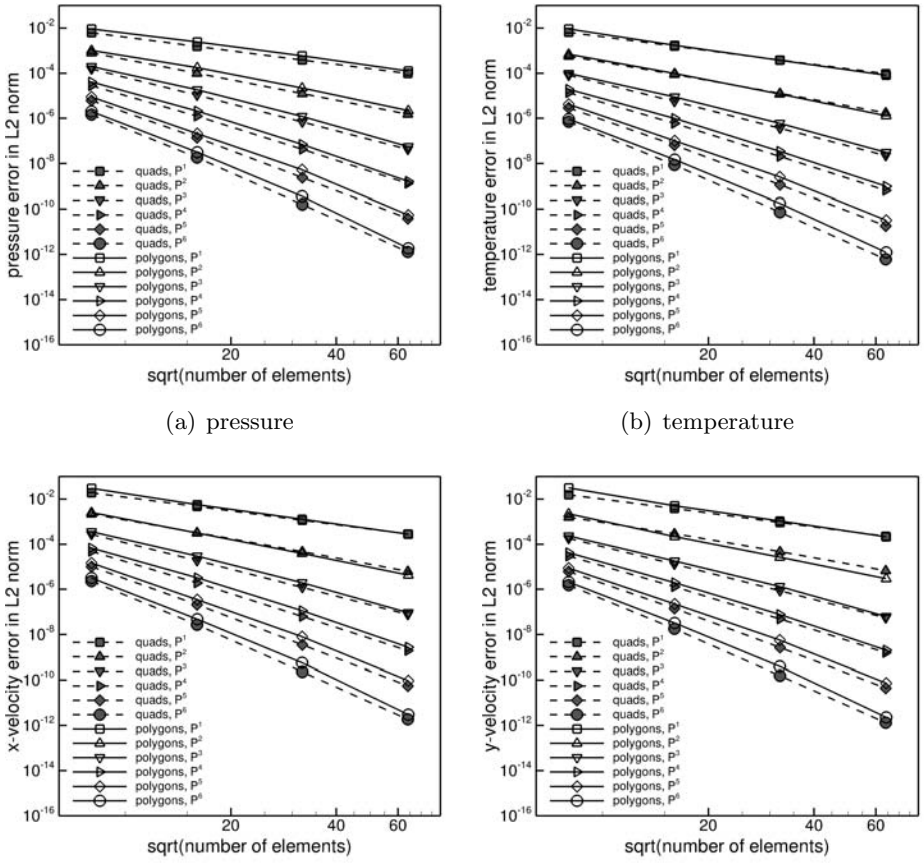
(a) 64, quadrilateral elements

(b) 64, polygonal elements

(c) 256, quadrilateral elements

(d) 257, polygonal elements

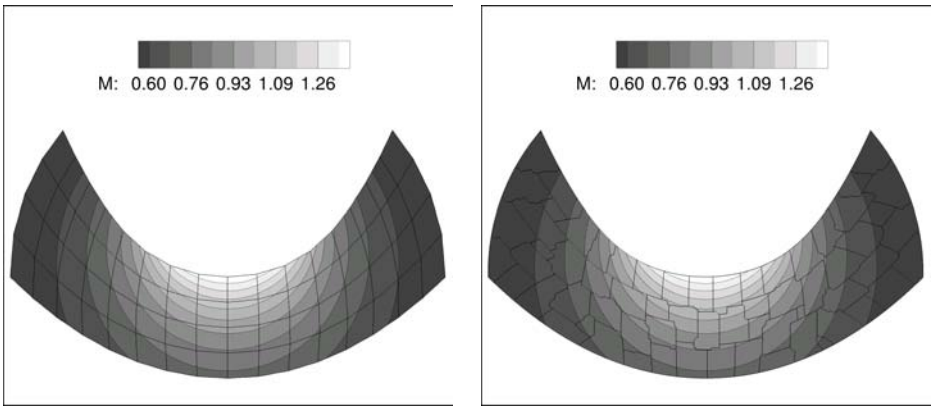
Figure 4.2: Ringleb flow. Quadrilateral and polygonal grids for the Ringleb flow test case.



(a) pressure (b) temperature

(c) x-velocity (d) y-velocity

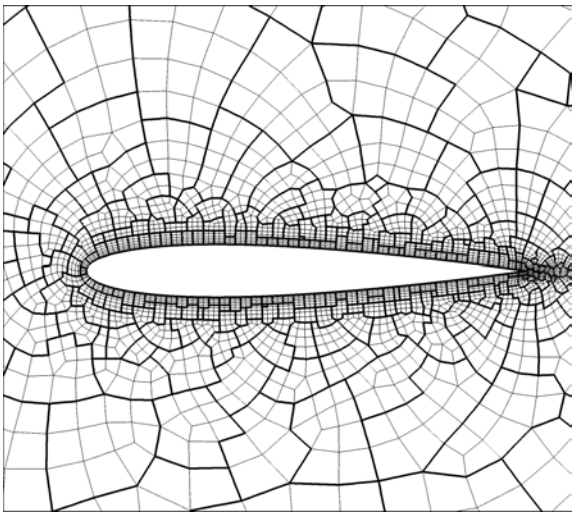
Figure 4.3: Ringleb flow. Convergence rates on uniform grids of quadrilaterals (dashed lines) and polygonal grids obtained by agglomeration (solid lines), $\mathbb{P}^{1\to 6}$.



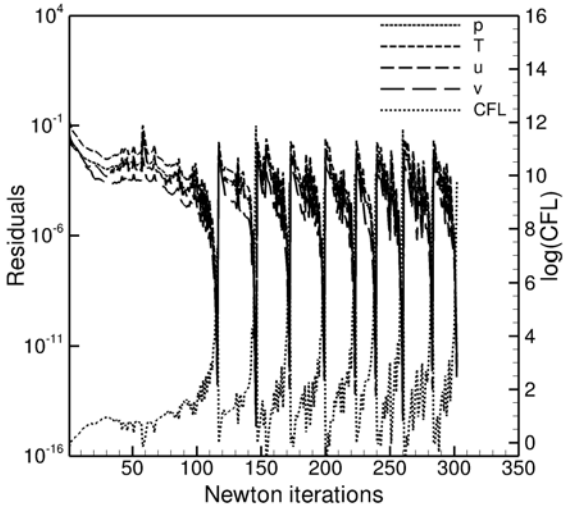
(a) 64, quadrilateral elements (b) 64, polygonal elements

Figure 4.4: Ringleb flow. Mach contours, $\mathbb{P}^6$ solutions on the standard grid and on the agglomerated grid, see text for details.

project [51]. Computations have been carried out up to $\mathbb{P}^9$ polynomial degree on a 452 polymorphic elements mesh built on top of a grid of 4417 13-node quadrilateral elements. Figure 4.5 shows the agglomerated grid superimposed to the underlying fine grid and the residuals convergence history for the sequence of $\mathbb{P}^{1\to 9}$ solutions. The computational effort is shown in terms of number of Newton steps and the corresponding growth of the CFL number has been also displayed. For the purpose of computational efficiency, higher order solutions have been started from the lower order ones. Figure 4.6 displays the Mach number contours with the polymorphic grid superimposed and the pressure coefficient C_p distribution. The discontinuities across elements interfaces diminish when increasing the order of the discretization, suggesting that accuracy increases with the order of the polynomial approximation. Moreover, the values of force coefficients reported in Tab. 4.1 are in line with those of other high-order methods reported by Taube and Munz in [51]. Figure 4.7 displays both the entropy contours and the total pressure losses distribution on the airfoil. We remark



(a)



(b)

Figure 4.5: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. 4.5(a): Agglomerated grid of 452 elements (thick lines) and standard hybrid grid of 4417 13-node quadrilateral elements (thin lines). 4.5(b): Residuals convergence history of $\mathbb{P}^{1\to 9}$ solutions.

	Agglomerated grid 452 el.		
k	C_d	C_l	C_m
1	0.027467	0.315766	0.026349
2	0.022225	0.335675	0.032818
3	0.022044	0.342824	0.035679
4	0.022451	0.351749	0.038364
5	0.022025	0.344492	0.036097
6	0.021979	0.343740	0.035788
7	0.021889	0.341899	0.035372
8	0.021859	0.341349	0.035240
9	0.021908	0.342363	0.035522

Table 4.1: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Values of drag, lift and momentum coefficients.

that spurious entropy deviation at the leading edge decreases dramatically when raising the degree of polynomial approximation. In conclusion, these results show that the employed shock-capturing approach is capable of resolving with good accuracy shocks even on very coarse polymorphic grids.

Viscous flow

NACA0012 $\alpha_\infty = 10°$, $M_\infty = 2$, $Re_\infty = 106$

We consider the laminar supersonic flow around the NACA0012 airfoil at an angle of attack 10°, freestream Mach number of 2, Reynolds number based on the freestream velocity and the airfoil chord equal to 106. Furthermore, the wall temperature is imposed equal to the freestream total temperature. A distinguishing feature of this test is the presence of a detached bow shock in front of the profile. This problem has been considered as validation test case for Navier-Stokes codes in a GAMM workshop and the benchmark

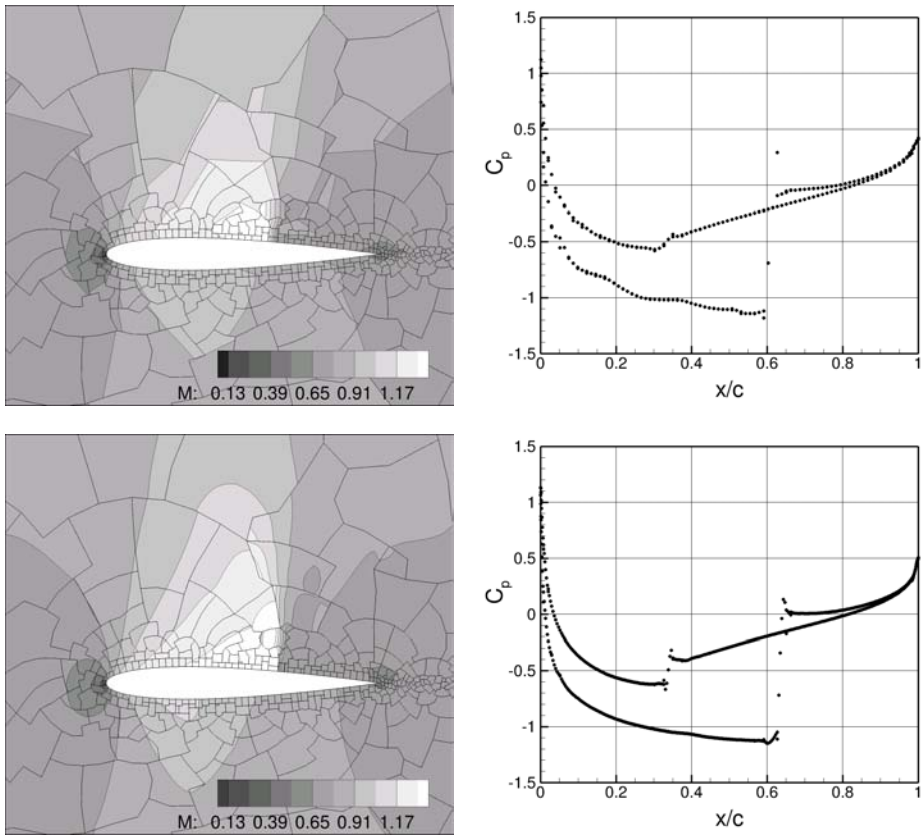


Figure 4.6: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Mach contours and pressure coefficient C_p , $\mathbb{P}^1$ and $\mathbb{P}^3$ solutions (*top/bottom row*).

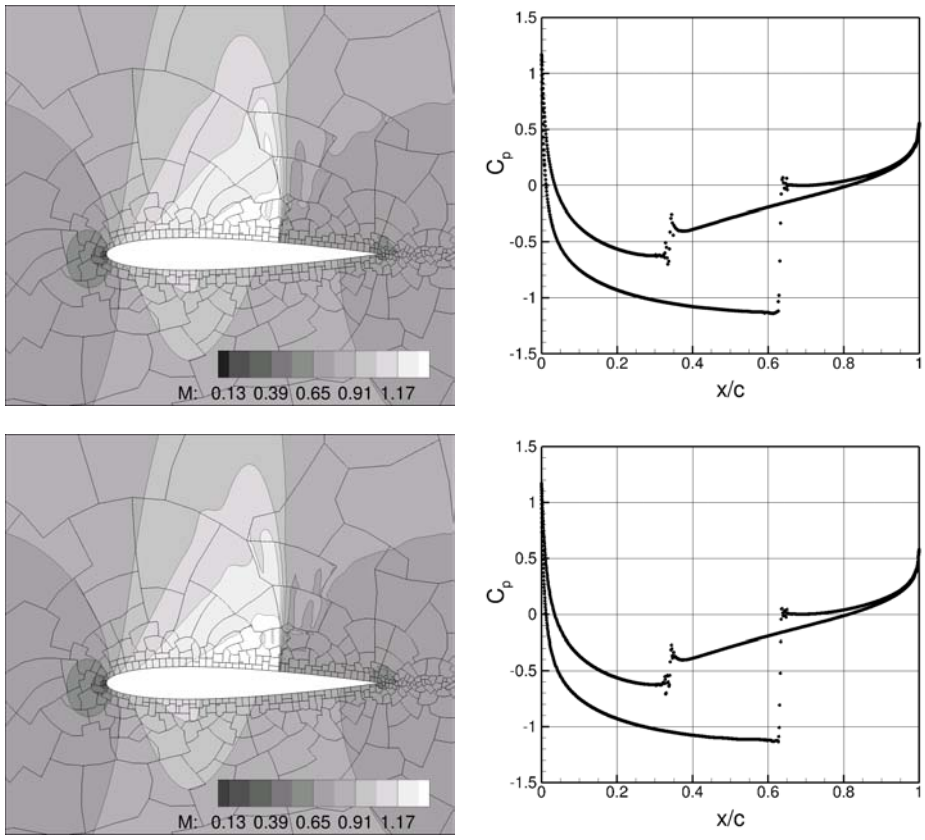


Figure 4.6: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $\mathrm{M}_\infty = 0.8$. Mach contours and pressure coefficient $\mathrm{C_p}$, $\mathbb{P}^6$ and $\mathbb{P}^9$ solutions (*top/bottom row*).

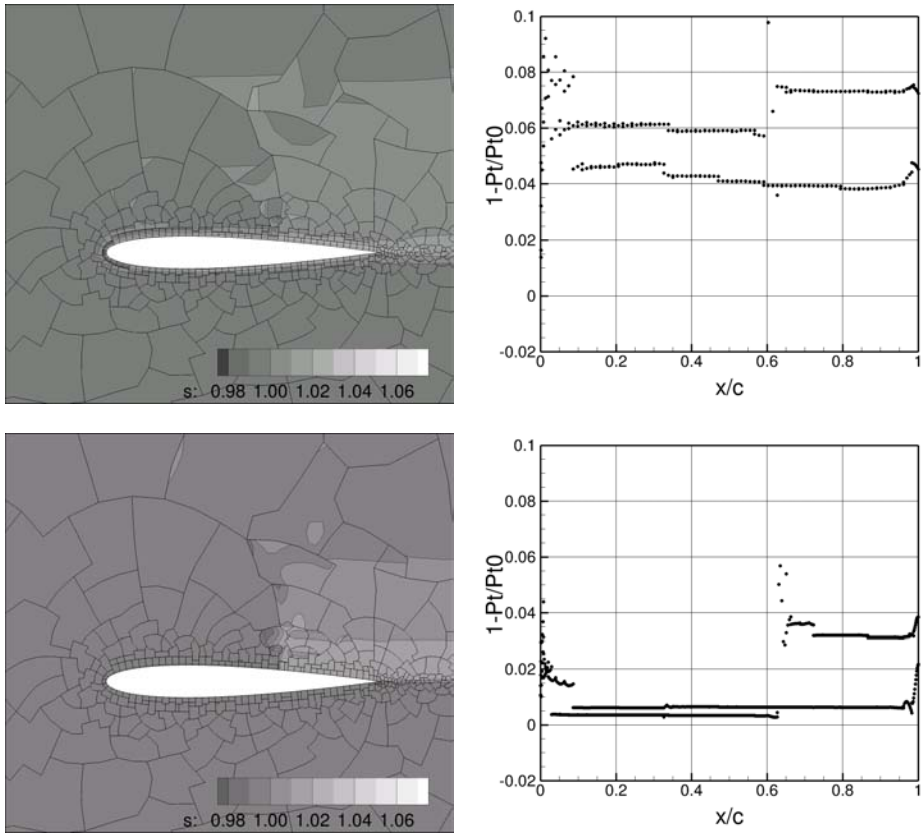


Figure 4.7: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Entropy contours and total pressure losses, $\mathbb{P}^1$ and $\mathbb{P}^3$ solutions (*top/bottom row*).

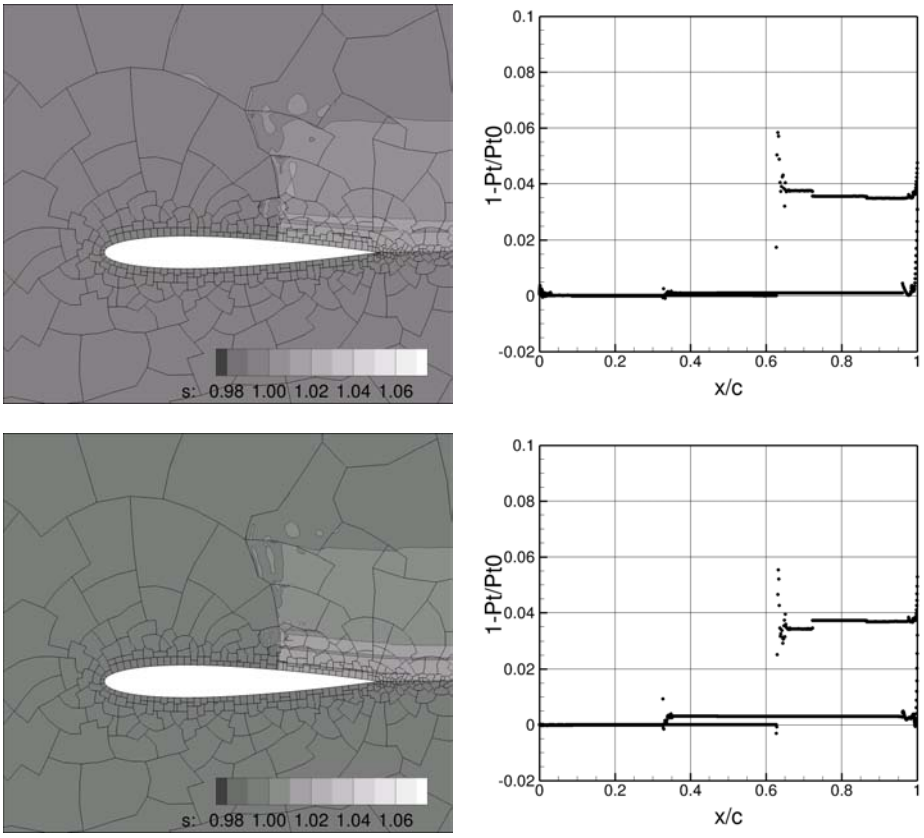


Figure 4.7: NACA0012 airfoil, $\alpha_\infty = 1.25°$, $\mathrm{M}_\infty = 0.8$. Entropy contours and total pressure losses, $\mathbb{P}^6$ and $\mathbb{P}^9$ solutions (*top/bottom row*).

	Agglomerated grid 258 el.		Standard grid 1197 el.	
k	C_d	C_l	C_d	C_l
1	0.507823	0.302784	0.567005	0.320164
2	0.545888	0.318946	0.560081	0.322146
3	0.551031	0.317560	0.561572	0.322742
4	0.556510	0.319238	0.562257	0.323027
5	0.558241	0.320987	0.562497	0.323199
6	0.559662	0.320810	0.562507	0.323130
7	0.559837	0.320863		
8	0.560385	0.320989		
9	0.560673	0.321795		
10	0.561028	0.322037		
11	0.561141	0.322123		
12	0.561220	0.322156		

Table 4.2: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 2$, $Re_\infty = 106$. Values of drag and lift coefficients.

results are very well documented in literature [24, 13].

The solution has been carried out up to $\mathbb{P}^{12}$ polynomial degree on a 259 polymorphic elements mesh built on top of a grid of 1197 13-node quadrilateral and 10-node triangular elements. Figure 4.8(b) reports the residuals convergence history for the sequence of $\mathbb{P}^{1\to12}$ solutions. The coefficients distributions displayed in Figure 4.9 and 4.10, are in good agreement with those available in literature, see [13]. However, in Figure 4.9 some pressure coefficient oscillation are discernible near the trailing edge even when high-degree polynomial approximation is employed, these oscillations are due to a too coarse discretization of the area. The very high-order solutions obtained on the uniformly agglomerated meshes have been compared with high-order solutions computed on the underlying standard grid. Computations have been carried out up to $\mathbb{P}^6$ on the 1197 elements grid employed

for agglomeration. The force coefficients, presented in Tab. 4.2, compare favorably with the most accurate solution obtained on the finer standard grid.

4.6 DG formulation of RANS+k-$\widetilde{\omega}$ equations

4.6.1 Governing equations

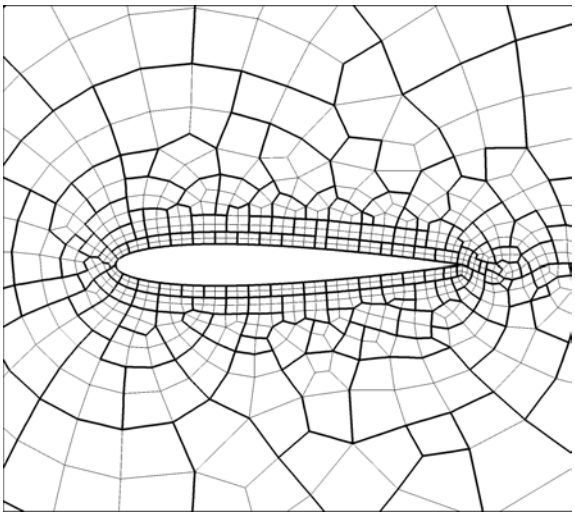
The turbulent flows are here treated by employing the Reynolds averaged Navier–Stokes (RANS) equations with closure provided by k-ω turbulence model, as described by Wilcox [54]. Following the approach proposed by Bassi and Rebay in [15], the stability of the coupled RANS and k-ω high-order computations can be greatly enhanced by means of a nonstandard implementation of the two equations model, where $\widetilde{\omega} = log(\omega)$ is used in place of ω itself. In fact, the use of the logarithmic variable $\widetilde{\omega}$ intrinsically guarantees the positivity of the specific dissipation rate providing also a much smoother near wall distribution. A further improvement in robustness and efficiency of turbulent computations is achieved by enforcing the respect of "realizability conditions" for the turbulent stresses. As described by Bassi, Crivellini, Rebay and Savini in [11] this means assuring that the turbulence model predicts positive normal turbulent stresses and fulfills the Schwarz inequality for the shear turbulent stresses. Below, for the sake of clarity let us refer to the implementation proposed by Bassi, Crivellini, Rebay and Savini in [11] as k-$\widetilde{\omega}$ model.

Let us consider the two-dimensional RANS+k-$\widetilde{\omega}$ equations written in compact conservation form

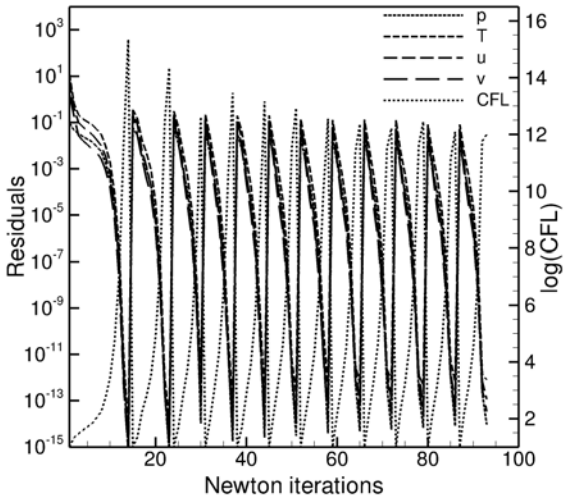
$$\frac{\partial \mathbf{u}}{\partial t} \; + \; \boldsymbol{\nabla} \cdot \mathbf{F}_c(\mathbf{u}) \; + \; \boldsymbol{\nabla} \cdot \mathbf{F}_v(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) \; + \; \mathbf{s}(\mathbf{u}, \boldsymbol{\nabla}\mathbf{u}) = \mathbf{0}, \tag{4.25}$$

where $\mathbf{u}, \mathbf{s} \quad \mathbb{R}^M$ denote the vectors of the M conservative variables and source terms, $\mathbf{F}_c, \mathbf{F}_v \quad \mathbb{R}^M \otimes \mathbb{R}^2$ denote the inviscid and viscous flux functions respectively.

The conservative variables $\mathbf{u}$ and the Cartesian components $\mathbf{f}_c(\mathbf{u})$ and



(a)



(b)

Figure 4.8: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 2$, $Re_\infty = 106$. 4.8(a): Agglomerated grid of 259 elements (thick lines) and standard hybrid grid of 1197 13-node quadrilateral and 10-node triangular elements (thin lines). 4.8(b): Residuals convergence history of $\mathbb{P}^{1\to 12}$ solutions.

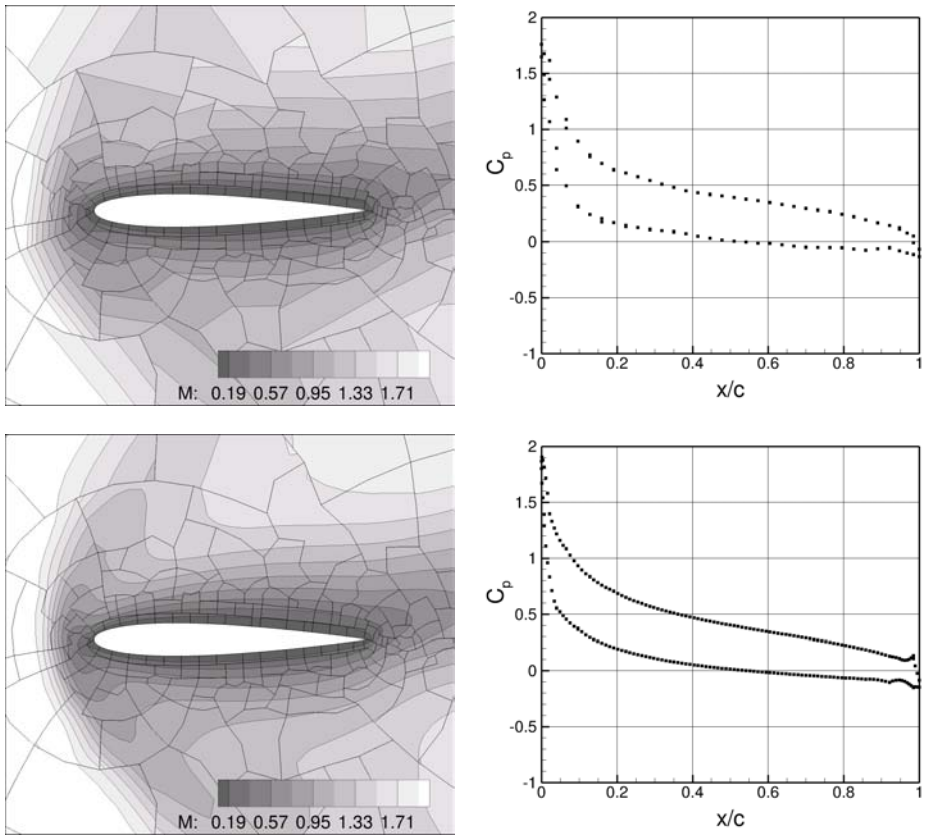


Figure 4.9: NACA0012 airfoil, $\alpha_\infty = 10°$, $\mathrm{M}_\infty = 2$, $\mathrm{Re}_\infty = 106$. Mach contours and pressure coefficient $\mathrm{C_p}$, $\mathbb{P}^1$ and $\mathbb{P}^3$ solutions (*top/bottom row*).

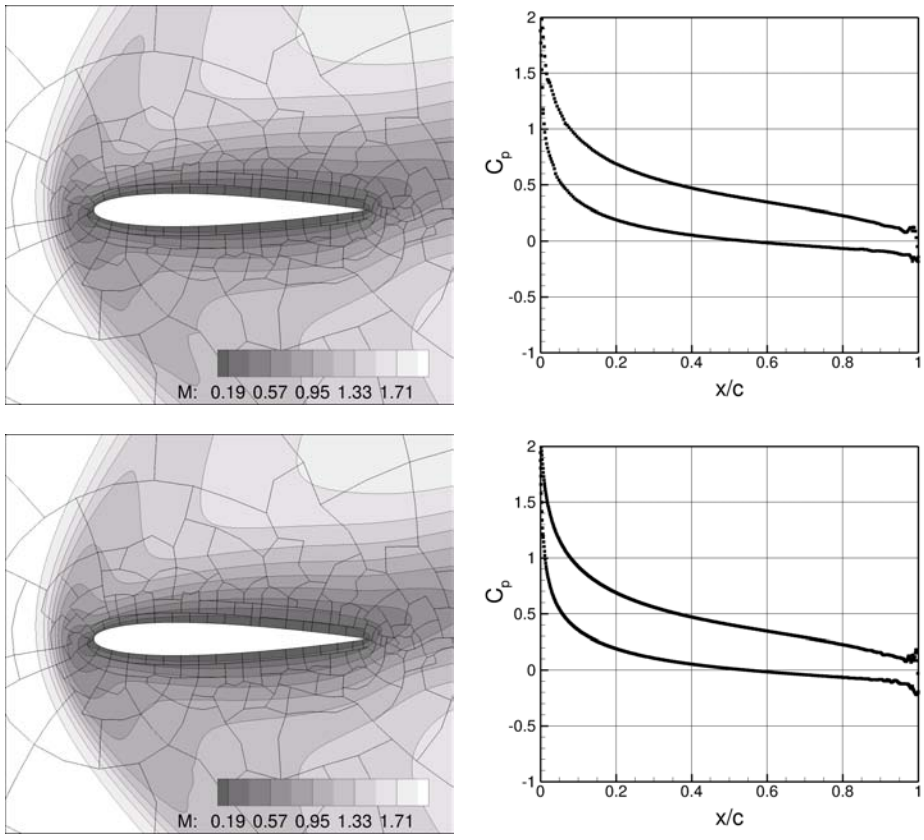


Figure 4.9: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 2$, $Re_\infty = 106$. Mach contours and pressure coefficient C_p , $\mathbb{P}^6$ and $\mathbb{P}^{12}$ solutions (*top/bottom row*).

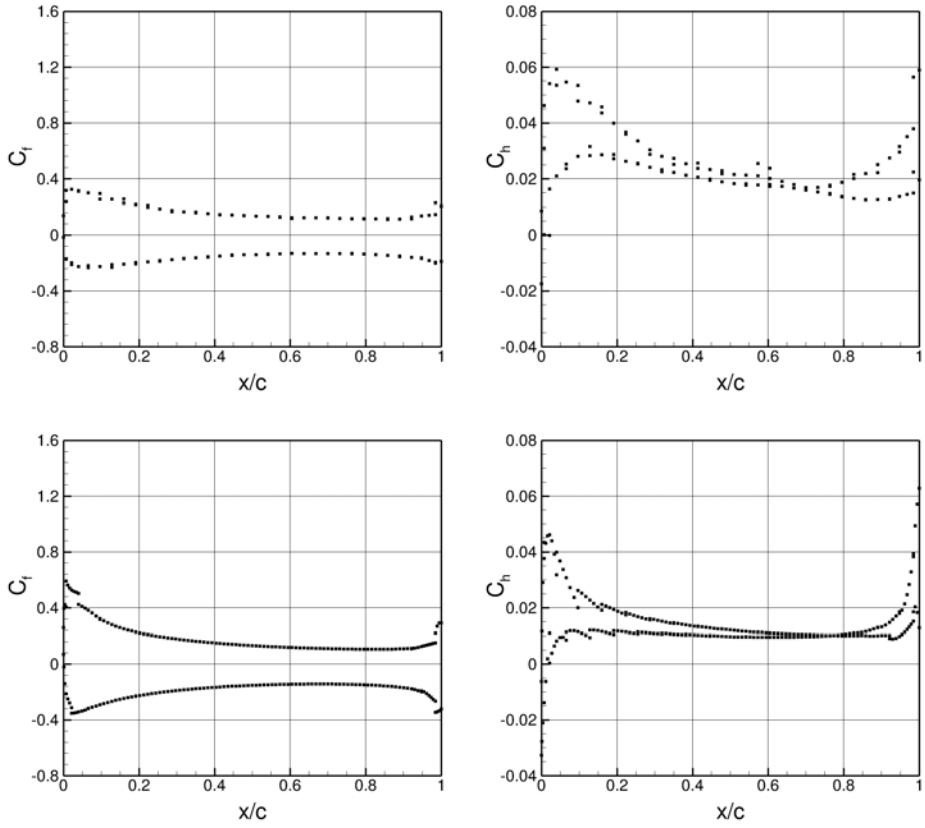


Figure 4.10: NACA0012 airfoil, $\alpha_\infty = 10°$, $\mathrm{M}_\infty = 2$, $\mathrm{Re}_\infty = 106$. Friction, $\mathrm{C_f}$, and heat flux, $\mathrm{C_h}$, coefficients distribution, $\mathbb{P}^1$ and $\mathbb{P}^3$ solutions (*top/bottom row*).

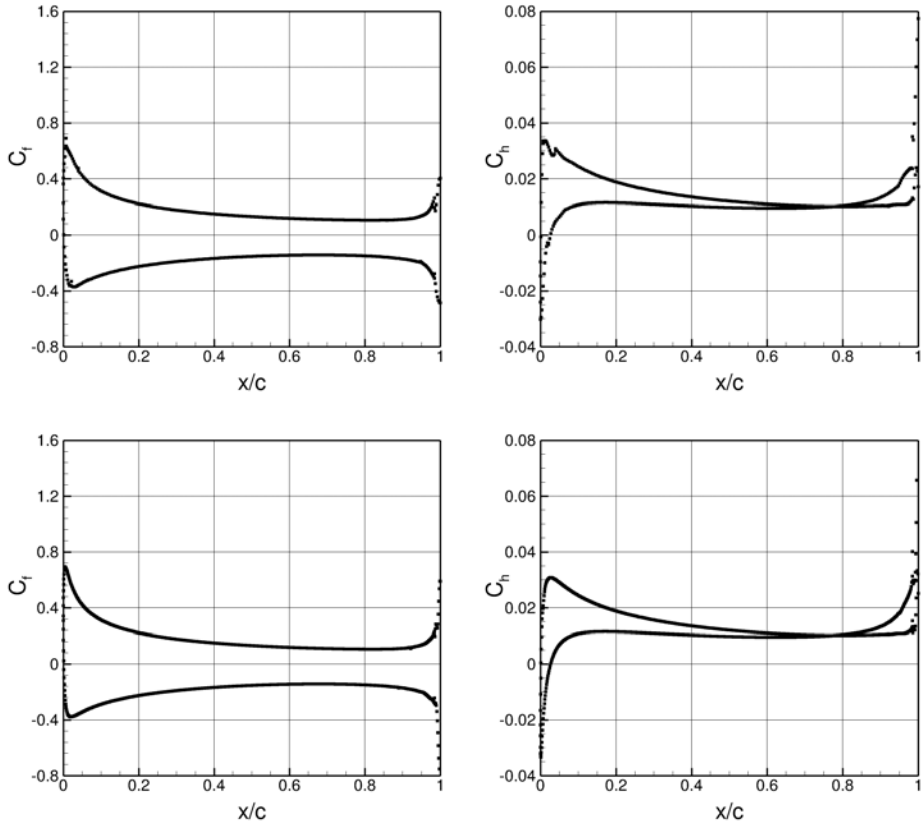


Figure 4.10: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 2$, $Re_\infty = 106$. Friction, C_f , and heat flux, C_h , coefficients distribution, $\mathbb{P}^6$ and $\mathbb{P}^{12}$ solutions (*top/bottom row*).

$\mathbf{g}_c(\mathbf{u})$ of the inviscid $\mathbf{F}_c(\mathbf{u})$ flux function, equal

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho e_0 \\ \rho u \\ \rho v \\ \rho k \\ \rho\widetilde{\omega} \end{bmatrix}, \quad \mathbf{f}_c(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho h_0 u \\ \rho uu + p \\ \rho vu \\ \rho ku \\ \rho\widetilde{\omega}u \end{bmatrix}, \quad \mathbf{g}_c(\mathbf{u}) = \begin{bmatrix} \rho v \\ \rho h_0 v \\ \rho uv \\ \rho vv + p \\ \rho kv \\ \rho\widetilde{\omega}v \end{bmatrix}, \tag{4.26}$$

where ρ is the fluid density, p is the pressure, u and v are the velocity components, e_0 and h_0 are the total energy and enthalpy per unit mass, k is the turbulent kinetic energy and $\widetilde{\omega}$ is the logarithm of specific dissipation rate. The total enthalpy per unit mass is defined as $h_0 = e_0 + p/\rho$. By assuming that the fluid obeys to the perfect gas state equation, p can be computed as $p = (\gamma - 1)\,\rho\left[e_0 - \left(u^2 + v^2\right)/2\right]$ where γ indicates the ratio between the specific heats of the fluid.

The Cartesian components $\mathbf{f}_v(\mathbf{u},\boldsymbol{\nabla}\mathbf{u})$ and $\mathbf{g}_v(\mathbf{u},\boldsymbol{\nabla}\ \mathbf{u})$ of the viscous flux function $\mathbf{F}_v(\mathbf{u},\boldsymbol{\nabla}\ \mathbf{u})$, are given by

$$\mathbf{f}_v(\mathbf{u},\boldsymbol{\nabla}\mathbf{u}) = -\begin{bmatrix} 0 \\ u\hat{\sigma}_1 + v\hat{\tau}_{12} + \left(\frac{\mu}{\mathrm{Pr}} + \frac{\overline{\mu}_t}{\mathrm{Pr_t}}\right)\partial_x h \\ \hat{\sigma}_1 \\ \hat{\tau}_{12} \\ (\mu + \sigma^*\overline{\mu}_t)\,\partial_x k \\ (\mu + \sigma\overline{\mu}_t)\,\partial_x\widetilde{\omega} \end{bmatrix}, \tag{4.27}$$

$$\mathbf{g}_v(\mathbf{u},\boldsymbol{\nabla}\mathbf{u}) = -\begin{bmatrix} 0 \\ u\hat{\tau}_{21} + v\hat{\sigma}_2 + \left(\frac{\mu}{\mathrm{Pr}} + \frac{\overline{\mu}_t}{\mathrm{Pr_t}}\right)\partial_y h \\ \hat{\tau}_{21} \\ \hat{\sigma}_2 \\ (\mu + \sigma^*\overline{\mu}_t)\,\partial_y k \\ (\mu + \sigma\overline{\mu}_t)\,\partial_y\widetilde{\omega} \end{bmatrix}, \tag{4.28}$$

and the source term is

$$\mathbf{s}(\mathbf{u},\boldsymbol{\nabla}\mathbf{u}) = -\begin{bmatrix} 0 \\ -\left[\sigma_1\partial_x u + \tau_{12}\partial_y u + \tau_{21}\partial_x v + \sigma_2\partial_y v\right] + \beta^*\rho\overline{k}e^{\widetilde{\omega}_r} \\ 0 \\ 0 \\ \sigma_1\partial_x u + \tau_{12}\partial_y u + \tau_{21}\partial_x v + \sigma_2\partial_y v - \beta^*\rho\overline{k}e^{\widetilde{\omega}_r} \\ \left\{\frac{\alpha}{\overline{k}}\left[\sigma_1\partial_x u + \tau_{12}\partial_y u + \tau_{21}\partial_x v + \sigma_2\partial_y v\right]\right. \\ \left. -\beta^*\rho e^{\widetilde{\omega}_r} + (\mu+\sigma\overline{\mu}_t)\left[(\partial_x\widetilde{\omega})^2 + (\partial_y\widetilde{\omega})^2\right]\right\} \end{bmatrix}, \tag{4.29}$$

where

$$\sigma_1 = \overline{\mu}_t\left[2\partial_x u + \lambda\left(\partial_x u + \partial_y v\right)\right] - {}^2\!/_3\rho\overline{k}, \tag{4.30}$$

$$\sigma_2 = \overline{\mu}_t\left[2\partial_y v + \lambda\left(\partial_x u + \partial_y v\right)\right] - {}^2\!/_3\rho\overline{k}, \tag{4.31}$$

$$\tau_{12} = \tau_{21} = \overline{\mu}_t\left(\partial_x v + \partial_y u\right), \tag{4.32}$$

$$\hat{\sigma}_1 = (\mu+\overline{\mu}_t)\left[2\partial_x u + \lambda\left(\partial_x u + \partial_y v\right)\right] - {}^2\!/_3\rho\overline{k}, \tag{4.33}$$

$$\hat{\sigma}_2 = (\mu+\overline{\mu}_t)\left[2\partial_y v + \lambda\left(\partial_x u + \partial_y v\right)\right] - {}^2\!/_3\rho\overline{k}, \tag{4.34}$$

$$\hat{\tau}_{12} = \hat{\tau}_{21} = (\mu+\overline{\mu}_t)\left(\partial_x v + \partial_y u\right), \tag{4.35}$$

$$\overline{k} = \max(0,k), \tag{4.36}$$

$$\overline{\mu}_t = \alpha^*\rho\overline{k}e^{-\widetilde{\omega}_r}, \tag{4.37}$$

with ∂_x and ∂_y we refer to partial derivatives with respect to variables x and y, μ is the dynamic viscosity coefficient, Pr and $\mathrm{Pr_t}$ are the molecular and turbulent Prandtl numbers respectively, and, using the Stokes hypothesis, $\lambda = -\frac{2}{3}$. The variable $\widetilde{\omega}_r$ in the source terms in Eq. (4.29) and in the eddy viscosity, $\overline{\mu}_t$, defined by Eq. (4.37) is introduced to indicate that $\widetilde{\omega}_r$ fulfills suitably defined "realizability" conditions, see [11]. Furthermore, to deal with possible negative values of k, the limited value $\overline{k}$ and the related eddy viscosity $\overline{\mu}_t$ have been used. The values of the closure parameters α, α^*, β, β^*, σ, σ^* are those of the high- or low-Reynolds number k-ω model of Wilcox, [54]. Notice that a source term is present in mean-flow energy equation because the total energy e_0 and enthalpy h_0 do not include the turbulent kinetic energy, see *e.g.*, [27].

4.6.2 DG discretization of the RANS+k-$\widetilde{\omega}$ equations

A DG discretization of the RANS+k-$\widetilde{\omega}$ equations can be obtained similarly to the procedure described for Navier–Stokes equations in §4.2.2. The DG formulation for the RANS+k-$\widetilde{\omega}$ equations reads as follows:
find $\mathbf{u}_h \quad V_h$ such that

$$\sum_{T\in\mathcal{T}_h}\int_T v_{h,j}\frac{\partial u_{h,j}}{\partial t}\,\mathrm{d}\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \boldsymbol{\nabla}_h v_{h,j}\cdot\mathbf{F}_j\left(\mathbf{u}_h,\mathbf{z}_h\right)\,\mathrm{d}\mathbf{x} + \sum_{\sigma\in\mathcal{F}_h^\sigma}\int_\sigma [\![v_{h,j}]\!]\cdot\widehat{\mathbf{F}}_j\left(\mathbf{u}_h^{\pm},\mathbf{z}_{h\,\sigma}^{\pm}\right)\,\mathrm{d}\sigma + \sum_{T\in\mathcal{T}_h}\int_T v_{h,j}\mathbf{s}_j\left(\mathbf{u}_h,\mathbf{z}_h\right)\,\mathrm{d}\mathbf{x} = 0 \qquad \mathbf{v}_h \quad V_h. \tag{4.38}$$

Implicit time integration, performed via backward Euler scheme, and time marching strategy are employed as reported for the Navier–Stokes equations in §4.3.

Surface boundary condition for $\widetilde{\omega}$

As reported by Wilcox [54] the ω equation provides an analytical solution in the viscous sublayer, that for $\widetilde{\omega}$ can be written as

$$\widetilde{\omega} = \log\left(\frac{6\nu}{\beta}\right) - 2\log\left(y + \sqrt{\frac{6\nu}{\beta e^{\widetilde{\omega}_w}}}\right), \tag{4.39}$$

where y is the local coordinate normal to the wall and $\widetilde{\omega}_w$ is the value of $\widetilde{\omega}$ at the wall ($y = 0$). For $\widetilde{\omega}_w$ the solution is singular and is considered the appropriate solution for smooth walls, whereas non-singular solutions are those corresponding to finite values of $\widetilde{\omega}_w$ and provide a way to include effects of surface roughness through surface boundary conditions. According to the so-called rough-wall method of Wilcox the surface values

of $\widetilde{\omega}$ can be simply set by means of the following correlation that recovers the smooth wall solution when the surface roughness tends to zero

$$\widetilde{\omega}_w = \log\left(S_r \frac{u_\tau^2}{\nu_w}\right), \tag{4.40}$$

where $u_\tau = \sqrt{\tau_w/\rho_w}$ is the friction velocity and τ_w, ρ_w and ν_w are the shear stress, the density and the kinematic viscosity at the wall, respectively. The non-dimensional function S_r is defined as

$$S_r = \begin{cases} (50/k_r^+)^2 & \text{if} \quad k_r^+ < 25, \\ 100/k_r^+ & \text{if} \quad k_r^+ \geq 25, \end{cases} \tag{4.41}$$

where $k_r^+ = k_r u_\tau/\nu_w$ denotes the non-dimensional equivalent sand-roughness height. For rough surfaces with prescribed values of k_r, Eq. (4.40) allows to compute the values ω_w to be set at the wall surface. Of course, the grid density or the degree of polynomial approximation should be high enough to provide accurate solutions.

On the other hand, the implementation of the smooth wall boundary condition for ω requires special care in the numerical treatment of the singularity. Two popular approaches have been proposed by Wilcox [54] and Menter [41]. The former strategy involves to replace the perfectly smooth surface with an hydraulically smooth surface, obtaining a so-called "slightly-rough-wall" boundary condition. This approach requires to properly set a value of k_r low enough to guarantee that $k_r^+ < 5$, *i.e.*, it should ensure that the surface is hydraulically smooth with roughness peaks lying within the viscous sublayer. The latter strategy consists in setting at the wall the analytical solution computed at y_1, that represents the distance to the next grid point away from the wall, multiplied by a factor 10, or put another way, the analytical solution computed at $y = \alpha_M y_1$ where $\alpha_M = 1/\quad\overline{10}$.

Hellsteen [35] observed that Menter's approach could be interpreted in terms of a slightly-rough-wall roughness value function of y_1. Moreover, he proposed to optimize the fraction of the height of cell α_M by means of an accurate near-wall numerical study. In the framework of the DG method, a

similar approach has been presented by Bassi, Crivellini, Rebay and Savini in [11], where the α_M coefficient of Menter has been replaced by a lower value given by $\alpha = 0.3\sqrt{6/\beta}/50$.

However, as higher degree polynomials can follow closer and closer the exact near wall distribution of ω, it seems reasonable to make α dependent on the degree k of the polynomial approximation. In this direction the approach here adopted is that proposed and numerically validated by Bassi, Botti, Colombo, Ghidoni and Rebay in [8], where the value $\widetilde{\omega}_w$ is again related to the exact solution computed at $y = \alpha y_1$ but taking into account the polynomial degree k by setting

$$\widetilde{\omega}_w^k = \log\left(\frac{6\nu_w}{\beta\left(\alpha^k h\right)^2}\right), \tag{4.42}$$

where

$$\alpha^k = e^{-\sum_{n=1}^{k}\frac{1}{n}}. \tag{4.43}$$

As Eq. (4.42) holds for hydraulically smooth surfaces, we remark that the implied slightly-rough-wall roughness should satisfy the condition $k_r^+ < 5$. Therefore, if locally this condition is not satisfied, then $\widetilde{\omega}_w$ is computed by means of Eq. (4.40) with $k_r^+ = 5$. This could be the case, for instance, of low-degree polynomial solutions computed on relatively coarse grids targeted at high-degree polynomial approximations.

Remarks on elements agglomeration for turbulent computations

High Reynolds number turbulent flows are characterized by near wall steep gradients of flow quantities. Hence, first layers of cells near solid walls must provide fine spatial resolution and regular growing rate suited to cope with the numerical difficulties related to the high-order solution of the RANS and k-ω equations near solid surfaces.

As the agglomeration process can not guarantee sufficient regularity of cells height in the first few layers of elements near solid walls, agglomeration has been prevented in this zone and the underlying fine grid has been employed.

4.7 Numerical example

In this section the solution of the of RANS+k-$\widetilde{\omega}$ equations on agglomerated grid is reported. In the following computation the surface has been assumed hydraulically smooth, and the $\widetilde{\omega}_w$ values have been set according with Eq. (4.42).

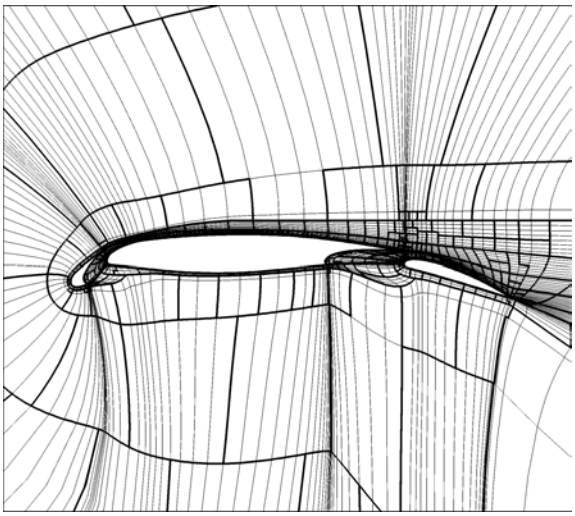
L1T2 $\alpha_\infty = 20.18°$, $M_\infty = 0.197$, $Re_\infty = 3.52 \times 10^6$.

The flow around the three elements airfoil configuration L1T2 has been computed with a farfield Mach number $M_\infty = 0.197$, angle of attack 20.18° and chord-based Reynolds number $Re_\infty = 3.52 \times 10^6$. The main difficulty of this problem is in the flow complexity that involves strongly interacting wakes. This test case has been computed up to $\mathbb{P}^6$ polynomial approximation on a 1131 polymorphic mesh built on top of a grid, suitable for turbulent computations, that consists of 4740 13-node quadrilateral elements, see Figure 4.11(a). In accordance with the foregoing the first three layers of elements at the wall boundary have not been agglomerated. Figures 4.12 and 4.13 display the Mach number, the turbulence intensity contours and the pressure coefficient C_p distribution, showing that for $\mathbb{P}^6$ polynomial approximation the C_p is in good agreement with available experimental data.

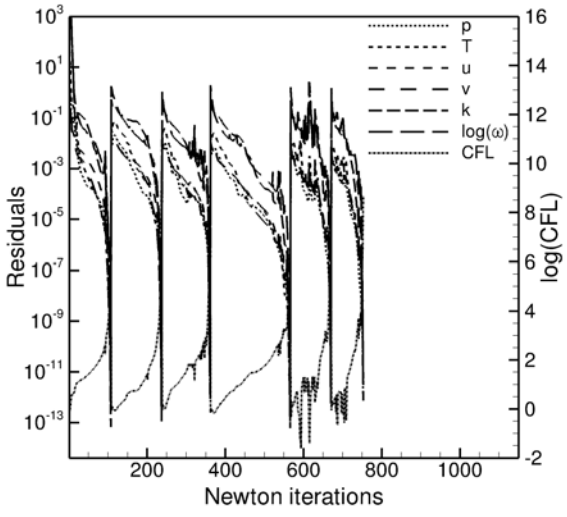
4.8 Conclusions

In this chapter the robustness and accuracy of an implicit solution of the Euler and Navier-Stokes equations on grids that consist of arbitrarily shaped elements have been assessed by means of classical compressible test cases up to very high-order approximations. Moreover, transonic computations evidenced that the shock-capturing approach of [5] is capable of resolving with good accuracy shocks even on coarse polymorphic meshes. Early stage results concerning the solution of RANS+k-$\widetilde{\omega}$ equations on polymorphic grids have been shown, demonstrating the feasibility of this approach even for turbulent problems. These results should be considered as an important

starting point since the robust and accurate computation on polymorphic elements is the key ingredient when targeting to solution algorithms that can involve agglomerated elements of unpredictable shape, such h-multigrid and h-adaptivity.



(a)



(b)

Figure 4.11: L1T2 three elements airfoil, $\alpha_\infty = 20.18°$, $\mathrm{M}_\infty = 0.197$, $\mathrm{Re}_\infty = 3.52 \times 10^6$. 4.11(a): Agglomerated grid of 1160 elements (thick lines) and standard hybrid grid of 4740 13-node quadrilateral elements (thin lines). 4.11(b): Residuals convergence history of $\mathbb{P}^{1\to6}$ solutions.

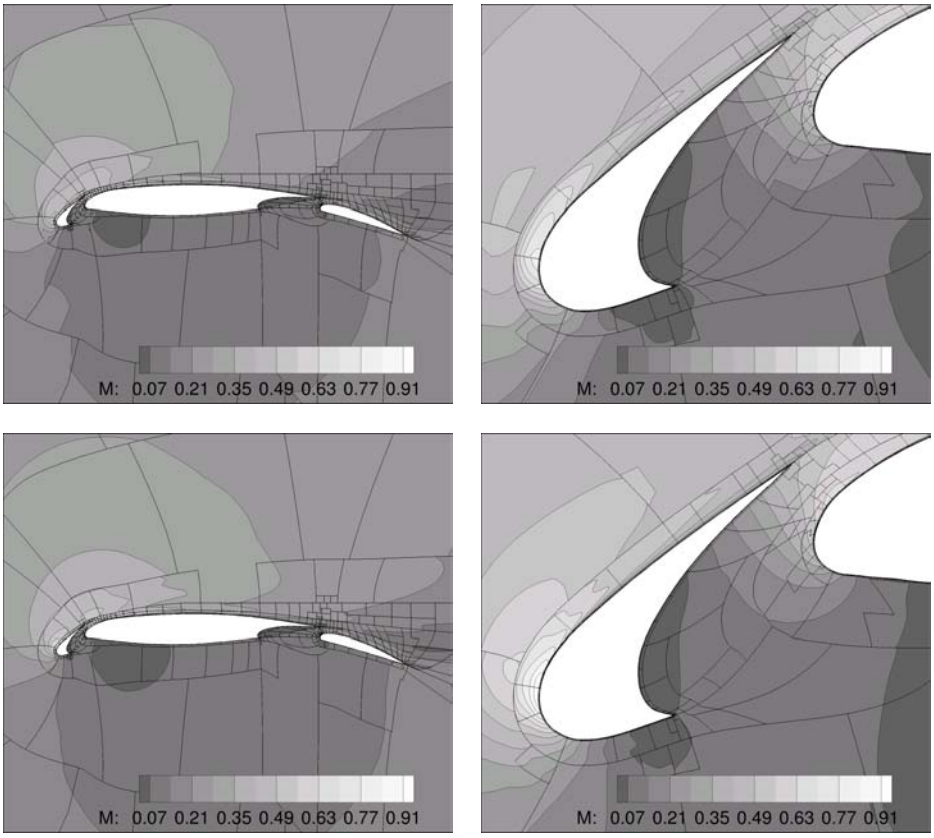


Figure 4.12: L1T2 three elements airfoil, $\alpha_\infty = 20.18°$, $\mathrm{M}_\infty = 0.197$, $\mathrm{Re}_\infty = 3.52 \times 10^6$. Mach contours, $\mathbb{P}^3$ and $\mathbb{P}^6$ solutions (*top/bottom row*).

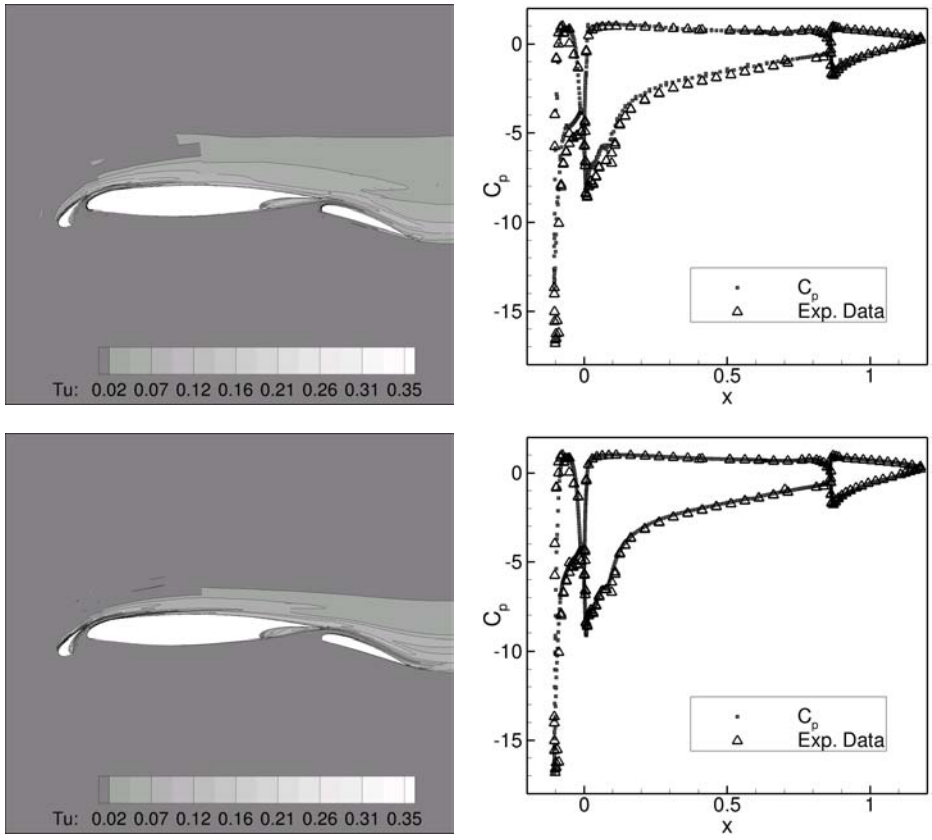


Figure 4.13: L1T2 three elements airfoil, $\alpha_\infty = 20.18°$, $M_\infty = 0.197$, $Re_\infty = 3.52 \times 10^6$. Turbulence intensity contours and pressure coefficient distribution C_p , $\mathbb{P}^3$ and $\mathbb{P}^6$ solutions (*top/bottom row*).

Chapter 5

Decoupling geometry representation from solution approximation

5.1 Introduction

In this chapter an alternative approach to high-order grid generation is presented. Once a standard fine grid able to provide an accurate domain discretization has been produced by means of standard low-order grid generation tools, a computational mesh suitable for the desired accuracy and computationally affordable can be obtained via agglomeration while keeping the boundary resolution of the fine grid, see also Bassi, Botti, Colombo, Di Pietro and Tesini in [6] and Bassi, Botti, Colombo and Rebay in [7]. The effectiveness of this approach in representing the geometry of the domain is numerically assessed both on a Poisson model problem and on challenging inviscid and viscous test cases.

5.2 Potential of agglomeration strategy

In general, the computational mesh is built as a tassellation of an approximation Ω_h of the actual domain Ω and boundary conditions are set on the approximate geometry. This introduces a consistency error with respect to the unknown exact solution that can dominate the approximation error and degrade convergence properties [6]. Former papers, see *e.g.*, Bassi and Rebay [14] and more recently Krivodonova and Berger [40], have emphasized the importance of an accurate discretization of domain boundaries in order to obtain accurate high-order DG solutions. Clearly the higher is the desired accuracy of the solution the higher must be the domain representation accuracy provided by the computational mesh. A trivial strategy is to improve the representation of Ω decreasing the mesh step size at least in the elements adjacent to the boundary. However, in the context of high-order discretization this approach is suboptimal due to the significant increase in the number of degrees of freedom. In fact, for the sake of efficiency, high-order solutions should be computed on grids with a relatively low number of large elements. In turn, this implies to devise an accurate description of a (possibly complex) boundary geometry by means of large element faces.

The classical approach of the so called *iso-parametric* finite elements, derived in the context of standard Galerkin discretizations, consists in increasing the order of the geometry discretization together with the order of the approximation in each mesh cell. However, the generation of high-order meshes is by no means a trivial task, especially for highly stretched meshes like those employed in CFD problems. More recently, other approaches based on Computer Aided Design (CAD) representations of the domain boundaries by means of Non-Uniform Rational B-Splines (NURBS) have been proposed, see *e.g.*, Hughes, Cottrell and Bazilevs [36] or the more conservative work of Sevilla, Fernández-Méndez and Huerta [50]. These are interesting techniques and certainly deserves more attention. Study of these approaches will be left for future work.

Here a simple alternative based on grid agglomeration is employed. Once a standard fine grid $\mathcal{R}$, able to provide an accurate domain discretization has been produced, a computational mesh suitable for the intended so-

lution approximation can be obtained via agglomeration. The underlying grid is built by means of standard low-order (*e.g.*, second order) grid generation tools and then, in order to obtain a computationally affordable mesh, the elements are clustered together while keeping the boundary resolution of the fine grid. Let us remind that, in agglomerated grids the boundary faces Σ of an aggregated element $T \quad \mathcal{T}_h$ can be defined as a portion of ∂T such that $\Sigma = \partial T \setminus \partial\Omega_h$, cfr. §3.2. Moreover, for every face Σ there exist a set $\sigma_h^\Sigma \subset \mathcal{F}_h^\mathrm{b}$ collecting the facets partitioning Σ, *i.e.*,

$$\overline{\Sigma} = \bigcup_{\sigma \in \sigma_h^\Sigma} \overline{\sigma}.$$

Clearly, increasing the number of facets composing a mesh face Σ enhances the boundary discretization. Indeed, not only the location of $\partial\Omega_h$, but also the accuracy in the computation of boundary normals and curvature is improved. Note that, the boundary representation can be refined without affecting the number of elements, and hence the number of degrees of freedom, of the agglomerated mesh. However, the price to pay for such flexibility is an increased cost of numerical integration as the number of elements of the underlying fine grid has an impact on the overall number of quadrature points, see 2.2.3.

5.3 Numerical examples

The following tests are designed to demonstrate how the devised approach allows to obtain accurate very high-order results on adequately coarse grids even when only low-order mesh generation tools are available. The following computations involve grids with the same cardinality and the same location of elements vertices, obtained by means of two different approaches. On the one side, meshes are created directly from low-order grid generator and their elements sides are hence entirely described by a polynomial, let us denote these grids as standard, see Figure 5.1 left. On the other side, agglomerated elements are built on top of a very fine grid with low-order

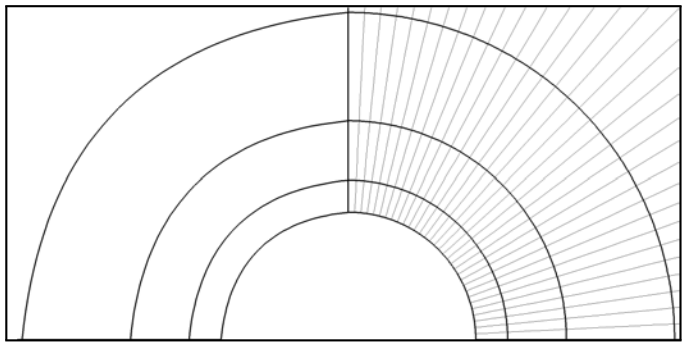


Figure 5.1: Circle. 4×8 8-node quadrilateral elements standard grid (circumferential × radial number of elements) compared with an agglomerated one built on top of a 128×8 8-node quadrilateral elements fine mesh (*left-/right*), sub-elements are depicted in thin line.

representation of sides, this entails that agglomerated elements faces are piecewise polynomial curved segments, see Figure 5.1 right.

The accuracy of the approach presented above is validated by means of a Poisson model problem and of challenging inviscid and viscous test cases.

Model test case for the Poisson problem

The solution of the test case proposed by Gobbert and Yang in [29] is here reported and the k-convergence rate of the BRMPS method, cfr. §3, when dealing with curved boundaries, is assessed [6].

Let us consider a fine mesh $\mathcal{R}$ that consists of 8-node quadrilateral elements approximating the unit annulus $\Omega = \ 0.5 < x^2 + y^2 < 1.5$, we test against the following exact solution of the Poisson equation (3.1):

$$u = \cos(\pi\sqrt{x^2 + y^2}), \tag{5.1}$$

with suitable forcing term f; see Figure 5.2. The solution (5.1) vanishes on the exact boundary $\partial\Omega$. The following discretizations of the homogeneous

boundary condition are considered:

$$\text{Choice 1:} \qquad u_{\partial\Omega_h} = \cos(\pi\sqrt{x^2+y^2}), \tag{5.2a}$$

$$\text{Choice 2:} \qquad u_{\partial\Omega_h} = 0. \tag{5.2b}$$

The choice (5.2a) neglects any influence of the domain discretization since the exact solution on $\partial\Omega_h$ is imposed; instead, the choice (5.2b) introduces a consistency error as $\partial\Omega_h = \partial\Omega$. For $i \quad 0,\ldots,3$, we generate an agglomerated grid $\mathcal{T}_{h,i}$ composed of 32×32 elements starting from a fine mesh $\mathcal{R}_i$ that consists of $(32\cdot2^i)\times32$ 8-node quadrilateral elements (circumferential $\times$ radial number of elements). For all $i \quad 0,\ldots,3$ the number of sub-elements clustered in each agglomerated element $T \quad \mathcal{T}_{h,i}$ equals 2^i. Clearly the choice $i=0$ leads to a standard non-agglomerated grid. Let us remark that while the number of agglomerated elements does not change, $\partial\Omega_h$ approximates $\partial\Omega$ more closely by increasing i.

The convergence results shown in Figure 5.2 demonstrate the ability of the boundary condition in (5.2b) (which is the one used in practice) to provide, increasing i, convergence results closer to the exponential convergence resulting from choice (5.2a). Instead, the lack of consistency for $i=0$ is clearly appreciable as the discretization is unable to provide an error in L^2-norm lower that 10^{-5} employing the boundary condition in (5.2b).

Inviscid test cases

The inviscid test cases include the flow around a circle and the Ringleb flow. The inviscid problems here considered are isentropic and we use the L_2-norm of the entropy error

$$e_{s\ 2} \;=\; \left(\frac{\int_{\Omega_h}(s-s_{\text{ex}})^2\ \mathrm{d}\mathbf{x}}{\Omega_h}\right)^{\frac{1}{2}}, \tag{5.3}$$

to evaluate the deviations of the computed entropy from the exact solution s_{ex}. Indeed, a poor geometric discretization of the boundary gives rise to errors in the numerical solution which, for the Euler equations in conservation form, appear as spurious entropy production near solid walls, see Figure 5.3.

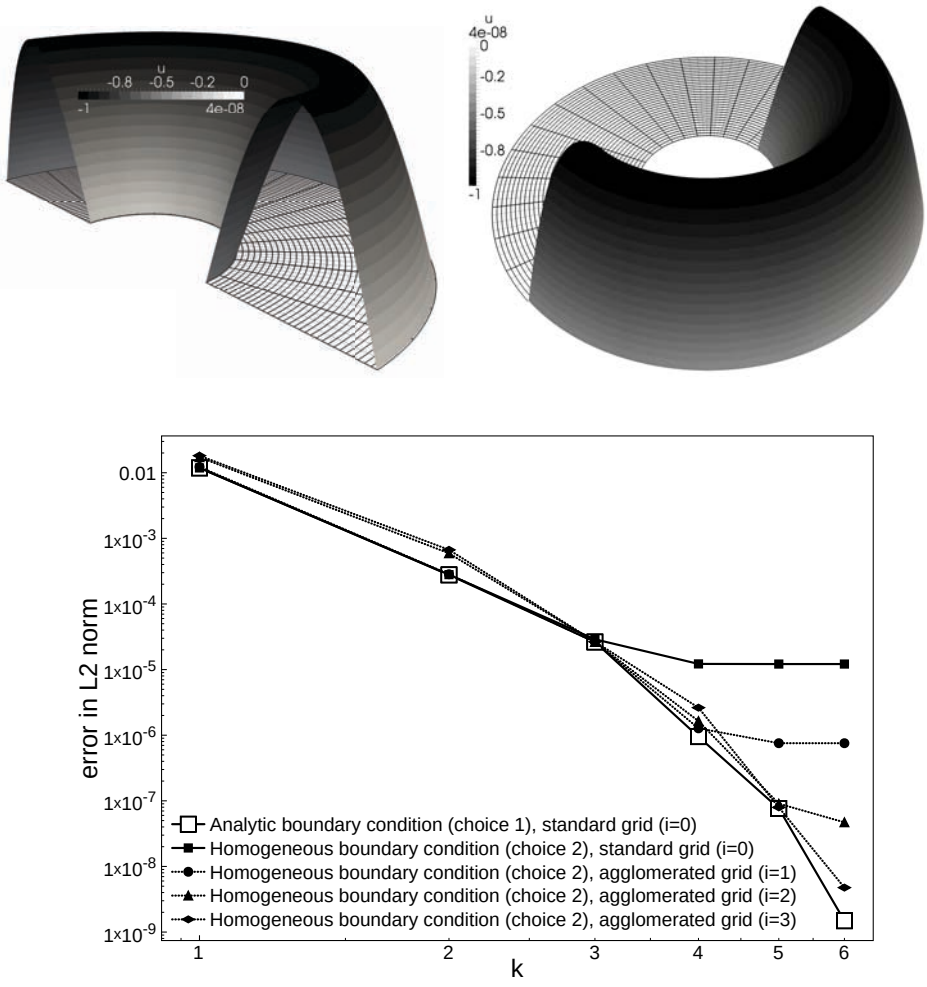


Figure 5.2: Test case proposed by Gobbert and Yang [29]. *Top row*, Solution on a 32×32 grid (thick lines) agglomerated on top of a 128×32 quadratic quadrilateral mesh (thin lines), $\mathbb{P}^6$ solution (half of the domain is sketched). *Bottom row*, k-convergence of the BRMPS discretization using a 32×32 agglomerated grid and various underlying fine meshes and boundary conditions, see text for details.

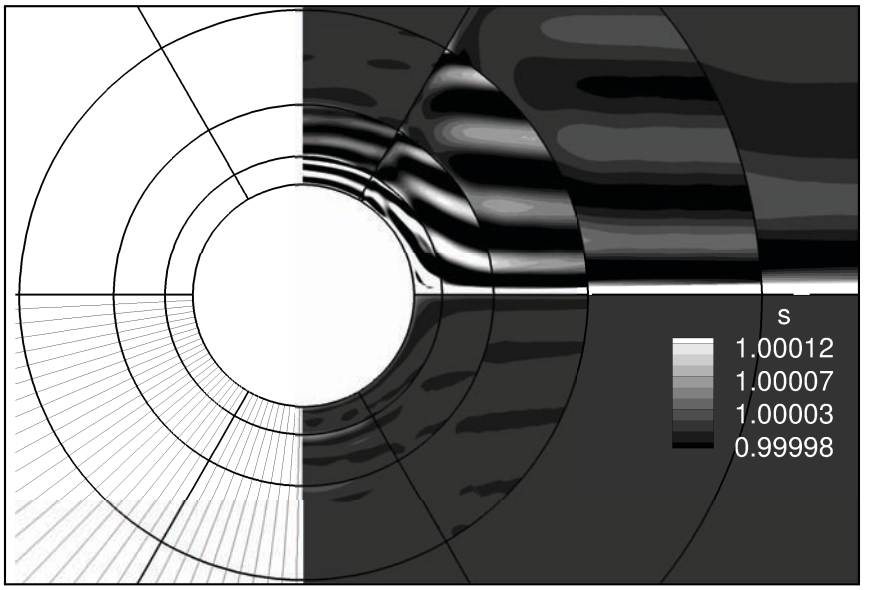


Figure 5.3: Quadratic 6×8 standard grid and agglomerated 6×8 built on top of a second order 126×8 fine mesh (II quadrant/III quadrant), subelement are represented with thin lines. Entropy contour on the standard and agglomerated grid (I quadrant/IV quadrant), $\mathbb{P}^{10}$ solutions. Ten equally spaced entropy levels ranging between the maximum and minimum value of the agglomerated grid flow field are considered.

Flow around a circle

Let us consider the subsonic flow around the unit circle $\Omega = \ 1 < x^2 + y^2 < 20$ with $M_\infty = 0.38$, a typical test case considered for the evaluation of numerical methods for the Euler equations. Solutions have been computed on standard and agglomerated meshes containing 4×8, 6×8, and 8×8 elements (circumferential × radial number of elements). All the computations have been performed up to $\mathbb{P}^{10}$ polynomial approximations. Characteristic and slip boundary conditions are applied at the outer boundary and on the circle surface, respectively. The geometric shape of standard grids

elements is defined by means of 8-node quadrilaterals for the 6×8 and 8×8 grids and 13-node quadrilaterals for the 4×8 grid. Geometric approximation of element faces is thus quadratic in the former cases and cubic in the latter. The higher-order approximation of boundary faces in the 4×8 grid was found necessary to avoid convergence difficulties or even breakdown of computations. The underlying fine grid of the 4×8 and 8×8 agglomerated meshes consists of 128×8 8-node quadrilaterals, that is for all $\Sigma \quad \partial\Omega_h$, $\mathrm{card}(\sigma_h^\Sigma) = 32$ and $\mathrm{card}(\sigma_h^\Sigma) = 16$, respectively. The underlying fine grid of the 6×8 mesh consists of 126×8 8-node quadrilaterals and for all $\Sigma \quad \partial\Omega_h$, $\mathrm{card}(\sigma_h^\Sigma) = 21$, see Figures 5.3. Figures 5.4 and 5.5 show the results, in terms of Mach isolines, computed on the 4×8 and 6×8 standard and agglomerated grids. The influence of the geometry representation on the solutions can be plainly appreciated. While up to $\mathbb{P}^8$ polynomial approximation the 6×8 standard grid solutions display an unphysical wake developing downstream of the circle, the $\mathbb{P}^{10}$ solutions seem to agree much better. Nevertheless, Figure 5.3 clearly highlights the improved entropy distribution of the $\mathbb{P}^{10}$ solution computed on the 6×8 agglomerated grid. The entropy errors obtained using the standard and the improved boundary approximation have been summarized in Figure 5.9(a) for all the polynomial approximations and grid densities. Figures 5.6 and 5.7 compare the pressure coefficient and total pressure loss surface distributions of the $\mathbb{P}^5$ and $\mathbb{P}^{10}$ solutions computed on all the standard and agglomerated grids employed in this test. Standard grid solutions clearly suffer from insufficient accuracy of boundary discretization, as is evident from the more and more localized spikes of aerodynamic quantities at element interfaces. These discontinuities reduce but do not disappear when the polynomial degree of the solution is raised. The spikes at elements junction vanish in the agglomerated mesh solutions thus demonstrating the effectiveness of the agglomeration-based approach to overcome the lack of accuracy due to the poor representation of the geometry. The gain of accuracy is tangible even for the 4×8 grids where the piecewise quadratic discretization of sides used for the generation of the agglomerated mesh is compared with a standard grid that consists of elements with cubic sides.

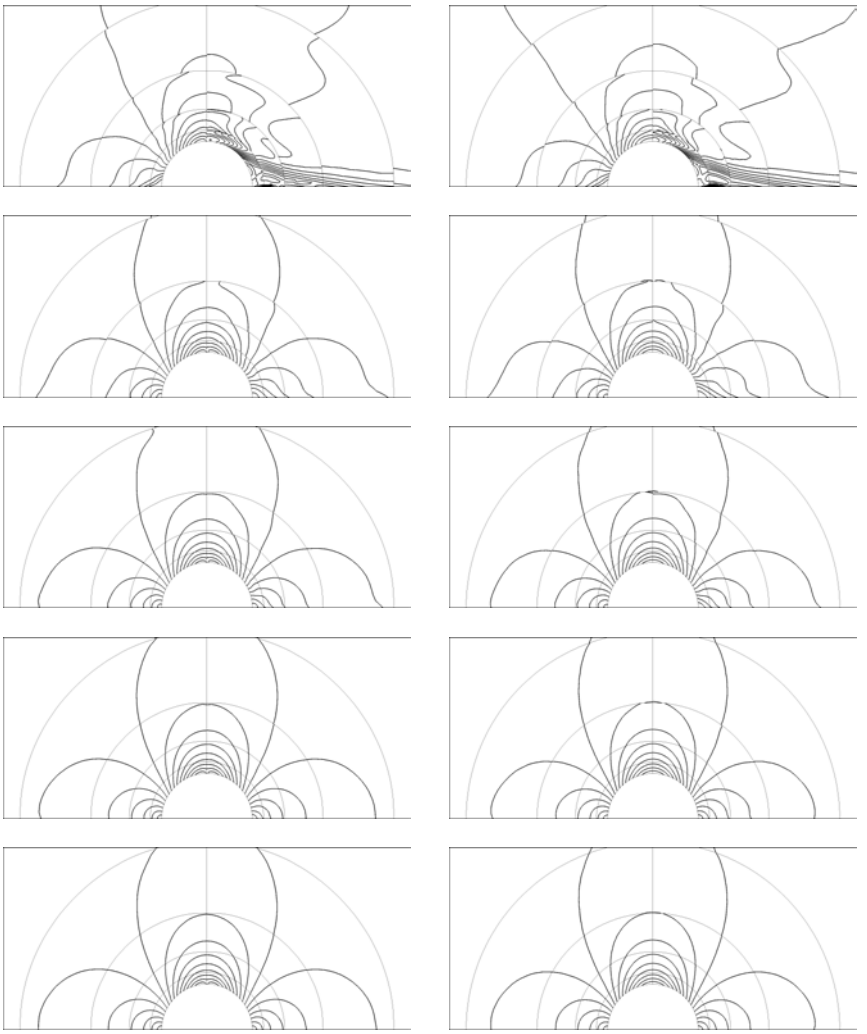


Figure 5.4: Flow around the unit circle, Mach contours. *Left column, top to bottom.* $\mathbb{P}^{6\to 10}$ approximations, 4×8 13-node quadrilateral elements grid *Right column, top to bottom.* $\mathbb{P}^{6\to 10}$ approximations, 4×8 mesh agglomerated on top of the 128×8 8-node quadrilateral elements grid.

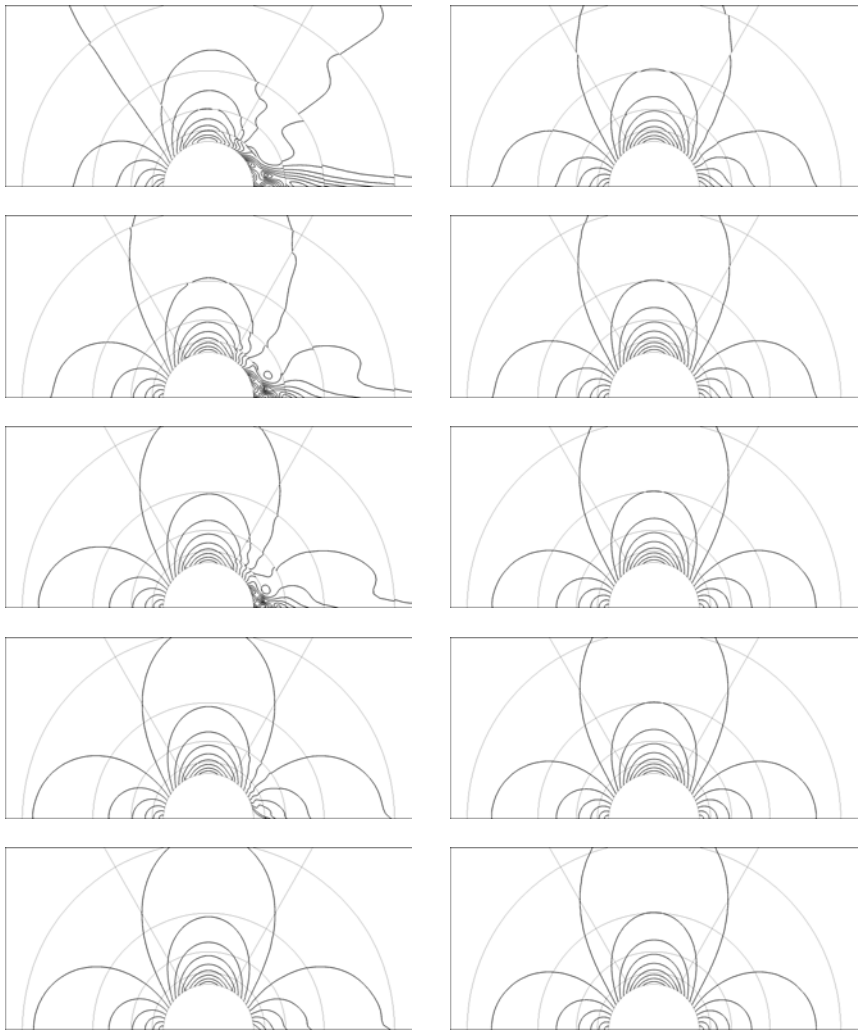


Figure 5.5: Flow around the unit circle, Mach contours. *Left column, top to bottom.* $\mathbb{P}^{6\to10}$ approximations, 6×8 8-node quadrilateral elements grid *Right column, top to bottom.* $\mathbb{P}^{6\to10}$ approximations, 6×8 mesh agglomerated on top of the 126×8 8-node quadrilateral elements grid.

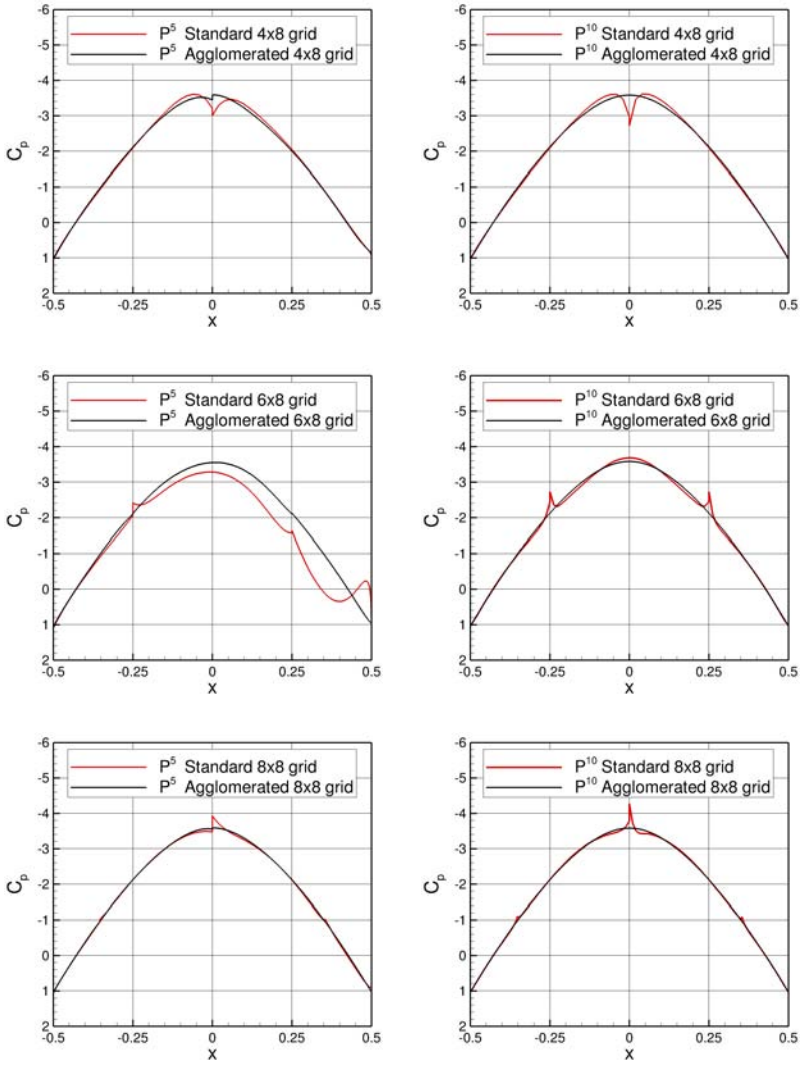


Figure 5.6: Pressure coefficient distribution around the circle on 4×8, 6×8 and 8×8 standard and agglomerated grids (*top/bottom*) for $\mathbb{P}^5$ and $\mathbb{P}^{10}$ solutions (*left/right*).

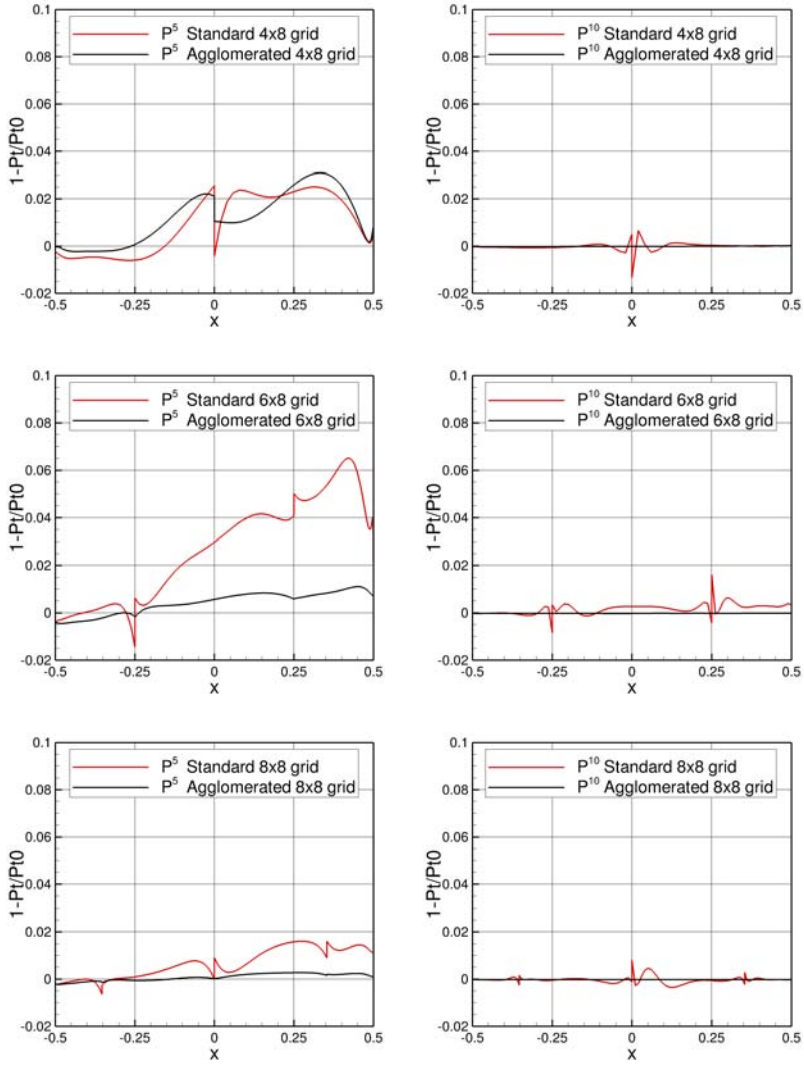


Figure 5.7: Total pressure loss distribution around the circle on 4×8, 6×8 and 8×8 standard and agglomerated grids (*top/bottom*) for $\mathbb{P}^5$ and $\mathbb{P}^{10}$ solutions (*left/right*).

Ringleb flow

We next present the results obtained for the widely known Ringleb flow test case, which corresponds to an analytical transonic smooth solution of the Euler equations obtained by means of the hodograph method (see, *e.g.*, [1]). The problem is particularly sensitive to an accurate treatment of the slip boundary conditions and requires careful discretization of the boundary representing the streamline where the velocity becomes supersonic. This is the reason why the Ringleb flow is often used to test convergence properties of numerical schemes replacing the slip condition on $\partial\Omega_h$ with the exact solution. In the following computations is also employed this technique to evaluate the reference convergence behaviour of the DG scheme as a function of the degree of polynomial approximation on a given grid consisting of 16×4 elements.

Let us remark that, while setting the exact solution on $\partial\Omega_h$ eliminates any influence of the boundary discretization, the slip boundary condition set on $\partial\Omega_h$ rather than on $\partial\Omega$ introduces a consistency error that should be kept small enough so as not to impair the accuracy of the solution [6, 7]. At the inflow and outflow boundaries the analytical solution is imposed.

For $i \quad 0, 2, 3, 4$, we construct an agglomerated mesh $\mathcal{T}_{h,i}$ consisting of 16×4 elements starting from a fine mesh $\mathcal{R}_i$ containing $(16 \cdot 2^i) \times (4 \cdot 2^i)$ 8-node quadrilateral elements. For all $i \quad 0, 2, 3, 4$ the number of sub-elements clustered in each agglomerated element $T \quad \mathcal{T}_{h,i}$ equals 2^{2i}, while, for all $\Sigma \quad \partial\Omega_h$, $\mathrm{card}(\sigma_h^{\Sigma}) = 2^i$. Solutions on the agglomerated grids have been computed up to $\mathbb{P}^{10}$ polynomial approximation while solutions beyond $\mathbb{P}^6$ using the slip boundary condition on the standard grid failed to converge.

Figure 5.8 highlights the much better accuracy of the solutions computed on agglomerated grids with respect to the one computed on the standard grid. As for the flow around a circle, the numerical solutions on standard grid is spoiled by a large spurious entropy production at solid boundary and at the junction of some elements. The improvements achievable by means of grid agglomeration can be nicely appreciated in Figure 5.9(b), where the k-convergence of entropy errors on agglomerated and standard grids using the slip boundary condition are compared to the reference con-

vergence curve. This Figure shows that, raising i, the convergence of the entropy error on agglomerated grids gets closer to the reference convergence curve, whilst for $i = 0$ the consistency error dominates and prevents the entropy error to fall below 10^{-3}.

Viscous test case

NACA0012 $\alpha_\infty = 10°$, $M_\infty = 0.8$, $Re_\infty = 73$

We consider the transonic, laminar flow around a NACA0012 airfoil with freestream flow conditions $\alpha_\infty = 10°$, $M_\infty = 0.8$ and $Re_\infty = 73$, based on airfoil chord. The temperature at the airfoil surface is set equal to the freestream total temperature. This is one of the problems proposed in a GAMM workshop, [24], for the validation of Navier-Stokes codes. As for the inviscid problems, the proposed test case aims at assessing the ability of large agglomerated element to suitably approximate the computational domain boundaries, as described in § 5.2. Here, several elements of the standard grid are deliberately grouped to form very elongated elements along the NACA0012 profile resulting in a more efficient distribution of degrees of freedom. We have computed up to $\mathbb{P}^{12}$ solutions on a grid of 178 elements resulting from the agglomeration of 1197 elements of an hybrid grid with 13-node quadrilaterals and 10-node triangles, see Figure 5.10. Lacking exact reference solutions, we compare very high-order solutions on the agglomerated grid with high-order solutions computed on the underlying standard grid. Computations have been carried out up to $\mathbb{P}^6$ on the 1197 elements grid employed for agglomeration. Let us remark that the two meshes provide exactly the same geometric resolution of the airfoil surface. Figure 5.10(a) shows the agglomerated grid superimposed to the underlying fine grid. Despite the quite arbitrary shape of agglomerated elements, with several re-entrant angles, the convergence to steady state of the $\mathbb{P}^{1\to12}$ solutions, shown in Figure 5.10(b), is pretty satisfactory. The Mach number contours and coefficients distributions shown in Figure 5.11 confirm the good quality of the $\mathbb{P}^{12}$ solution on the agglomerated grid. The force coefficients, reported both in Figure 5.12 and in Tab. 5.1, are in very

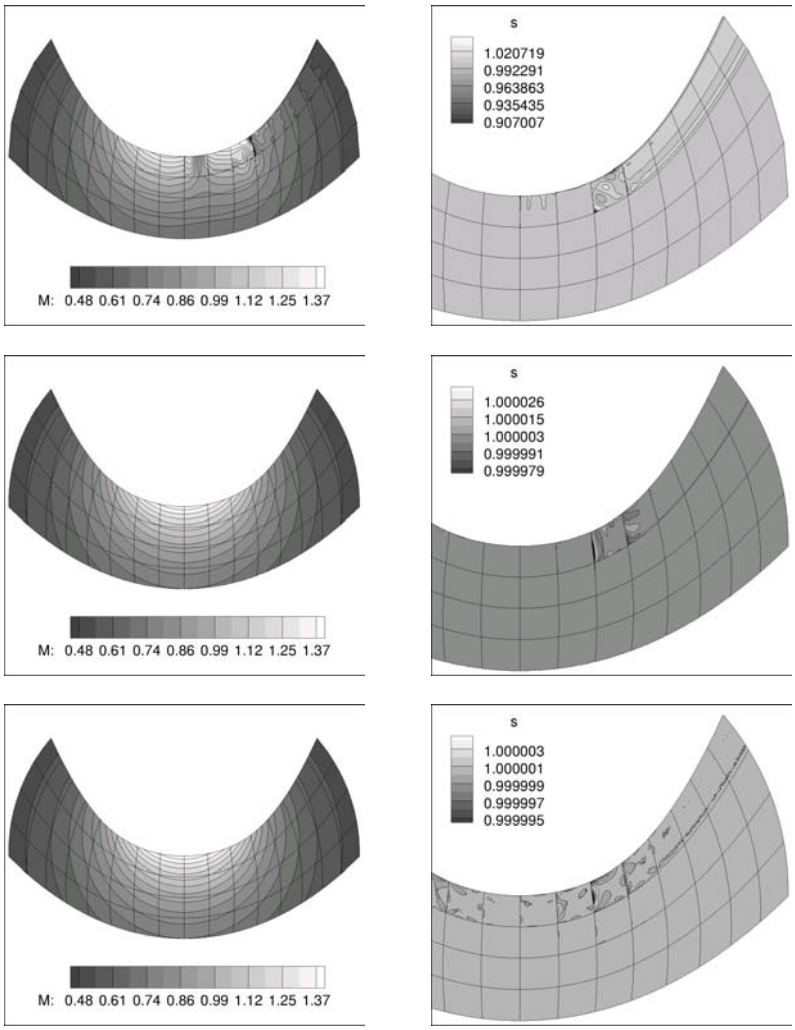
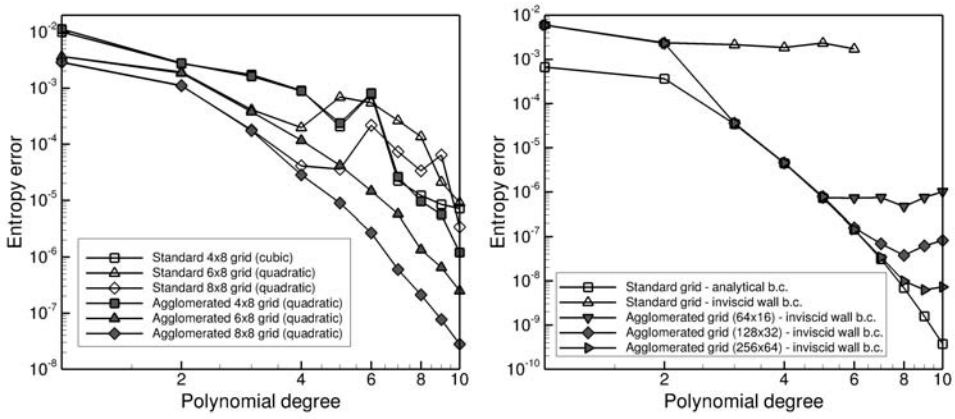


Figure 5.8: Ringleb flow. Mach number (*left column*) and entropy (*right column*) contours of the $\mathbb{P}^6$ solutions on the standard grid ($i = 0$, *1*$^{\text{st}}$*row*) and on the agglomerated grids ($i = 2, 3$, *2*$^{\text{nd}}$ *and 3*$^{\text{rd}}$*row*), see text for details. We consider ten equally spaced entropy contours between maximum and minimum value of each solution.



(a) (b)

Figure 5.9: Entropy error vs. polynomial degree of the solution approximation. 5.9(a): Flow past a circle. 5.9(b): Ringleb flow.

good agreement with those computed on the fine standard grid. The results of this test case show that it is possible to exploit the accuracy related to high-order discretizations without dealing with high-order mesh generation techniques. Nevertheless, the importance of grid topology must not be underestimated. The agglomerated mesh of this problem had to be adjusted by trial-and-error in order to recover good agreement of force coefficients computed on the agglomerated and standard grids. As a matter of fact, grid resolution near leading and trailing edges has a significant impact on the resulting force coefficients and in real life computations mesh agglomeration should be driven by *a posteriori* error estimators.

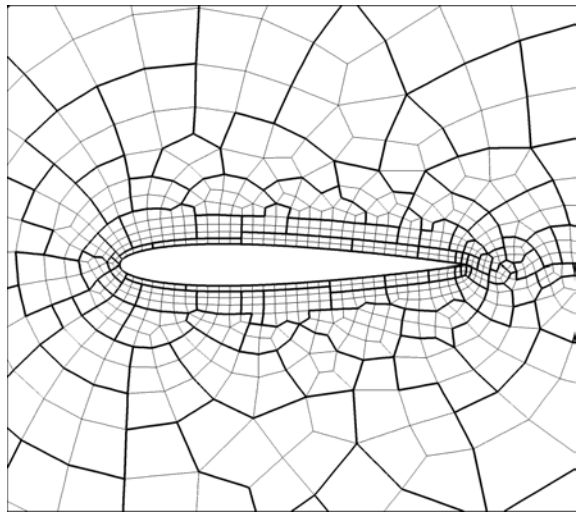
5.4 Conclusions

In this chapter an alternative approach to high-order mesh generation has been proposed and assessed by means of several test cases. Once a fine grid that guarantees a reliable representation of the geometries is available, a

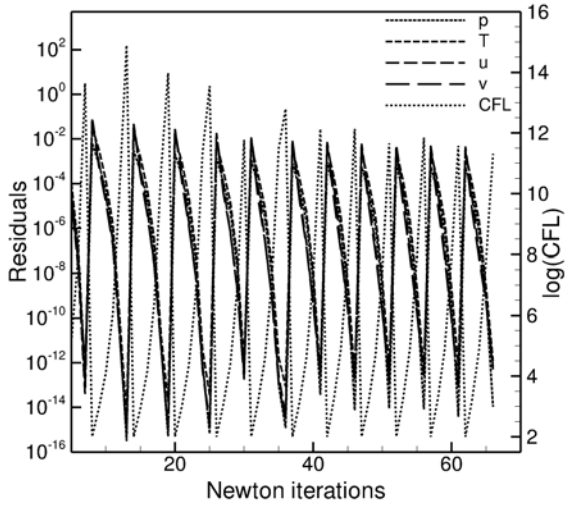
	Agglomerated grid 178 el.		Standard grid 1197 el.	
k	C_d	C_l	C_d	C_l
1	0.673659	0.628496	0.660408	0.570540
2	0.679781	0.592885	0.671731	0.556436
3	0.674850	0.571296	0.673134	0.555534
4	0.674957	0.560756	0.673192	0.555497
5	0.676370	0.559205	0.673155	0.555587
6	0.675686	0.558778	0.673158	0.555736
7	0.674577	0.557623		
8	0.673865	0.556783		
9	0.673537	0.556431		
10	0.673267	0.556242		
11	0.673349	0.556347		
12	0.673194	0.556219		

Table 5.1: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 0.8$, $Re_\infty = 73$. Values of drag and lift coefficients reported in Fig. 5.12.

coarse grid computationally affordable and suitable for a very high-order method can be obtained by means of an agglomeration strategy while keeping the boundary resolution of the fine grid, This technique can be considered convenient since the low-order meshes employed in the agglomeration process are easily obtained with current grid generators. Moreover, once the fine grid is defined, the agglomerated mesh can be changed very simply to fit into the available computational resources or to achieve a better error distribution (h-adaptivity), without modifying at all the geometry representation.

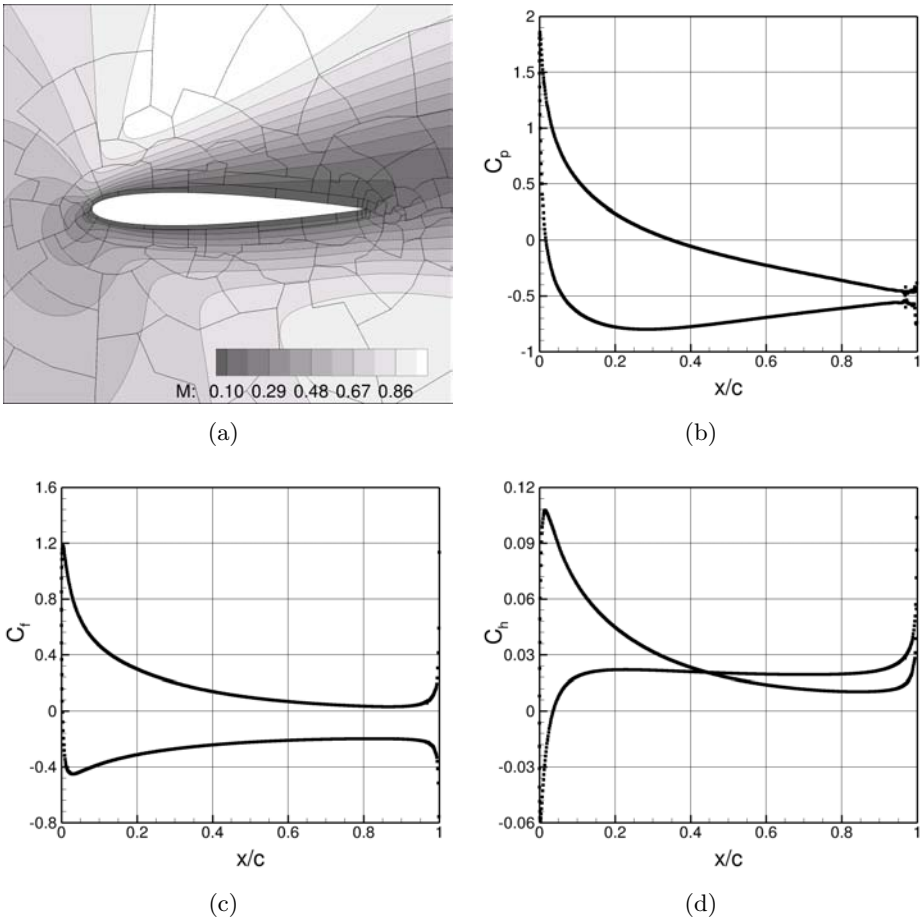


(a)



(b)

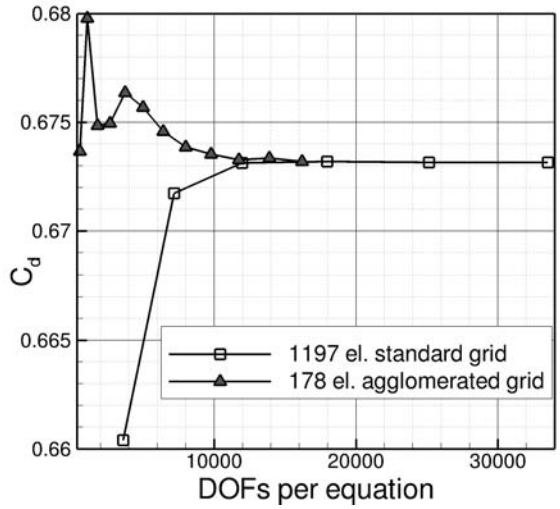
Figure 5.10: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 0.8$, $Re_\infty = 73$. 5.10(a): Agglomerated grid of 178 elements (thick lines) and standard hybrid grid of 1197 13-node quadrilateral and 10-node triangular elements (thin lines). 5.10(b): Residuals convergence history of $\mathbb{P}^{1\to12}$ solutions.



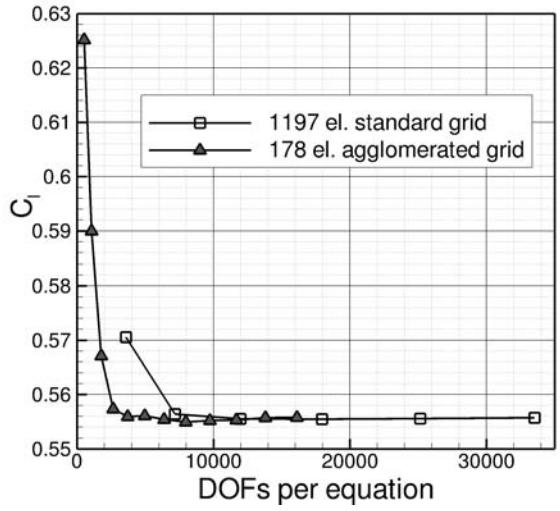
(a) (b)

(c) (d)

Figure 5.11: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 0.8$, $Re_\infty = 73$. $\mathbb{P}^{12}$ solution. 5.11(a): Mach number contours. 5.11(b): Pressure coefficient C_p . 5.11(c): Skin friction coefficient C_f . 5.11(d): Heat flux coefficient C_h .



(a)



(b)

Figure 5.12: NACA0012 airfoil, $\alpha_\infty = 10°$, $M_\infty = 0.8$, $Re_\infty = 73$. k-convergence of the aerodynamic coefficients on the standard and agglomerated grids of Figure 5.10. 5.12(a): Drag coefficient. 5.12(b): Lift coefficient.

Chapter 6

Grid adaptivity based on agglomeration

6.1 Introduction

The main goal of adaptivity is to maximize the solution accuracy for a given computational cost by tuning the grid density and/or the polynomial degree by means of a suitable error estimator. In this chapter a nonstandard approach to h-adaptivity based on agglomeration of a fine grid elements is investigated and its capabilities are validated through numerical tests, see also Bassi, Botti, Colombo, Di Pietro and Tesini [6].

6.2 Motivation of agglomeration-based adaptation

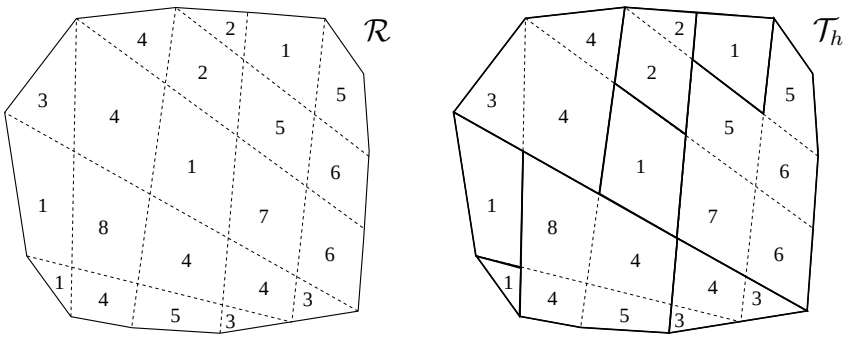
The interest in mesh adaptivity in the context of finite element discretizations has been strong since the pioneering work of Babuška and Suri [3]. The flexibility associated to DG methods allows to rely on many mesh modification strategies based on element subdivision, see *e.g.*, Hartmann and Leicht [34], or conforming grid modification, see *e.g.*, Dolejší and Felcman [23]. Here a different approach to adaptation, relying on the DG

discretization for arbitrarily shaped elements, is proposed. An h-adaptivity strategy based on agglomeration coarsening of a fine mesh can be attractive from the implementation viewpoint as the underlying mesh does not change, no nodes nor elements need to be dynamically added, moved or removed, and no hierarchic structures are required to track refinement levels.

6.3 The agglomeration algorithm

As mentioned in §2.2.2, general meshes can be obtained by means of standard agglomeration algorithms originally developed in the framework of geometric multigrid. In particular, this work relies on the well-known multilevel agglomeration library MGridGen [42, 43] developed by Moulitsas and Karypis. Let us remind that the cardinality of the agglomerated grid can be driven by means of a user-defined agglomeration factor $\mathcal{A}_c$ that specifies an upper bound for the number of sub-elements in an aggregated element. To drive the algorithm towards h-adaptivity, the procedure has been modified introducing the possibility to locally set the agglomeration factor. In practice the number of elements of the starting mesh $\mathcal{R}$ that concur in forming an element $T \quad \mathcal{T}_h$ is specified for each $E \quad \mathcal{R}$ providing an indication of $\mathrm{card}(\mathcal{R}_T)$ through the parameter $\mathcal{A}_c(E)$. Let us point out that, in practice, this requirement cannot be satisfied exactly for each element $T \quad \mathcal{T}_h$, and the agglomeration indication should be rather considered as an upper bound for $\mathrm{card}(\mathcal{R}_T)$. Figure 6.1 displays the distribution of the agglomeration factor $\mathcal{A}_c(E)$ on a starting fine grid $\mathcal{R}$ and a possible resulting aggregated grid $\mathcal{T}_h$. In an h-adaptivity context the actual distribution of the agglomeration factor must be related to the desired spatial distribution of accuracy of the computed solution.

The effectiveness of the above procedure to generate adapted grids is assessed by employing it in two simple approaches to adaptivity. The first approach, referred to as *error-based adaptivity*, involves the use of an exact estimator to drive the grid adaptation to obtain better error distribution over the domain. The second approach, referred to as *feature-based adaptivity*, aims at refining the mesh in proximity of shocks in order to improve



(a) Fine starting grid, $\mathcal{R}$. $\mathcal{A}_c(E)$ distribution.

(b) Agglomerated grid, $\mathcal{T}_h$

Figure 6.1: Example of a starting fine grid $\mathcal{R}$ with $\mathcal{A}_c(E)$ distribution superimposed and a coarse grid $\mathcal{T}_h$ obtained by means of the adaptive agglomeration process.

the resolution of discontinuities.

6.4 Error-based adaptivity

The devised h-adaptive strategy based on cells agglomeration is driven by the error in L^2-norm with respect to analytical solution of a model Poisson problem. Indeed, thanks to the definition of discrete polynomial spaces over arbitrarily shaped cells the grid can be easily adapted to the solution behavior.

To this end, for each element $E \quad \mathcal{R}$ an indication of the agglomeration factor $\mathcal{A}_c(E)$ representing a strict upper bound for the number of sub-elements in an aggregated element has to be given. The mesh obtained by means of the agglomeration process is hence such that, for all $T \quad \mathcal{T}_h$, there holds

$$\text{card}\,(R_T) \leq \mathcal{A}_c(E), \qquad E \quad \mathcal{R}_T. \tag{6.1}$$

Starting from a constant value over the fine mesh $\mathcal{R}$, the agglomeration

factor is modified during the refinement and coarsening processes according to the local error in the L^2-norm with respect to the analytical solution. The error is defined as follows

$$T \quad \mathcal{T}_h, \qquad e_T \stackrel{\text{def}}{=} \quad u - u_h \quad _{L^2(T)}, \tag{6.2}$$

$$T \quad \mathcal{T}_h, \quad E \quad R_T, \qquad e_E \stackrel{\text{def}}{=} \quad u - u_h \quad _{L^2(E)} + e_T. \tag{6.3}$$

The definition of e_E is intended to increase the probability that the element $E \quad \mathcal{R}$ with the largest error lies inside the aggregated element $T \quad \mathcal{T}_h$ with the largest error. The actual error metric is obtained by normalizing the above quantities

$$E \quad \mathcal{R}, \qquad \epsilon_E = \frac{e_E - \min_{E \in \mathcal{R}} e_E}{\max_{E \in \mathcal{R}} e_E - \min_{E \in \mathcal{R}} e_E}, \tag{6.4}$$

$$T \quad \mathcal{T}_h, \qquad \epsilon_T = \frac{e_T - \min_{T \in \mathcal{T}_h} e_T}{\max_{T \in \mathcal{T}_h} e_T - \min_{T \in \mathcal{T}_h} e_T}, \tag{6.5}$$

so that ϵ_E and ϵ_T are both contained in the interval $[0, 1]$.

Two different adaptation strategies borrowed from the open source library `libMesh` [39] are considered. The first one is best suited to optimize the discretization error while maintaining fixed computation requirements in terms of memory and execution time, while the second one can be used in place of global h-refinement in order to perform convergence studies and evaluate the discretization accuracy.

1st adaptation strategy Refinement and coarsening are performed according to two user-supplied parameters, say $\%_{\text{ref}}$ and $\%_{\text{coar}}$, corresponding to the relative errors triggering refinement and coarsening respectively. The normalized errors in each cell $T \quad \mathcal{T}_h$ and $E \quad \mathcal{R}_T$ are compared with the defined thresholds and, if necessary, adaptivity performed. At each

adaptation phase the agglomeration factors $\mathcal{A}_c(E)$, are then modified in accordance with the procedures below

1: Refinement
2: **for** $T \quad \mathcal{T}_h$ **do**
3: **if** $\epsilon_T \geq (1 - \%_{\text{ref}})$ **then**
4: **for** $E \quad \mathcal{R}_T$ **do**
5: **if** $\epsilon_E \geq (1 - \%_{\text{ref}})$ **then**
6: $\mathcal{A}_c(E) = \max\left(1, \frac{\text{card}(\mathcal{R}_T)}{2}\right)$
7: **end if**
8: **end for**
9: **end if**
10: **end for**

1: Coarsening
2: **for** $T \quad \mathcal{T}_h$ **do**
3: **if** $\epsilon_T \leq \%_{\text{coar}}$ **then**
4: **for** $E \quad \mathcal{R}_T$ **do**
5: **if** $\epsilon_E \leq \%_{\text{coar}}$ **then**
6: **if** $\mathcal{A}_c(E)/\text{card}(\mathcal{R}_T) \leq 2$ **then**
7: $\mathcal{A}_c(E) = 2\,\text{card}(\mathcal{R}_T)$
8: **end if**
9: **end if**
10: **end for**
11: **end if**
12: **end for**

In the coarsening process, line 6 accounts for the potential inability of the aggregation process to saturate the agglomeration factor, possibly resulting in $\text{card}(\mathcal{R}_T) \ll \mathcal{A}_c(E)$ with $E \quad \mathcal{R}_T$.

2nd adaptation strategy The approach presented below employs only a refinement phase. The process is performed on a fixed user-defined percentage $\%_{\text{ref}}$ of $\text{card}(\mathcal{T}_h)$. In practice, the elements are sorted in increasing order according to ϵ_T and numbered starting from zero to reflect the element position Pos_T in the sorted elements vector. Every time adaption is performed the refinement of touched elements is obtained by modifying the agglomeration factors $\mathcal{A}_c(E)$, $E \quad \mathcal{R}$, as follows:

1: **for** $T \quad \mathcal{T}_h$ **do**
2: **if** $\text{Pos}_T \geq (1 - \%_{\text{ref}})\,\text{card}(\mathcal{T}_h)$ **then**
3: $\mathcal{A}_c(E) = \max\left(1, \frac{\text{card}(\mathcal{R}_T)}{2}\right)$
4: **end if**
5: **end for**

Note that in both strategies, refinement and coarsening act on the real number of elements of the starting mesh $\mathcal{R}$ that concur in forming each element of $\mathcal{T}_h$ and not on the agglomeration factor $\mathcal{A}_c(E)$.

In real-life applications, when an analytical solution is not available, the above strategies can however be used by simply replacing the quantities ϵ_T, $T \quad \mathcal{T}_h$, and ϵ_E, $E \quad \mathcal{R}$, by suitable *a posteriori* error estimates.

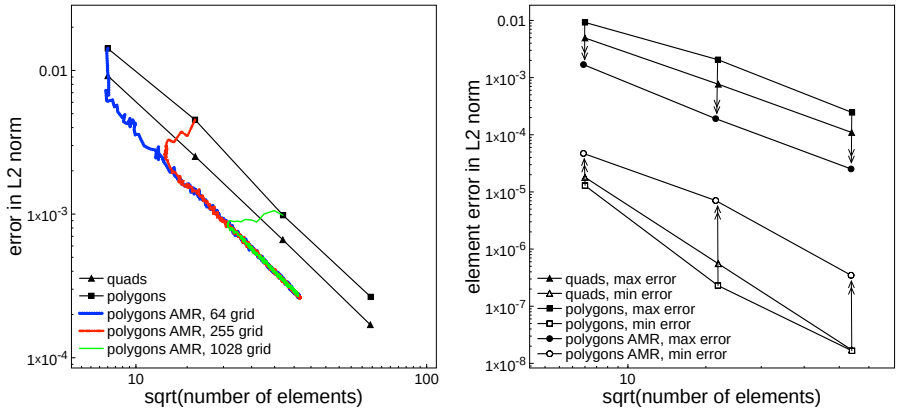
6.4.1 Numerical examples

The effectiveness of the h-adaptive strategy based on cells agglomeration is here assessed by comparing the resulting convergence rate with those of uniformly refined meshes. Convergence results for the BRMPS method outlined in §3.1 using the exact solution (3.16) and performing agglomeration-based adaptive mesh refinement (AMR) are reported. Approximations based on the space of piecewise affine functions, *i.e.*, $V_h = \mathbb{P}^1_d(\mathcal{T}_h)$ are here considered. Figure 6.2 displays the results obtained by applying the former adaptation strategy with $\%_{\text{ref}} = 0.1$ and $\%_{\text{coar}} = 0.15$ to the 64, 255 and 1028 polygonal elements grids in Figure 3.6. It is interesting to note that both $\text{card}(\mathcal{T}_h)$ and the L^2-norm of the error decrease in the first part of con-

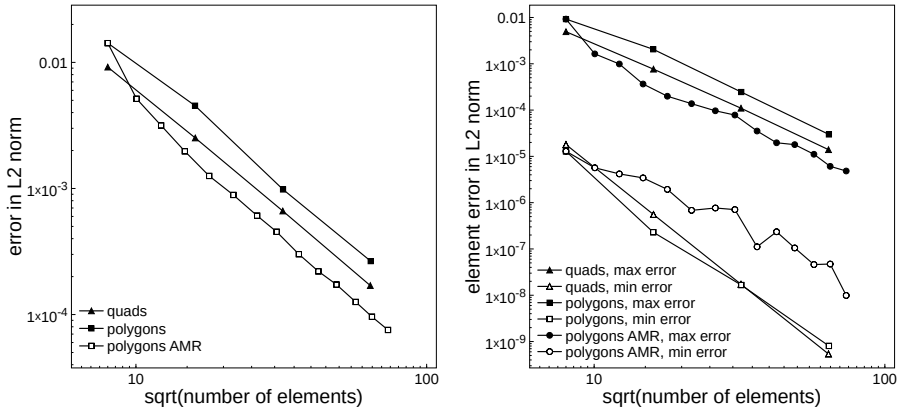
vergence thanks to the combined action of coarsening and refinement; then, once the error over the domain is redistributed, the AMR process maintain a second order convergence rate. Independently of the initial mesh, this strategy is able to drive the discretization towards analogue error values as the lines associated to different starting grids are perfectly overlapping once the same card($\mathcal{T}_h$) is reached. Moreover, the global L^2-norm of the error is always smaller on the grids resulting from adaptive h-refinement. The right panel of Figure 6.2 displays the values of $\max_{T\in\mathcal{T}_h} e_T$ and $\min_{T\in\mathcal{T}_h} e_T$ that would have been obtained stopping the refinement process as soon as card($\mathcal{T}_h$) is comparable with the number of elements in the starting mesh. The resulting h-adapted grids as well as the initial polygonal grids are shown in Figure 6.3. A comparison between uniform and adapted meshes with the same number of elements highlights the difference in the maximum and minimum error values, their location, and the better distribution of the error over the domain in the adaptive case. Concerning the second adaptation strategy the results shown in Figure 6.2 have been obtained by employing $\%_i\text{ref} = 0.17$ and the 64 polygonal elements grid displayed in Figure 3.6. The number of elements increases with a constant ratio allowing to quickly improve the discretization accuracy. Up to card($\mathcal{T}_h$) = 300, in the error redistribution phase, the first order discretization shows a third order convergence rate and, afterwards, the theoretical second order convergence is maintained. Although the error in L^2-norm as well as the maximum element error are higher than the ones obtained by employing the previous adaptation strategy, the benefits of h-adaptivity are still present in this raw and fast refinement algorithm.

6.5 Feature-based adaptivity

Feature-based adaptive mesh refinement is a simple and attractive strategy that aims at refining grid around some critical flow features whose resolution is crucial to the accuracy of computations results. In this work mesh adaptation is performed around shocks. For this purpose, a strategy to detect the troubled cells, namely those elements where discontinuities or steep

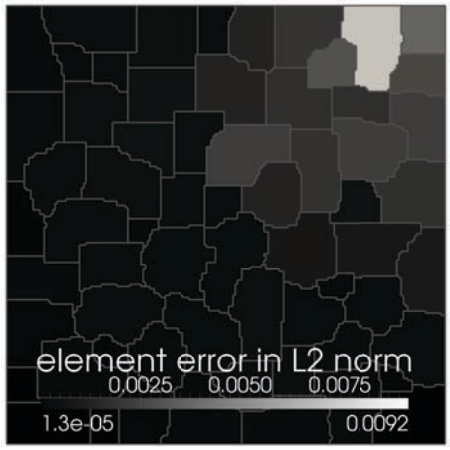


(a) 1st strategy

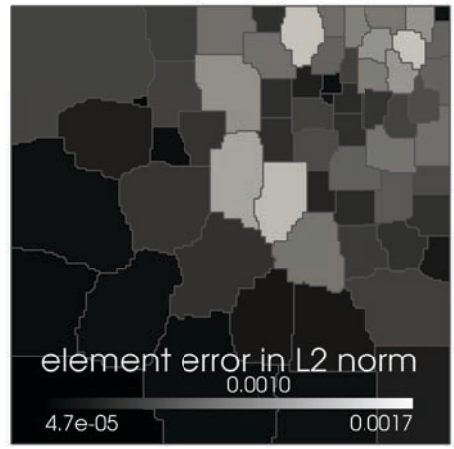


(b) 2nd strategy

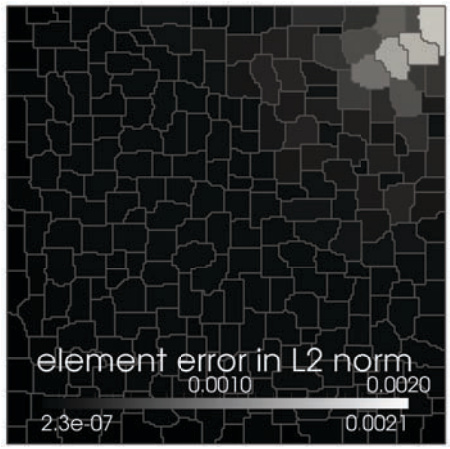
Figure 6.2: BRMPS solution for the test case of Karniadakis and Sherwin [38], $\mathbb{P}^1$. Comparison of the convergence rates using uniform Cartesian meshes, uniform polygonal meshes, and adaptively refined polygonal meshes. 6.2(a): Three distinct adaptive processes are considered starting from 64, 255, and 1028 polygonal elements. 6.2(b): Single adaptive process starting from 64 polygonal elements. *Left.* Error in L^2-norm. *Right.* Maximum and minimum error in L^2-norm computed over the mesh elements. The arrows indicate the action of AMR from starting polygonal grids to adapted grids with the same card($\mathcal{T}_h$).



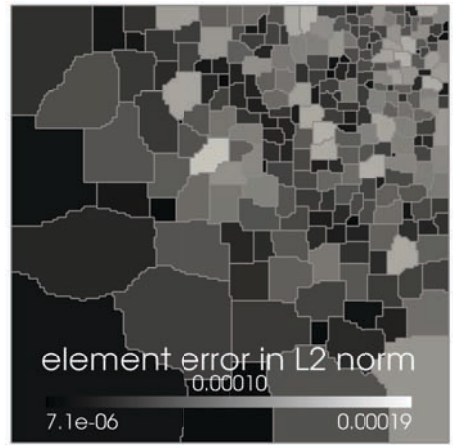
(a) $\mathrm{card}(\mathcal{T}_h) = 64$, uniform



(b) $\mathrm{card}(\mathcal{T}_h) = 63$, adaptive

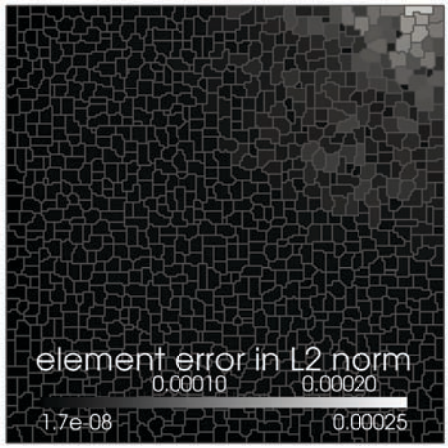


(c) $\mathrm{card}(\mathcal{T}_h) = 255$, uniform

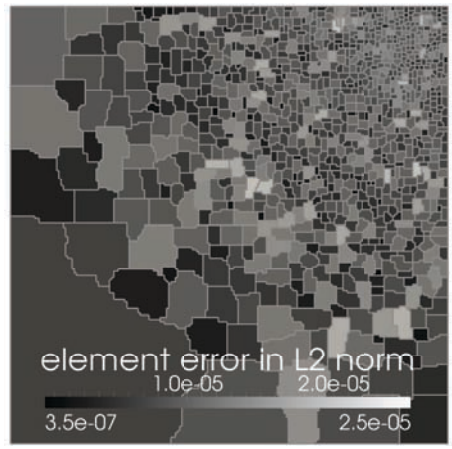


(d) $\mathrm{card}(\mathcal{T}_h) = 249$, adaptive

Figure 6.3: Polygonal grids on top of a 200×200 uniform quadrilateral grid, first degree polynomial approximation and distribution of the error in the L^2-norm, 1^{st} adaptation strategy.

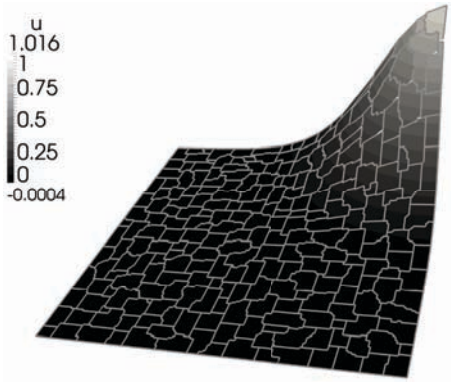


(e) card($\mathcal{T}_h$) = 1028, uniform

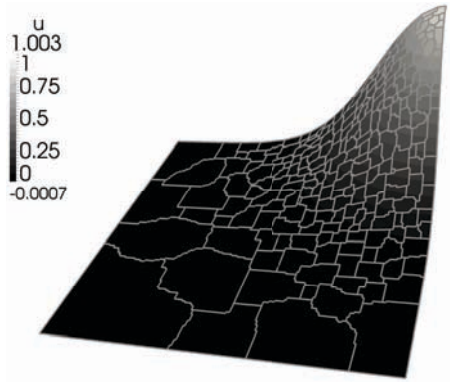


(f) card($\mathcal{T}_h$) = 1018, adaptive

Figure 6.3: Polygonal grids on top of a 200×200 uniform quadrilateral grid, first degree polynomial approximation and distribution of the error in the L^2-norm, 1$^{\text{st}}$ adaptation strategy.



(a) $\mathbb{P}^1$, 255 elements, uniform



(b) $\mathbb{P}^1$, 249 elements, adaptive

Figure 6.4: BRMPS solution for the test case of Karniadakis and Sherwin [38], 1$^{\text{st}}$ adaptation strategy.

gradients are present, is needed. Several so-called troubled cell detectors are available in literature. In this work the indicator adopted by Persson and Peraire in [45] is employed.

Discontinuity sensor and adaptation strategy

A suitable sensor for discontinuities is that adopted in [45], where within each element $T \quad \mathcal{T}_h$ the following indicator is defined

$$s_T = \frac{\int_T (u_h - \widehat{u}_h)^2 \, \mathrm{d}\mathbf{x}}{\int_T u_h^2 \, \mathrm{d}\mathbf{x}}, \tag{6.6}$$

where, in terms of hierarchical orthogonal basis functions, u_h represents the approximation of order k and $\widehat{u}_h$ is a truncated polynomial expansion, only containing terms up to order $k-1$

$$u_h = \sum_{i=1}^{N_{\mathrm{dof}}^k} = \mathbf{U}_i^T \varphi_i^T, \qquad \widehat{u}_h = \sum_{i=1}^{N_{\mathrm{dof}}^{k-1}} = \mathbf{U}_i^T \varphi_i^T. \tag{6.7}$$

The basic idea is that for smooth solutions the coefficients in the expansion are expected to decay very quickly, while in presence of discontinuities or steep gradients their strength dictates the rate of decay of the expansion coefficients. In the following pressure is the variable used in the shock detecting procedure. The shock sensor is then normalized so that $\tilde{s}_T$ $[0,1]$, *i.e.*,

$$T \quad \mathcal{T}_h, \qquad \tilde{s}_T = \frac{s_T - \min_{T \in \mathcal{T}_h} s_T}{\max_{T \in \mathcal{T}_h} s_T - \min_{T \in \mathcal{T}_h} s_T}. \tag{6.8}$$

Starting from a constant value over the fine mesh $\mathcal{R}$, the agglomeration factor $\mathcal{A}_c(E) \quad E \quad \mathcal{R}_T$, $T \quad \mathcal{T}_h$ is modified during the refinement process in relation with the local normalized shock sensor $\tilde{s}_T$ and a small user-supplied threshold $\tilde{s}_T^0$.

1: **for** $T \quad \mathcal{T}_h$ **do**
2: **if** $\tilde{s}_T \geq \tilde{s}_T^0$ **then**
3: **for** $E \quad \mathcal{R}_T$ **do**
4: $\mathcal{A}_c(E) = \max\left(1, \dfrac{\mathcal{A}_c(E)}{2}\right)$
5: **end for**
6: **end if**
7: **end for**

The h-refinement procedure takes place every time a prescribed number of iteration it_k or a given threshold on the lowest L^2-norm of the equations residuals res_t is reached.

6.5.1 Numerical examples

NACA0012, $\alpha_\infty = 1.25°$**,** $\mathbf{M}_\infty = 0.8$

This test case aims at demonstrating the capability of this feature-based refinement process by computing the inviscid transonic flow around the NACA0012 airfoil at an angle of attack 1.25° and freestream Mach number of 0.8. This configuration gives rise to a steady solution with a strong shock on the suction side and a weak one on the pressure side, see *e.g.*, [51]. The starting grid is an uniformly agglomerated mesh of 284 arbitrarily shaped elements, built on top of a mesh that consists of 8192 10-node triangular elements, see Figure 6.5. The results presented in the following have been obtained by employing $\text{res}_t = 1e-3$, $\text{it}_k = 100$ and $\tilde{s}_T^0 = 2.5e-2$. Solution has been carried out up to $\mathbb{P}^6$ polynomial degree and residuals convergence history is shown in Figure 6.5. All the computations have been performed in parallel on four partition, starting higher-order solutions from the lower-order ones. Moreover, the agglomerated adapted grid of the low-order solution has been used as starting grid for the refinement process of the next higher-order computation. Solutions have been advanced in time using the linearly implicit backward Euler method and the resulting linear system has been solved using the default solver available in PETSc, *i.e.*,

the restarted preconditioned GMRES algorithm. Figure 6.6 displays the solution on the starting uniformly aggregated grid, and the first two steps of the adaptation process. Moreover, the pressure coefficient distribution C_p and the normalized shock sensor contours are reported. Note that, with $\mathbb{P}^1$ approximation the shock detector and the given threshold $\tilde{s}_T^0$ are not able to detect the weak shock on the pressure side of the airfoil. When the polynomial approximation is raised both shocks in the flow field are recognized and well resolved on adapted grids as reported in Figure 6.7.

6.6 Conclusions

A nonstandard approach to h-adaptivity based on agglomeration coarsening of a fine underlying mesh has been shown and validated by means of different numerical examples. This technique has advantages that makes it an attractive alternative to other adaptations strategies. Indeed, once a suitable underlying fine grid is available, the agglomerated mesh density can be tuned to maximize the solution accuracy for a given computational cost while keeping the boundary resolution of the fine grid. Therefore, h-adaptivity could be performed without the need to improve the accuracy of domain discretization during the adaption steps. From an implementation viewpoint since h-adaptive agglomeration coarsening relies on a fixed underlying mesh it does not need to dynamically add, move or remove nodes or elements and it avoids any hierarchical structures to track refinement levels. Although this work does not deal with error estimate, the devised strategy to handle mesh via adaptive agglomeration can be trivially extended to employ any suitable *a posteriori* error estimator that would be considered more appropriate to real-life applications, see *e.g.*, Hartmann and Leicht [34], Hartmann, Held, Leicht and Prill [32] and Wang and Mavriplis [53].

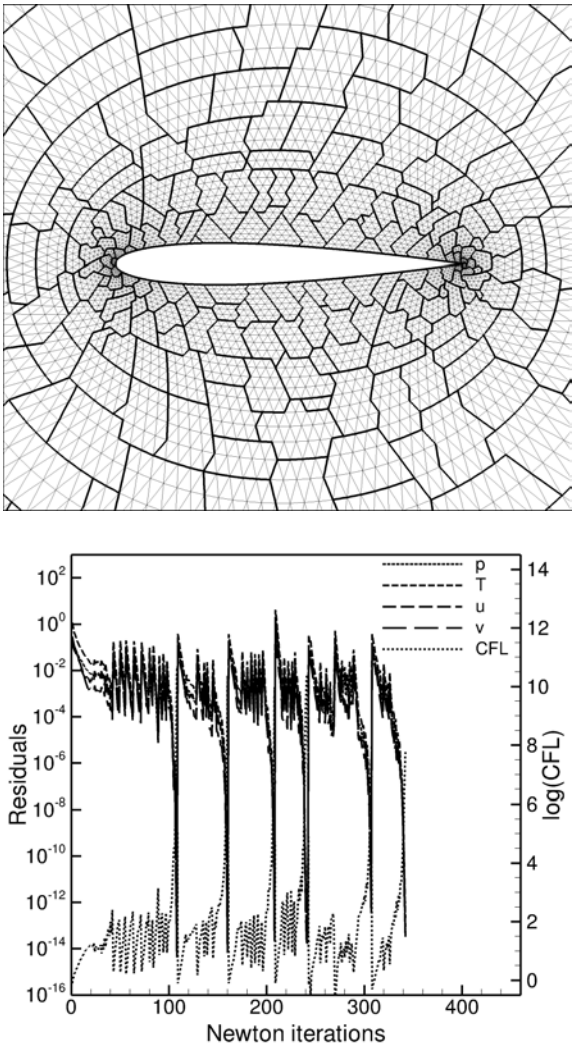
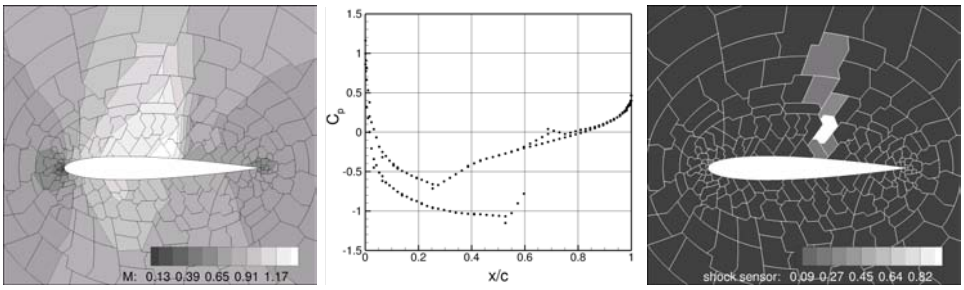
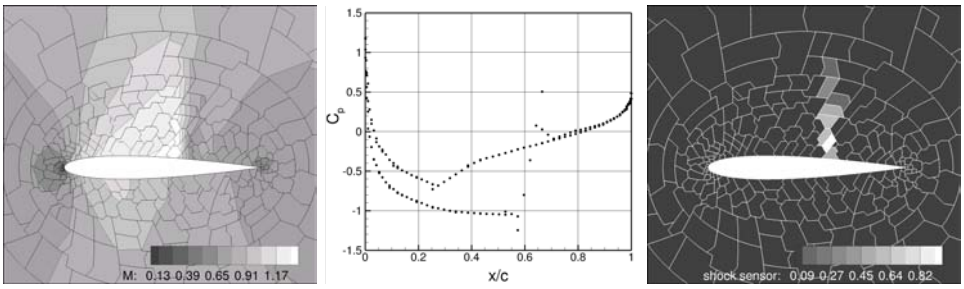


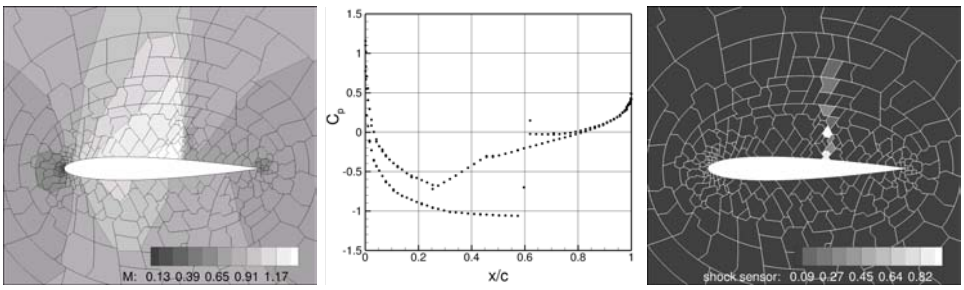
Figure 6.5: NACA0012 $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Starting uniformly agglomerated grid of 284 arbitrarily shaped elements and residual convergence history $\mathbb{P}^{1\to6}$.



(a) *step 0*: starting uniformly agglomerated grid, 284 elements



(b) *step 1*: first adapted grid, 314 elements



(c) *step 2*: second adapted grid, 347 elements

Figure 6.6: NACA0012 airfoil $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Adaptation process towards $\mathbb{P}^1$ solution. Mach contours, pressure coefficient C_p distribution and shock sensor $\tilde{s}_T$ contours (*left/right*). Adaptation step 0 2 (*top/bottom*).

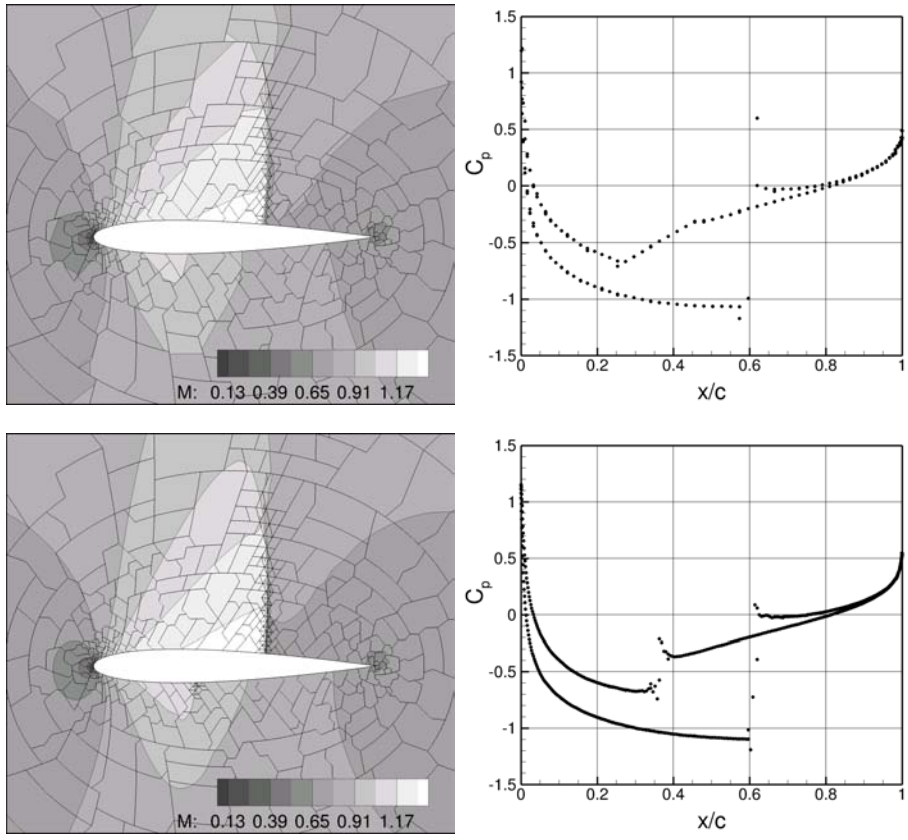


Figure 6.7: NACA0012 airfoil $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Solutions converged up to machine accuracy on adapted grids. Mach contours, pressure coefficient C_p (*left/right*). $\mathbb{P}^1$ 417 elements, $\mathbb{P}^4$ 452 elements solutions (*top/bottom*).

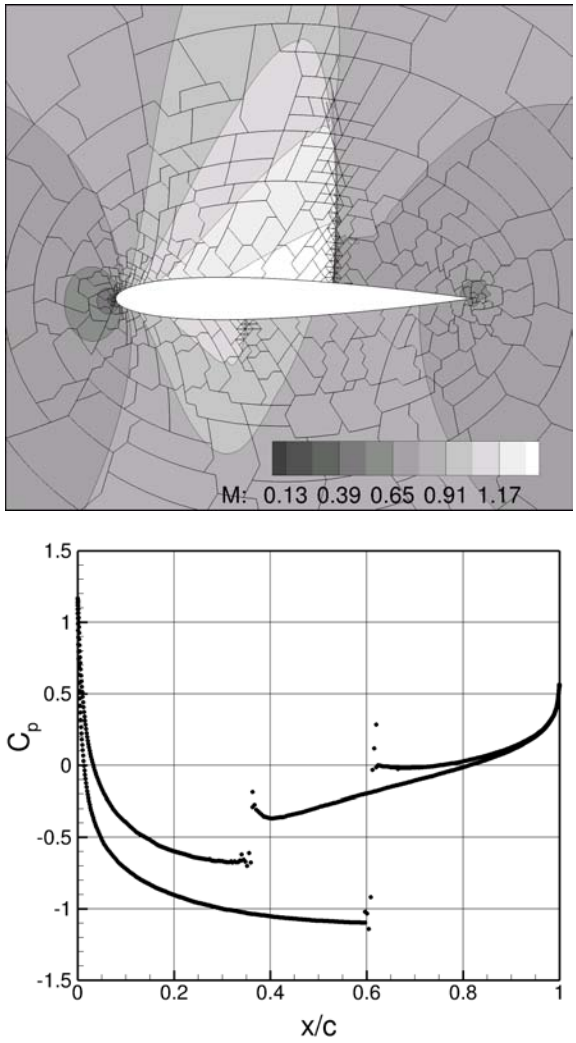


Figure 6.7: NACA0012 airfoil $\alpha_\infty = 1.25°$, $M_\infty = 0.8$. Solutions converged up to machine accuracy on adapted grid. Mach contours, pressure coefficient C_p (*left/right*). $\mathbb{P}^6$ 459 elements solutions (*top/bottom*).

Chapter 7

Conclusion and future work

In this work the BRMPS method has been extended to arbitrarily shaped elements proposing and validating two different implementations for grids that consist of agglomerated cells. Afterwards, the capabilities and robustness of an implicit DG method on elements of general shape have been assessed by solving the compressible Navier-Stokes and RANS+k-ω equations on classical aerodynamics test cases. The use of agglomerated meshes gives a high flexibility in handling the grid that actually becomes a degree of freedom of the computation. This flexibility has been exploited to propose an alternative approach to high-order mesh generation that relies on the agglomeration of fine underlying grids that guarantee an accurate discretization of the domain. This technique can be considered convenient since the meshes employed for the agglomeration process are easily obtained by current low-order grid generators. Moreover, once the fine grid is defined, the agglomerated mesh can be changed very simply to fit into the available computational resources or to achieve a better error distribution (h-adaptivity), without modifying at all the geometry representation. A nonstandard approach to h-adaptivity based on adaptive coarsening of a fine underlying mesh has been presented and its feasibility demonstrated by means of different strategies and numerical examples.

The main drawback of DG implementations that rely on arbitrarily shaped elements can be ascribed to the cost of quadrature rules. Indeed, when integrals on polymorphic elements are computed as the sum of the integrals on standard sub-elements the computational effort due to integration can easily become very high. To this end, future work will be devoted to the development of more efficient quadrature rules for elements of general shape. On the side of mesh adaptation, the devised strategies to handle mesh via adaptive agglomeration can be trivially extended to employ any suitable *a posteriori* error estimator that would be considered appropriate to real-life applications. In this regard, the implementation of an *adjoint-based* error estimator to perform goal-oriented mesh refinement will be considered, *e.g.*, Hartmann and Leicht [34], Hartmann, Held and Leicht [31] and Wang and Mavriplis [53].

Bibliography

[1] AGARD. Test cases for inviscid flow field methods. Tech. rep., AGARD-AR-221, Ancelle 92200, Neuilly sur Seine, France, 1985.

[2] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal. 39*, 5 (2002), 1749–1779.

[3] Babuška, I., and Suri, M. The *h-p* version of the finite element method with quasiuniform meshes. *Math. Mod. Numer. Anal. 21* (1987), 199–238.

[4] Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[5] Bassi, F., Botti, L., Colombo, A., Crivellini, A., Franchina, N., Ghidoni, A., and Rebay, S. Very high-order accurate discontinuous Galerkin computation of transonic turbulent flows on aeronautical configurations. In *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sørensen, Eds., vol. 113 of *Notes on Nu-*

merical Fluid Mechanics and Multidisciplinary Design. Springer Berlin / Heidelberg, 2010, pp. 25–38.

[6] Bassi, F., Botti, L., Colombo, A., Di Pietro, D. A., and Tesini, P. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *Submitted to Journal of Computational Physics* (2011). http://hal.archives-ouvertes.fr/hal-00562219/en/.

[7] Bassi, F., Botti, L., Colombo, A., and Rebay, S. Agglomeration based discontinuous Galerkin discretization for the Euler and Navier–Stokes equations. *Submitted to Computers & Fluids* (2011).

[8] Bassi, F., Botti, L., Colombo, N., Ghidoni, A., and Rebay, S. Discontinuous Galerkin for turbulent flows. In *Adaptive high-order methods in computational fluid dynamics*, Z. J. Wang, Ed., vol. 2 of *Advances in Computational Fluid Dynamics*. World Scientific, Spring 2011.

[9] Bassi, F., and Crivellini, A. A three-dimensional parallel discontinuous Galerkin solver for acoustic propagation studies. *International Journal of Aeroacoustics 2* (April 2003), 157–173.

[10] Bassi, F., Crivellini, A., Di Pietro, D. A., and Rebay, S. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations. *J. Comput. Phys. 218* (2006), 794–815.

[11] Bassi, F., Crivellini, A., Rebay, S., and Savini, M. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations. *Comput. & Fluids 34* (2005), 507–540.

[12] Bassi, F., and Rebay, S. Accurate 2D Euler computations by means of a high order discontinuous finite element method. vol. 453 of *Lecture Notes in Physics*, Springer, pp. 234–240. XIV ICNMFD, Bangalore, India, 11–15 July 1994.

[13] Bassi, F., and Rebay, S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys. 131* (1997), 267–279.

[14] Bassi, F., and Rebay, S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys. 138* (1997), 251–285.

[15] Bassi, F., and Rebay, S. A high order discontinuous Galerkin method for compressible turbulent flows. In *Discontinuous Galerkin Methods. Theory, Computation and Applications* (2000), vol. 11 of *Lecture Notes in Computational Science and Engeneering*, Springer-Verlag, pp. 77–88. *First International Symposium on Discontinuous Galerkin Methods,* May 24–26, 1999, Newport, RI, USA.

[16] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., and Savini, M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In *Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics* (Antwerpen, Belgium, March 5–7 1997), R. Decuypere and G. Dibelius, Eds., Technologisch Instituut, pp. 99–108.

[17] Brezzi, F., Manzini, G., Marini, D., Pietra, P., and Russo, A. Discontinuous Galerkin approximations for elliptic problems. *Numer. Methods Partial Differential Equations 16* (2000), 365–378.

[18] Cockburn, B., Hou, S., and Shu, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54 (1990), 545–581.

[19] Cockburn, B., Lin, S., and Shu, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws III: One dimensional systems. *J. Comput. Phys.*, 84 (1989), 90–113.

[20] Cockburn, B., and Shu, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework. *Math. Comp.*, 52 (1989), 411–435.

[21] Di Pietro, D. A., and Ern, A. *Mathematical Aspects of Discontinuous Galerkin Methods.* Maths & Applications. Springer-Verlag, 2011. In press.

[22] Dolejší, V. Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. *Commun. Comput. Phys. 4* (2008), 231–274.

[23] Dolejší, V., and Felcman, J. Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numerical Methods for Partial Differential Equations 20* (2004), 576–608.

[24] GAMM. Numerical simulation of compressible Navier–Stokes equations – External 2D flows around a NACA0012 airfoil. In *GAMM Workshop.* (Edt. INRIA, Centre de Rocquefort, de Rennes et de Sophia-Antipolis, 1986), Nice, France, December 4–6 1985, pp. 247–254.

[25] Gassner, G. J., Lörcher, F., and Munz, C.-D. A discontinuous Galerkin scheme based on a space-time expansion II. Viscous flow equations in multi dimensions. *J. Sci. Comput. 34* (2008), 260–286.

[26] Gassner, G. J., Lörcher, F., Munz, C.-D., and Hesthaven, J. S. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comput. Phys. 228* (2009), 1573–1590.

[27] Gerolymos, G. A., and Vallet, I. Implicit computation of the three-dimensional compressible Navier–Stokes equations. *AIAA Journal 34*, 7 (July 1996), 1321–1330.

[28] Giraud, L., Langou, J., and Rozloznik, M. On the loss of orthogonality in the Gram-Schmidt orthogonalization process. Technical Report No. TR/PA/03/25, CERFACS, 2003.

[29] Gobbert, K., and Yang, S. Numerical demonstration of finite element convergence for Lagrange elements in COMSOL Multiphysics. In *Proceedings of the COMSOL Conference 2008 Boston* (2008).

[30] Hänel, D., Schwane, R., and Seider, G. On the accuracy of upwind schemes for the solution of the Navier–Stokes equations. AIAA Paper 87-1105 CP, AIAA, July 1987. Proceedings of the AIAA 8th Computational Fluid Dynamics Conference.

[31] Hartmann, R., Held, J., and Leicht, T. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations. *Journal of Computational Physics In Press, Corrected Proof* (2010), –.

[32] Hartmann, R., Held, J., Leicht, T., and Prill, F. Discontinuous Galerkin Methods for Computational Aerodynamics - 3D Adaptive Flow Simulation with the DLR PADGE Code. *Aerospace Sci. Tech. 14* (2010), 512–519.

[33] Hartmann, R., and Houston, P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys. 183* (2002), 508–532.

[34] Hartmann, R., and Leicht, T. Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations. *J. Comput. Phys. 229* (2010), 7344–7360.

[35] Hellsten, A. On the solid-wall boundary condition of ω in the k-ω - type turbulence models. Technical Report B–50, Helsinky University of Technology, Laboratory of Aerodynamics, 1998.

[36] Hughes, T., Cottrell, J., and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering 194*, 39-41 (2005), 4135 – 4195.

[37] Jaffre, J., Johnson, C., and Szepessy, A. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Math. Models Methods Appl. Sci. 5*, 3 (1995), 367–386.

[38] Karniadakis, G. E., and Sherwin, S. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scietific Computation. Oxford University Press, USA, 2005.

[39] Kirk, B., Peterson, J. W., Stogner, R. H., and Carey, G. F. libmesh: A c++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng. Comput. 22* (2006), 237–254.

[40] Krivodonova, L., and Berger, M. High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics 211*, 2 (2006), 492 – 512.

[41] Menter, F. R. Two-equation eddy-viscosity turbulence model for engineering applications. *AIAA Journal 32*, 8 (August 1994), 1598–1605.

[42] Moulitsas, I., and Karypis, G. MGridGen/ParmGridGen, Serial/Parallel library for generating coase meshes for multigrid methods. Technical Report Version 1.0, University of Minnesota, Department of Computer Science/Army HPC Research Center, 2001. http://www-users.cs.umn.edu/∼moulitsa/software.html.

[43] Moulitsas, I., and Karypis, G. Multilevel algorithms for generating coarse grids for multigrid methods. In *Supercomputing 2001 Conference Proceedings* (2001).

[44] Mousavi, S., Xiao, H., and Sukumar, N. Generalized Gaussian quadrature rules on arbitrary polygons. *Int. J. Numer. Meth. Engng. 00* (2009), 1–26.

[45] Persson, P., and Peraire, J. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit* (January 2006), no. AIAA-2006-112.

[46] Qiu, B., and Shu, C.-W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case. *J. Comput. Phys. 193* (2005), 115–135.

[47] Qiu, B., and Shu, C.-W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: two dimensional case. *Comput. & Fluids 34* (2005), 642–663.

[48] Qiu, B., and Shu, C.-W. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM J. Sci. Comput. 26* (2005), 907–929.

[49] Rashid, M. M., and Selimotic, M. A three-dimensional finite element method with arbitrary polyhedral elements. *Int. J. Numer. Meth. Engng. 67* (2006), 226–252.

[50] Sevilla, R., Fernández-Méndez, S., and Huerta, A. NURBS-enhanced finite element method for Euler equations. *International Journal for Numerical Methods in Fluids 57* (July 2008), 1051–1069.

[51] Taube, A., and Munz, C.-D. Synthesis report on shock capturing strategies. In *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sørensen, Eds., vol. 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer Berlin / Heidelberg, 2010, pp. 195–207.

[52] Tesini, P. *An* h*-Multigrid Approach for High-Order Discontinuous Galerkin Methods*. PhD thesis, Università degli Studi di Bergamo, January 2008.

[53] Wang, L., and Mavriplis, D. J. Adjoint-based *h-p* adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *J. Comput. Phys. 228* (November 2009), 7643–7661.

[54] Wilcox, D. C. *Turbulence Modelling for CFD*. DCW industries Inc., La Cañada, CA 91011, USA, 1993.

[55] Xiao, H., and Gimbutas, Z. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Comput. Math. Appl. 59* (January 2010), 663–676.

Index

ISBN: 978-88-97413-01-1