POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Fluidodinamica – XXVII ciclo

Tesi di Dottorato

Computational fluid dynamics for aerospace propulsion systems: an approach based on discontinuous finite elements

Andrea Ferrero
matricola: 189152

Tutor
prof. Francesco Larocca

Coordinatore del corso di dottorato
prof. Gaetano Iuso

December 2014

Summary

The purpose of this work is the development of a numerical tool devoted to the study of the flow field in the components of aerospace propulsion systems. The goal is to obtain a code which can efficiently deal with both steady and unsteady problems, even in the presence of complex geometries.

Several physical models have been implemented and tested, starting from Euler equations up to a three equations RANS model. Numerical results have been compared with experimental data for several real life applications in order to understand the range of applicability of the code. Performance optimization has been considered with particular care thanks to the participation to two international Workshops in which the results were compared with other groups from all over the world.

As far as the numerical aspect is concerned, state-of-art algorithms have been implemented in order to make the tool competitive with respect to existing softwares. The features of the chosen discretization have been exploited to develop adaptive algorithms (p, h and hp adaptivity) which can automatically refine the discretization. Furthermore, two new algorithms have been developed during the research activity. In particular, a new technique (Feedback filtering [1]) for shock capturing in the framework of Discontinuous Galerkin methods has been introduced. It is based on an adaptive filter and can be efficiently used with explicit time integration schemes. Furthermore, a new method (Enhance Stability Recovery [2]) for the computation of diffusive fluxes in Discontinuous Galerkin discretizations has been developed. It derives from the original recovery approach proposed by van Leer and Nomura [3] in 2005 but it uses a different recovery basis and a different approach for the imposition of Dirichlet boundary conditions. The performed numerical comparisons showed that the ESR method has a larger stability limit in explicit time integration with respect to other existing methods (BR2 [4] and original recovery [3]). In conclusion, several well known test cases were studied in order to evaluate the behavior of the implemented physical models and the performance of the developed numerical schemes.

Acknowledgements

I'm very grateful to my supervisor, Professor Francesco Larocca for all the time he dedicated to my doubts and questions.
I would like to thank my family for the support and Francesca for her patience and comprehension.

Contents

Summary III

Acknowledgements V

I A numerical approach to aerospace propulsion **1**

1 Introduction 3
- 1.1 CFD for aerospace propulsion systems 3
- 1.2 Motivations and description of the work 4

2 Physical models 9
- 2.1 Euler equations 9
- 2.2 Navier-Stokes equations 10
- 2.3 Reynolds Averaged Navier-Stokes equations 11
  - 2.3.1 Spalart-Allmaras model 12
  - 2.3.2 Wilcox $k-\omega$ model 14
  - 2.3.3 Laminar Kinetic Energy model 20
- 2.4 Boundary conditions 22
  - 2.4.1 BCs for Euler equations 23
  - 2.4.2 BCs for Navier-Stokes equations 23
  - 2.4.3 BCs for Spalart-Allmaras model 24
  - 2.4.4 BCs for Wilcox $k-\omega$ model 24
  - 2.4.5 BCs for LKE model 26

II Discretization **27**

3 Discontinuous Galerkin spatial discretization 29
- 3.1 Variational approach 30
- 3.2 Basis functions 31

3.3 Mappings 33
3.3.1 Quadrilaterals 33
3.3.2 Triangles 34
3.4 Choice of the basis 35
3.5 Convective fluxes 38

4 Diffusive fluxes 41
4.1 BR2 method 43
4.2 Original recovery method 44
4.3 Enhanced stability recovery method 46
4.3.1 Derivation of the method 46
4.3.2 Extension to 3D 52
4.3.3 BCs for ESR 52
4.3.4 Dirichlet BC's in strong form 53
4.3.5 Dirichlet BCs in weak form 55
4.3.6 Implementation and computational cost 56
4.3.7 Interface integral computational cost 56
4.3.8 Volume integral computational cost 57
4.4 Tests and comparisons for the ESR method 58
4.4.1 1D Poisson equation 58
4.4.2 1D Convection Diffusion problem 59
4.4.3 2D Heat equation 60
4.4.4 Laminar boundary layer on a flat plate 63
4.5 Considerations on implicit time integration 65

5 Time discretization 67
5.1 Explicit Runge-Kutta methods 68
5.2 Implicit methods 69
5.2.1 Jacobian evaluation 70
5.2.2 Iterative solver 71
5.2.3 CFL evolution laws for steady problems 71

6 Shock capturing 73
6.1 Feedback filtering 74
6.1.1 Exponential filter 74
6.1.2 Smoothness indicator 75
6.1.3 Feedback filtering 76
6.1.4 Sod problem 78
6.1.5 Shock-bubble interaction 82
6.1.6 Ringleb flow 85
6.1.7 Transonic NACA0012 86
6.2 Artificial viscosity 90

6.2.1 Future works 90

7 Adaptive algorithms 93
7.1 Sensors and error indicators 94
7.2 p-adaptive algorithm 95
7.2.1 Heat equation and *p*-adaptivity 96
7.2.2 Vortex shedding around a circular cylinder 99
7.3 h-adaptive algorithm 102
7.3.1 Transonic NACA0012: h-adaptivity 103
7.4 hp-adaptive algorithm 108

III Numerical results **111**

8 Performances 113
8.1 Acceleration for steady problems 113
8.2 Workshop test problems 116
8.2.1 Subsonic flow in a channel with a smooth bump 116
8.2.2 Transonic Ringleb flow 119
8.2.3 Vortex transport by uniform flow 123
8.2.4 Unsteady viscous flow over tandem airfoils 128

9 Aerospace propulsion examples and validation 131
9.1 VKI-LS59 turbine cascade 131
9.2 T106c turbine cascade 132

10 Conclusions 141

A Adimensionalization 143

B Normalization of performances 145

C High-order visualizations 147

D Meshes 149

E List of symbols 151

Bibliography 155

List of Tables

4.1 Condition number of the matrix $\left([\mathbf{R}]^T[\mathbf{R}]\right)$ for the interface of Figure 4.4 without and with orthonormalization (ON) of the recovery basis 51
4.2 1D Poisson problem: stability limit (p=1, forward Euler) 59
4.3 1D Poisson problem: L^2 errors and convergence order (p=1) 59
4.4 1D Convection-diffusion problem: L^2 errors at steady state (p=1) . 60
4.5 L^2-norm of the temperature error for the ESR and the BR2 methods. 62
4.6 Stability limit with explicit time integration for 2D heat diffusion on unstructured meshes. 62
4.7 Drag coefficient (C_D) for flate plate problem ($Re_\infty = 10^6, M_\infty = 0.5 - 0.2$). 65
6.1 Shock-bubble problem: initial data. 82
6.2 Ringleb flow: entropy accuracy test for feedback filtered DG methods. 86
6.3 Ringleb flow: pressure accuracy test for feedback filtered DG methods. 87
7.1 Average drag coefficient and Strouhal number for cylinder flow ($Re_\infty = 10^2, M_\infty = 0.2$). 100
7.2 Experimental results from the literature for cylinder flow ($Re_\infty = 10^2$) 100

List of Figures

2.1 Velocity profile in turbulent boundary layer obtained by the proposed discretization with the Spalart-Allmaras model 15
2.2 Skin friction coefficient for turbulent flat plate: comparison between the implemented Spalart-Allmaras model and correlations from the literature 15
2.3 Skin friction coefficient on flat plate: transition from laminar to turbulent. Blue=Blasius solution, Light blue=turbulent correlation, Green= implemented Wilcox $k - \omega$ low Re model 19
3.1 Serendipity mappings for quadrilateral elements. 33
3.2 Mach field for the inviscid subsonic flow in a channel. 36
3.3 Unstructured mesh with cubic quadrilateral elements. 37
3.4 Unstructured mesh with cubic triangular elements. 37
3.5 Entropy error vs equivalent length scale for the inviscid channel flow (p=2) 37
3.6 Entropy error vs reconstruction order for Osher and Local Lax-Friedrichs (LLF) fluxes. 39
4.1 Interface reference frame for constructing the recovery basis. 48
4.2 Preliminary recovery basis before orthonormalization for the case $p_A = p_B = 2$, in the Enhanced Recovery approach. 48
4.3 Preliminary recovery basis with the original recovery approach for the case $p_A = p_B = 2$. 48
4.4 Interface between two distorted elements for the test of Table 4.1. . 51
4.5 Exact temperature field for the 2D heat diffusion problem. 61
4.6 L^2-norm of the temperature error vs the equivalent length scale h. 63
4.7 Wall friction coefficient (c_f) for the flat plate ($Re_\infty = 10^6, M_\infty = 0.2$) 64
4.8 Adimensionalized L^2-norm of the residual vs computational time [s]. 66
6.1 Adimensionalized L^2-norm of the residual vs computational time [s]. 80
6.2 Adimensionalized L^2-norm of the residual vs computational time [s]. 80
6.3 Detail of the contact surface in the Sod problem (Feedback filtering). 81
6.4 Detail of the shock in the Sod problem (Feedback filtering). 81
6.5 Domain and initial condition for shock-bubble interaction 82

6.6 Numerical Schlieren images for shock-bubble interaction at $t = 0.071$, $t = 0.21$ and $t = 0.33$ (Feedback filtered DG2 scheme). 83
6.7 Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme). 84
6.8 Ringleb flow: constant speed lines (dashed) and computational domain Ω. 85
6.9 Mach field on the NACA0012 airfoil ($M_\infty = 0.8$,$\alpha_\infty = 1.25^o$): feedback filtered DG1 scheme. 88
6.10 Mach field on the NACA0012 airfoil ($M_\infty = 0.8$,$\alpha_\infty = 1.25^o$): feedback filtered DG5 scheme. 88
6.11 Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme). 89
6.12 Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme). 89
7.1 Temperature field for the 2D heat equation problem, with $k_w = 5$. 96
7.2 Example of order distribution with p-adaptivity and the 8x8x2 mesh ($\alpha_p = 0.3$) 97
7.3 L^2-norm of the temperature error vs total number of DOF's (possible orders: $p_e = 1,2,3,4,5$). 97
7.4 L^2-norm of the temperature error vs total number of DOF's (possible orders: $p_e = 1,3,5$) 98
7.5 Distribution of the reconstruction order with the entropy sensor. . 101
7.6 Distribution of the reconstruction order with the sensor based on velocity gradients. 101
7.7 Distribution of the reconstruction order using both the entropy and the velocity gradient sensors. 102
7.8 Entropy field around a circular cylinder with p-adaptivity ($Re_\infty = 10^2$). 102
7.9 Isotropic splitting on unstructured meshes: initial (black) and adapted (red) meshes. 103
7.10 Initial mesh (1060 elements). 106
7.11 Adapted mesh after 5 h-refinements (11806 elements). 106
7.12 Transonic NACA0012: Mach field on the initial mesh (p=1). . . . 107
7.13 Transonic NACA0012: Mach field on the adapted mesh after 5 h-refinements (p=1). 107
7.14 Computational domain for NACA0012 problem. 108
7.15 Transonic NACA0012: Cd error vs length scale 109
7.16 Transonic NACA0012: Cl error vs length scale 109
7.17 Transonic NACA0012: Cd error vs work units 110
7.18 Transonic NACA0012: Cl error vs work units 110
8.1 Channel with smooth bump: acceleration by p-refinement. 115
8.2 Channel with a smooth bump. 116

8.3 Channel with smooth bump: entropy error vs equivalent length scale. 118
8.4 Channel with smooth bump: entropy error vs work units. 118
8.5 Domain for the transonic Ringleb flow. 120
8.6 Mach field for the transonic Ringleb flow: initial solution. 121
8.7 Mach field for the transonic Ringleb flow: spurious shock. 121
8.8 Example of structured mesh for the transonic Ringleb flow. 121
8.9 Transonic Ringleb flow: convergence history. 122
8.10 Transonic Ringleb flow: entropy error vs equivalent length scale. . 122
8.11 Transonic Ringleb flow: entropy error vs work units. 123
8.12 Vortex transport: final velocity distribution on regular mesh. . . . 125
8.13 Vortex transport: error vs equivalent length scale on regular meshes. 125
8.14 Vortex transport: error vs work units on regular meshes. 126
8.15 Vortex transport: example of regular and perturbed mesh. 126
8.16 Vortex transport: error vs equivalent length scale on perturbed meshes. 127
8.17 Vortex transport: error vs work units on perturbed meshes. 127
8.18 Detail of the finest mesh used for the tandem airfoils test. 128
8.19 Entropy field for the tandem airfoils test at the end of the simulation (DG5RK4LS finest mesh) 129
8.20 Drag coefficient for the trailing airfoil (finest mesh p=3,4,5). 130
8.21 Lift coefficient for the trailing airfoil (finest mesh p=3,4,5). 130
9.1 VKI-LS59 cascade: detail of the mesh. 132
9.2 VKI-LS59 cascade: Mach field. 133
9.3 VKI-LS59 cascade: wall isentropic Mach distribution. 133
9.4 T106c cascade: example of hybrid mesh. 135
9.5 T106c cascade: Mach at $Re_{2s} = 185000$, $M_{2s} = 0.65$ 136
9.6 T106c cascade: Mach at $Re_{2s} = 80000$, $M_{2s} = 0.65$ 136
9.7 T106c cascade: laminar kinetic energy at $Re_{2s} = 185000$, $M_{2s} = 0.65$ 137
9.8 T106c cascade: laminar kinetic energy at $Re_{2s} = 80000$, $M_{2s} = 0.65$ 137
9.9 T106c cascade: turbulent kinetic energy at $Re_{2s} = 185000$, $M_{2s} = 0.65$ 138
9.10 T106c cascade: turbulent kinetic energy at $Re_{2s} = 80000$, $M_{2s} = 0.65$ 138
9.11 T106c cascade: wall isentropic Mach distribution ($M_{2s} = 0.65$). . . 139
C.1 Curvilinear computational meshes (blue) and visualization meshes (black) with different subdivisions for triangles and quadrilaterals. 148

Part I

A numerical approach to aerospace propulsion

Chapter 1

Introduction

1.1 CFD for aerospace propulsion systems

The mechanics and thermodynamics of fluid flows can be effectively investigated by both experimental and numerical approaches. The results obtained by experimental activities are the basis for the knowledge of the phenomena which characterize the motion of fluids. Starting from experimental results, it is possible to define mathematical models which try to reproduce the observed phenomena. Usually, these mathematical models require a numerical solution.

A comparison between the experimental approach and the numerical approach shows that they are affected by different kind of uncertainties. The results obtained by an experiment are influenced by the uncertainty related to the sensors and by the assumptions made in order to evaluate quantities which cannot be directly measured.

On the other hand, the numerical approach introduces two kind of errors: a discretization error and a modeling error. The discretization error is related to the numerical solution of the mathematical model and tends to zero when the discretization is refined. In contrast, the modeling error cannot be avoided and is independent from the numerical discretization. The modeling error represent the intrinsic approximation which is hidden in the equations and which can bring to an unphysical prediction. The effects of the modeling errors become quite evident in turbulence modeling for RANS equations, in which usually each turbulence model gives a different result. A possible remedy to this problem is to switch to simpler physical models in which less arbitrary assumptions are included. As far as turbulent flows are concerned, this can be done by performing Direct Numerical Simulations in which only the basic conservation principles are assumed and all the turbulent scales are directly solved. This approach is quite similar to perform a numerical experiment but, unfortunately, its cost becomes prohibitive for the typical Reynolds numbers which characterize the flows in aerospace propulsion systems ($10^5 - 10^7$).

Starting from these considerations, the experimental approach seems to be more reliable, especially if turbulent flows are considered. However, the experimental study of flows related to aerospace propulsion systems can be challenging. This is due to the fact that they are usually characterized by high speed and very large jumps in pressure and temperature (e.g. in rockets or jet engines). These features make the building of an experimental facility very expensive and complex. For example, it is difficult to reproduce experimentally the working conditions of a blade inside a turbomachine because of the unsteady interactions between the stages. Furthermore the numerical approach gives the full distribution of the state variables in the domain. This is very useful in several cases. For example, the numerical approach can be used in conjugated heat transfer problems in order to obtain directly the solution of the temperature distribution in both the flow and in the immersed body.

Finally, the numerical approach is interesting in the framework of optimization processes during the development of an industrial product. Indeed, simulations can be efficiently used to drive the optimization process because they make it possible to test several different configurations without the need to build a large set of prototypes. The experimental approach can then be used as a complementary tool which allows to validate the numerical results and to quantify the error related to the numerical predictions.

1.2 Motivations and description of the work

The study of the flows in aerospace propulsion systems is characterized by several difficulties related to both physical modeling and numerical solution of equations. These problems have been progressively addressed during the research activity.

Nowadays, the CFD community is divided by a debate about the use of high-order methods in fluid dynamics. In the literature it is possible to find papers with different views on this topic (see for example [5] and [6]). In this work, high-order methods have been studied and tested for several problems in the aerospace field. The results obtained during the research activity show that high-order methods are more efficient than low-order methods in smooth problems. In other words, the same error level can be obtained with a lower computational cost if high-order methods are employed. However, if the solution contains shock waves or other singularities, then low-order methods (for example classical second order finite volume methods) can become more efficient. Since the flow field is usually characterized by both smooth and singular regions, an optimal choice may be to use an adaptive approach which changes the reconstruction order according to the local flow features.

In the preliminary study performed in this work, this last approach outperforms both uniform low-order and uniform high-order discretizations (see Chapter 7.4).

In order to explore the possibilities offered by high-order methods, several high-order finite volume schemes have been implemented during the first six months of the research activity. In particular, experiments were performed with the WENO method [7], the compact WENO method [8] and the third order limiter by Cada and Torrilhon [9]. This preliminary activity gave goods results on cartesian and regular structured meshes. In particular, the tested finite volume schemes show excellent shock capturing properties. They are able to describe moving shock waves with very sharp profiles and negligible oscillations. However, the extension of these schemes to more complex problems is not trivial. This is mainly due to the fact that in finite volume schemes the reconstruction is performed through the use of information from neighbouring elements. When unstructured meshes are considered the reconstruction process can become involved. Further complications appear in the presence of non-conforming meshes which can result from h-adaptive algorithms (see 7.3) or sliding meshes approaches (for example in studies on the interaction between stator and rotor in turbomachinery flows). These problems become even more complicate when high-order reconstructions are employed. In fact, in finite volume schemes the increase in the reconstruction order is related to an enlargement of the numerical stencil.
For these reasons an alternative approach has been explored, following one of the current trends in the CFD community. Indeed, the second half of the first year has been devoted to the implementation of a discontinuous Galerkin method. This approach brings some simplifications in the previously described situations because it makes it possible to obtain high-order reconstructions by using only the information which is defined inside each element. In this way the stencil remains fixed, even if high-order reconstructions are chosen. This aspect is not only important because it greatly simplifies the implementation but also because it allows good scalability in parallel computations.
During the first year the discontinuous Galerkin approach was initially implemented for a linear scalar one-dimensional convection problem. After that, the model was extended to the Burger equation and 1D Euler equations. The introduction of non-linear terms in the convective fluxes required the implementation of a proper shock capturing technique. At the beginning, some experiments were performed with the Total Variation Bounded in the Mean (TVBM, see [10]) limiter which gave good results mainly for low order reconstructions. After that, a filter-based approach has been tested. In particular, an adaptive procedure named Feedback filtering [1] has been developed in this phase. The goal of this procedure is to obtain a shock capturing method which is as problem independent as possible. See Section 6.1 for details. In this phase the extension of the code to 2D problems was also performed.

The code developed during the first year was used to participate to the Second International Workshop on High Order CFD methods [11]. It was a very useful experience because it makes it possible to compare the developed tool with state of art codes from all over the world. A first comparison was performed by considering the error level as a function of the total number of degrees of freedom. Good results were obtained in this case, especially in the transonic Ringleb flow (8.2.2). A second comparison involved the computational time which was properly normalized in order to take into account the features of the particular hardware used by each participant. This second comparison showed that the preliminary code was significantly slower than other existing codes in steady problems. The main reason for this weakness was the lack of a proper accelerating technique for steady problems. Indeed, while all the other participants used implicit time integration schemes, the preliminary code could use only explicit time integration schemes which can be quite inefficient for steady problems. The workshop experience influenced several of the following choices.

During the first year, only inviscid compressible flows were considered. This made it possible to develop a solid numerical background for the following work and to focus the attention on the problems related to shock capturing.

During the second year, diffusive fluxes were introduced in the model. Heat equation, Navier-Stokes equations and Reynolds Averaged Navier-Stokes equations were studied in this phase. The work was mainly related to the development of a new method for the discretization of the diffusive terms. The method, named Enhanced Stability Recovery [2], is related to the original recovery method introduced by van Leer and Nomura [3] in 2005. The original recovery method shows an elegant way to compute diffusive fluxes at the interface between two adjacent elements. It is based on a procedure in which a smooth differentiable recovery solution is obtained starting from the discontinuous solution of the two elements. The main idea is to find a recovery solution which is the weak sense indistinguishable from the original discontinuous solution. The ESR method is based on the same apporach but is uses a different recovery basis and a different procedure for the imposition of Dirichlet boundary conditions. The performed numerical comparisons showed that the ESR method has a larger stability limit in explicit time integration with respect to the original recovery method but it introduces a larger discretization error (see Section 4.3).

As far as turbulence modeling is concerned, the Spalart-Allmaras one-equation model has been implemented during the second year. The model showed good results on turbomachinery flows characterized by high Reynolds number and fully turbulent behavior (see Section 9.1). Furthermore it appears particularly suitable for the use with implicit time integration schemes because of the low stiffness of its source terms (if compared to Wilcox $k - \omega$ model).

However, the weaknesses of this model appear when it is used in very low Reynolds number transitional flows.

In order to investigate also low Reynolds number flows, other two turbulence models have been implemented during the third year. In particular, the several versions of the Wilcox $k - \omega$ model (1988 [12], 1998 [13], 2006 [14]) have been implemented and tested. This phase was very useful to gain experience in transitional flows. Indeed, several sensitivity studies have been performed in order to understand the effects of boundary conditions and calibration terms in turbulence modeling. The attention was focused to a particular test case in the turbomachinery field: the transitional flow in the T106c turbine cascade. The numerical experiments performed on this airfoil suggested the introduction of another turbulence model, focused on transitional flows. In particular a three equation model based on the $k - \omega$ model and the laminar kinetic energy concept has been implemented and tested, following the approach of [15].
The stiffness related to the source terms of these turbulence models makes the explicit time integration of the equations quite expensive. For this reason an implicit time integration scheme has been implemented. This new feature improves significantly the performances of the code in steady problems, bringing them to a level comparable to that of other research codes presented at the Workshop.
The last months of the third year have been dedicated to the development of adaptive algorithms. The discontinuous Galerkin approach offers the possibility to easily perform adaptation in both the element order (p-adaptivity) and the element size (h-adaptivity). Several experiments were done on both smooth flows and flows with singularities. The most interesting results have been obtained by means of an hybrid hp-adaptive approach.

Chapter 2

Physical models

In a CFD simulation the choice of the underlying physical model is the first fundamental step required to obtain results which can reproduce real-world phenomena. In particular the physical model should be chosen in order to obtain a good compromise between the cost and the ability to reproduce the physics. For example, the industrial design process of an aerospace component is characterized by several steps in which the physical model is gradually refined and improved. In the following, the five physical models implemented in the code are described putting in evidence the typical flow features which they can describe.
All the models' equations are adimensionalized following the procedure described in Appendix A.

2.1 Euler equations

The Euler equations describe the behavior of compressible flows. They show all the relevant phenomena which are related to the propagation of nonlinear waves. Indeed, the solution of Euler equations can shown expansion and compression fans, shock waves or contact surfaces.
Consider a 2D domain Ω. The equations can be written in conservative form as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} \tag{2.2}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j(E+P)) = 0 \tag{2.3}$$

Here ρ, P, E, u, v denote density, pressure, total energy per unit volume and cartesian components of velocity normalized to conventional reference values (see Appendix A). The set of equations is closed by the introduction of the following equation of state:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \tag{2.4}$$

in which the ratio of specific heats γ has been set equal to 1.4 for all the considered test cases.

2.2 Navier-Stokes equations

The compressible Navier-Stokes equations describe the behavior of compressible flows in the presence of diffusive effects. The equations can be written in the following form:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.5}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{1}{Re_{ref}}\frac{\partial \tau_{ij}}{\partial x_j} \tag{2.6}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j(E + P)) = \frac{1}{Re_{ref}}\frac{\partial}{\partial x_j}\left[u_i \tau_{ij} - q_j\right] \tag{2.7}$$

The heat flux is described through the linear Fourier law:

$$q_i = -\frac{\gamma}{(\gamma - 1)}\frac{\mu}{Pr}\frac{\partial T}{\partial x_i} \tag{2.8}$$

where the heat conductivity λ has been expressed in terms of the dynamic viscosity μ and the Prandtl number Pr. In this work the heat conductivity and the dynamic viscosity are assumed constant. However, it is possible to easily implement the Sutherland's law which describes the dependency of dynamic viscosity from temperature. The components of the shear-stress tensor τ_{ij} are computed according to the Boussinesq approach and assuming the Stoke's hypothesis for bulk viscosity:

$$\tau_{ij} = 2\mu\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{2.9}$$

where S_{ij} is the mean strain rate tensor:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.10}$$

2.3 Reynolds Averaged Navier-Stokes equations

The Navier-Stokes equations can be used to predict the behavior of turbulence flows with Direct Numerical Simulations in which the 3D equations are integrated on a sufficiently fine mesh. In this way, all the scales of turbulence are solved and their effect on the solution is directly taken in account. However, this approach becomes prohibitively expensive for high Reynolds number flows. For this reason, several alternatives approaches have been introduced in order to reduce the required computation effort. The key idea of these methods is that only the larger scales of turbulence have to be solved while the effects of the smaller scales can be described by a proper closure model.

For example in Large Eddy Simulations the governing equations are obtained by the application of a filter to the Navier-Stokes equations. The obtained filtered equations contain new terms which represents the effects of the scales removed by the filter. Usually the cut-off length of the filter is assumed to be proportional to the size of the mesh elements and so the filtered flow field can be seen as the restriction of the flow field to those components which can be resolved by the mesh. The practical consequence of this approach is that the new terms (subgrid terms) which appear in the governing equation are a function of the mesh size. When the mesh size tends to zero the subgrid terms also tend to zero and so the Navier-Stokes equations are obtained as an asymptotic limit.

An alternative approach is to perform a phase average (or a time average for steady problems) of the Navier-Stokes equations in order to obtain governing equations which describe the evolution of the average properties of the flow. This is the main idea on which Reynolds Averaged Navier-Stokes (RANS) equations are based. In particular, the new terms which appear after the averaging procedure are expressed as a function of the averaged quantities. This is the weak point of the RANS approach: the closure model tries to reproduce the effects of scales which are not resolved, using the information available in the averaged flow field. This approach requires some strong hypothesis on the behavior of the flow which are not usually satisfied in real problems. For example, several turbulence models assume that the turbulent energy cascade is in equilibrium or they assume isotropic properties for the turbulent structures.

From an engineering perspective, this approach can be acceptable because in practical problems there is often the need to know only some averaged or integral properties of the flow field (for example the lift and drag coefficients of a body). Furthermore, the cost of the RANS approach is several orders of magnitude lower than a corresponding DNS computation. This difference grows quickly when the Reynold number of the flow is increased. Is is also interesting to note that the RANS approach can be used to perform 2D simulations in problems which show a direction in which the averaged quantities are constant (for example turbine cascades). This makes it possible to perform very cheap estimates of the properties of

these flow fields. In contrast, the LES and DNS approaches require usually a 3D simulation in order to describe properly the dynamics of vorticity, even if there is a direction in which the averaged fields are constant.
Because of their low computational cost RANS models are widely used. In this work, three models have been implemented and compared on several flow fields. As it can be expected, each model works better than others in some problems and so a general rule cannot be found. In the literature there is a trend in proposing new models based on a growing number of equations. The goal is that the use of more equations can reduce the number of calibration coefficients in the model. This is not always true.
Since this work is devoted to the study of compressible flows, all the implemented models are based on a Favre average of the Navier-Stokes equations. In the following the full system of conservation equations is reported for the different models. This makes it possible to put in evidence some approximations which can be done during the coupling of the turbulence model with the original conservation equations. For example, some models involve the definition of a turbulent kinetic energy variable and this require some modification in the energy equation: the total energy can be redefined in order to include also the turbulent kinetic energy or, as it has been done in this work, an additive source term can be introduced in the original energy equation. In contrast, there are models (like for example the Spalart-Allmaras model) which do not involve the definition on a turbulent kinetic energy variable. In these cases the energy equation remains unaltered, with the exception of the inclusion of turbulent terms in the diffusive fluxes. As far as turbulent heat fluxes are concerned they are computed thanks to the introduction of a turbulent Prandtl number Pr_t, which has been set $Pr_t = 0.9$ in all simulations.

2.3.1 Spalart-Allmaras model

The Spalart-Allmaras (SA) model [16] is one of the most used RANS models. This is due to its good behavior on fully turbulent high Reynolds number flows. Furthermore, it requires the introduction of only one transport equation for the modified turbulent viscosity $\tilde{\nu}$. Since its introduction, several versions have been proposed. A comprehensive review can be found in [17]. Spalart et al. [18] report that confusion exists in the literature over the formulation of the SA model for compressible flows. In this work the version of the model described in [18] has be implemented. It can be obtained by combining the mass conservation equation with the incompressible version of the SA model. The complete set of equations is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \qquad (2.11)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial \hat{\tau}_{ij}}{\partial x_j} \tag{2.12}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j(E+P)) = \frac{\partial}{\partial x_j}\left[u_i \hat{\tau}_{ij} - q_j\right] \tag{2.13}$$

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_j}(\rho u_j \tilde{\nu}) = \rho\left(\hat{P} - \hat{D}\right) + \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[\rho(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right] + \frac{cb_2}{\sigma Re_{ref}}\rho\frac{\partial \tilde{\nu}}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k} + \\ - \frac{1}{\sigma Re_{ref}}\frac{\partial \rho}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k} \tag{2.14}$$

The heat flux includes the turbulent contribution which is modeled by the Reynolds analogy and is expressed through the introduction of the turbulent Prandtl number Pr_t:

$$q_i = -\frac{\gamma}{(\gamma-1)}\left(\frac{\mu}{Pr Re_{ref}} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x_i} \tag{2.15}$$

The turbulent stress tensor $\bar{\tau}_{ij}$ is:

$$\bar{\tau}_{ij} = 2\mu_t\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{2.16}$$

The total (laminar+turbulent) stress tensor is:

$$\hat{\tau}_{ij} = \frac{2\mu}{Re_{ref}}\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] + \bar{\tau}_{ij} \tag{2.17}$$

The turbulent viscosity μ_t is defined as:

$$\mu_t = \rho\tilde{\nu} f_{v1} \tag{2.18}$$

The damping function f_{v1} is given by:

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{vi}^3} \qquad \chi = \frac{\tilde{\nu}}{\nu} \tag{2.19}$$

The production and destruction terms $\hat{P}$ and $\hat{D}$ can be expressed as:

$$\hat{P} = c_{b1}(1-f_{t2})\tilde{S}\tilde{\nu} \qquad \hat{D} = \frac{1}{Re_{ref}}c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 \tag{2.20}$$

Here d is the distance from the closest wall while $\tilde{S}$ is the modified vorticity:

$$\bar{S} = \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{2.21}$$

$$\tilde{S} = \begin{cases} S + \bar{S} & \text{if } \bar{S} \geq -c_{v2} S \\ S + \frac{S\left(c_{v2}^2 S + c_{v3}\bar{S}\right)}{(c_{v3} - 2cv2)S - \bar{S}} & \text{if } \bar{S} < -c_{v2} S \end{cases} \tag{2.22}$$

where S is the magnitude of vorticity. In Eq. 2.22 the modified vorticity has been limited following the approach proposed in [18]. In that work Spalart et al. observe that the modified vorticity $\tilde{S}$ should always be positive and greater than $0.3S$ in physically relevant situations. In the discretized equations it is possible for $\tilde{S}$ to become zero or negative due to the fact that f_{v2} is itself negative over a range of χ. Negative values of $\tilde{S}$ would introduce problems in the other correlations of the model.

The function f_w is:

$$f_w = g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6} \qquad g = r + c_{w2}(r^6 - r) \qquad r = min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, r_{lim} \right) \tag{2.23}$$

In the original SA model a trip term and a laminar suppression term are introduced in the $\tilde{\nu}$ equation. This makes it possible to force the transition of the flow in a particular point specified by the user. Since in most of the applications the location of the transition point is not known and the model is considered not appropriate for transitional flows by Rumsey and Spalart [19], the trip and laminar suppression terms have not been included in the previous equations.

The constants of the model are $c_{b1} = 0.1355, \sigma = 2/3, c_{b2} = 0.622, \kappa = 0.41, c_{w1} = c_{b1}/\kappa^2, c_{w2} = 0.3, c_{w3} = 2, c_{v1} = 7.1, c_{t1} = 1, c_{t2} = 2, c_{t3} = 1.2, c_{t4} = 0.5, r_{lim} = 10, c_{v2} = 0.7, c_{v3} = 0.9$.

Some constants of the model (like for example κ) come from the constants of the theoretical boundary layer law. Indeed, the tests performed on the velocity profile for a turbulent boundary layer on a flat plate show good agreement with the boundary layer law (see Figure 2.1). In Figure 2.2 the computed skin friction coefficient for the turbulent flow on a flat plate is reported and compared with two correlations from the literature.

2.3.2 Wilcox $k - \omega$ model

In this work several versions of the Wilcox k-ω model have been considered (1988 [12], 1998 [13], 2006 [14]). They have been implemented following the non-standard

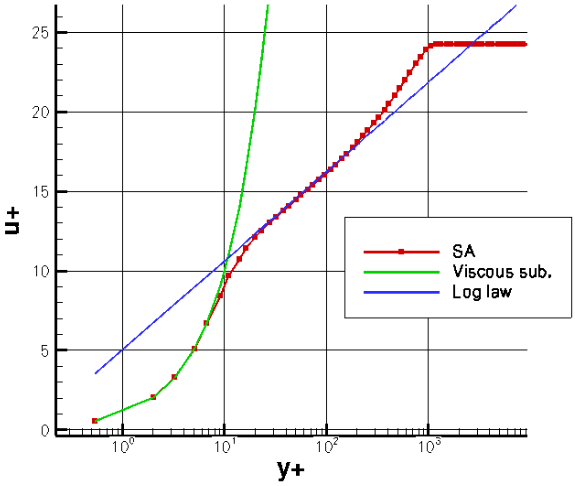

Figure 2.1. Velocity profile in turbulent boundary layer obtained by the proposed discretization with the Spalart-Allmaras model

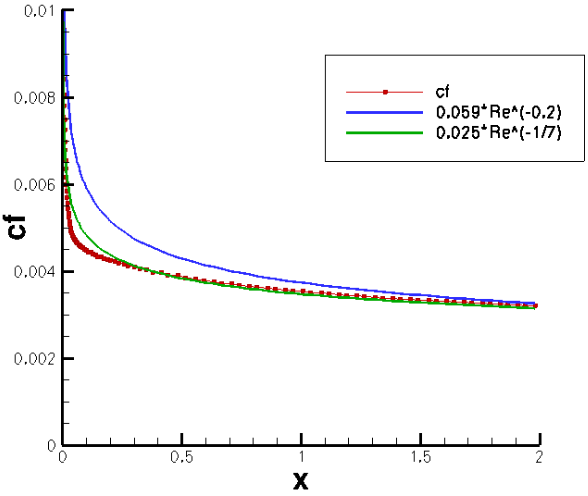

Figure 2.2. Skin friction coefficient for turbulent flat plate: comparison between the implemented Spalart-Allmaras model and correlations from the literature

approach proposed by Bassi et al. [20]. In this way the equations are solved for the logarithm of ω ($\tilde{\omega} = \log(\omega)$) instead of ω itself: this makes the discretization more stable. Furthermore, both k and $\tilde{\omega}$ are limited on the basis of physical considerations. This helps in reducing the stiffness of the discretized equations which can lead to unphysical values during strong transients. This problem is particularly significant when the equations are integrated in time by an implicit scheme with a very large time step.
In the implemented approach the conservative variable E includes only the internal energy and the kinetic energy of the mean flow. The energy is intended per unit volume. The conservation of the total energy (internal energy+kinetic energy of the mean flow+turbulent kinetic energy) is guaranteed by the presence of coupling source terms in the energy equation and in the k equation.
Wilcox proposed two versions of his models, named high Reynolds and low Reynolds. The names refer to a local Reynols number (Re_T) obtained by the turbulence variables. While in the high Re version some coefficients are constant, in the low Re version these coefficients become functions of Re_T. The correlations of the low Re model tend to the high Re model for high Re_T. Since some applications considered in this work involve low Reynolds number transitional flows, the low Re versions of the models have been considered.
In the following the set of equations for the low Re k-ω(1998) model is reported in the form implemented in the code:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.24}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial \hat{\tau}_{ij}}{\partial x_j} \tag{2.25}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j(E+P)) = \frac{\partial}{\partial x_j}\left[u_i \hat{\tau}_{ij} - q_j\right] - \tau_{ij}\frac{\partial u_i}{\partial x_j} + \beta^* \rho \bar{k} e^{\tilde{\omega}_r} \tag{2.26}$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta^* \rho \bar{k} e^{\tilde{\omega}_r} + \frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Re_{ref}} + \sigma_k \bar{\mu}_t\right)\frac{\partial k}{\partial x_j}\right] \tag{2.27}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho \tilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \tilde{\omega}) = \frac{\alpha}{\bar{k}}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta \rho e^{\tilde{\omega}_r} + \left(\frac{\mu}{Re_{ref}} + \sigma_\omega \bar{\mu}_t\right)\frac{\partial \tilde{\omega}}{\partial x_k}\frac{\partial \tilde{\omega}}{\partial x_k} + \\ + \frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Re_{ref}} + \sigma_\omega \bar{\mu}_t\right)\frac{\partial \tilde{\omega}}{\partial x_j}\right]\end{aligned} \tag{2.28}$$

The heat flux includes also the turbulent contribution:

$$q_i = -\frac{\gamma}{(\gamma-1)}\left(\frac{\mu}{PrRe_{ref}} + \frac{\bar{\mu}_t}{Pr_t}\right)\frac{\partial T}{\partial x_i} \tag{2.29}$$

The turbulent stress tensor $\bar{\tau}_{ij}$ is:

$$\bar{\tau}_{ij} = 2\bar{\mu}_t \left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right] - \frac{2}{3}\rho \bar{k} \delta_{ij} \tag{2.30}$$

The total (laminar+turbulent) stress tensor is:

$$\hat{\tau}_{ij} = \frac{2\mu}{Re_{ref}} \left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right] + \bar{\tau}_{ij} \tag{2.31}$$

The turbulent viscosity $\bar{\mu}_t$ is defined as:

$$\bar{\mu}_t = \frac{\alpha^* \rho \bar{k}}{e^{\tilde{\omega}_r}} \tag{2.32}$$

The model constant, the auxiliary functions and the low Re correlations are:

$$\sigma_k = 0.5 \qquad \sigma_\omega = 0.5 \qquad \beta_0^* = 0.09 \qquad \beta_0 = 9/125$$

$$\alpha_0 = 1/9 \qquad \alpha_0^* = \beta_0/3 \qquad R_k = 6 \qquad R_\omega = 2.61$$

$$\alpha = 13/25 \left(\frac{\alpha_0 + Re_T/R_\omega}{1 + Re_T/R_\omega} \right) (\alpha^*)^{-1} \qquad \alpha^* = \frac{\alpha_0^* + Re_T/R_k}{1 + Re_T/R_k} \tag{2.33}$$

$$Re_T = \frac{\rho k}{\mu e^{\tilde{\omega}}} Re_{ref} \qquad \chi_\omega = \nabla k \cdot \nabla \tilde{\omega} / e^{2\tilde{\omega}} \tag{2.34}$$

$$\beta = \beta_0 f_\beta \qquad \beta^* = \beta_0^* f_{\beta^*} \tag{2.35}$$

$$f_\beta = \frac{1 + 70\chi_\omega}{1 + 80\chi_\omega} \qquad f_{\beta^*} = \begin{cases} 1 & \text{if } \chi_k < 0 \\ (1 + 680\chi_k^2)/(1 + 400\chi_k^2) & \text{if } \chi_k \geq 0 \end{cases} \tag{2.36}$$

$$\chi_\omega = \left| \frac{\Omega_{ij}\Omega_{jk}\Omega_{ki}}{\beta_0^* e^{3\tilde{\omega}}} \right| \qquad \chi_k = \frac{1}{e^{3\tilde{\omega}}} \nabla k \cdot \nabla \tilde{\omega} \tag{2.37}$$

where Ω is the vorticity. Note that the term χ_ω is identically zero for 2D flows. For simplicity the correlations and the constants for the 1988 and 2006 versions are not reported. They can be found in [12] and [14].

In the previous equations the limited turbulent viscosity $\bar{k}$ appears in the source terms and in the eddy viscosity. It is defined as

$$\bar{k} = \max(0, k) \tag{2.38}$$

Notice that $\bar{k}$ has been limited exactly to zero (and not to an arbitrary small value) because, after the appropriate substitutions, no term in the previous equations is divided by $\bar{k}$.
Furthermore the source terms and the eddy viscosity are computed by using the variable $\tilde{\omega}_r$ instead of $\tilde{\omega}$. The variable $\tilde{\omega}_r$ can be obtained from $\tilde{\omega}$ by the imposition of some realizability conditions. In particular, the positivity of the normal turbulent stresses 2.39 and the Schwarz inequality for the shear turbulent stresses 2.40 are imposed, following the approach of Bassi et al.[20]:

$$\overline{\rho u_i''^2} \geq 0 \tag{2.39}$$

$$\left(\overline{\rho u_i'' u_j''}\right)^2 \leq \overline{\rho u_i''^2} \cdot \overline{\rho u_j''^2} \tag{2.40}$$

These requirements are satisfied thanks to the definition of a lower bound $\tilde{\omega}_{r0}$ for $\tilde{\omega}$:

$$\tilde{\omega}_r = \max(\tilde{\omega}, \tilde{\omega}_{r0}) \tag{2.41}$$

The lower bound $\tilde{\omega}_{r0}$ can be obtained by substituting the modeled turbulent stresses (Eq.2.30)in Eq.2.39 and Eq. 2.40:

$$\frac{2}{3}\rho\bar{k} - 2\bar{\mu}_t\left(S_{ii} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right) \geq 0 \qquad i = 1{,}2 \tag{2.42}$$

$$\begin{gathered}(-2\bar{\mu}_t S_{ij})^2 \leq \left[\frac{2}{3}\rho\bar{k} - 2\bar{\mu}_t\left(S_{ii} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right)\right]\left[\frac{2}{3}\rho\bar{k} - 2\bar{\mu}_t\left(S_{jj} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right)\right] \\ i = 1{,}2 \quad i \neq j\end{gathered} \tag{2.43}$$

Recalling the definition of the turbulent viscosity (Eq. 2.32) it is possible to obtain the following set of realizability inequalities expressed in terms of the unknown $\frac{e^{\tilde{\omega}}}{\alpha^*}$:

$$\frac{e^{\tilde{\omega}}}{\alpha^*} - 3\left(S_{ii} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right) \geq 0 \qquad i = 1{,}2 \tag{2.44}$$

$$\begin{aligned}&\left(\frac{e^{\tilde{\omega}}}{\alpha^*}\right)^2 - 3\left(S_{ii} + S_{jj} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\right)\left(\frac{e^{\tilde{\omega}}}{\alpha^*}\right) + \\ &- 9\left[S_{ij}^2 - \left(S_{ii} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right)\left(S_{jj} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\right)\right] \geq 0 \qquad i = 1{,}2 \qquad i \neq j\end{aligned} \tag{2.45}$$

In the following, the maximum value between the zeros of Eqs. 2.44 and 2.45 (the unknown is $\frac{e^{\tilde{\omega}}}{\alpha^*}$) will be denoted as $\hat{a}$. If $\frac{e^{\tilde{\omega}}}{\alpha^*} > \hat{a}$ then Eqs. 2.44 and 2.45 are both satisfied. The lower bound $\tilde{\omega}_{r0}$ that guarantees realizable turbulent stresses can be obtained by:

$$\frac{e^{\tilde{\omega}_0}}{\alpha^*} = \hat{a} \tag{2.46}$$

Since the low Re version of the $k-\omega$ model is considered here, α^* depends on the local turbulent Reynolds number Re_k according to Eq. 2.33. In conclusion, $\tilde{\omega}_{r0}$ can be obtained by the solution of the following second degree equation:

$$\left(e^{\tilde{\omega}_{r0}}\right)^2 + \left(\frac{\rho\bar{k}Re_{ref}}{\mu R_k} - \hat{a}\alpha_0^*\right) e^{\tilde{\omega}_{r0}} - \frac{\hat{a}\rho\bar{k}Re_{ref}}{\mu R_k} = 0 \tag{2.47}$$

In order to show an example of the behavior of the Wilcox $k-\omega$ low Re model in the presence of transition the plot of Figure 2.3 is reported. It shows the skin friction coefficient distribution on a flat plate obtained by the present implementation. The obtained results show how the behavior change from the laminar Blasius law (blue line) to the turbulent correlation (light blue). The transition point is strongly influenced by the freestream turbulence level and by the wall roughness which determines the boundary condition for ω.

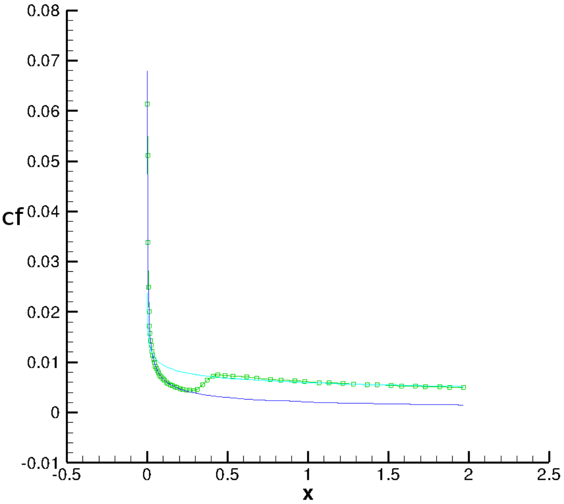

Figure 2.3. Skin friction coefficient on flat plate: transition from laminar to turbulent. Blue=Blasius solution, Light blue=turbulent correlation, Green= implemented Wilcox $k-\omega$ low Re model

2.3.3 Laminar Kinetic Energy model

The study of transitional flows by RANS models is very difficult because the hypothesis on which the RANS approach is based are usually not satisfied in transitional flows. This is the typical field in which LES and DNS methods find their natural application. This is due to the fact that the prediction of transition becomes important for low Reynolds number flows and the computational cost of LES or DNS can be acceptable when the Reynolds number is sufficiently low. However, several works have been proposed in the literature also on the study of transitional flows with RANS models.

In particular, there are several transition models which are based on the intermittency concept. They use an additive transport equation for the intermittency variable (usually named γ) which can usually assume values in the range 0-1. The intermittency variable is used as a weighting factor to control the intensity of the production term in classic RANS model. This approach requires the introduction of empirical correlations which have to be calibrated on the particular flow which is under study. An important work in this field was made by Langtry and Menter [21] who introduced the $\gamma - Re_\theta$ model. In particular their model has the remarkable feature of being truly local. Babajee [22] performed a detailed analysis on the use of the $\gamma - Re_\theta + SST$ model for the low pressure turbine cascade T106c, which has been studied also in this work (Chapter 9.2). In particular, Babajee [22] showed that it is very difficult to obtain results in good accordance with the experimental data when the $\gamma - Re_\theta$ model is used for very low Reynolds numbers.

In the last years, several transition models based on an alternative idea have been developed. In particular, the laminar kinetic energy (LKE) concept seems to be promising. This approach was initially introduced by Mayle and Schulz [23] in order to take into account the pretransitional rise of fluctuations in transitional boundary layers and their subsequent breakdown to turbulence. They introduced an additional equation in order to describe the evolution of the energy which is associated to these pretransitional fluctuations. This idea has been exploited by Walters and Leylek [24] and Lardeau et al. [25] to develop a model for natural and by-pass transition. The concept has been used also by Pacciani et al. [15] for the study of transitional flows in low pressure gas turbines. In particular, they have focused the attention on the transition induced by laminar separation. In their approach, a three equations RANS model has been developed and tested with a second order finite volume solver. Their model is based on the low Re Wilcox (1998) k-ω model plus an additive equation for the laminar kinetic energy. The model contains coupling terms between the laminar kinetic energy equation and the turbulent kinetic energy equation which are responsible for the transition process and the exchange of energy from the pretransitional fluctuations to the turbulent field.

In this work the LKE model [26] has been implemented and tested in the DG

framework. The numerical experiments performed on a low pressure turbine cascade (see Section 9.2) showed that the LKE approach works well if used together with the 1998 version of the Wilcox $k-\omega$ model. Indeed, this version of the $k-\omega$ model underpredicts the turbulent kinetic energy and so the LKE equation introduces the missing contribution. In contrast, the 1988 and 2006 versions of the $k-\omega$ model seem to overpredict the turbulent kinetic energy and so the additional contribution given by the LKE equation degrades further the agreement with the experimental data.
In the following the equations of the implemented LKE model are reported:

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x_j}(\rho u_j)=0 \tag{2.48}$$

$$\frac{\partial}{\partial t}(\rho u_i)+\frac{\partial}{\partial x_j}(\rho u_i u_j)=-\frac{\partial P}{\partial x_i}+\frac{\partial \hat{\tau}_{ij}}{\partial x_j} \tag{2.49}$$

$$\frac{\partial E}{\partial t}+\frac{\partial}{\partial x_j}(u_j(E+P))=\frac{\partial}{\partial x_j}\left[u_i\hat{\tau}_{ij}-q_j\right]-\tau_{ij}\frac{\partial u_i}{\partial x_j}+\beta^*\rho\bar{k}e^{\tilde{\omega}_r}-\rho\nu_L S^2+2\frac{\mu}{Re_{ref}}\frac{k_L}{d^2} \tag{2.50}$$

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_j}(\rho u_j k)=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\bar{k}e^{\tilde{\omega}_r}+\frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Re_{ref}}+\sigma_k\bar{\mu}_t\right)\frac{\partial k}{\partial x_j}\right]+\rho R \tag{2.51}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho\tilde{\omega})+\frac{\partial}{\partial x_j}(\rho u_j\tilde{\omega})=\frac{\alpha}{\bar{k}}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta\rho e^{\tilde{\omega}_r}+\left(\frac{\mu}{Re_{ref}}+\sigma_\omega\bar{\mu}_t\right)\frac{\partial\tilde{\omega}}{\partial x_k}\frac{\partial\tilde{\omega}}{\partial x_k}+\\+\frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Re_{ref}}+\sigma_\omega\bar{\mu}_t\right)\frac{\partial\tilde{\omega}}{\partial x_j}\right]\end{aligned} \tag{2.52}$$

$$\frac{\partial}{\partial t}(\rho k_L)+\frac{\partial}{\partial x_j}(\rho u_j k_L)=\frac{\partial}{\partial x_j}\left(\frac{\mu}{Re_{ref}}\frac{\partial k_L}{\partial x_j}\right)+\rho\nu_L S^2-2\frac{\mu}{Re_{ref}}\frac{k_L}{d^2}-\rho R \tag{2.53}$$

Here d is the distance from the closest wall.
It can be seen from Eqs. 2.51 and 2.53 that in the LKE approach a coupling term (ρR) is added in the k and k_L equations. This term is responsible for the transition from laminar to turbulent flow: when this term is activated there is a transfer of energy from the fluctuating laminar field to the turbulent field.
The production term in the k_L equation depends on the "laminar eddy viscosity" ν_L:

$$\nu_L=C_1 f_1\sqrt{k_L}\,\delta_\Omega \tag{2.54}$$

The constant C_1 has to be deduced from calibration. In [15] it is set as $C_1 = 0.01$. In the present work the value $C_1 = 0.009$ is chosen.
The function f_1 is

$$f_1(T_u) = \max\left(0.8{,}2\tanh\left(\sqrt{T_u/4.5}\right)\right) \tag{2.55}$$

where T_u is the turbulence level. It is defined as $T_u = 100\sqrt{(2/3)k/q^2}$ where q is the velocity magnitude.
The variable δ_Ω is the vorticity thickness. It is the most problematic variable in this model because it is non local. Indeed, Pacciani et al. [15] suggest that it can be computed as

$$\delta_\Omega = \frac{q_\infty}{2}\left(\frac{\partial q}{\partial y}\right)^{-1}_{\max_y} \tag{2.56}$$

where q_∞ is the "freestream" velocity magnitude and $\frac{\partial q}{\partial y}$ is the derivative of the velocity magnitude with respect to the wall normal coordinate y. The value of q_∞ can be estimated by the local velocity value at the edge of the boundary layer.
The transfer term R is modeled as:

$$R = C_2 f_2 \omega \beta^* k_L \tag{2.57}$$

The damping function f_2 is

$$f_2 = 1 - e^{-\Psi/C_3} \qquad \Psi = \max(0, R_y - C_4) \tag{2.58}$$

The constant C_2 regulates the intensity of the energy transfer from k_L to k and is set as $C_2 = 0.3$. The constant C_4 represents a threshold for the activation of transition and it is set as $C_4 = 10$. This threshold is compared with the variable $R_y = Re_{ref}\sqrt{k}d/\nu$ which can be seen as a local Reynolds number based on the turbulent kinetic energy k and the wall distance d. Finally, the constant C_3 is set as $C_3 = 8$.
Heat fluxes and shear stresses are computed following the same approach used for the $k - \omega$ model.

2.4 Boundary conditions

The evolution of the computed flow field is regulated by the imposed boundary conditions. For this reason, it is fundamental to take care of the physical phenomena which are responsible for the propagation of the informations in the fluid. This is particular evident in Euler equations in which the wave propagation effects are described by the characteristics signals.
In the presence of diffusive effects further conditions have to be imposed in order

to compute the viscous fluxes (F_v and G_v).

As far as RANS equations are concerned, the choice of the proper boundary conditions can be quite difficult because often there are not experimental data available for the variables on which the turbulence model works. In this case, it is necessary to introduce empirical correlations which link the available boundary information to the variables requested by the turbulence model.

2.4.1 BCs for Euler equations

The Euler equations are dominated by the presence of wave propagation effects. It is possible to find characteristic lines in the space-time along which signals are transported. In order not to violate physics, BCs have to be imposed coherently with the propagation of these signals. In particular, the number of BCs which can be imposed is equal to the number of signals which enter in the computational domain.

Inlet and outlet boundaries are treated by defining a ghost state with some information from the computational domain and some from the external environment, according to the number of characteristic signals which enter in the domain. In this work, total temperature, total pressure and flow direction are imposed at subsonic inlets. All the data are imposed at supersonic inlets. Pressure is the only quantity imposed at subsonic outlets. Alternatively, the incoming signals can be directly imposed in subsonic inlets or outlets in order to obtain a non-reflecting boundary condition.

If the boundary is a solid wall then the tangency condition is enforced by defining a ghost state characterized by a velocity vector specular with respect to the internal velocity.

The previous considerations hold for the Euler equations but can also be used for the computation of boundary convective fluxes (F and G) in Navier-Stokes or RANS equations.

2.4.2 BCs for Navier-Stokes equations

The presence of diffusive terms in the Navier-Stokes equations requires the introduction of additive BCs with respect to the Euler equations. In particular, in the presence of solid walls the no-slip condition can be imposed on the momentum variables.

As far as the energy equation is concerned, it is possible to impose Dirichlet, Neumann or Robin BCs depending on the properties of the wall.

2.4.3 BCs for Spalart-Allmaras model

The Spalart-Allmaras equation works on the modified turbulent viscosity $\tilde{\nu}$ which is transported by the flow field. At solid walls an homogeneous Dirichlet boundary condition can be imposed because the velocity fluctuations are zero here and so also the turbulent viscosity has to be zero. However, the situation is more complex at the inlet where the incoming turbulent viscosity has to be specified. The turbulent eddy viscosity is a quantity which cannot be directly measured in an experiment. It can be obtained from empirical correlations based on the turbulence intensity and the turbulence characteristic length scale.
However, Rumsey and Spalart [19] recommend to use this model only for fully turbulent flows and not for low Reynolds transitional flows. In particular, the recommended inlet turbulent viscosity for fully turbulent flows is $3\nu \leq \tilde{\nu} \leq 5\nu$ [17] where ν is the molecular kinematic viscosity.

2.4.4 BCs for Wilcox $k - \omega$ model

At solid walls the turbulent kinetic energy k has to be set equal zero because there are no turbulent fluctuations here.
In contrast, the wall value for the specific dissipation ω is not zero. A possible approach to find the proper wall boundary condition for ω is the so-called rough-wall method proposed by Wilcox [14]. This approach makes it possible to describe both smooth and rough walls. In the case of rough walls, it requires the definition of a non-dimensional equivalent sand-roughness height $k_r^+ = k_r u_\tau / \nu$, where u_τ is the friction velocity defined as $u_\tau = \sqrt{\tau_w / \rho_w}$. Then the boundary condition is:

$$\omega_w = S_r \frac{u_\tau^2}{\nu} \tag{2.59}$$

where the correlation S_r is

$$S_r = \begin{cases} (50/k_r^+)^2 & \text{if } k_r^+ < 25 \\ 100/k_r^+ & \text{if } k_r^+ \geq 25 \end{cases} \tag{2.60}$$

Wilcox proposed to use the same approach also for smooth walls with the so-called "slightly-rough-wall" boundary condition. In practice he suggested to substitute the perfectly smooth wall surface with an hydraulically smooth surface. In this way the boundary condition becomes:

$$\omega_w = 2500 \frac{\nu}{k_r^2} \tag{2.61}$$

where, according to Wilcox, k_r should be 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.

An alternative approach for smooth walls was proposed by Menter [27] with the following relation which links the wall specific dissipation (ω_w) to the distance of the first grid point off the wall (y_1):

$$\omega_w = 10 \frac{6\nu}{\beta y_1^2} \tag{2.62}$$

The condition described by Eq.2.62 means setting at the wall the analytical solution computed at $y = \alpha_M y_1$ where $\alpha_M = 1/\sqrt{10}$.

A combination between Eq.2.61 and Eq.2.62 gives a linear relation between the prescribed wall roughness (k_r) and the distance of the grid point nearest to the wall (y_1):

$$k_r = C_w y_1 \qquad C_w = \sqrt{2500 \alpha_M^2 \frac{\beta_0}{6}} = 1.77 \tag{2.63}$$

where the constant C_w depends on the numerical scheme and the reconstruction order. Hellsten [28] proposed to optimize the factor 10 of Menter's formula by means of an accurate near-wall numerical study of the ω solution and by comparing skin friction distributions of flat plate flows computed on differently refined grids. The value of the factor proposed by Hellsten is 1.25 instead of 10. With this value the constant of Eq. 2.63 becomes $C_w = 2.5$. Another different optimal value ($C_w = 0.3$) has been found by Bassi et al. [20].

Bassi et al. [29] performed a study in order to understand how the term α_M depends on the reconstruction order p_e in the framework of discontinuous Galerkin schemes. They proposed the following boundary condition:

$$\omega_w = Re_{ref} \frac{6\nu}{\beta (h \alpha_M)^2} \tag{2.64}$$

where

$$\alpha_M = e^{-\sum_{n=1}^{p_e} \frac{1}{n}} \tag{2.65}$$

They suggested to set h equal to the distance between the wall and the centroid of the element.

In this work, the boundary condition described by Eq. 2.64 has always been used in the presence of smooth walls.

As far as the inlet is concerned, both k and ω have to be assigned. The inlet turbulence intensity is usually known from experimental data and so k_{in} can be easily assigned as a function of the inlet turbulence level T_u:

$$k = \frac{3}{2} q_{in}^2 \left(\frac{T_u}{100} \right)^2 \tag{2.66}$$

where q_{in} in the inlet velocity magnitude.

On the contrary, the specific dissipation rate ω_{in} is usually not directly measured and so it should be obtained indirectly. For example, in this work it is obtained starting from the turbulent kinetic energy k_{in} and the characteristic turbulence length scale l_t:

$$\omega_{in} = \frac{\sqrt{k_{in}}}{l_t} \tag{2.67}$$

At the outlet the values of k and ω are extrapolated from the domain.

2.4.5 BCs for LKE model

In the LKE model the same boundary conditions described in 2.4.4 can be used for k and ω. As far as k_L is concerned, it can be set equal to zero at solid walls. In [15] the recommended value for k_L at the inlet is $k_{Lin} = k_{in}$. At the outlet k_L is extrapolated from the interior of the domain.

Part II

Discretization

Chapter 3

Discontinuous Galerkin spatial discretization

The Discontinuous Galerkin (DG) method was introduced first by Hill and Reed in 1973 [30] for the solution of the transport equation in neutronics. Nowadays they are widely used for the numerical solution of conservation laws thanks to some of their features.

Indeed, DG methods seem to join the main advantages of finite volumes (FV) methods and finite elements (FE) methods. From FV methods they have inherited several numerical fluxes which make it possible to define fluxes based on upwind considerations: when hyperbolic partial differential equations are considered, this approach obtains good results since it takes into account physical wave propagation phenomena.

On the other hand, the strategy used to obtain high accuracy in the space discretization is very different in FV and DG methods. In FV methods the only information that is known inside a cell is the average value of the conservative variables and so high-order reconstructions can be obtained only through interpolation of data coming from neighbouring cells. In contrast, DG methods obtain high-order reconstructions introducing several degrees of freedom (DOFs) inside the element. This alternative approach greatly simplify reconstruction since all the required information is already inside the element. Furthermore, all the integrals in the governing equations can be easily calculated by mapping the generic element in the physical space to a reference element in the computational space. This aspect makes DG methods very similar to FE methods and so all the knowledge on element mapping developed for FE methods can also be used for DG methods. This topic is of considerable interest with regard to the proper treatment of boundary conditions. Indeed high-order schemes require a careful representation of solid wall

boundaries in order to preserve solution accuracy as was shown by Bassi and Rebay in 1997 [31]. For this reason curvilinear elements have to be introduced at solid walls.
Furthermore, the local and discontinuous nature of the DG reconstruction makes it possible to easily introduce adaptive strategies as will be shown in Chapter 7.

3.1 Variational approach

The DG space discretization will be described in the case of a general 2D governing equation for the variable u:

$$\frac{\partial u}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=Q \qquad (x,y)\in\Omega\subset R^2, t\in R_0^+ \tag{3.1}$$

where F and G are the fluxes (convective+diffusive) and Q is a generic source term. Assume that the domain Ω is subdivided into a collection of non overlapping elements Ω_e. Consider a generic element Ω_e. Consider the functional space V_h defined by

$$V_h=\{v\in L^2(\Omega):v|_{\Omega_e}\in P^p(\Omega_e)\quad \forall\Omega_e\in T_h\} \tag{3.2}$$

where $P^p(\Omega_e)$ is the space of polynomials of degree up to p defined in Ω_e. The space V_h will be used to approximate the solution inside Ω_e. There are no continuity requirements across the element edges in the space V_h and so the reconstruction order p can vary from element to element.
Consider a proper basis for V_h which contains N_e basis functions Φ_i. The numerical approximation of the solution (u_h) belongs to the space V_h and can be represented using a linear combination of the basis functions:

$$u_h(\mathbf{x},t)=\tilde{\mathbf{u}}\cdot\mathbf{\Phi}=\sum_{i=1}^{N_e}\tilde{u}_i\Phi_i \tag{3.3}$$

If u_h is substituted in the governing equation (3.1) the residual R_h is obtained:

$$R_h=\frac{\partial u_h}{\partial t}+\frac{\partial F_h}{\partial x}+\frac{\partial G_h}{\partial y}-Q \tag{3.4}$$

The Galerkin's approach requires that the residual projection on the V_h space is set equal to zero:

$$\int_{\Omega_e}R_h\nu dxdy=\int_{\Omega_e}\frac{\partial u_h}{\partial t}\nu dxdy+\int_{\Omega_e}\frac{\partial F_h}{\partial x}\nu dxdy+\int_{\Omega_e}\frac{\partial G_h}{\partial y}\nu dxdy-\int_{\Omega_e}Q\nu dxdy=0 \quad \forall\nu\in V_h \tag{3.5}$$

The weak form of the discontinuous Galerkin discretization is derived by using integration by parts:

$$\int_{\Omega_e} \frac{\partial u_h}{\partial t} \nu dxdy + \int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \nu ds - \int_{\Omega_e} (\frac{\partial \nu}{\partial x} F_h + \frac{\partial \nu}{\partial y} G_h) dxdy - \int_{\Omega_e} Q\nu dxdy = 0 \tag{3.6}$$

where n_x and n_x are the cartesian components of the outward-pointing unit normal vector and s is the element boundary curve. The last equation shows that the integral over the element boundary $\partial\Omega_e$ requires the knowledge of fluxes ($\hat{F}_h$ and $\hat{G}_h$) at the interface between neighbouring elements. This problem is solved by the introduction of a proper numerical flux. The convective and diffusive contributions to this flux are evaluated separately. They are described in Chapter 3.5 and 4 respectively.

The fact that Eq. 3.6 has to be valid for $\forall v$ can be imposed by considering Eq. 3.6 for all the N_e functions which define the element basis, i.e. setting $v = \Phi_i$ with $1 \leq i \leq N_e$. In this way N_e equations can be obtained and used to describe the time evolution of the N_e solution coefficients inside the element Ω_e:

$$\begin{aligned} &\int_{\Omega_e} \sum_{i=1}^{N_e} \frac{\partial \tilde{u}_i}{\partial t} \Phi_i \Phi_j dxdy + \int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \Phi_j ds - \int_{\Omega_e} (\frac{\partial \Phi_j}{\partial x} F_h + \frac{\partial \Phi_j}{\partial y} G_h) dxdy + \\ &- \int_{\Omega_e} Q \Phi_j dxdy = 0 \quad 1 \leq j \leq N_e \end{aligned} \tag{3.7}$$

The system can be rewritten in compact form through the introduction of the element mass matrix ([**M**])

$$[\mathbf{M}]_{ij} = \int_{\Omega_e} \Phi_i \Phi_j dxdy \tag{3.8}$$

$$[\mathbf{M}] \frac{\partial \tilde{\mathbf{u}}}{\partial t} = -\int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \mathbf{\Phi} ds + \int_{\Omega_e} (\frac{\partial \mathbf{\Phi}}{\partial x} F_h + \frac{\partial \mathbf{\Phi}}{\partial y} G_h) dxdy + \int_{\Omega_e} q \mathbf{\Phi} dxdy \tag{3.9}$$

3.2 Basis functions

The choice of the basis functions strongly affects the behavior of the numerical discretization. Indeed, the time evolution of the numerical solution depends on the linear system (3.9) in which the mass matrix appears. A bad choice of the basis functions can lead to an ill-conditioned mass matrix and this can deteriorate the accuracy of the numerical solution. In the DG framework there are several possibilities about the choice of the element basis. Both nodal and modal basis functions can be adopted. In this work, only modal basis have been considered because of the freedom related to the choice of modal functions. In particular, this makes it possible to choose a set of basis functions which shows two positive features: hierarchy and orthonormality. In Chapter 7.2 the advantages related to the use of a hierarchical basis will be explained.

The second important feature of the chosen bases is the fact that they are orthonormal. When the basis is orthonormal the element mass matrix is the identity and so its condition number is one, independently from the shape of the element. Furthermore, this choice helps to reduce the computational cost with explicit time integration because it is not necessary to multiply the right hand side of (3.9) for the inverse of the mass matrix (which is the identity) at each time step.

In order to obtain an orthonormal element basis the modified Gram-Schmidt procedure has been implemented, following the approach of Bassi et al. [32]. This algorithm requires to be initialized with a starting basis. In this work, two different starting bases have been implemented for this purpose. The first one is obtained by a tensor product of Legendre polynomials defined on the reference square. With this basis, an element with a reconstruction of order p contains $N_e = (p+1)^2$ DOFs in 2D. In the following of this work this basis will be referred to as BASIS A.

Alternatively, it is possible to define a starting basis from a set of monomials defined in the physical space on a reference frame centered on the element's center of mass and aligned to its axis of inertia. This last choice has been proposed by Bassi et al. [32] for general shaped elements. With this basis, an element with a reconstruction of order p contains $N_e = (p+1)(p+2)/2$ DOFs in 2D. In the following of this work this basis will be referred to as BASIS B.

3.3 Mappings

Integrals are approximated through the use of Gauss quadrature formulas (from [33]) defined on the transformed reference element. The number of quadrature points is chosen in order to exactly integrate polynomials of order $2p$ on the reference element.

3.3.1 Quadrilaterals

The mapping between the general curvilinear quadrilateral element (implemented up to quartic elements) and the reference square are computed by the Serendipity mappings (Figure 3.1) reported by Oñate [34].

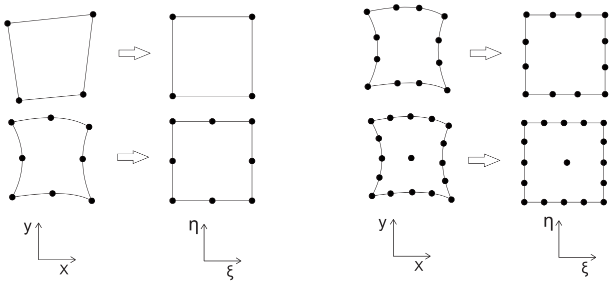

Figure 3.1. Serendipity mappings for quadrilateral elements.

The transformation is

$$\begin{aligned} x &= \textstyle\sum_{i=1}^{n_{nod}} N_i(\xi,\eta)x_i \\ y &= \textstyle\sum_{i=1}^{n_{nod}} N_i(\xi,\eta)y_i \end{aligned} \tag{3.10}$$

where n_{nod} is the number of nodes which define the geometry of the element and N_i are the shape functions.

Linear element ($n_{nod} = 4$)

$$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta) \qquad \xi_i = \pm 1 \quad \eta_i = \pm 1 \tag{3.11}$$

Quadratic element ($n_{nod} = 8$)

$$\begin{aligned}
N_i &= \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)(\xi_i\xi+\eta_i\eta-1) && \xi_i = \pm 1 \quad \eta_i = \pm 1 \\
N_i &= \frac{1}{2}(1-\xi^2)(1+\eta_i\eta) && \xi_i = 0 \quad \eta_i = \pm 1 \\
N_i &= \frac{1}{2}(1+\xi_i\xi)(1-\eta^2) && \xi_i = \pm 1 \quad \eta_i = 0
\end{aligned} \tag{3.12}$$

Cubic element ($n_{nod} = 12$)

$$\begin{aligned}
N_i &= \frac{1}{32}(1+\xi_i\xi)(1+\eta_i\eta)(9(\xi^2+\eta^2)-10) && \xi_i = \pm 1 \quad \eta_i = \pm 1 \\
N_i &= \frac{9}{32}(1-\xi^2)(1+\eta_i\eta)(1+9\xi_i\xi) && \xi_i = \pm\frac{1}{3} \quad \eta_i = \pm 1 \\
N_i &= \frac{9}{32}(1+\xi_i\xi)(1-\eta^2)(1+9\eta_i\eta) && \xi_i = \pm 1 \quad \eta_i = \pm\frac{1}{3}
\end{aligned} \tag{3.13}$$

Quartic element ($n_{nod} = 17$)

$$\begin{aligned}
N_i &= \frac{1}{12}(1+\xi\xi_i)(1+\eta\eta_i)[4(\eta^2-1)\xi\xi_i+4(\eta^2-1)\eta\eta_i+3\xi\eta\xi_i\eta_i] && \xi_i = \pm 1 \quad \eta_i = \pm 1 \\
N_i &= 2(1+\xi\xi_i)(\eta^2-1)(\eta^2-\frac{\xi\xi_i}{4}) && \xi_i = \pm 1 \quad \eta_i = 0 \\
N_i &= 2(1+\eta\eta_i)(\xi^2-1)(\xi^2-\frac{\eta\eta_i}{4}) && \xi_i = 0 \quad \eta_i = \pm 1 \\
N_i &= \frac{4}{3}(1+\xi\xi_i)(1-\eta^2)(\eta^2+\eta\eta_i) && \xi_i = \pm 1 \quad \eta_i = \pm\frac{1}{2} \\
N_i &= \frac{4}{3}(1+\eta\eta_i)(1-\xi^2)(\xi^2+\xi\xi_i) && \xi_i = \pm\frac{1}{2} \quad \eta_i = \pm 1 \\
N_i &= (1-\xi^2)(1-\eta^2) && \xi_i = 0 \quad \eta_i = 0
\end{aligned} \tag{3.14}$$

3.3.2 Triangles

Curvilinear triangular elements (implemented up to cubic elements) are transformed into the reference triangle by standard mappings also from [34]. Consider a point P inside a triangle with the vertices 1,2 and 3. This defines three subareas A_1, A_2 and A_3 corresponding to the triangles $P23$, $P13$ and $P12$. The area coordinates L_1, L_2 and L_3 are defined as

$$L_1 = \frac{A_1}{A} \qquad L_2 = \frac{A_2}{A} \qquad L_3 = \frac{A_3}{A} \tag{3.15}$$

where A is the area of the triangle.
As it can be expected only two area coordinates are sufficient to define the position

of a point inside the triangle. Indeed, the three coordinates are not independent:

$$L_1 + L_2 + L_3 = 1 \tag{3.16}$$

The mapping between the area coordinates and the physical coordinates is:

$$\begin{array}{ll} x & = \sum_{i=1}^{n_{nod}} N_i(L_1, L_2, L_3)x_i \\ y & = \sum_{i=1}^{n_{nod}} N_i(L_1, L_2, L_3)y_i \end{array} \tag{3.17}$$

In the following there are the shape functions for linear, parabolic and cubic elements.

Linear element ($n_{nod} = 3$)

$$N_i \quad = \quad L_i \qquad 1 \leq i \leq 3 \tag{3.18}$$

Quadratic element ($n_{nod} = 6$)

$$\begin{array}{lll} N_1 & = & (2L_1 - 1)L_1 \qquad N_2 = (2L_2 - 1)L_2 \qquad N_3 = (2L_3 - 1)L_3 \\ N_4 & = & 4L_1L_2 \qquad N_4 = 4L_2L_3 \qquad N_6 = 4L_1L_3 \end{array} \tag{3.19}$$

Cubic element ($n_{nod} = 10$)

$$\begin{array}{lll} N_1 & = & \frac{1}{2}L_1(3L_1 - 1)(3L_1 - 2) \qquad N_2 = \frac{1}{2}L_2(3L_2 - 1)(3L_2 - 2) \\ N_3 & = & \frac{1}{2}L_3(3L_3 - 1)(3L_3 - 2) \qquad N_4 = \frac{9}{2}(3L_1 - 1)L_1L_2 \\ N_5 & = & \frac{9}{2}(3L_2 - 1)L_1L_2 \qquad N_6 = \frac{9}{2}(3L_2 - 1)L_2L_3 \\ N_7 & = & \frac{9}{2}(3L_3 - 1)L_2L_3 \qquad N_8 = \frac{9}{2}(3L_3 - 1)L_1L_3 \\ N_9 & = & \frac{9}{2}(3L_1 - 1)L_1L_3 \qquad N_{10} = 27L_1L_2L_3 \end{array} \tag{3.20}$$

The position of the nodes on the reference triangle is described in [34]. Note that the shape function N_9 for cubic triangles reported in [34] contains a typo.

3.4 Choice of the basis

Several tests have been carried out with both BASIS A and BASIS B. From these experiments, some guidelines have been established. When complex geometries are studied, triangular elements with the BASIS B can be interesting thanks to their ability to adapt to the geometry. On the other hand, quadrilateral elements with the BASIS A can be very efficient in some problems, especially when structured mesh are employed.

In order to show the effects related to the choice of the basis, some results on the inviscid flow in a channel with a smooth bump are reported in the following. More details on this test case can be found in Chapter 8.2.1. Two sequences of meshes have been used for this experiment: a sequence of unstructured meshes with quadrilateral elements and a sequence of unstructured meshes with triangular elements. The elements at wall are cubic in all the meshes. A reconstruction order $p = 2$ is assigned in all the elements and the Osher [35] numerical flux is used to evaluate the convective fluxes. In Figure 3.3 and Figure 3.4 the initial meshes are reported: finer meshes are obtained by isotropic splitting. In Figure 3.2 the Mach field on one of the finest meshes is reported. In Figure 3.5 a plot of the L2 entropy error versus the equivalent length scale is reported. The equivalent length scale h is defined in Appendix B. Results are proposed for three different discretizations: quadrilateral elements with BASIS A, quadrilateral elements with BASIS B and triangular elements with BASIS B. It is clear that the BASIS A is much more convenient than the BASIS B in the presence of quadrilateral elements.

The comparison between the results obtained with triangles and quadrilaterals is more difficult, even in this simple test case. Indeed, when the nDOFs is low the quadrilaterals with the BASIS A are significantly more convenient than the triangles with the BASIS B. However, this behavior changes when the nDOFs is increased. In conclusion, these results show that there is not a choice which is always the best for all the situations. Triangles with BASIS B or quadrilaterals with BASIS A can be both good choices, depending on the problem which is under study. The only certain guidelines that can be obtained from this test is that BASIS A is more efficient than BASIS B when quadrilaterals are employed.

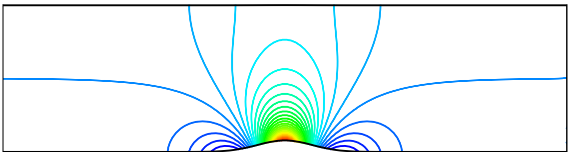

Figure 3.2. Mach field for the inviscid subsonic flow in a channel.

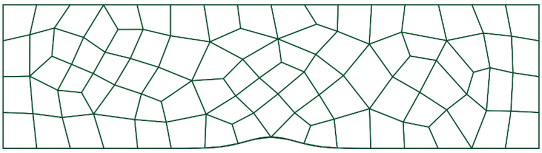

Figure 3.3. Unstructured mesh with cubic quadrilateral elements.

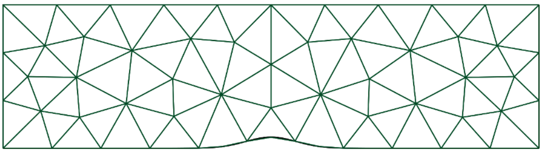

Figure 3.4. Unstructured mesh with cubic triangular elements.

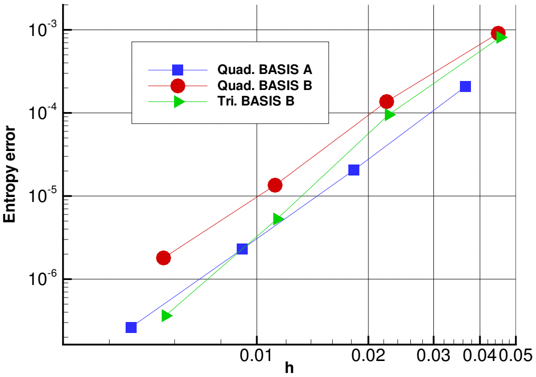

Figure 3.5. Entropy error vs equivalent length scale for the inviscid channel flow (p=2)

3.5 Convective fluxes

The convective fluxes represent the transport of conservative variables due to the inviscid terms of the equations. In this work, they coincide with the fluxes of the Euler equations. The numerical approximation of the convective fluxes in the DG framework can be done with one of the several numerical fluxes developed for finite volume schemes.

Since Euler equations are characterized by wave propagation phenomena, a large family of upwind numerical fluxes has been developed through the years. In the code developed for this work, four upwind fluxes have been implemented: Osher, AUSM+, Roe and rotated RHLL. These fluxes take into account the propagation of characteristic signals and introduce a numerical dissipation which is controlled by physical phenomena associated to the hyperbolic nature of the Euler equations.
The Osher flux was proposed by Osher and Solomon [35]. In this work, it has been implemented following the approach proposed by Pandolfi [36]. This numerical flux is based on an approximate solution of a Riemann problem at the interface between two elements. First of all, kinematic variables are projected in the normal and tangential direction of the interface. Then the Riemann problem is approximately solved in this frame of reference. The main idea of this method is to treat the compression waves which can appear in the Riemann problem as isentropic waves. In this way it is possible to obtain an algebraic solution of the Riemann problem based on compatibility relations. This solution shows how signals propagate across the interface and how they contribute to the value of the numerical flux.
The AUSM+ [37] flux is one of the several upgraded versions of the original Advection Upstream Splitting Method (AUSM) [38]. It is an upwind method in which the advection and the pressure contributions to the flux are computed separately.
The Roe [39] flux is based on the exact solution of a linearized Riemann problem. In the original version, this numerical flux gives unphysical expansion shocks. In order to prevent this behavior, the entropy fix proposed by Harten [**?**] has been implemented in this work.
The rotated RHLL [41] numerical flux is obtained by combining together the Roe flux and the HLL flux with a rotated-Riemann solver approach. The purpose of this approach is to obtain a numerical flux which is very robust for nonlinear instabilities. Indeed, it does not suffer from the carbuncle problem in the presence of strong shock waves. More details on the carbuncle problem can be found in [42].

On the other hand, several numerical fluxes based on the central approach have been used in the framework of DG methods. For example, the local Lax-Friedrichs ([43],[44]) flux is widely used and it has been implemented in this work. The great advantage of this flux is related to its simplicity. On the other hand, it is one of the most dissipative numerical fluxes.

In the following, a comparison between the local Lax-Friedrichs and the Osher fluxes is reported. Computations were carried out on the inviscid channel problem (see Chapter 8.2.1) with the mesh of Figure 3.3. The L2 entropy error is reported as a function of the reconstruction order p in Figure 3.6. It can be seen that the Osher flux gives an error which is always lower than the error given by the local Lax-Friedrichs flux. However, when the order is increased, the absolute difference between the two errors becomes negligible.

This behavior is true for smooth flows. However, when there are discontinuities in the solution the upwind fluxes can show further advantages. Indeed, Wheatley et al. [45] compared several numerical fluxes on discontinuous problems with DG discretizations. They observed that in the presence of shock waves there is a degradation in the numerical order of accuracy of the DG method and so it behaves like low order methods. In this situation, the better accuracy of upwind solvers becomes important.

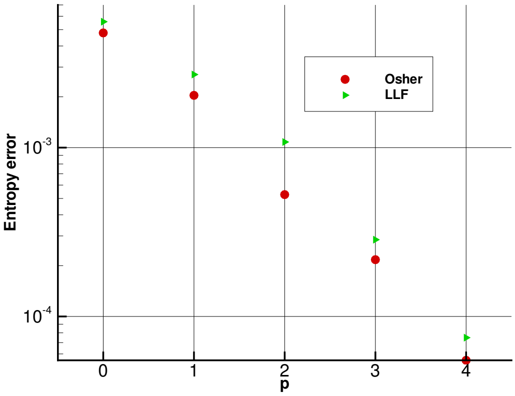

Figure 3.6. Entropy error vs reconstruction order for Osher and Local Lax-Friedrichs (LLF) fluxes.

Chapter 4

Diffusive fluxes

In the DG approach the global solution is not continuous but has jumps at the interfaces between the elements. At first glance, these methods may seem quite unsuitable for the discretization of elliptic equations because the discontinuous behavior of the numerical solution does not match well with the smoothness of the solution of an elliptic problem. However, the application of the DG framework to the solution of elliptic problems has proved to be remarkably successful. Several approaches have been proposed for the study of elliptic problems in the DG framework, see [46] for a comprehensive review. The first works related to these problems appeared several years ago. Lions [47] proposed the use of a penalization approach for the imposition of Dirichlet boundary conditions (BCs) in elliptic problems. His idea was to substitute the exact Dirichlet BC with a weak BC which reproduces the same effect of the original BC when a penalization factor tends to infinity. Nitsche [48] proposed an improved technique which was proven to be consistent. These ideas were subsequently adopted to impose inter-element continuity in the context of internal penalization methods (see e.g. the work of Babuska and Zlamal [49]) and were used to compute the diffusive fluxes exchanged between adjacent elements. Bassi and Rebay [50] proposed a DG method (BR1) for the discretization of the compressible Navier-Stokes equations which was subsequently generalized by Cockburn and Shu [51] with the Local Discontinuous Galerkin (LDG) method. Bassi, Rebay et al. [4] proposed also a second version of their scheme (BR2) in which a compact stencil is obtained thanks to the use of lifting operators. Similar motivations drove Peraire and Persson [**?**] to introduce the Compact Discontinuous Galerkin (CDG) method which can be seen as a compact evolution of the LDG. A contribution was also given by Baumann and Oden [52] with a method based on a non-symmetric bilinear form. A different path was chosen by Gassner, Lorcher and Munz [53] who proposed a numerical flux for the diffusion terms in finite volume or DG schemes which is based on an exact solution of the diffusion equation with piecewise polynomial initial data.

In 2005 van Leer and Nomura [3] proposed the recovery method (RDG) in which diffusive fluxes are computed after a recovery procedure which gives a smooth solution that in the weak sense is indistinguishable from the piecewise continuous discrete solution. A similar approach was proposed by Dumbser [54] with the $P_N P_M$ scheme in which both convective and diffusive fluxes are computed through a reconstruction of order higher than the local element reconstruction. Luo et al. [55] presented a work in which they applied a reconstruction method for the Navier-Stokes equations, following the approach originally introduced by Dumbser et al. [56] in the context of Euler equations.

Other methods recently proposed exploit the idea of recovery. Huynh [57] performed a stability study for several of these schemes. He also proposed a 1D method which can be extended to 2D on quadrilateral meshes via tensor products. Borrel and Ryan [58] proposed the elastoplast method in which the recovery idea is applied to a small rectangle overlapping the interface between two elements. French et al. [59] introduced a modified recovery scheme which makes use of a penalty term. Moreover, in time several versions of the original recovery scheme were proposed by van Leer and co-workers. While the first version was based on a weak formulation twice partially integrated (RDG2x), a second version based on a weak formulation (RDG1x), requiring a single partial integration, was subsequently introduced. Lo and van Leer [60] suggested that this second version could be more suited to the discretization of the non linear Navier-Stokes equations, thanks to some simplifications in the switching to primitive variables. For this reason, the scheme proposed in this work will be compared mainly with the RDG1x scheme.

In this brief historical review it is possible to see some common lines which have driven the evolution of DG methods for elliptic problems. One of these is the compactness of the discretization. A compact stencil is very useful to simplify the implementation of codes on unstructured grids, and to improve scalability in parallel computations. Other important aspects are flexibility and robustness in dealing with highly distorted meshes and p-adaptive algorithms.

During the research activity a new method (ESR) for the computation of diffusive terms has been developed. It has been derived from the original recovery method (RDG1x) focusing the attention on high order discretizations of convection-diffusion problems.

In particular, the ESR method is characterized by a recovery basis which makes it flexible and allows a robust implementation of p-adaptive algorithms. This basis has less terms than the original recovery basis and remains well conditioned also in the presence of highly distorted meshes. Furthermore the proposed discretization shows a larger stability limit than the RDG1x method and the implementation is simplified (Section 4.3.6). Moreover, a general procedure for the weak imposition of Dirichlet BCs in the framework of recovery-based methods (Section 4.3.3) is proposed and tested in the presence of curvilinear wall elements. The proposed

boundary procedure is truly local and acts only on the boundary element. In contrast, the original procedure [61] proposed for the RDG1x method needs information also from the neighbouring elements.
The results obtained with the proposed discretization were compared not only with the original RGD1x method but also with the widely used BR2 method, which represents a reference in this field.

4.1 BR2 method

In this Section the BRMPS method introduced by Bassi, Rebay, Mariotti, Pedinotti and Savini [4] is described. The method is also known in the literature as BR2 since it can be seen as an evolution of the BR1 method proposed by Bassi and Rebay [50]. However, the BR2 method is compact and consistent for pure diffusion problems while the BR1 method is not.
The BR2 method has been implemented in the code in order to make a comparison with the proposed ESR approach.
The main idea of the BR2 approach is to introduce lifting operators which take into account the influence of the jumps at the interfaces when computing the gradients required by diffusive fluxes. Here the DG formulation already proposed in Eq.3.9 is reported for a diffusion problem after the application of the BR2 method:

$$\begin{aligned}[\mathbf{M}]\frac{\partial \tilde{\mathbf{u}}}{\partial t} = &-\int_{\partial\Omega_e}\left(\hat{F}_h(\mathbf{u}_h^{\pm},\mathbf{z}_h|_{\partial e}^{\pm})n_x+\hat{G}_h(\mathbf{u}_h^{\pm},\mathbf{z}_h|_{\partial e}^{\pm})n_y\right)\mathbf{\Phi}ds+\\ &+\int_{\Omega_e}\left(\frac{\partial \mathbf{\Phi}}{\partial x}F_h(\mathbf{u}_h,\mathbf{z}_h)+\frac{\partial \mathbf{\Phi}}{\partial y}G_h(\mathbf{u}_h,\mathbf{z}_h)\right)dxdy+\int_{\Omega_e}Q\mathbf{\Phi}dxdy\end{aligned} \tag{4.1}$$

For simplicity, a pure diffusion problem is considered and so F and G denote diffusive fluxes. In the presence of convection-diffusion equations the convective fluxes are computed separately as described in Section 3.5.

In Eq. 4.1 the dependence of the diffusive fluxes on the element solution ($\mathbf{u}_h$) and on the modified gradients ($\mathbf{z}_h$ and $\mathbf{z}_h|_{\partial e}^{\pm}$) is stated. In the BR2 method the fluxes across the faces of the element ($\hat{F}_h$ and $\hat{G}_h$) are computed in a central way as:

$$\hat{F}_h = \frac{F(\mathbf{u}_h^{+},\mathbf{z}_h|_{\sigma}^{+})+F(\mathbf{u}_h^{-},\mathbf{z}_h|_{\sigma}^{-})}{2} \tag{4.2}$$

$$\hat{G}_h = \frac{G(\mathbf{u}_h^{+},\mathbf{z}_h|_{\sigma}^{+})+G(\mathbf{u}_h^{-},\mathbf{z}_h|_{\sigma}^{-})}{2} \tag{4.3}$$

where σ is a generic face of the element and the data from the two side of the face are denoted by ±.

The modified gradients are computed as:

$$\mathbf{z}_h = \nabla_h \mathbf{u}_h + \mathbf{R} \qquad \mathbf{z}_h|_\sigma = \nabla_h \mathbf{u}_h + \eta_\sigma \mathbf{r}_\sigma \tag{4.4}$$

The previous equation shows that the gradients are obtained by the gradient of the element solution ($\nabla_h \mathbf{u}_h$) augmented by the global ($\mathbf{R}$) or local ($\mathbf{r}_\sigma$) lifting operators. The coefficient η_σ is a penalty parameter which should be chosen larger than the average number of faces of the elements sharing the interface σ on which the flux is evaluated.
For each face of the element a local lifting operator $\mathbf{r}_\sigma : [L^2(\sigma)]^2 \to [\mathbb{P}_2^p(T_h)]^2$ is defined by the following integral relation:

$$\int_{\Omega_e} \boldsymbol{\tau}_h \cdot \mathbf{r}_\sigma(\mathbf{v}) dxdy = -\int_\sigma (\frac{\boldsymbol{\tau}_h^+ + \boldsymbol{\tau}_h^-}{2}) \cdot \mathbf{v} ds \qquad \forall \boldsymbol{\tau}_h \in [P_2^k(T_h)]^2 \tag{4.5}$$

where Eq. 4.5 is valid for all $\mathbf{v} \in [L^2(\sigma)]^2$.
The global lifting operator $\mathbf{R}$ is the sum of all the local lifting operators of the element:

$$\mathbf{R}(\mathbf{v}) = \sum_{\forall \sigma} \mathbf{r}_\sigma((v)) \tag{4.6}$$

The BR2 method in convection-diffusion problems takes advantage of the facts that the jumps at the interfaces are already computed for the evaluation of the convective fluxes and so they can be immediately used to compute the local lifting operators. Furthermore, all the functions which are used can be expressed in terms of element basis functions and so no additive functions have to be introduced: this is very useful in problems with deforming domains. In contrast, the ESR method which will be described in the following requires the definition of recovery functions which depends on the geometry and so have to be recomputed at each time step when the mesh changes in time.

4.2 Original recovery method

The recovery method (RDG) proposed by van Leer and Nomura [3] is an elegant way to obtain the numerical diffusive fluxes.

Consider the interface between two elements Ω_A and Ω_B. The goal is to recover a smooth differentiable function (u^r) on the union of these two elements ($\Omega_C = \Omega_A \cup \Omega_B$) in order to compute gradients on the quadrature points along the interface. The key idea is that the recovered function has to be equivalent in the weak sense to the original solution on the two elements, namely

$$\int_{\Omega_A} u^r \Phi_n dxdy = \int_{\Omega_A} u_h \Phi_n dxdy \qquad 1 \le n \le N_A$$

$$\int_{\Omega_B} u^r \Phi_n dxdy = \int_{\Omega_B} u_h \Phi_n dxdy \qquad 1 \leq n \leq N_B \tag{4.7}$$

A set of basis functions $\boldsymbol{\Psi}$ with N_r degrees of freedom can be introduced for the recovered solution:

$$u^r = \tilde{\mathbf{u}}^{\mathbf{r}} \cdot \boldsymbol{\Psi} = \sum_{j=1}^{N_r} \tilde{u}_j^r(t) \Psi_j(x,y) \tag{4.8}$$

Substituting (4.8) and (3.3) into (4.7) the following two sets of linear equations are obtained:

$$[\mathbf{R}_A]\, \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{M}_A]\, \tilde{\mathbf{u}}_A$$

$$[\mathbf{R}_B]\, \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{M}_B]\, \tilde{\mathbf{u}}_B \tag{4.9}$$

in which $[\mathbf{M}_A]$, $[\mathbf{M}_B]$, $\tilde{\mathbf{u}}_A$ and $\tilde{\mathbf{u}}_B$ are the mass matrices and the vectors of the degrees of freedom of the two elements A and B, respectively. The recovery matrices $[\mathbf{R}_A]$ and $[\mathbf{R}_B]$ are defined as

$$[\mathbf{R}_A]_{nj} = \int_{\Omega_A} \Psi_j \Phi_n dxdy \qquad 1 \leq n \leq N_A, \quad 1 \leq j \leq N_r$$

$$[\mathbf{R}_B]_{nj} = \int_{\Omega_B} \Psi_j \Phi_n dxdy \qquad 1 \leq n \leq N_B, \quad 1 \leq j \leq N_r. \tag{4.10}$$

Introducing the interface recovery matrix $[\mathbf{R}]$, the set of equations (4.9) can be expressed in a more compact form:

$$\begin{bmatrix} [\mathbf{R}_A] \\ \hline [\mathbf{R}_B] \end{bmatrix} \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{R}]\tilde{\mathbf{u}}^{\mathbf{r}} = \begin{bmatrix} [\mathbf{M}_A] & 0 \\ 0 & [\mathbf{M}_B] \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \hline \tilde{\mathbf{u}}^{\mathbf{B}} \end{Bmatrix}. \tag{4.11}$$

If an orthonormal basis is used, the mass matrix within each element is the identity, and so the matrix multiplication on the right hand side can be removed,

$$[\mathbf{R}]\tilde{\mathbf{u}}^{\mathbf{r}} = \begin{Bmatrix} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \hline \tilde{\mathbf{u}}^{\mathbf{B}} \end{Bmatrix}. \tag{4.12}$$

From the previous derivation it can be seen that the recovery matrix $[\mathbf{R}]$ has dimension $(N_A + N_B) \times N_r$. The most difficult task in the recovery approach is the choice of the recovery basis $\boldsymbol{\Psi}$.

Van Leer and Lo [61] considered the case in which $N_A = N_B$, and they set $N_r = (N_A + N_B)$. They introduced a provisional recovery basis $\boldsymbol{\Psi}$ which makes the recovery matrix non singular. The elegance of their idea is related to the next step.

They used the columns of the inverse recovery matrix to define a final recovery basis $\hat{\mathbf{\Psi}}$ which is in the weak sense indistinguishable from the elements discontinuous basis (van Leer and Lo [62]). In this way the degrees of freedom of the two elements which share an interface can be used to put in evidence both the discontinuous hyperbolic nature or the continuous parabolic nature of the underlying equations simply by switching from a basis ($\mathbf{\Phi}$) to another ($\hat{\mathbf{\Psi}}$).

$$u^r = \sum_{j=1}^{N_A} \tilde{u}_j^A \hat{\Psi}_j + \sum_{j=1}^{N_B} \tilde{u}_j^B \hat{\Psi}_{N_A+j}. \tag{4.13}$$

4.3 Enhanced stability recovery method

4.3.1 Derivation of the method

The approach proposed by van Leer, Nomura and van Raalte [61] makes it possible to enhance the accuracy of the viscous fluxes discretization because the recovered solution has twice the number of degrees of freedom with respect to the element solution. This advantage is important in pure diffusion problems but can be lost in convection-diffusion problems if the convective fluxes are computed using the element internal reconstruction. There are some approaches in which the information from the neighbouring elements is used to improve also the accuracy of the convective terms, see for example the PNPM method [54] or the approach proposed by Nourgaliev et al. [63]. One of the main reasons for the current success of DG methods is the possibility of obtaining high order reconstructions without the need of data from neighbouring elements. For this reason, a classical approach has been adopted in this work and so the convective fluxes are computed using the internal element reconstruction. As a consequence the effects of a higher order approximation of the diffusion terms are mitigated by the lower order approximation of the convective terms.

Furthermore, in the original recovery approach it is necessary to choose a provisional recovery basis (Ψ) which makes the recovery matrix ($[\mathbf{R}]$) square and invertible. This is not a trivial task in multidimensional problems, especially when high order reconstructions are used on irregular unstructured meshes. For example, van Leer et al. [64] reported that the RDG1x scheme becomes unstable for $p > 2$, if the gradient in the volume integral is computed with one of the approaches described in [60]. In contrast, the discretization proposed in this work was successfully tested up to $p = 6$ (see Section 4.4.3).

Van Leer, Nomura and van Raalte [61] chose to use an anisotropic monomial recovery basis in which the reconstruction orders in the direction normal and tangential to the interface are $(2p + 1)$ and p, respectively. Even if this approach improves the order of the scheme on cartesian meshes, a standard $(p + 1)$ convergence order

is obtained on general unstructured meshes [62].
One of the final goals of this work is the discretization of convection-diffusion problems in which p-adaptivity strategies can bring important benefits. In this context, it is not clear how to extend the RDG1x approach (Eq. 4.13 which requires $N_r = N_A + N_B$) to problems in which the reconstruction order changes from one element to another ($N_A \neq N_B$). In the literature there are no applications of the original recovery approach with p-adaptive algorithms to the best of our knowledge.
Thus an alternative approach has been chosen in this work in order to obtain a general and flexible scheme (Enhanced Stability Recovery (ESR) [2]), which can be easily adopted in the context of p-adaptive algorithms. In particular the constraint of a square recovery matrix is removed (i.e. it is set $N_r < N_A + N_B$) and the over-determined system (4.12) is solved by the least square method:

$$[\mathbf{R}]^T[\mathbf{R}]\tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{R}]^T \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} \tag{4.14}$$

Thanks to a proper choice of the recovery basis (defined in the following) the least square problem is well conditioned and the matrix $[\mathbf{R}]^T[\mathbf{R}]$ is invertible:

$$\tilde{\mathbf{u}}^{\mathbf{r}} = \left([\mathbf{R}]^T[\mathbf{R}]\right)^{-1} [\mathbf{R}]^T \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} = [\tilde{\mathbf{R}}] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} \tag{4.15}$$

In this way it is sufficient to perform a matrix-vector product for each interface and the recovery coefficients are obtained. Then these coefficients can be multiplied by the gradient of the recovery basis functions in order to obtain the diffusive fluxes at the interface quadrature points (first term on the right hand side of Eq. (3.9)). If the domain is fixed, the rectangular recovery matrix ($[\tilde{\mathbf{R}}]$) can be precomputed since it depends only on the mesh. In Section 4.3.6 more details on the implementation are given.
In the proposed approach, the recovered solution is used only for the evaluation of the diffusive fluxes at the interface. Indeed, the diffusive fluxes within the volume integral (second term on the right hand side of Eq. (3.9)) are computed inside each element, and therefore are given by the product between the element degrees of freedom and the gradient of the element basis functions.
The most important aspect of the method is the choice of the recovery basis Ψ. An orthonormal basis obtained by the modified Gram-Schmidt procedure is used for this purpose. The orthonormalization uses the L^2 inner product $(f,g) = \int_{\Omega_A \cup \Omega_B} fg\,dxdy$ in which the integral is evaluated on the union of the two elements. This approach makes it possible to ensure that the condition number of the recovery system remains low.
In this work, the orthonormalization procedure is initialized with a particular starting monomial basis defined according to the following rules, depending on the degree p_A and p_B of the polynomials in the two elements A and B, defining the

interface. Three cases will be described depending on the order of the two elements. Consider a reference frame centered on the mid-point of the interface and with the axis aligned to the normal and tangential directions (ξ, η) as shown in Figure 4.1:

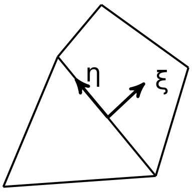

Figure 4.1. Interface reference frame for constructing the recovery basis.

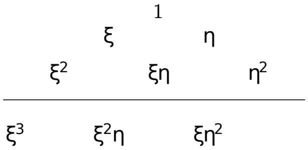

Figure 4.2. Preliminary recovery basis before orthonormalization for the case $p_A = p_B = 2$, in the Enhanced Recovery approach.

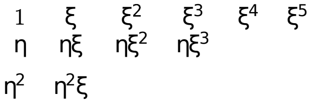

Figure 4.3. Preliminary recovery basis with the original recovery approach for the case $p_A = p_B = 2$.

- $p_A = p_B = p$
  First of all we consider the case in which the two elements have the same order p. The starting recovery basis for the initialization of the orthonormalization procedure is obtained by considering the monomial basis of order p in the variables (ξ, η) and adding all the monomials with order $p+1$ in which ξ appears. In other words these additive functions consist in the first $p+1$ monomials of the $(p+1)th$ row of the Pascal triangle. The tests done in this work have shown that these additive functions make the difference. Indeed, the preliminary experiments showed that if the additive functions are not used or if only the term ξ^{p+1} is included, the convergence rate of the scheme can be compromised.

Since the order of the recovery reconstruction is reduced with respect to the original recovery approach, it is easier to control the condition number of the recovery matrix. Moreover, this choice is related to the improvement in the stability limit observed in the numerical experiments presented below.
An example of the proposed starting basis in the case $p = 2$ is shown in Figure 4.2. For the sake of comparison, Figure 4.3 reports the provisional basis for $p = 2$ obtained according to the original recovery approach.

- $|p_A - p_B| = 1$
  The starting recovery basis is obtained by considering the monomial basis of order $\tilde{p} = \max(p_A, p_B)$ and adding all the monomials of order $\tilde{p} + 1$ in which ξ appears. It is possible to use the maximum order between p_A and p_B if there is enough information in the union of the two elements to obtain a recovered solution with this accuracy, i.e. $N_r \leq N_A + N_B$. This condition is satisfied if $min(p_A, p_B) \geq 1$. Thus, the p-adaptive strategy avoids the use of elements with $p = 0$.

- $|p_A - p_B| > 1$
  It could happen that a p-adaptive algorithm introduces a jump of two or more orders between neighbouring elements. In this case it is impossible to choose the recovery basis with $\tilde{p} = max(p_A, p_B)$. Indeed, in the case $|p_A - p_B| > 1$, the recovery degrees of freedom would be more than the sum of the degrees of freedom in the two elements if $\tilde{p} = max(p_A, p_B)$. For this reason, the order of the recovery is locally reduced, choosing $\tilde{p} = max(p_A, p_B) - 1$, i.e. at the interface between elements with a different accuracy, the algorithm chooses an intermediate value for the degree of the polynomial being reconstructed. This reflects the fact that the adaptive algorithm assigns a jump in the accuracy across the interface, so it seems reasonable to choose an intermediate state at the interface. Thus, in this case, the starting basis is built from a monomial basis of order $\tilde{p} = \max(p_A, p_B) - 1$ plus all the monomials of order $\tilde{p} + 1$ in which ξ appears. In the performed tests it was found that the presence of strong jumps in the order distribution across the domain should be prevented whenever possible (see Section 7.2.1). For this reason, the condition $|p_A - p_B| > 1$ should be avoided during the simulations by choosing a more regular order distribution. However, if the condition is verified the algorithm is able to manage this case, as described above.

Therefore, the starting recovery basis has always $N_r = (\tilde{p}+1)(\tilde{p}+2)/2 + (\tilde{p}+1)$ degrees of freedom, where $\tilde{p}$ is equal to p if a uniform order distribution is used. Otherwise $\tilde{p}$ can be set equal to $max(p_A, p_B)$ or $max(p_A, p_B) - 1$, depending on

the jump in the order.
As far as numerical quadrature is concerned, the integral on the interface is computed with a suitable quadrature formula in order to satisfy the accuracy requirements of both elements. In particular, if the two elements require M_A and M_B points for the integrals on their edges, the formula with $M = max(M_A, M_B)$ is adopted.

Finally, a comment on the use of piecewise constant reconstructions. When $p_A = p_B = 0$ the recovery reconstruction has only two degrees of freedom: a constant term and a linear term related to the direction normal to the interface. From this information, only the normal derivative can be obtained from the recovery solution. If the heat equation is considered this information is enough to compute the fluxes across the interface. On the other hand, if a more complex system of equations is considered (i.e. Navier-Stokes), not only the normal derivative but also the tangential derivative has to be computed in order to evaluate the fluxes. For this reason, the presence of elements with $p = 0$ is avoided in the domain. Note that the DG method is usually chosen for its high accuracy. Thus, the occurrence of piecewise constant reconstructions is expected to be unlikely in most applications.

It is important to emphasize the use of the Gram-Schmidt orthonormalization not only on the element basis but also on the recovery basis. This approach makes the method robust and able to deal with highly distorted meshes. For example the effect of the recovery basis orthonormalization on the 2-norm condition number of the matrix $\left([\mathbf{R}]^T[\mathbf{R}]\right)$ which has to be inverted in Eq. 4.15 is reported in Table 4.1. The data refer to the interface between the two distorted elements shown in Figure 4.4 for the case $p_A = p_B$. Orthonormalization is applied on the element basis functions in both cases, but only in the second case it is also applied to the recovery basis. These results suggest that the use of orthonormalization on the proposed recovery basis becomes fundamental when high-order reconstructions are employed on irregular meshes. The condition number of the matrix was estimated with the free software package GNU Octave [65].
It can be expected that the use of the orthonormalization procedure on the recovery basis can be useful also for other recovery-based schemes, since they all involve the definition of a recovery procedure which is influenced by the behavior of the recovery basis.

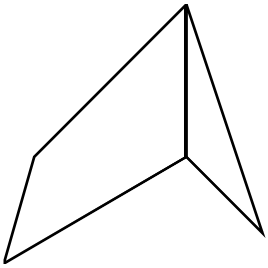

Figure 4.4. Interface between two distorted elements for the test of Table 4.1.

Scheme	Cond. number without ON	Cond. number with ON
DG1	4.61E+02	1.97E+00
DG2	1.05E+04	3.87E+00
DG3	2.52E+05	5.36E+00
DG4	6.25E+06	6.62E+00
DG5	1.58E+08	7.87E+00
DG6	5.03E+09	9.09E+00

Table 4.1. Condition number of the matrix $\left([\mathbf{R}]^T[\mathbf{R}]\right)$ for the interface of Figure 4.4 without and with orthonormalization (ON) of the recovery basis

4.3.2 Extension to 3D

In this Section some considerations on the extension of the method to 3D are reported. In particular it is necessary to define how to build the recovery basis in 3D. Consider a reference frame with the origin in the geometric center of the interface. Its axis are aligned with the normal and tangential directions (ξ, η, ζ).
First of all, the case $p_A = p_B = \tilde{p}$ is studied. The starting recovery basis can be obtained by considering the monomial basis of order $\tilde{p}$ and adding all the monomials with order $\tilde{p}+1$ in which ξ appears. With this approach the size of the recovery basis is $N_r = (\tilde{p}+1)(\tilde{p}+2)(\tilde{p}+3)/6 + (\tilde{p}+1)(\tilde{p}+2)/2$.
The cases $|p_A - p_B| = 1$ and $|p_A - p_B| > 1$ can be managed in a way similar to the 2D approach.
The same problems described for $p = 0$ in the 2D context are present also in 3D. Another important issue which can become important in 3D is the memory requirement. In Section 4.3.6 there are suggestions on implementation details which can help in saving memory. Future work will be devoted to the assessment of the method in 3D.

4.3.3 BCs for ESR

In this Section the procedure for the imposition of boundary condition for diffusive equations is described in the framework of the Enhanced Recovery Method. Neumann BCs are simply satisfied by setting the exact value of the fluxes on the boundary integral. Dirichlet BCs require further considerations.

In the framework of discontinuous Galerkin methods it is possible to impose Dirichlet BCs in both strong or weak form [66]. The strong approach consists in forcing the solution reconstruction to exactly satisfy a prescribed condition on the boundary. The weak approach is based on the idea that the error in the enforcement of the BC's and the error in the interior solution can be of the same size. In other words, the accuracy required in the interior of the domain is the same as the accuracy imposed at the boundary: in this approach therefore there is an error between the value prescribed and the solution reconstructed on the boundary, but this error vanishes in the limit of an infinite mesh refinement, with the same rate of the error in the interior of the domain.

From a practical point of view, it can be shown that the use of strong Dirichlet BCs can introduce stability limitations and spurious oscillations in the numerical solution (see for example [67]). Furthermore several discretization methods allow the weak imposition of Dirichlet BCs in a natural way. For example, in the BR2 scheme this is simply done through the definition of a proper ghost state in all the boundary quadrature points. The extension to this approach in recovery-based

schemes could be non trivial. Indeed, the recovery procedure is applied to the union of two elements and so a ghost element would be required in this case, not just a local ghost state. The definition of the proper values in the ghost element can be obtained through symmetry considerations between the boundary element and the ghost element. For example Borrel and Ryan use this approach in their paper [58] on the elastoplast method. The problem is that this procedure may become ambiguous in the presence of curvilinear elements, because in this case the boundary curve is not a symmetry axis between the boundary element and the ghost element.
Another path was followed by van Leer, Nomura and van Raalte [61]. They proposed to impose the BCs on the recovery solution. They substitute the missing information in the boundary element with data from one or more interior elements. They observed that this procedure may reduce the stability range of the method. Furthermore this approach is not trivial when unstructured meshes are used, because there are multiple possible choices.

Two of the main reasons for the success of DG methods are the compactness of the reconstruction and the simplification in the treatment of BC's when high order reconstructions are employed. In order to keep these positive features, an alternative approach has been developed in this work which makes it possible to impose both strong or weak Dirichlet BC's in recovery-based DG methods, without the need to use ghost elements or data from other interior elements.

4.3.4 Dirichlet BC's in strong form

In the proposed approach boundary conditions are imposed on the element internal reconstruction, not on the recovery solution.
The strong imposition of Dirichlet BC's requires that the numerical solution at the quadrature points of the boundary elements is equal to the prescribed values. In other words, the degrees of freedom inside the element have to be altered in order to satisfy the BC's. Consider for example a discontinuous Galerkin scheme with N_e degrees of freedom inside the element and M quadrature points on the edge ($M < N_e$). The imposition of the BC's on the quadrature points gives M equations for each element on the boundary:

$$u_w^m = u_w(\mathbf{x}_q^m) = \sum_{i=1}^{N_e} (\tilde{u}_i + \Delta\tilde{u}_i)\Phi_i(\mathbf{x}_q^m) \qquad 1 \le m \le M \tag{4.16}$$

where u_w^m represents the prescribed wall value in the quadrature point m.
If the procedure is forced to be conservative the mean value has to be preserved ($\Delta\tilde{u}_1 = 0$). In this way the system (4.16) has $N_e - 1$ unknowns which describe the perturbation of the elements degrees of freedom ($\Delta\tilde{u}_i, 2 \le i \le N_e$). In general

the set of equations (4.16) is under-determined and so more information has to be imposed. In this work it has been chosen to minimize the L^2-norm of the perturbation $\boldsymbol{\Delta\tilde{u}}$:

$$min(|\boldsymbol{\Delta\tilde{u}}|_2) = min(\sum_{i=2}^{N_e} \Delta\tilde{u}_i^2). \tag{4.17}$$

The minimization problem can be solved using a Lagrange multiplier, μ. In this approach, equation (4.17) is the goal and equations (4.16) are the constraints:

$$\frac{\partial}{\partial\Delta\tilde{u}_i}\left[\sum_{i=2}^{N_e}\Delta\tilde{u}_i^2 + \sum_{m=1}^{M}\mu_m\left(u_w^m - \sum_{i=1}^{N_e}(\tilde{u}_i + \Delta\tilde{u}_i)\Phi_i(\mathbf{x}_q^m)\right)\right] = 0. \tag{4.18}$$

In other words it is necessary to look for the smallest perturbation which is able to satisfy the BC's. The set of equations (4.18) and (4.16) defines the linear system (4.19) in which there are $(N_e - 1 + M)$ equations and $(N_e - 1 + M)$ unknowns: the perturbations of the element degrees of freedom (with the exception of $\Delta\tilde{u}_1$ which is set to zero and is already known) plus the Lagrange multipliers. The boundary matrix $[\mathbf{B}]$ and the boundary vector $\{\mathbf{W}\}$ are explicitly written in (4.20) and (4.21):

$$[\mathbf{B}]\begin{Bmatrix}\Delta u_2\\ \vdots\\ \Delta u_{N_e}\\ \mu_1\\ \vdots\\ \mu_M\end{Bmatrix} = \{\mathbf{W}\} \tag{4.19}$$

$$[\mathbf{B}] = \begin{bmatrix}
2 & 0 & \cdots & 0 & -\Phi_2(\mathbf{x}_q^1) & \cdots & -\Phi_2(\mathbf{x}_q^M)\\
0 & 2 & 0 & \vdots & \vdots & \cdots & \vdots\\
\vdots & 0 & \ddots & 0 & \vdots & \cdots & \vdots\\
0 & \cdots & 0 & 2 & -\Phi_{N_e}(\mathbf{x}_q^1) & \cdots & -\Phi_{N_e}(\mathbf{x}_q^M)\\
\Phi_2(\mathbf{x}_q^1) & \cdots & \cdots & \Phi_{N_e}(\mathbf{x}_q^1) & 0 & \cdots & 0\\
\vdots & \cdots & \cdots & \vdots & \vdots & \cdots & \vdots\\
\Phi_2(\mathbf{x}_q^M) & \cdots & \cdots & \Phi_{N_e}(\mathbf{x}_q^M) & 0 & \cdots & 0
\end{bmatrix} \tag{4.20}$$

$$\{\mathbf{W}\} = \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ u_{w1} - \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i(\mathbf{x}_q^1) \\ \vdots \\ u_{wM} - \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i(\mathbf{x}_q^M) \end{Bmatrix} . \qquad (4.21)$$

The solution of this system gives the perturbations which have to be applied to the element. In particular, the Lagrange multipliers are not needed and they can be eliminated. Thus it is possible to define a reduced boundary matrix $[\hat{\mathbf{B}}]$ such that:

$$\begin{Bmatrix} \Delta u_2 \\ \vdots \\ \Delta u_{N_e} \end{Bmatrix} = [\hat{\mathbf{B}}] \begin{Bmatrix} u_w^1 - \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i(\mathbf{x}_q^1) \\ \vdots \\ u_w^M - \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i(\mathbf{x}_q^M) \end{Bmatrix} = [\hat{\mathbf{B}}] \{\hat{\mathbf{W}}\} . \qquad (4.22)$$

The reduced boundary matrix $[\hat{\mathbf{B}}]$ can be obtained by the first $N_e - 1$ rows and the last M columns of the inverse of $[\mathbf{B}]$. At this point the gradient required by the boundary fluxes can be computed with the product between the perturbed degrees of freedom and the gradient of the element basis functions.

4.3.5 Dirichlet BCs in weak form

In the previous paragraph a procedure for the strong imposition of Dirichlet BC's has been presented. The element's degrees of freedom are perturbed in such a way that the element internal reconstruction exactly satisfies the condition on the boundary. After that, the gradient of the element solution is computed at the boundary quadrature points, yielding the boundary fluxes.

A similar procedure is adopted also for the weak imposition of the BC's. The only difference is that, after the fluxes computation, the perturbation previously applied to the element is removed. In this way the BC's are weakly imposed in the sense that they determine the value of the boundary fluxes, but there is not a direct alteration of the element's degrees of freedom. The procedure can be summarized as follows

- Compute the element's perturbation $\Delta\tilde{\mathbf{u}}$ with Eq. (4.22).
- Apply the perturbations to $\tilde{\mathbf{u}}$, $\tilde{\mathbf{u}} := \tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}$.
- Compute the boundary gradients $\nabla u = \sum_{i=1}^{N_e} \tilde{u}_i \nabla\Phi_i$.
- Compute the boundary fluxes $\hat{F}_h(\nabla u)$

- Remove the perturbations $\tilde{\mathbf{u}} := \tilde{\mathbf{u}} - \Delta\tilde{\mathbf{u}}$.

All numerical results obtained with the ESR method which are presented in the following were computed with this weak enforcement of BC's.

4.3.6 Implementation and computational cost

In this Section the implementation of the method is briefly described in order to identify the main contributions to the computational cost. In particular some considerations are made in order to compare the proposed approach with the RDG1x recovery method. Both schemes are compact in the sense that the computation of viscous fluxes requires the knowledge of information only from the first neighbours of the element. For simplicity a scalar 2D problem in which all the elements have the same order (p) is considered. In this case $N_e = (p+1)(p+2)/2$ and $N_r = (p+1)(p+2)/2 + (p+1)$. The domain is supposed to be fixed, so that the mesh does not change in time. The cost required by the computation of the integrals appearing in Eq. (3.9) can be decomposed in different contributions, analyzed below.

4.3.7 Interface integral computational cost

The analysis starts from the computational cost due to the evaluation of the diffusive fluxes on an interface with M quadrature points.
If the ESR method is used the following operations are required for each interface quadrature point:

$$\nabla u = [\mathbf{D}_r]\tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{D}_r][\tilde{\mathbf{R}}]\left\{ \begin{smallmatrix} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \tilde{\mathbf{u}}^{\mathbf{B}} \end{smallmatrix} \right\} = [\mathbf{D}_r\tilde{\mathbf{R}}]\left\{ \begin{smallmatrix} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \tilde{\mathbf{u}}^{\mathbf{B}} \end{smallmatrix} \right\}.$$

The matrix with the derivatives of the recovery basis functions $[\mathbf{D}_r]$ and the recovery matrix $[\tilde{\mathbf{R}}]$ can be computed and multiplied at the beginning of the simulation since they depend only on the mesh. The cost for each interface is proportional to $M \times 2 \times 2N_e$.

In the RDG1x scheme the recovery coefficients $\tilde{\mathbf{u}}^{\mathbf{r}}$ coincide with the degrees of freedom inside the two elements and so the gradient can be obtained as

$$\nabla u = [\mathbf{D}_r]\left\{ \begin{smallmatrix} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \tilde{\mathbf{u}}^{\mathbf{B}} \end{smallmatrix} \right\}.$$

Thus, the cost for each interface is proportional to $M \times 2 \times 2N_e$ also in this case.

4.3.8 Volume integral computational cost

The computational cost of the volume integral will be evaluated assuming that L quadrature points are used.
The cost for the ESR method is related to the following operations:

- $\nabla u = [\mathbf{D}]\tilde{\mathbf{u}}$
  In each quadrature point the fluxes are computed using the internal element solution. The cost is proportional to $L \times 2 \times N_e$. The matrix with the derivatives of the element basis functions $[\mathbf{D}]$ can be precomputed since it depends only on the mesh.

The cost for the RDG1x method is related to the following operations:

- $\nabla u = \frac{1}{N_F} \sum_{i=1}^{N_F} [\mathbf{D}_r]_i \tilde{\mathbf{u}}_i^{\mathbf{r}}$
  In each quadrature point the gradient is computed as an average between all the gradients obtained by the several available recovery solutions (one for each face of the element). In this case the cost is proportional to $L \times N_F \times 2 \times 2N_e$ where N_F is the number of faces of the element. Lo and van Leer [60] proposed also another approach which improves accuracy but is more involved. In practice they use the recovery solution only for evaluating the derivative in the coordinate normal to the interface and, since there are more directions than dimensions, they introduce a least square approach. This implies that, in each internal quadrature point, a least square problem has to be solved. The cost in this case is greater than in the previous approach and the implementation is significantly more involved.

The previous considerations suggest that the main difference between the cost of the ESR method and the RDG1x method is related to the computation of the volume integral. Indeed, in the described implementation the cost per interface is the same for the two methods even if they use recovery bases with a different size. However, the RDG1x method requires more work than the ESR method in the volume integral, because it uses all the several recovery functions defined in the element (one for each face). In contrast, the ESR method is based on a very simple evaluation of the volume integral since it directly uses the gradient of the element solution. From a practical point of view, the CPU time per iteration required by the two methods might be similar but it is strongly influenced by the implementation (cache effects can significantly alter the speed) and by possible memory limitations which can forbid the storage of precomputed matrices.
Since the test cases considered in this work are 2D, there are no memory limitations, and all quantities that can be precomputed will be stored in memory at the beginning of each computation. However, memory requirements can become an important issue in 3D problems. In order to reduce memory usage it is useful to

avoid the storage of the interface recovery matrix ($[\tilde{\mathbf{R}}]$). Indeed, it is more convenient to pre-compute and store the product between the matrix of the derivatives of the recovery basis functions ($[\mathbf{D}_r]$) and the recovery matrix ($[\tilde{\mathbf{R}}]$), for each quadrature point.
In the example described in this Section a scalar problem is studied. If a system of equations is considered (i.e. Navier-Stokes) the recovery approach gives the gradients of the conservative variables. In order to compute the fluxes it is necessary to switch to the gradients of the primitive variables. This can be done by computing the values of the conservative recovered variables at the interface. In this case too, it is possible to save memory, avoiding the storage of the recovery matrix and storing the product between the matrix containing the recovery functions, evaluated at the interface, and the recovery matrix.

The previous analysis considers only a problem with a fixed grid. If deforming meshes are employed, the recovery functions and the related matrices have to be re-computed during the simulation. This additional cost can be significant for both the RDG1x and the ESR methods. In this kind of problems recovery-based methods might be less efficient than other existing approaches, like, for example, the BR2 method.

4.4 Tests and comparisons for the ESR method

4.4.1 1D Poisson equation

In order to evaluate the behavior of the proposed discretization a 1D Poisson problem was studied with a time-marching approach. The problem has the following exact solution which also defines Dirichlet BC's:

$$u(x) = 1 - x + \sin(2\pi x) \qquad 0 \leq x \leq \ 1. \tag{4.23}$$

Van Leer et al. [64] reported a comparison between several schemes (with $p = 1$ and forward Euler) for this test case. In Table 4.2 the dimensionless time step $\nu\Delta t/\Delta x^2$ obtained for the ESR method are reported and compared with the stability limit for the BR2 and then RDG1x methods from [64]. Table 4.3 shows the L^2 error on the temperature and the convergence order obtained in the present implementation for RDG1x, BR2 ($\eta_{BR2} = 1$) and the proposed ESR scheme. All results shown in the table, were obtained with $p = 1$ and forward Euler time integration.
Particular attention was dedicated to the BC's procedures originally proposed by the different authors. In particular for the RDG1x scheme, BC's are imposed on the recovery solution, exploiting information from the boundary and the first two elements close to it, as in [61]. For the BR2 scheme a proper ghost state computed as a function of the BC's is defined. For the ESR method the weak minimization

procedure proposed in Section 4.3.5 is used. The performed numerical experiments show that the ESR method has a significantly larger stability limit than the other schemes. On the other hand RDG1x gives excellent results and is fourth order accurate ($2(p+1)$) while the other two schemes are second order accurate ($p+1$). This advantage of the RDG1x method is lost on multidimensional unstructured meshes in which its convergence order reduces to $p+1$ [61]. Even if the ESR method obtains the expected convergence order, it can be seen that its error is 60% larger than the error obtained by BR2 when $p = 1$. On the other hand, the time step for ESR can be chosen 65% larger than it is possible in BR2. From this first comparison, it can be seen that in pure diffusion problems with cartesian meshes the RDG1x method is the most accurate scheme between the three tested.

Scheme	$\nu\Delta t/\Delta x^2$
RDG1x (from [64])	0.08
BR2 (from [64])	0.10
ESR	0.165

Table 4.2. 1D Poisson problem: stability limit (p=1, forward Euler)

Mesh	RDG1x L^2 error	Order	BR2 L^2 error	Order	ESR L^2 error	Order
8	8.90E-04	-	3.06E-02	-	5.48E-02	-
16	4.24E-05	4.39	8.67E-03	1.82	1.44E-02	1.93
32	2.18E-06	4.28	2.25E-03	1.95	3.69E-03	1.96
64	1.21E-07	4.17	5.66E-04	1.99	9.35E-04	1.98

Table 4.3. 1D Poisson problem: L^2 errors and convergence order (p=1)

4.4.2 1D Convection Diffusion problem

The final goal of this work is the application of the proposed discretization in convection-diffusion problems. For this reason the following 1D model problem was studied with a time marching approach:

$$\frac{\partial u}{\partial t} + 10\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = 0 \qquad 0 \le x \le 1 \qquad u(0) = 1, u(1) = 0. \tag{4.24}$$

Simulations were carried out until the residual reached machine precision. Then, the converged numerical solution was compared with the exact steady state solution

$$u(x) = \frac{\exp(10x) - \exp(10)}{1 - \exp(10)}. \tag{4.25}$$

Several computations were performed with the RDG1x, the BR2 and the ESR methods ($p = 1$) and the first order forward Euler time integration scheme. Three non uniform meshes with 5, 10 and 20 elements were used. The meshes are clustered in order to reproduce the aspect of the typical grids used in Navier-Stokes boundary layer simulations. The size of the element near the boundary $x = 1$ is $\Delta x = 0.08, 0.04$ and 0.02 for the different meshes. In Table 4.4 the L^2-errors are reported for the three schemes. While in the previous test case the RDG1x scheme gave errors which are several orders of magnitude lower than the other schemes, now the errors for the different methods are significantly closer. The ratio between the error of the ESR and the RDG1x methods is equal to approximately 3-4 for the different meshes. Furthermore the stability limit of the discretization is dominated by the diffusive terms and so the ESR method can benefit from its double time step with respect to the RDG1x (the data in Table 4.2 still hold). These results are limited to a scalar linear problem with $p = 1$ but they suggest that when convective effects appear in the equations, the global errors obtained by the ESR and the RDG1x methods become closer because the bottle neck is represented by the convective discretization. This difference could become even smaller in high Reynolds number industrial flows, in which the convective terms are particularly strong and the meshes are often highly distorted.

Mesh	RDG1x L^2 error	BR2 L^2 error	ESR L^2 error
5	5.58E-03	1.39E-02	1.77E-02
10	1.58E-03	4.25E-03	6.08E-03
20	4.25E-04	1.56E-03	1.82E-03

Table 4.4. 1D Convection-diffusion problem: L^2 errors at steady state (p=1)

4.4.3 2D Heat equation

In this Section the behavior of the proposed scheme is evaluated in 2D for several accuracies, including some very high order schemes. The linear 2D heat equation was solved in a square domain (0,1) × (0,1) with unstructured triangular meshes. The meshes were obtained starting from uniform cartesian grids in which quadrilaterals were split in triangles. The following Dirichlet BC's were prescribed:

$$\begin{cases} T = 0 & \text{on } x = 1, y = 0 \text{ and } y = 1 \\ T = \sin(k_w \pi y) & \text{on } x = 0. \end{cases} \tag{4.26}$$

For the ESR method, BC's were imposed with the weak approach, based on the minimization procedure of Section 4.3.5. The parameter k_w is an integer which represents the number of semi-waves in the y-direction. In this test it was set to

$k_w = 1$. The numerical solution was initialized with a uniform field $T_0 = 1$ and computations were carried out until the residuals reached machine precision. The L^2 norm of the temperature error was computed through a comparison between the obtained steady state results and the following exact analytical solution, which is reported in Figure 4.5:

$$T = \frac{\sinh(k_w \pi (1-x)) \sin(k_w \pi y)}{\sinh(k_w \pi)}. \tag{4.27}$$

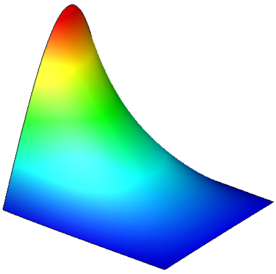

Figure 4.5. Exact temperature field for the 2D heat diffusion problem.

The test was also performed with the widely used BR2 scheme in order to compare the behavior of the two methods. The stabilization parameter associated with this last scheme was set to $\eta_{BR2} = 4$ ($\eta_{BR2} > N_F$ according to [32]). The BC's for the BR2 scheme are weakly imposed by means of proper ghost values on the boundary quadrature points.

The mesh refinement analysis is reported in Table 4.5. It can be seen that even with few elements, both schemes reach quickly the theoretical convergence order. In Figure 4.6 a direct comparison of the two methods is presented. The error is given as a function of the equivalent length scale (h), which depends on the number of total degrees of freedom ($nDOFs$): $h = (nDOFs)^{-0.5}$. The plot shows that the error for the ESR method is greater than the error for the BR2 method but, when the order is increased, the two schemes give comparable errors.

These numerical experiments were performed with a time marching approach in which explicit integration schemes were employed. The stability limits for the ESR method and the BR2 method are reported in Table 4.6. It is interesting to see that the allowed time step for the ESR method is approximately 4 times larger than the allowed time step for the BR2 method when $p = 1$. The ratio tends to grow when the order is increased and it is near 8 for $p = 6$. This behavior, together with the fact that for high orders the two schemes give comparable errors, suggest that

Scheme	Mesh	Error ESR	Order ESR	Error BR2	Order BR2
DG1	$2 \times (2 \times 2)$	7.33E-02	-	2.98E-02	-
	$2 \times (4 \times 4)$	2.02E-02	**1.86**	1.23E-02	**1.28**
	$2 \times (8 \times 8)$	4.92E-03	**2.04**	3.88E-03	**1.66**
DG2	$2 \times (2 \times 2)$	2.15E-02	-	8.79E-03	-
	$2 \times (4 \times 4)$	2.00E-03	**3.43**	9.77E-04	**3.17**
	$2 \times (8 \times 8)$	2.47E-04	**3.02**	1.14E-04	**3.10**
DG3	$2 \times (2 \times 2)$	1.75E-03	-	1.08E-03	-
	$2 \times (4 \times 4)$	9.17E-05	**4.25**	7.38E-05	**3.87**
	$2 \times (8 \times 8)$	5.32E-06	**4.11**	4.79E-06	**3.95**
DG4	$2 \times (2 \times 2)$	1.45E-04	-	1.23E-04	-
	$2 \times (4 \times 4)$	4.90E-06	**4.89**	4.25E-06	**4.86**
	$2 \times (8 \times 8)$	1.45E-07	**5.08**	1.41E-07	**4.91**
DG5	$2 \times (2 \times 2)$	1.51E-05	-	1.24E-05	-
	$2 \times (4 \times 4)$	2.96E-07	**5.67**	2.12E-07	**5.87**
	$2 \times (8 \times 8)$	4.53E-09	**6.03**	3.44E-09	**5.95**
DG6	$2 \times (2 \times 2)$	1.25E-06	-	8.82E-07	-
	$2 \times (4 \times 4)$	9.74E-09	**7.00**	7.60E-09	**6.86**
	$2 \times (8 \times 8)$	7.79E-11	**6.97**	6.15E-11	**6.95**

Table 4.5. L^2-norm of the temperature error for the ESR and the BR2 methods.

Scheme	$\mathrm{CFL}_{\mathrm{max}}$ESR	$\mathrm{CFL}_{\mathrm{max}}$ BR2
DG1-RK2TVD	6.2E-02	1.4E-02
DG2-RK3TVD	3.9E-02	8.5E-03
DG3-RK4LS	5.6E-02	8.6E-03
DG4-RK4LS	3.5E-02	5.0E-03
DG5-RK4LS	2.4E-02	3.2E-03
DG6-RK4LS	1.6E-02	2.1E-03

Table 4.6. Stability limit with explicit time integration for 2D heat diffusion on unstructured meshes.

the ESR method becomes competitive for explicit time integration of high order reconstructions. On the other hand, the fact that the BR2 method gives a lower error for all the reconstruction orders considered here, makes it a general purpose scheme, with a wider applicability than the ESR method.

It is very hard to compare the efficiencies of the different schemes. A possible approach might be to evaluate the total computational cost required to obtain a certain error level. However, the CPU time obtained is strongly related to the particular implementation chosen, and to the nature of the problem under study. Indeed, the presence of memory limitations (typically in 3D) or deforming grids can dramatically increase the computational cost, since several quantities cannot be stored but have to be re-computed during the simulation. Furthermore, the choice of the time integration scheme can enhance or cancel some benefits of the different methods. For example, in steady state problems, implicit integration schemes are commonly used. Their large stability domain can reduce the advantages observed in the explicit time integration with the ESR method.

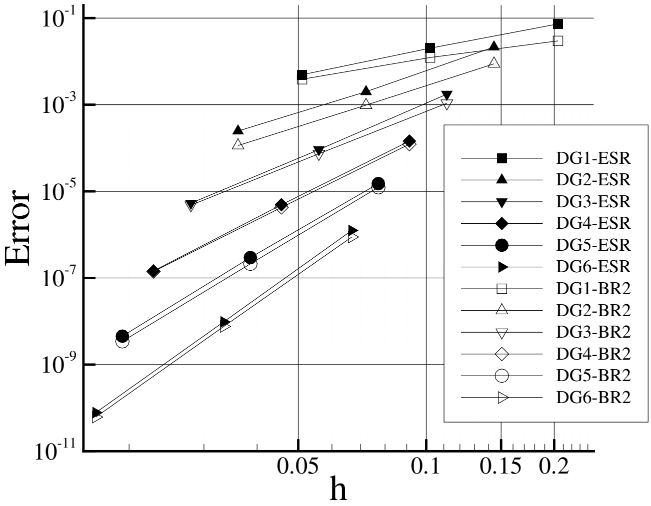

Figure 4.6. L^2-norm of the temperature error vs the equivalent length scale h.

4.4.4 Laminar boundary layer on a flat plate

This test case was proposed at the First, Second and Third International Workshop on High-Order CFD Methods ([68], [11] and [69]). The flow is characterized by a Reynolds number based on the plate length equal to $Re_\infty = 10^6$. The angle of attack is 0° and the free-stream Mach number is $M_\infty = 0.5$. The plate is adiabatic and has unit length.

The purpose of this test simulations is to evaluate the behavior of the ESR method in the presence of highly clustered meshes. In particular one of the meshes proposed by Bassi and Colombo was used. It can be downloaded from the Workshops website (file *a1-125-2s.msh*, [11]). This grid contains 560 quadrilateral elements. The distance between the inlet and the leading edge is 1.25. The distance between the freestream and the plate is 2. The size of the elements near the wall is $3.75 \cdot 10^{-4}$ in the normal direction. The most stretched element in the domain has a length/thickness ratio equal to 877.

A *p*-refinement study was performed on this mesh in order to evaluate the convergence of the solution. The simulations make it possible to compare the results with those presented by several groups at the workshop, which are reported in [68]. In Figure 4.7 the wall friction coefficient distribution is shown for several orders of accuracy. Also, Blasius' incompressible solution is reported. The effect of *p*-refinement can be easily seen in the region near the leading edge. We performed a second set of simulations with a lower free-stream Mach number ($M_\infty = 0.2$) in order to understand the influence of compressibility. In Table 4.7 the drag coefficient (C_D) is reported as a function of the total number of degrees of freedom per equation ($nDOFs/eq$). Even when the computations were performed using one of the meshes with less elements between those proposed at the Workshop, the *p*-refinement quickly obtained a converged solution. The ESR method behaves well also in the presence of highly clustered meshes with very stretched elements.

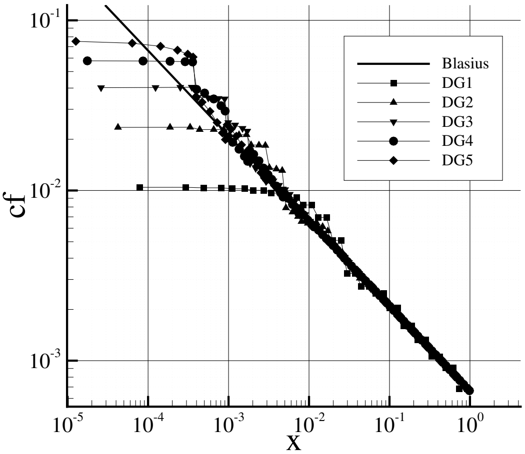

Figure 4.7. Wall friction coefficient (c_f) for the flat plate ($Re_\infty = 10^6, M_\infty = 0.2$)

Scheme	nDOFs/eq	C_D ($M_\infty = 0.5$)	C_D ($M_\infty = 0.2$)
DG1-RK2TVD	1680	1.277	1.282
DG2-RK3TVD	3360	1.306	1.324
DG3-RK4LS	5600	1.308	1.327
DG4-RK4LS	8400	1.307	1.327
DG5-RK4LS	11760	1.307	1.327
Blasius	-	1.328	1.328

Table 4.7. Drag coefficient (C_D) for flate plate problem ($Re_\infty = 10^6, M_\infty = 0.5 - 0.2$).

4.5 Considerations on implicit time integration

In the previous paragraph, explicit time integration schemes have been considered in order to evaluate the stability limits of the different methods. This aspect is particular important in time dependent problems. However, implicit schemes can be much more efficient than explicit ones when steady problems are studied. In this Section we describe how to implement the ESR method with implicit time integration. Implicit schemes require the evaluation of the jacobian of the system. If an approach based on the analytical computation of the jacobian is used, the ESR method shows some interesting features which can be useful during the implementation. A linear heat diffusion problem with a fixed mesh is considered as example.

In order to evaluate each term of the jacobian it is necessary to perform the derivative of the residuals with respect to each degree of freedom. This derivative contains contributions from the volume and the boundary integrals which appear in Eq. (3.9). In the proposed approach, the gradients required by the volume integral are approximated using the internal reconstruction. This simplifies significantly the computation of the jacobian. In contrast, both the RDG1x method and the BR2 method compute these gradients using information also from the interfaces. As far as the boundary integral is concerned, there are two possible cases. Indeed, an interface can be inside the domain or it can belong to the domain boundary. In the first case, the required gradients are computed with the recovery approach. This is done through matrix-vector multiplications in which all the matrices depend only on the mesh and not on the solution:

$$\nabla u = [\mathbf{D}]\tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{D}][\tilde{\mathbf{R}}] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\}. \tag{4.28}$$

The terms in the jacobian related to this contribution can be immediately computed from the constant terms in the previous relation. On the other hand, if the interface belongs to a Dirichlet boundary, the procedure of Section 5 has to be considered. In particular the gradient on the boundary is computed as:

$$\nabla u = [\mathbf{D}]\left(\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}\right) = [\mathbf{D}]\tilde{\mathbf{u}} + [\mathbf{D}][\hat{\mathbf{B}}]\hat{\mathbf{W}}. \tag{4.29}$$

Again, the contribution to the jacobian can be directly obtained from the constant terms which appear in the previous relation.
A performance test was carried out on the 2D heat diffusion problem of Section 4.4.3. In particular an unstructured mesh with 512 triangles is considered. All the elements are fifth order accurate ($p = 4$). The solution is initialized with a uniform field and time integration is performed with both the RK4-LS explicit scheme and the backward Euler (BE) implicit scheme. In Figure 4.8 the L^2 norm of the residuals is reported as a function of the computational time. As expected, the plot shows the significant advantage offered by the implicit approach for steady problems.

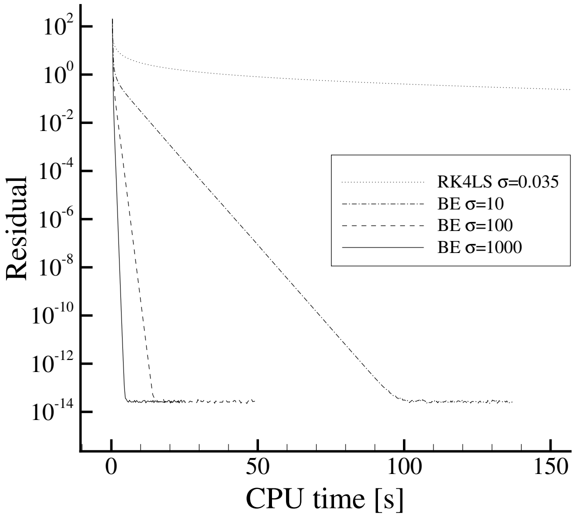

Figure 4.8. Adimensionalized L^2-norm of the residual vs computational time [s].

Chapter 5

Time discretization

Several time integration methods have been tested during the research activity. During the first year both explicit multistep and explicit Runge-Kutta algorithms have been implemented. Multistep methods [70] require the computation of fluxes only once for each time step but they need to keep in the memory the numerical solution at several previous steps. Furthermore, their stability limit becomes lower when the order of accuracy is increased. On the other hand, Runge-Kutta methods introduce several stages inside each step but they show a larger stability domain when the order is increased.

Some experiments have also been performed with the ADER scheme [71]: it is based on a particular procedure which gives a time dependent solution for the fluxes at the interfaces between elements. The time evolution of the solution is then obtained by a proper quadrature of the time dependent fluxes.

In order to increase the performances of the code in steady problems the implicit backward Euler scheme has been implemented.

It is interesting to note that there are current trends in the literature which suggest the use of implicit time integration schemes also for unsteady problem. A method which seems to be very promising has been proposed by Nigro et al. [72].

In the following, only the implemented explicit Runge-Kutta methods and the backward Euler scheme are described because most of this work has been performed with these methods.

In Chapter 3 the discontinuous Galerkin semidiscrete formulation (Eq. 3.9) has been derived for a generic element. It describes the evolution in time of the degrees of freedom of the element. The global set of equations for all the elements in the domain can be expressed as:

$$[\mathbb{M}]\,\frac{d\tilde{\mathbf{u}}}{dt} = -\mathbb{R}(\tilde{\mathbf{u}}) \tag{5.1}$$

where $[\mathbb{M}]$ is the block diagonal global mass matrix and $\tilde{\mathbf{u}}$ is the global vector of degrees of freedom. The vector $\mathbb{R}(\tilde{\mathbf{u}})$ is the global vector of the residuals obtained by the boundary and volume integrals of Eq. 3.9. In order to simplify the notation, the different methods will be presented by considering a reference problem described by an equivalent system of ordinary differential equations:

$$\frac{d\mathbf{u}}{dt} = L(\mathbf{u}) \tag{5.2}$$

In all the simulations a global time stepping strategy is employed. In the general case in which convection-diffusion equations are considered the diffusion and convection stability limits are evaluated separately in each element:

$$\Delta t_d = CFL \frac{H_e^2}{\kappa(2p_e + 1)} \tag{5.3}$$

$$\Delta t_c = CFL \frac{H_e}{c(2p_e + 1)} \tag{5.4}$$

where CFL, c, κ, H_e and p_e are the stability coefficient, maximum signal speed, diffusivity coefficient and element characteristic size and order. Then, the minimum allowable time step is chosen in all the domain and it is used as global time step. Cockburn [73] reports that a CFL number equal to 1 can be used in Eq. 5.4 for convection problems when a DG reconstruction of order p is integrated by an explicit Runge-Kutta algorithm with an accuracy order equal to $p + 1$.

5.1 Explicit Runge-Kutta methods

The explicit Euler method is used for first order accurate time integrations:

Forward Euler

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t L(\mathbf{u}^n) \tag{5.5}$$

Two and three stages Total Variation Diminishing Runge-Kutta (TVD RK) methods proposed by Gottlieb and Shu [74] have been implemented in order to achieve second and third-order accuracy. These are schemes particularly suited for problems containing discontinuities since they avoid the introduction of spurious oscillations connected to the time integration.

Second-order two stages TVD-RK

$$\begin{aligned} \mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t L(\mathbf{u}^n) \\ \mathbf{u}^{n+1} &= \frac{1}{2}\mathbf{u}^n + \frac{1}{2}\mathbf{u}^{(1)} + \frac{1}{2}\Delta t L(\mathbf{u}^{(1)}) \end{aligned} \tag{5.6}$$

Third-order three stages TVD-RK

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t L(\mathbf{u}^n) \\
\mathbf{u}^{(2)} &= \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\mathbf{u}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{u}^{(1)}) \\
\mathbf{u}^{n+1} &= \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{u}^{(2)}) \qquad (5.7)
\end{aligned}$$

Fourth-order accuracy has been obtained with three different Runge-Kutta algorithms: the classical four stages method [75], the Strong Stability Preserving (SSP) five stages method [76] and the low memory five stages method [77].
Most of the simulations have been performed with the SSP RK4 which is reported in the following. Notice that the presence of the fifth stage requires a greater computational effort in comparison with the classical fourth-order/four stages scheme, but this is paid by a larger allowable time step.

Fourth-order five stages SSP-RK

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + 0.39175222700392\Delta t L(\mathbf{u}^n) \\
\mathbf{u}^{(2)} &= 0.44437049406734\mathbf{u}^n + 0.55562950593266\mathbf{u}^{(1)} + 0.36841059262959\Delta t L(\mathbf{u}^{(1)}) \\
\mathbf{u}^{(3)} &= 0.62010185138540\mathbf{u}^n + 0.37989814861460\mathbf{u}^{(2)} + 0.25189177424738\Delta t L(\mathbf{u}^{(2)}) \\
\mathbf{u}^{(4)} &= 0.17807995410773\mathbf{u}^n + 0.82192004589227\mathbf{u}^{(3)} + 0.54497475021237\Delta t L(\mathbf{u}^{(3)}) \\
\mathbf{u}^{n+1} &= 0.00683325884039\mathbf{u}^n + 0.51723167208978\mathbf{u}^{(2)} + 0.12759831133288\mathbf{u}^{(3)} + \\
&+ 0.34833675773694\mathbf{u}^{(4)} + 0.08460416338212\Delta t L(\mathbf{u}^{(3)}) + \\
&+ 0.22600748319395\Delta t L(\mathbf{u}^{(4)}) \qquad (5.8)
\end{aligned}$$

5.2 Implicit methods

In this work the implicit Backward Euler method has been implemented. It can be obtained by linearizing Eq. 5.1:

$$[\mathbb{M}]\,\frac{\mathbf{\tilde{u}^{n+1}} - \mathbf{\tilde{u}^{n}}}{\Delta t} = -\left(\mathbb{R}(\mathbf{\tilde{u}^{n}}) + [\mathbb{J}](\mathbf{\tilde{u}^{n+1}} - \mathbf{\tilde{u}^{n}})\right) \qquad (5.9)$$

where the matrices $[\mathbb{J}]$ and $[\mathbb{M}]$ are the jacobian matrix and the global mass matrix. The global mass matrix is a block diagonal matrix in which each block is obtained by an element mass matrix. The jacobian matrix is a block sparse matrix which contains the derivatives of the residual with respect to the degrees of freedom:

$$\mathbb{J}_{ij} = \frac{\partial \mathbb{R}_i}{\partial \tilde{u}_j} \qquad (5.10)$$

In Eq. 5.9 $\mathbf{\tilde{u}}$ represents the global vector of degrees of freedom which has size $M\sum_{i=1}^{n_{ele}} N_i$, where M, n_{ele} and N_i are the number of governing equations, the number of elements and the number of degrees of freedom in the element i, respectively. The equation can be rewritten in order to put in evidence the following linear system:

$$\left(\frac{1}{\Delta t}[\mathbb{M}] + [\mathbb{J}]\right)(\mathbf{\tilde{u}^{n+1}} - \mathbf{\tilde{u}^{n}}) = -\mathbb{R}(\mathbf{\tilde{u}^{n}}) \tag{5.11}$$

$$[\mathbf{A}](\mathbf{\tilde{u}^{n+1}} - \mathbf{\tilde{u}^{n}}) = -\mathbb{R}(\mathbf{\tilde{u}^{n}}) \tag{5.12}$$

which has to be solved at each time step in order to update the vector of degrees of freedom. The matrix A resulting from Eq. 5.12 is memorized in the Compressed Sparse Row (CSR) format [78].

5.2.1 Jacobian evaluation

The terms in the jacobian matrix represents the dependency of the numerical residuals on the degrees of freedom of the discretization. A possible way to obtain the jacobian matrix is to perform the analytical differentiation of the discretized equations. This approach is very efficient and can be easily parallelized. It has been used in this work for the implicit integration of the unsteady linear heat equation. However, this approach requires a lot of time for the implementation and the maintenance of the code. Indeed, if a new physical model or a new numerical flux is introduced, then the code for the computation of the jacobian has to be rewritten. In order to simplify this task it is possible to use automatic differentiation tools which automatically generates the source code for the terms which contribute to the jacobian.

An alternative approach is based on the numerical approximation of the jacobian by means of finite differences. This is done by the application of a perturbation ϵ on each of the degrees of freedom of the numerical solution. The residuals are then evaluated before and after the application of the perturbation, simply by calling the same subroutines. This approach is very interesting from the point of view of code maintenance because it does not require any change in the subroutine which evaluate the jacobian when a new physical model is introduced. However, there are several drawbacks. First of all, great care must be taken in parallelization because it is necessary to avoid the perturbation of two neighbouring elements (and also the neighbours of the neighbours) at the same time.

Furthermore, the perturbation should be chosen sufficiently small to give a good approximation of the derivative but not so small to introduce round-off problems. In this work, the perturbation ϵ for the $i-th$ degree of freedom is chosen according to the following rule:

$$\epsilon = \max(\epsilon_{min}, \epsilon_{rel} u_i) \tag{5.13}$$

where the coefficients ϵ_{min} and ϵ_{rel} are usually chosen equal to 10^{-10} and 10^{-8}. A similar approach has been proposed by Brown and Saad [79].

5.2.2 Iterative solver

The linear system 5.12 has to be solved at each time step. For this purpose the iterative solver GMRES [80] has been employed. In particular, the PARALUTION [81] library has been used. This library offers several iterative solvers and preconditioners. Furthermore, the library is parallelized by OpenMp directives for computations on multi-core machines.
In all the simulations the Incomplete Lower Upper preconditioner with zero fill-in (ILU0) has been employed to improve the convergence of the GMRES algorithm. Some tests have been performed also with the restarted GMRES algorithm.

5.2.3 CFL evolution laws for steady problems

It is interesting to note that the backward Euler method applied to the unsteady equations 5.1 tends to the Newton method applied to the steady equations when $\Delta t \to +\infty$. This means that, for sufficiently high values of the time step, it is possible to obtain quadratic convergence towards the steady solution. The presence of the unsteady term in the equations plays the same role of the damping term in damped Newton methods applied on the steady equations.
The time step cannot be chosen too high at the beginning of the simulation because unphysical values can be obtained when the transient is too strong. For this reason, it is necessary to introduce a CFL evolution strategy which represents a compromise between the robustness and the efficiency of the integration. This approach is called pseudo-transient continuation strategy.
Several approaches have been implemented in this work.
One of them is a robust method proposed by Bassi et al. [82] in the context of turbulent problems. It uses the L^2 and the L^∞ norms of the residual to control the CFL evolution. Let x_{L^2} and x_{L^∞} be defined as

$$x_{L^2} = \max \frac{|R_i|_{L^2}}{|R_{i0}|_{L^2}} \qquad x_{L^\infty} = \max \frac{|R_i|_{L^\infty}}{|R_{i0}|_{L^\infty}} \qquad 1 \leq i \leq M \tag{5.14}$$

where R_{i0} and R_i are the residuals of the i-th equation at the first time step and at the current iteration. M is the number of governing equations which depends on the chosen physical model and the dimensionality of the problem.

$$CFL = \begin{cases} \frac{CFL_0}{x^\alpha}, & \text{if } x \leq 1 \\ CFL_{exp} + (CFL_0 - CFL_{exp}) e^{\alpha \frac{CFL_0}{CFL_0 - CFL_{exp}}(1-x)}, & \text{if } x > 1 \end{cases} \tag{5.15}$$

where CFL_0, CFL_{exp} and α are the minimum CFL number, the maximum allowable CFL number of explicit schemes and a parameter (usually ≤ 1) which controls the growing rate, respectively. The chosen CFL number is then limited under a maximum value (usually $CFL_{max} = 10^{10} - 10^{20}$). The variable x is defined as:

$$x = \begin{cases} \min(x_{L^2},1) & \text{if } x_{L^\infty} \leq 1 \\ x_{L^\infty} & \text{if } x_{L^\infty} > 1 \end{cases} \tag{5.16}$$

An alternative approach is the Switched Evolution Relaxation proposed by Mulder and van Leer [83]. In this approach the CFL number at the step n is obtained from the CFL number at the previous step according to the following relation:

$$CFL^n = \min(CFL^{n-1}\frac{|R^{n-1}|_{L^2}}{|R^n|_{L^2}}, CFL_{max}) \tag{5.17}$$

Finally, the Residual Difference Method by Bucker et al. [84] has been implemented. The algorithm reads as follows:

$$CFL^n = \min(CFL^{n-1} \cdot \beta^{\frac{|R^{n-1}|_{L^2} - |R^n|_{L^2}}{|R^{n-1}|_{L^2}}}, CFL_{max}) \tag{5.18}$$

where β is a user-defined coefficient ($\beta > 1$) which controls the growing rate. Also a monotonic variant of this algorithm has been considered by setting the exponent of Eq. 5.18 to zero when $|R^n|_{L^2} > |R^{n-1}|_{L^2}$.

In conclusion, all the previously described strategies have to be carefully calibrated. Indeed, the total computational time for a steady problem is strongly influenced by the CFL evolution law. It is easy to change the computational time by several orders of magnitude simply by choosing a strategy instead of another or changing one of the parameters which control the growing rate.

Chapter 6

Shock capturing

The treatment of discontinuities which can appear in compressible flows requires great care. Where the solution is discontinuous numerical methods of order greater than one tend to show spurious oscillations. Several techniques have been proposed to deal with these problems in the DG framework. A review of them can be found in the synthesis report of the ADIGMA european project [85].
A widely used approach consists in adding artificial viscosity to the governing equations. This artificial term should be activated only in the regions in which there are discontinuities ([86], [82], [89], [90]). The technique provides good results but if used with explicit time integration schemes can introduce a significant limitation on the maximum allowable time step and it also produces some oscillations when used with high-order reconstructions. Furthermore great care must be taken for the definition of the algorithm which controls the amount of numerical viscosity, in order to keep it as problem independent as possible. On the other hand, the artificial viscosity approach works very well with implicit time integration in steady problems because the artificial term is directly introduced in the jacobian and so convergence to machine precision can be reached.
A different approach has been derived from shock capturing techniques developed in the context of FV methods; it consists in limiting the solution in a element by the use of functions which compare the average value inside the element with the average values of the neighbouring elements [91]. This approach suffers from accuracy degradation on local extremes. The problem can be avoided by the introduction of Total Variation Bounded in the Mean limiters [10] which require the definition of a threshold that is problem dependent. Momentum limiters [92] have also been proposed: they generalize the classic limiters approach extending it to high-order reconstructions. Finally the WENO scheme [93] developed for FV methods has been applied also to DG methods. All the techniques based on

limiters share the need to know information from neighbouring cells and so they require special treatment for boundary elements or hanging-nodes meshes. A further way of dealing with discontinuities is to filter the numerical solution by reducing high-order modal coefficients, which are responsible for oscillations. This operation shows results similar to those obtained by the artificial viscosity approach but does not introduce any additional stability restriction for explicit integration. Furthermore, the filter acts only on the element solution and does not require information from neighbouring elements. It is possible to apply filtering directly to the solution modal coefficients ([94]), but also to the time derivatives of these coefficients [95]. In any case the main problem is to understand how much filtering is needed: enough to avoid spurious oscillations but not so much as to degrade accuracy. In order to give a more precise answer to this question, a feedback filtering procedure has been developed during the research activity. It is described in Section 6.1. The aim of the proposed approach is to maintain the locality of the algorithm and to make it as problem independent as possible.
Both limiters and filters work well in time-accurate calculations but they can introduce convergence problems in steady flows.

6.1 Feedback filtering

6.1.1 Exponential filter

In the present work an exponential filter has been used to keep under control oscillations induce by shock waves. The filter multiplies the solution's modal coefficients according to the approach described by Gottlieb and Hesthaven [96]. In particular, the filter has been implemented in the explicit version of the code. It acts as a post-processing step after each stage of the Runge-Kutta time integration algorithm. In 1D the filtered solution coefficients are obtained in the following way:

$$(\tilde{u}_n)_{filtered} = \sigma(\eta_n)\tilde{u}_n \qquad 0 \leq n \leq p \tag{6.1}$$

$$\sigma = e^{(-\alpha\eta^r)} \qquad \eta_n = \frac{n}{p} \tag{6.2}$$

where n is the degree of the generic basis function and p is the order of the reconstruction inside the element. The extension to 2D is performed in the following way:

$$(\tilde{u}_{nm})_{filtered} = \sigma(\eta_n)\sigma(\eta_m)\tilde{u}_{nm} \qquad 0 \leq n \leq p \quad 0 \leq m \leq p \tag{6.3}$$

$$\eta_n = \frac{n}{p} \qquad \eta_m = \frac{m}{p} \tag{6.4}$$

where n and m are the degrees of the basis function in the two coordinate variables. The parameters r (filter order) and α (filter intensity) have to be chosen carefully in order to eliminate oscillations while preserving the accuracy of the numerical solution in smooth regions. In the literature there are several works (see e.g. Hesthaven and Warburton [97]) which suggest to set $\alpha = -log(eps)$ where eps is the machine precision (for double precision $\alpha \simeq 36$). Other indications are given by Boyd [98] who suggests the use of high-order filters in smooth regions and low-order filters in the proximity of discontinuities.
The numerical experiments performed in this work show that the choice of $\alpha = -log(eps)$ degrades significantly the solution accuracy, even when high-order filters are used. These results are in line with the work previously carried out by Ueckermann and Lermusiaux [99]. They introduced a function which measures the smoothness of the numerical solution inside each cell: the signal obtained by this function is compared to a threshold and, if the signal is greater than the threshold, an exponential filter is applied in the considered cell. In their computations they set $\alpha = -log(0.01) \simeq 4.6$.
In the simulations performed in this work, this kind of approach appears to be not very robust since the choice of α and the threshold has to be done very carefully. Indeed, the optimal values for these parameters depend on the particular problem which is under study and depend also on the reconstruction order. For this reasons an adaptive procedure has been introduced in order to automatically adjust the filter intensity.

6.1.2 Smoothness indicator

One of the fundamental components of the proposed procedure is a smoothness indicator. In particular, the sensor developed by Persson and Peraire [86] for their shock capturing scheme is used in this work.
This sensor compares the numerical solution inside an element to a truncated numerical solution, which does not contain higher-order terms. In practice, it can be shown that where the solution is smooth the modal coefficients decay very quickly. For this reason the sensor shows weak signals in smooth regions. In contrast, the high-order coefficients earn importance near discontinuities and so the sensor gives intense signals where the solution is not smooth. In particular the following expression is obtained when the sensor is applied to a generic scalar variable u_h inside the element Ω_e :

$$S_{\Omega_e} = \frac{\int_{\Omega_e} (u - \bar{u})^2 dxdy}{\int_{\Omega_e} u^2 dxdy + \varepsilon} \tag{6.5}$$

In 1D problems the test variable $\bar{u}$ is obtained by truncating the reconstruction at the first $\bar{N}_e = p$ modes and is compared with the complete u obtained by all the

$N_e = p + 1$ modes.
This is generalized to 2D by setting $\bar{N}_e = p^2, N_e = (p+1)^2$ for BASIS A and $\bar{N}_e = p(p+1)/2$ and $N_e = (p+1)(p+2)/2$ for BASIS B.

$$\bar{u} = \sum_{i=1}^{\bar{N}_e} \tilde{u}_i \Phi_i \qquad u = \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i \tag{6.6}$$

In this way the sensor gives a signal which quantifies the smoothness of the numerical solution in each element. Numerical experiments on discontinuous problems show that the order of magnitude of the signal changes significantly when moving from smooth regions to the elements near a discontinuity. For this reason the logarithm of the smoothness indicator signal (SI) is monitored:

$$SI = \log_{10}(S_{\Omega_e} + \varepsilon) \tag{6.7}$$

In some cases the denominator of Eq. 6.5 and the logarithm's argument in Eq. 6.7 could become equal to zero, depending on which variable is monitored. In particular the sensor signal can become equal to zero when the truncated solution is able to exactly describe the physical solution. For example, this happen when a uniform field is considered. In order to avoid this problem an additive positive constant ϵ is introduced in Eqs. 6.5 and 6.7 and set equal to 1^{-10} in all simulations.

6.1.3 Feedback filtering

The key idea of feedback filtering [1] consists in setting a goal for the solution smoothness (SR = smoothness requirement) and in automatically finding the filter intensity needed to achieve this goal. This is done through an iterative procedure which measures the smoothness of the solution (SI). When SI is not satisfactory a weak filter is applied (for example an exponential filter with $\alpha_{iter} = 0.1$). These operations are repeated until the desired smoothness is reached. In other words, in each element at each stage of the Runge-Kutta algorithm the procedure described by the following pseudocode is applied:

```
Compute SI
do while (SI > SR)
    Apply weak filter
    Compute SI
end do
```

In this way the number of parameters which have to be calibrated is halved in comparison to the approach of Ueckermann and Lermusiaux: it is only necessary to set a requirement for the solution smoothness and the filter intensity will be adjusted automatically in order to satisfy it. This is the concept of feedback filtering: the

filter influences the smoothness of numerical solution but this in turn influences the filter intensity.
It is useful to say few words about idempotent filters (Kanevsky et al. [100]). If a filter is applied repeatedly to a solution providing always the same results, independently on how many times it is applied, the filter is referred to as idempotent. In other words an idempotent filter does not show cumulative effects when it is applied repeatedly. The exponential filter is not idempotent but if it is used inside the proposed feedback procedure, an idempotent filtering operation is obtained. Indeed when the procedure is applied for the first time the solution is filtered in order to satisfy the smoothness requirement: any subsequent application of the procedure does not change the solution, since it already satisfies the requirement.

The proposed filtering procedure can be implemented in several ways. As far as the choice of α_{iter} is concerned, it can be observed that an increase in α_{iter} reduces the number of iterations required to reach the goal and so it reduces the computational cost. However, α_{iter} should not be chosen too large otherwise the solution obtained after the last iteration could be over-filtered. In an over-filtered solution the high-order modes have been strongly reduced and only the first modal coefficient remains unaltered. Furthermore, high-order DG schemes are usually employed with relatively coarse meshes since they introduce several DOFs inside each element and so they require few elements. For this reasons, an over-filtered solution can appear quite inaccurate as it is close to a first order solution on a relatively coarse mesh (see Par. 5.1). In all the computations performed in this work the following parameters were used: $\alpha_{iter} = 0.1$ and $r = 2$.
As far as computational cost is concerned, it is important to keep in mind that the iterative filtering procedure is applied only in those elements in which the smoothness requirement is not satisfied, i.e. in the elements containing a discontinuity. They usually represent a small fraction of the total number of elements.

Some guidelines about the choice of the smoothness threshold SR can be given. In their work Persson and Peraire [86] observe that if the solution is based on a Fourier expansion the sensor S_{Ω_e} provides a signal which depends on the reconstruction order p according to the following relation:

$$S_{\Omega_e} = \frac{g}{p^4} \tag{6.8}$$

where g is a constant. They assume that this relation holds also for polynomial reconstructions. They use this result to adjust the artificial viscosity of their scheme. In the present work the Eq. 6.8 is used to understand how the threshold SR should depend on the reconstruction order. If the logarithm of Eq. 6.8 is considered the following relation is obtained:

$$SR = \log_{10}(g) - 4\log_{10}(p) \tag{6.9}$$

in which the constant g is the only parameter which the user has to adjust.

Some considerations can be done about the practical implementation of this procedure. There are several variables which can be monitored by the smoothness indicator in the framework of Euler equations. A computationally efficient choice could be to monitor only one variable, which should be significant for both shocks and contact surfaces: for example density could be a good candidate. When in an element the sensor applied to density provides a signal greater than the threshold then all the conservative variables are filtered. Alternatively each conservative variable could be monitored applying the filter only to those variables which do not satisfy the requirement. All the results shown in the following have been obtained with the first approach, by monitoring density. If the proposed algorithm is used in the context of the compressible Navier-Stokes equations the sensor variable should be chosen in order that the filter does not act on boundary layers or wakes. In particular, the pressure could be a better sensor variable for Navier-Stokes equations because it helps in preserving the accuracy of the boundary layer discretization.

Results obtained for four different test cases are presented in the following. The first two are characterized by the presence of discontinuities, both in 1D (Section 6.1.4) and 2D (Section 6.1.5). In the third test case (Section 6.1.6) a smooth flow is studied and the numerical solution is compared with the existing analytical solution in order to test whether feedback filtering preserves solution accuracy in smooth regions. In the fourth test case (Section 6.1.7) the transonic flow field around an airfoil is studied and results are compared with reference values from the literature. In order to verify the robustness of the proposed approach the same law is used for the smoothness requirement SR in all test cases :

$$SR = -2 - 4\log_{10}(k) \tag{6.10}$$

6.1.4 Sod problem

The first test case is the classic one-dimensional Sod shock tube problem. It consists in a Riemann problem in which there are two initial uniform fields A (left) and B (right) defined by the following conditions: $u_A = 0$, $P_A = 0.4$, $\rho_A = 1$, $u_B = 0$, $P_B = 0.04$, $\rho_B = 0.125$. The exact solution is characterized by the propagation of three waves: an expansion fan, a contact surface and a shock. Between the waves there are two uniform fields which can be computed by means of an iterative procedure in which the non-linear Rankine-Hugoniot jump relations are considered together with the compatibility relations for the expansion wave.

After the computation of these constant fields it is possible to find the exact solution inside the expansion fan by the use of compatibility relations. Further details can be found in Toro [87] and Shapiro [88].

Numerical results obtained by DG0, DG1, DG2 and DG3 schemes are compared in the following. Time integration was performed by Runge-Kutta schemes described in Section 5.1 in order to obtain the same accuracy order in space and time. The number of DOFs per equation was kept approximately equal to 100 in all simulations in order to make a fair comparison between different schemes. For this reason a uniform mesh with 100 elements was used for the DG0 scheme while coarser meshes were used for higher-order schemes. The numerical solution is represented by plotting in each element a number of points equal to the number of DOFs in that element.
First of all, some computations were performed with the exponential filter ($\alpha = 36$, $r = 2$) activated in all the elements. In Figure 6.1 the density field at time $t = 0.4$ is reported. The results show clearly the effects of over-filtering. The accuracy of high-order reconstructions is completely lost. Indeed, the DG0 solution on the finest mesh is more accurate than all the others high-order solutions on the coarser meshes.
A second set of simulations was performed with the proposed feedback filtering procedure. Results are reported in Figure 6.2. In this case the global solution shows that high-order schemes achieve a significant improvement over the DG0 scheme. This is particularly evident in the contact surface and at the edges of the expansion fan. A detail of the contact surface is presented in Figure 6.4. It can be seen that results obtained bt DG1, DG2 and DG3 schemes are very close and show a significant improvement with respect to the DG0 results. However, high-order schemes produce a weak undershoot near the discontinuity. Figure 6.3 shows a detail of the captured shock. The filtered high-order schemes introduce a slight deterioration in the quality of the captured shock with respect to the DG0 results. This behavior can be compensated by the introduction of an adaptive h-refinement strategy in order to increase shock resolution.
In conclusion this first test shows how feedback filtered high-order DG methods can capture discontinuities and keep spurious oscillations under control.

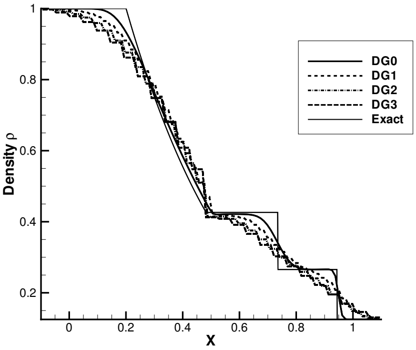

Figure 6.1. Adimensionalized L^2-norm of the residual vs computational time [s].

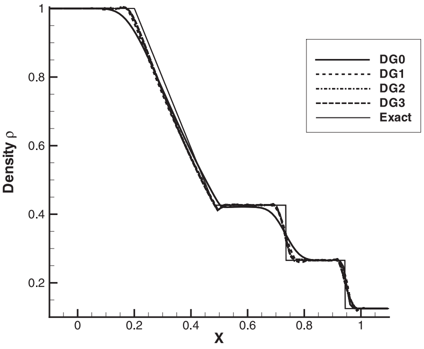

Figure 6.2. Adimensionalized L^2-norm of the residual vs computational time [s].

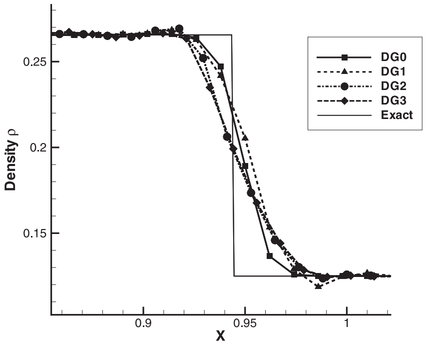

Figure 6.3. Detail of the contact surface in the Sod problem (Feedback filtering).

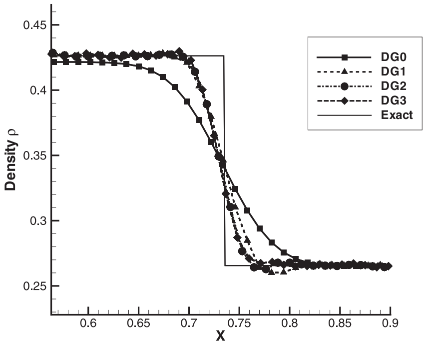

Figure 6.4. Detail of the shock in the Sod problem (Feedback filtering).

6.1.5 Shock-bubble interaction

The second test case is a two-dimensional problem characterized by the interaction between a normal shock and a cylindrical contact surface with radius equal to 0.2. The initial condition (Figure 6.5) shows a field at rest (B) containing a bubble (C) of hot gas. A normal shock wave coming from the left separates the field B from the supersonic uniform field A and interacts with the bubble. As a consequence the bubble's edge is deformed. After some time it breaks and gives rise to the formation of vortices. This test case makes it possible to carry out a qualitative analysis of numerical schemes. Indeed the solution shows vortex structures which come from the destabilization of the initial contact surface: the greater the numerical dissipation introduced by the numerical scheme, the less it will be able to capture small scale structures. As far as the boundary conditions are concerned, the left side is a supersonic inlet, the right side is a subsonic outlet while the upper and lower sides are solid walls. Since the problem is symmetric the computational domain was restricted to the upper part of the physical domain and solid wall boundary conditions were imposed on the symmetry axis. Initial data are summarized in Table 6.1.

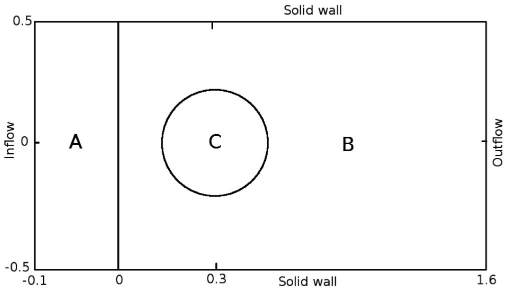

Figure 6.5. Domain and initial condition for shock-bubble interaction

Region	ρ	P	u	v
A	3.81×10^0	1.00×10^1	2.58×10^0	0.00×10^0
B	1.00×10^0	1.00×10^0	0.00×10^0	0.00×10^0
C	1.00×10^{-1}	1.00×10^0	0.00×10^0	0.00×10^0

Table 6.1. Shock-bubble problem: initial data.

The numerical solution is represented by computing the density gradient magnitude and plotting it with a nonlinear gray scale, in order to obtain a numerical

Schlieren image. Results can be compared with those reported by Cada and Torrilhon [9] and obtained by their third-order finite volume scheme. They use a very fine cartesian mesh with 1700x500 cells. The solution quality was evaluated by performing a simulation with the feedback filtered DG2 scheme on a 567x167 elements grid, in order to maintain approximately the same number of DOFs of the simulation carried out by Cada and Torrilhon. Results obtained by feedback filtering show the expected behavior: the evolution of the contact surface can clearly be seen in the sequence reported in Figure 6.6. A comparison between the structures visible in Figure 6.7 and those obtained by Cada and Torrilhon [9] shows that the discontinuities captured by the feedback filtered DG scheme have a less sharp profile than those captured by its finite volume counterpart. This is in line with the results obtained in the Sod test case, in which the filtered DG2 scheme captures the shock with a slight less sharp profile than the DG0 scheme. However, the feedback filtered DG method shows its accuracy in the description of contact surfaces and smooth regions. This feature permitted a good representation of the main structures of this flow. In Figure 6.6 it can be seen that the normal shock wave is followed by two numerical starting waves which appear at the beginning of the simulation. This numerical phenomenon is common also in finite volume computations and its origin is explained by Leveque [44].

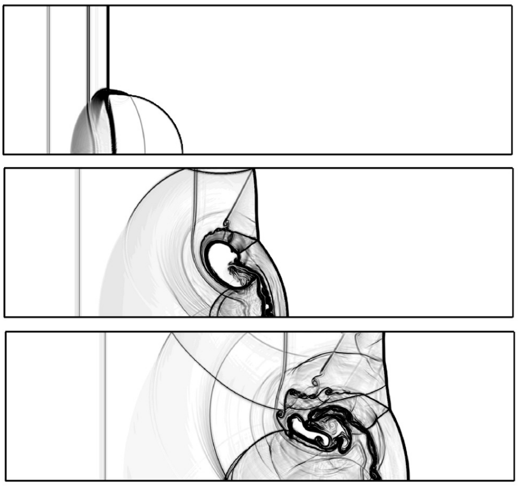

Figure 6.6. Numerical Schlieren images for shock-bubble interaction at $t = 0.071$, $t = 0.21$ and $t = 0.33$ (Feedback filtered DG2 scheme).

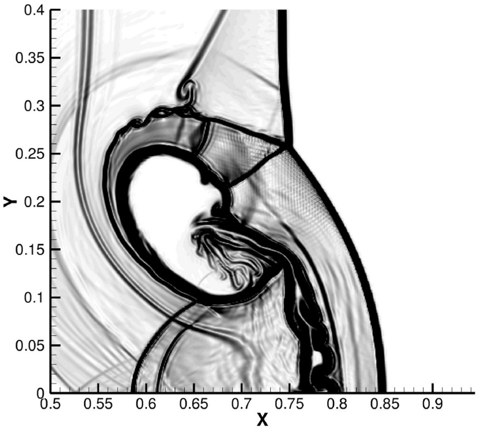

Figure 6.7. Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme).

6.1.6 Ringleb flow

The Ringleb flow is an inviscid steady isentropic two-dimensional flow. It can be seen as a shock-less transonic flow around a blunt body. This test case makes it possible to evaluate the accuracy order of a scheme by a comparison between the numerical solution and the exact solution, which can be obtained by the hodograph method [101]. In particular, the flow field is characterized by the following streamlines:

$$x = \frac{1}{2\bar{\rho}}\left(\frac{1}{\bar{q}^2} - \frac{2}{\kappa^2}\right) + \frac{J}{2} \tag{6.11}$$

$$y = \pm\frac{1}{\kappa\bar{\rho}\bar{q}}\sqrt{1 - \left(\frac{\bar{q}}{\kappa}\right)^2} \tag{6.12}$$

where

$$\kappa = \frac{1}{\Psi} \tag{6.13}$$

$$J = \frac{1}{\bar{a}} + \frac{1}{3\bar{a}^3} + \frac{1}{5\bar{a}^5} - \frac{1}{2}\ln\frac{1+\bar{a}}{1-\bar{a}} \tag{6.14}$$

$$\bar{a} = \sqrt{1 - \frac{\gamma - 1}{2}\bar{q}^2} \tag{6.15}$$

$$\bar{\rho} = \bar{a}^{\frac{2}{\gamma-1}} \tag{6.16}$$

In the previous relations $\bar{q}$, $\bar{a}$ and $\bar{\rho}$ are non-dimensional velocity magnitude, speed of sound and density. Here they are referred to the stagnation values of speed of sound and density respectively. Notice that in all the rest of this work, the normalization described in the Appendix A is used.

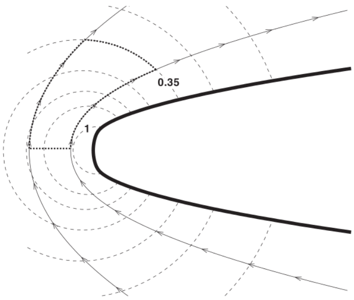

Figure 6.8. Ringleb flow: constant speed lines (dashed) and computational domain Ω.

The computational domain Ω is inside the subsonic region and it is shown in Figure 6.8. It is delimited by two streamlines with $\Psi = 1.15$ and $\Psi = 2$, where Ψ is the streamline constant which appears in Eq. 6.13. The corresponding boundaries are treated as solid walls. The inlet is chosen on the flow symmetry axis. The outlet corresponds to the iso-Mach line for which $\bar{q} = 0.35$. Curvilinear elements are used near solid walls and also near the outlet, on which pressure is imposed. The numerical solution is initialized with a uniform field at rest and then it is integrated up to steady state. Steady-state entropy (S) and pressure (P) errors in both norm-2 (L_2) and norm infinity (L_∞) are reported in Table 6.2 and Table 6.3 for the feedback filtered computations. Another set of computations without filtering gave exactly the same results. This suggests that during the transient the filter is applied in those elements in which discontinuities appear but when the smooth steady state field is reached the filter does not have any effect on the numerical solution and so accuracy is preserved.

Scheme	Grid	L_2S error	Order L_2S	L_∞S error	Order L_∞S
DG1B2	8x4	4.78E-04	-	1.69E-03	-
	16x8	1.14E-04	**2.07**	4.94E-04	**1.77**
	32x16	2.66E-05	**2.10**	1.33E-04	**1.89**
	64x32	6.31E-06	**2.08**	3.48E-05	**1.93**
DG2B3	8x4	1.41E-04	-	9.00E-04	-
	16x8	1.46E-05	**3.27**	1.52E-04	**2.57**
	32x16	1.43E-06	**3.35**	2.24E-05	**2.76**
	64x32	1.42E-07	**3.23**	3.05E-06	**2.88**
DG3B4	8x4	1.85E-05	-	1.55E-04	-
	16x8	1.02E-06	**4.18**	1.31E-05	**3.56**
	32x16	4.88E-08	**4.39**	9.58E-07	**3.77**
	64x32	2.25E-09	**4.44**	6.49E-08	**3.88**

Table 6.2. Ringleb flow: entropy accuracy test for feedback filtered DG methods.

6.1.7 Transonic NACA0012

This inviscid test case is characterized by a transonic flow field with shock waves on both the pressure and suction side of the airfoil. The freestream Mach number is $M_\infty = 0.8$ and the angle of attach is $\alpha = 1.25^o$.

Six simulations were performed with the reconstruction order varying from $p = 0$ to $p = 5$ and feedback filtering activated. Time integration was performed by EE (for $p = 0$), RK2 (for $p = 1$), RK3 (for $p = 2$) and SSPRK (for $p >= 3$) algorithms respectively. Computations were carried out on a structured C-mesh and domain

Scheme	Grid	L_2p error	Order L_2p	L_∞p error	Order L_∞p
DG1B2	8x4	1.44E-03	-	5.22E-03	-
	16x8	3.02E-04	**2.25**	1.12E-03	**2.22**
	32x16	9.87E-05	**1.61**	2.86E-04	**1.97**
	64x32	2.58E-05	**1.94**	8.75E-05	**1.71**
DG2B3	8x4	1.51E-04	-	3.84E-04	-
	16x8	1.45E-05	**3.38**	4.16E-05	**3.21**
	32x16	1.63E-06	**3.15**	5.26E-06	**2.98**
	64x32	2.03E-07	**3.01**	7.77E-07	**2.76**
DG3B4	8x4	1.67E-05	-	8.98E-05	-
	16x8	9.76E-07	**4.10**	8.23E-06	**3.45**
	32x16	5.81E-08	**4.07**	7.10E-07	**3.54**
	64x32	3.51E-09	**4.05**	5.19E-08	**3.77**

Table 6.3. Ringleb flow: pressure accuracy test for feedback filtered DG methods.

boundaries were fixed at 20 chords from the airfoil. This distance is too small to obtain results which are independent from the boundary distance. However, the main purpose of this test is to study the shock capturing properties of the proposed approach. A more accurate study (with freestream distance greater than 2000 chords) can be found in Sections 7.3.1.

The chosen mesh has 50 quartic curvilinear elements on both the pressure side and the suction side of the airfoil. In total the mesh contains 200x50 elements.

In Figure 6.9 and Figure 6.10 the Mach field for DG1 and DG5 simulations is reported. The DG1 results on this very coarse mesh show a significant generation of spurious entropy at wall which disappears in the DG5 results. On the other hand, the DG5 results show some oscillations behind the stronger shock. These oscillations are related to jumps in entropy production where the shock moves from an element to another. This behavior is probably related to the fact that the filter removes the higher-order modes in the elements crossed by the shock and so the solution in this region is nearly piecewise constant. It is interesting to observe that in the test case of Section 6.1.5 there are strong curvilinear shock waves but there are not significant oscillations in the region behind the shocks. However, in that case a very fine grid was employed while in this problem the mesh is relatively coarse. These considerations suggest that the use of local grid refinement in the shock region could help in reducing this kind of post-shock oscillations.

In Figure 6.11 and Figure 6.12 the wall pressure coefficient (Cp) distribution for the DG1 and DG5 computations is reported. For the purpose of comparison, DG5 results obtained by Barter [102] with the artificial viscosity approach of Persson and Peraire [86] are also plotted. It can be seen that the shock on the pressure side appears very similar with the two approaches, since in both cases there is an undershoot. On the other hand, the shock on the suction side is captured more sharply by the artificial viscosity approach. The magnitude of the post-shocks oscillations are comparable to those obtained by other common shock capturing DG schemes reported in the ADIGMA report [85] for this test case. The lift and drag coefficients reported in Table 4 converge when the order is increased. They are in good agreement with the reference values obtained by Vassberg and Jameson [103] with a second order finite volume method on a very fine grid (4096x4096).

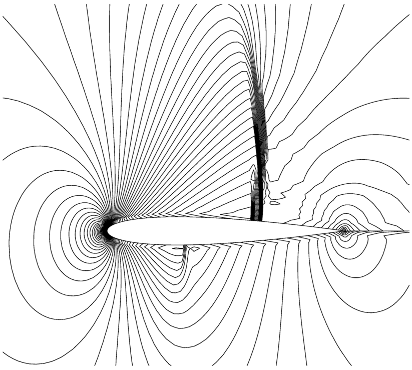

Figure 6.9. Mach field on the NACA0012 airfoil ($M_\infty = 0.8$,$\alpha_\infty = 1.25^o$): feedback filtered DG1 scheme.

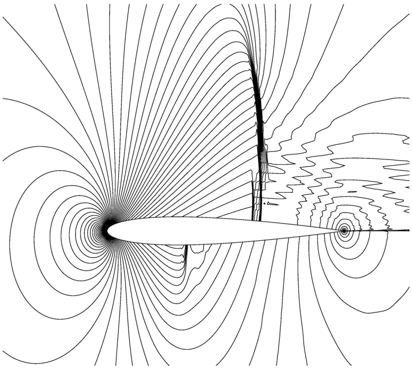

Figure 6.10. Mach field on the NACA0012 airfoil ($M_\infty = 0.8$,$\alpha_\infty = 1.25^o$): feedback filtered DG5 scheme.

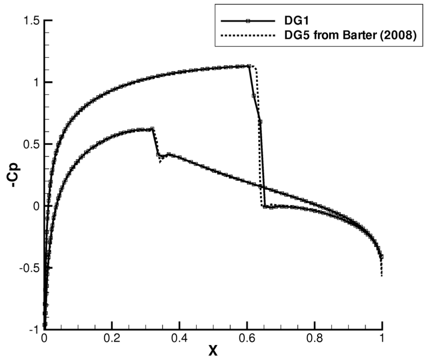

Figure 6.11. Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme).

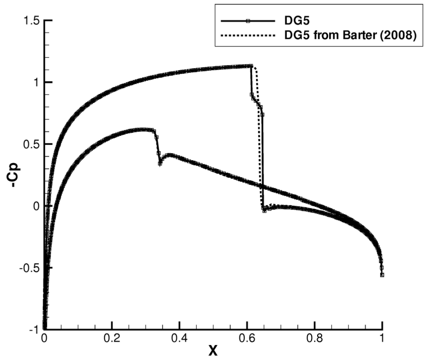

Figure 6.12. Numerical Schlieren image for shock-bubble interaction at $t = 0.18$ (Feedback filtered DG2 scheme).

6.2 Artificial viscosity

In this work the artificial viscosity method proposed by Nguyen and Peraire [89] has been implemented. It can be activated in transonic or supersonic problems characterized by the presence of shock waves. The method is based on a laplacian term which is added to the governing equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{F} - \nu_a \nabla \mathbf{u}) = 0 \tag{6.17}$$

where $\mathbf{u}$ and $\mathbf{F}$ are the vector of the conservative variables and the vector of the physical fluxes, respectively. The artificial viscosity ν_a can be computed as:

$$\nu_a = \epsilon_0 f \left(\frac{l \nabla \cdot \mathbf{q}}{a} \right) \tag{6.18}$$

where ϵ_0, h, $\mathbf{q}$ and a are a positive user-defined constant, a characteristic length, the velocity vector and the speed of sound, respectively. The characteristic lenght l depends on the element characteristic size h and on the distance from the closest wall d. It can be obtained by the following relation, which guarantees zero artificial viscosity at wall:

$$l = \min(h, 10d) \tag{6.19}$$

In the present work, when Eq. 6.19 is evaluated at an internal interface, an average element characteristic size is computed between the two elements.
The function f is:

$$f(x) = \alpha_1 \log\left(1 + e^{(\alpha_2 - x)/\alpha_1}\right) \tag{6.20}$$

where the constants are $\alpha_1 = 0.1$ and $\alpha_2 = -0.5$. It can be seen that the previous relation gives a strong artificial viscosity when the divergence of velocity becomes negative.

The described shock capturing approach requires the definition of a parameter ϵ_0 which controls the maximum amount of artificial viscosity that can be introduced in the discretization. This parameter is problem dependent. Some preliminary experiments performed during this work have shown that a bad choice of its value can significantly alter the results of the simulation, especially in the presence of boundary layers.
A possible value for inviscid transonic computations (see Chapter 7.3.1) can be $\epsilon_0 = 0.004$.

6.2.1 Future works

Some preliminary experiments were done with the implemented artificial viscosity method in viscous flow problems. In particular, the tests performed on the laminar

flow over a flat plate showed that the implemented artificial viscosity can significantly alter the velocity profile in the boundary layer if the parameter ϵ_0 is set to the value used for the transonic NACA0012 problem (see Chapter 7.3.1). This means that this artificial viscosity approach is strongly problem dependent.
An alternative approach could be the shock capturing term proposed by Bassi et al. [82]. This method introduces an artificial viscosity which depends on the inviscid residuals of the discretization. This approach makes the scheme consistent to the Euler equations but only asymptotically consistent to Navier-Stokes equations or RANS equations.
Hartmann [90] proposed an artificial viscosity approach which is based on the full residuals and so it is consistent also for Navier-Stokes and RANS equations. The introduction of these shock capturing approaches could bring several benefits.

Chapter 7

Adaptive algorithms

Fluid dynamics is characterized by problems in which the complexity of the solution can change significantly from zone to zone. Indeed, there are several structures (shock waves, boundary layers, wakes, turbulent structures,...) which are localized in small portions of the domain but which have great influence on the results. In order to increase the efficiency of the discretization, it is necessary to adapt the discretization to the physics of the problem. This can be done by concentrating the degrees of freedom in the regions which require more attention.
This approach is very important in unsteady problems in which the most interesting structures move across the domain. Furthermore, the possibility to automatically adapt the discretization is useful in the framework of simulation-based optimizations, in which several geometries have to be tested and it is not possible to manually adapt each mesh.
There are two main families of adaptive algorithms: p-adaptive algorithms and h-adaptive algorithms. The local nature of the discontinuous Galerkin reconstruction allows an easy implementation of both strategies.

P-adaptive algorithms exploit a local variation of the reconstruction order (p) across the domain without any modification to the mesh. In the framework of DG methods, this can be done without any change in the stencil of the reconstruction. Furthermore, if a hierarchical basis is employed, the implementation of a p-adaptive algorithm becomes very simple. Indeed, when an element needs to be refined, it is sufficient to add more degrees of freedom without the need to change the ones already known. On the other hand, if an element needs to decrease the reconstruction order, it is enough to eliminate the degrees of freedom related to the higher order modes.
If the p-adaptation process is driven by a proper error indicator, then an exponential convergence of the error can be obtained in sufficiently smooth problems.

On the other hand, p-adaptive algorithms are not the best approach in the presence of singular solutions. Indeed, the increase in the reconstruction order gives small benefits on shock waves or near singular geometries. In these cases the numerical convergence order is far from the theoretical one. For this reason, it is more convenient to keep the reconstruction order constant and to reduce the size of the elements. However, the error convergence rate obtained by this approach is not exponential but only algebraic.
Finally, it is possible to use mixed approaches in which both p-adaptive and h-adaptive strategies are employed. Several works are concerned with these algorithms in the framework of discontinuous Galerkin methods. An overview of some techniques is presented in the book which summarizes the results of the ADIGMA European project [104].

7.1 Sensors and error indicators

The adaptation process requires a suitable sensor to drive the algorithm. The sensor should be able to detect the main sources of errors and its evaluation should be as cheap as possible. In particular, the sensors which use quantities already computed by the discretization scheme are more suitable for unsteady problems in which the adaption has to be performed at each time step. In this work, several approaches have been tested.

Some "feature-based" sensors have been implemented as first step. They control some specific features of the solution which are indicative of the most interesting regions in the domain. These sensors are very easy to implement but are problem dependent. They have been used in this work mainly to test the behavior of the ESR method when the reconstruction order varies from an element to its neighbours. These algorithms are based on a sensor and a control law. The choice of the sensor depends on the particular set of equations which is considered. For example it can be related to the intensity of a variable or to its gradient.

A more rigorous approach can be obtained by the choice of an error indicator. There are two main families of error indicators: adjoint-based and residual-based error indicators. Adjoint-based error indicators are obtained by the definition of a dual problem. They identify the sources of error for a given target function. The aim of this approach is to minimize the error on a particular goal (for example the lift or drag coefficients of a wing) for a given number of degrees of freedom employed in the numerical solution. Several works based on this approach can be found in the literature, like for example [105].

Residual-based error indicators try to estimate the discretization error on the conservative variables and so they target at resolving all flow features, not only those related to a particular target function. Furthermore, they are very cheap because they make use of the residual of the governing equations which are already available. In this work, a residual-based error indicator has been implemented following the approach of Leicht and Hartmann [106]. In particular, here the error indicator η_e has been implemented for inviscid flows in the following way:

$$\eta_e = h_e|\mathbf{R}(\mathbf{u}_h)|_e + \sqrt{h_e}|\mathbf{r}_{\partial e}(\mathbf{u}_h)|_{\partial e} \tag{7.1}$$

where h_e is the characteristic element size while $\mathbf{R}(\mathbf{u}_h)$ and $\mathbf{r}_{\partial e}(\mathbf{u}_h)$ are the volume and surface residual defined as:

$$\mathbf{R}(\mathbf{u}_h) = -\frac{\partial F_h}{\partial x} - \frac{\partial G_h}{\partial y} \tag{7.2}$$

$$\mathbf{r}_{\partial e}(\mathbf{u}_h) = n_x F_h + n_y G_h - (\hat{F}_h n_x + \hat{G}_h n_y) \tag{7.3}$$

The notation of Eqs. 7.2 and 7.3 refers to Eq. 3.6.

7.2 p-adaptive algorithm

P-adaptive algorithms are very useful in smooth and regular problems. In this work, they have been tested on the linear heat equation and the compressible Navier-Stokes equations. The main purpose of these tests is to study the behavior of the ESR method in problems with an inhomogeneous order distribution. For this reason, a very simple p-adaptive strategy is used and a sensor based on some flow features is chosen. A more rigorous approach can be obtained by the use of an error indicator, similar to Eq. 7.1.

The implemented strategy makes use of a feature-based sensor and a control law. The control law is responsible for the choice of the element order p_e according to the signal obtained locally by the sensor s_e. The following very simple control law has been chosen:

$$p_e = \mathrm{Int}\left[p_{min} + (p_{max} - p_{min})\left(\frac{s_e - s_{min}}{s_{max} - s_{min}}\right)^{\alpha}_{p}\right] \tag{7.4}$$

where the function Int rounds its argument to the nearest integer number.

Eq. (7.4) shows that the reconstruction order in a generic element is allowed to vary between a minimum (p_{min}) and a maximum (p_{max}). The local value of the sensor (s_e) is compared to the minimum (s_{min}) and maximum (s_{max}) values found in the computational domain at the considered time step. The coefficient α_p is a positive number which controls the distribution of the order in the domain: the lower the α_p, the larger the number of elements in which high order reconstructions are employed. The chosen law helps in avoiding strong jumps in the order between neighbouring elements.

7.2.1 Heat equation and p-adaptivity

Several numerical experiments were performed on the 2D heat equation with the p-adaptive algorithm and the ESR method [2]. The influence of the parameter α_p was also investigated. The considered test case is similar to the one used before for the convergence analysis (Section 4.4.3), but now the number of semi-waves is increased up to 5 ($k_w = 5$) as shown in Figure 7.1. The exact solution is characterized by regions in which there are large gradients and regions in which the temperature distribution is close to constant. In a situation like this, the p-adaptive algorithm can show its advantages with respect to a uniform order discretization. The same unstructured meshes used in Section 4.4.3 were adopted for these experiments. The magnitude of the temperature gradient was chosen as sensor variable.

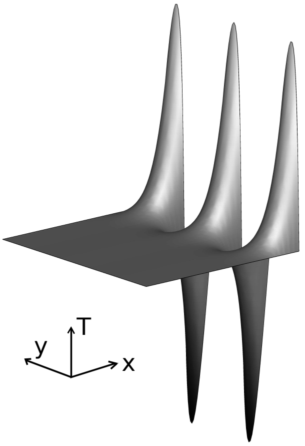

Figure 7.1. Temperature field for the 2D heat equation problem, with $k_w = 5$.

In the plot of Figure 7.3 the L^2-norm of the temperature error is reported as a function of the total number of degrees of freedom. DG1 and DG5 denote data obtained by a uniform order of accuracy, namely $p = 1$ and $p = 5$. The other data are obtained by the p-adaptive strategy (with $p_{min} = 1$ and $p_{max} = 5$). The control law defined by Eq.7.4 is used for this test. It can be seen that the p-adaptive

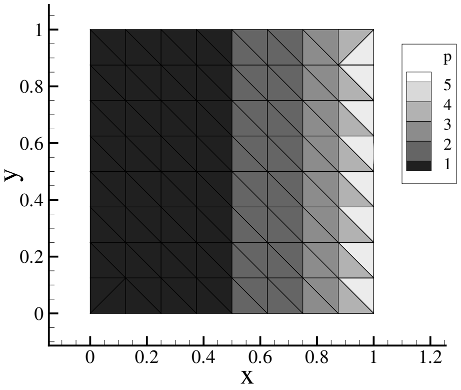

Figure 7.2. Example of order distribution with p-adaptivity and the 8x8x2 mesh ($\alpha_p = 0.3$)

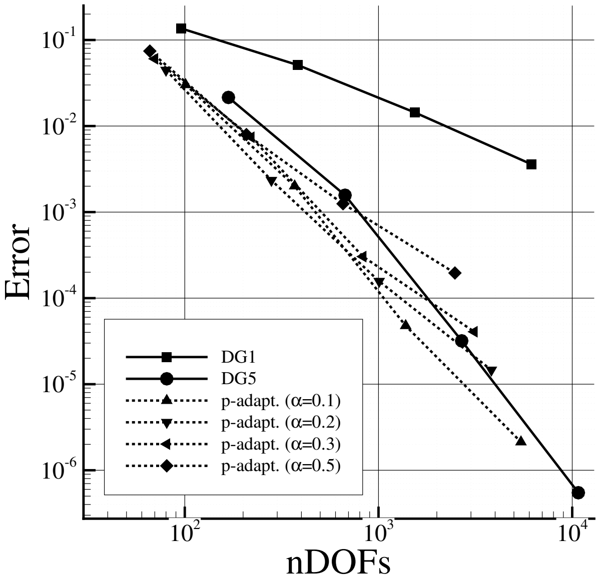

Figure 7.3. L^2-norm of the temperature error vs total number of DOF's (possible orders: p_e = 1,2,3,4,5).

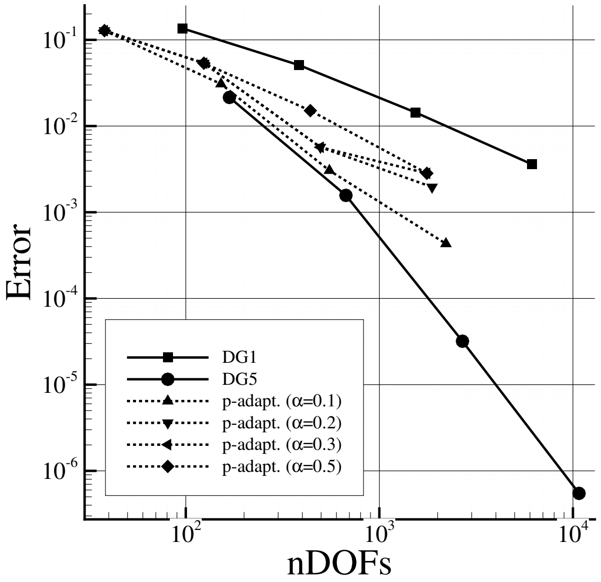

Figure 7.4. L^2-norm of the temperature error vs total number of DOF's (possible orders: p_e = 1,3,5)

algorithm is convenient with respect to the uniform order discretization if a sufficiently low value of α_p is chosen. The results suggest that the adaptive algorithm is competitive for $0.1 \leq \alpha_p \leq 0.2$.

Another test was performed in order to study the effects of the proposed discretization for the case $|p_A - p_B| > 1$, which is described in Section **??**. In particular, the control law (Eq.7.4) was modified in such a way that the algorithm can choose only the following reconstruction orders: $p_e = 1$, $p_e = 3$ or $p_e = 5$. In this way, when there is a jump in the order across an interface, the jump is always greater or equal than 2. The results for this somewhat artificial case are reported in Figure 7.4. A comparison with Figure 7.3 shows clearly that the presence of strong jumps in the order distribution is not convenient. For this reason, it is important to use a control law which chooses a smooth distribution of the order and which discourages the occurrence of the case $|p_A - p_B| > 1$.

7.2.2 Vortex shedding around a circular cylinder

In this classical test case, the unsteady flow field around a circular cylinder with $Re_\infty = 10^2$ and $M_\infty = 0.2$ is considered. Characteristic far field boundary conditions are prescribed on a circular boundary. The distance of the far field from the body is equal to 100 cylinder diameters. The solid wall is adiabatic. The domain is discretized with a structured O-mesh with 19x48 quadrilateral elements. Wall elements have quartic edges. Convective fluxes are computed by the Osher [35] numerical flux while diffusive fluxed are computed by the ESR method [2].

Five simulations were performed with schemes ranging from DG1 to DG5 in order to evaluate the behavior of the proposed discretization in the presence of unsteady flow fields. Computations were carried out until periodicity was reached. The Strouhal number (St) related to vortex shedding and the average drag coefficient were evaluated over a period of 10 cycles. After that, the p-adaptive algorithm was tested and the order of the elements was allowed to vary between $p_{min} = 1$ and $p_{max} = 5$ at each time step.

First of all, some experiments with the algorithm's sensor were performed. Figure 7.5 and Figure 7.6 show the order distribution in the region near the cylinder when the magnitude of the velocity gradient or the entropy are used as sensors. In the first case, it can be seen that the separated flow zone does not have strong velocity gradients, and as a consequence the algorithm chooses to use low order reconstructions here. On the other hand, if entropy is used as sensor, the wake and the separation zone are correctly detected, but low order reconstructions are selected in front of the cylinder. For this reasons a mixed approach was adopted: both sensors were used and the order of the element was set to the maximum between the orders suggested by the two sensors (see Figure 7.7). In all cases the control law described by Eq. (7.4) with $\alpha_p = 0.3$ was employed.

Scheme	nDOFs/eq	Average C_D	St
DG1-RK2TVD	2736	1.287	0.137
DG2-RK3TVD	5472	1.334	0.161
DG3-RK4LS	9120	1.334	0.163
DG4-RK4LS	13680	1.336	0.163
p-adapt.-RK4LS	5200-5300	1.335	0.163

Table 7.1. Average drag coefficient and Strouhal number for cylinder flow ($Re_\infty = 10^2, M_\infty = 0.2$).

Author	Average C_D	St
Norberg [107]	-	0.164
Tritton [108]	1.26	0.157-0.164
Wieselsberger [109]	1.43	-
Williamson [110]	1.33	0.160-0.164

Table 7.2. Experimental results from the literature for cylinder flow ($Re_\infty = 10^2$)

Table 7.1 reports the results obtained as a function of the total number of degrees of freedom per equation. Note that in the case of p-adaptivity the reported number of degrees of freedom refers to the range spanned in a period. The average drag coefficient and the Strouhal number are in line with the experimental data available in the literature (see Table 7.2). It can be seen from these results that the strong clustering of the mesh makes it difficult to resolve the wake in regions far from the cylinder. The natural remedy for this problem is the use of a hybrid *hp*-adaptive algorithm, which will be described in Section 7.4.

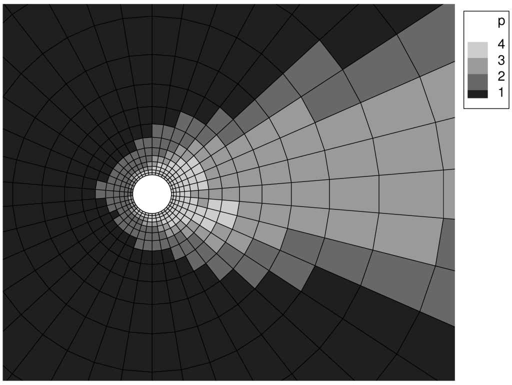

Figure 7.5. Distribution of the reconstruction order with the entropy sensor.

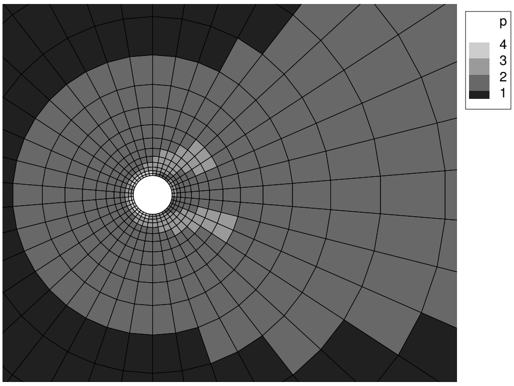

Figure 7.6. Distribution of the reconstruction order with the sensor based on velocity gradients.

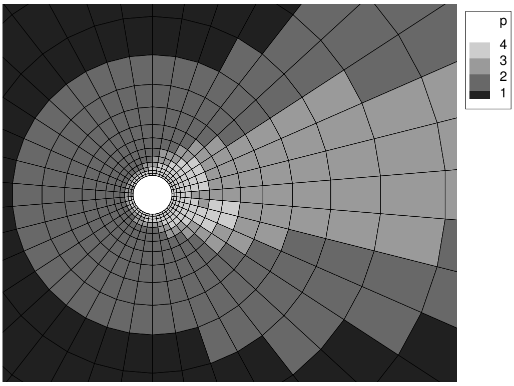

Figure 7.7. Distribution of the reconstruction order using both the entropy and the velocity gradient sensors.

Figure 7.8. Entropy field around a circular cylinder with p-adaptivity ($Re_\infty = 10^2$).

7.3 h-adaptive algorithm

Several flows in the field of aerospace propulsion are characterized by the presence of shock waves or other singular features. For this reason, an h-adaptive algorithm has been developed during the research activity. This algorithm is based on isotropic splitting of triangular meshes. When an element requires to be refined, it is split in four elements and then it is deactivated but kept in the memory. In particular, a hierarchical data structure has been introduced: each element knows his "sons" and his "father". In this way, it is possible not only to refine the mesh but also to re-agglomerate elements. This feature is useful in unsteady problems in which there are moving shock waves inside the domain.

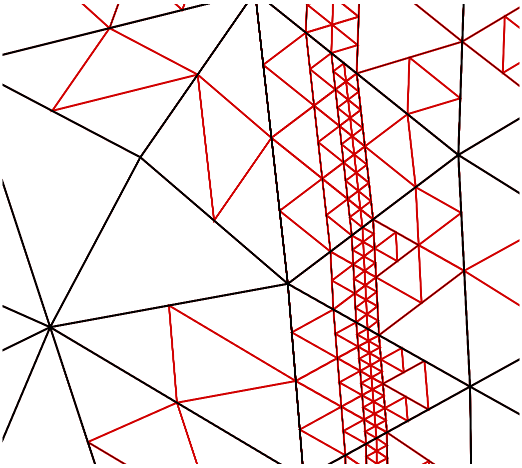

Figure 7.9. Isotropic splitting on unstructured meshes: initial (black) and adapted (red) meshes.

The meshes obtained by the application of this algorithm cointain hanging nodes (see Figure7.9). This is not a problem in the framework of discontinuous Galerkin methods thanks to the locality of the reconstruction. On the contrary, finite volume methods or continuous Galerkin finite elements methods require special treatments for the reconstruction on non-conforming meshes.

7.3.1 Transonic NACA0012: h-adaptivity

The described h-adaptive algorithm, together with the error indicator of Eq. 7.1,was used for an inviscid transonic test case on the NACA0012 airfoil. This test case was proposed at the First, Second and Third Workshops on High-Order CFD Methods ([68], [11], [69]). The computations performed during the research activity were submitted to the Third Edition of the Workshop (Orlando, 2005).
The flow field is characterized by a transonic inviscid flow on a NACA0012 airfoil. The freestream conditions are $M_\infty = 0.8$ and $\alpha_\infty = 1.25^o$. The original NACA0012 airfoil has a finite trailing edge of 0.252% chords. The following modified version

with a different x^4 coefficient is considered in order to have a trailing edge with zero thichness:

$$y = \pm 0.6\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 + 0.1036x^4\right) \quad 0 \leq x \leq 1 \tag{7.5}$$

The flow is characterized by the presence of two shock waves: a strong shock wave on the suction side and a weak shock wave on the pressure side. Furthermore, the sharp trailing edge introduces a singularity in the solution.

In order to efficiently use the backward Euler implicit integration scheme, the artificial viscosity approach described in Section 6.2 is chosen. The artificial viscosity parameter is set as $\epsilon_0 = 0.004$ in all the simulations for this test case. Convective fluxes are approximated by the Osher numerical flux [35] while diffusive fluxes are approximated by the ESR method [2].

The computational domain is reported in Figure 7.14. The freestream boundaries are at more than 2000 chords from the airfoil, according to the suggestions reported on the Workshop website [69].

In Figure 7.10 the initial mesh is reported: it is a very coarse mesh with 1060 triangular elements. Wall elements have curvilinear cubic edges. Since this is a steady problem, the adaptation process refines the initial mesh recursively, without the need to re-agglomerate elements.

In order to accelerate convergence, the solution is obtained through a sequence of progressively higher-order reconstructions according to the strategy described in Section 8.1. The adaptation procedure is performed when the final reconstruction order is reached. At each refinement step the algorithm splits 20 % of the elements which are characterized by the largest values of the error indicator. After that, a further check and splitting is performed in order to avoid too large jumps in the mesh size distribution. This last step is required by the employed visualization subroutines (see Appendix C) which are not able to deal with too large jumps in the mesh. However, it is not required by the numerical discretization and can be avoided by the use of an updated visualization subroutine.

When a wall element is refined, his sons are still curvilinear cubic elements: the extra points required to define the wall geometry are obtained by Eq.7.5. In Figure 7.11 the adapted mesh after 5 refinement steps is reported. The Mach field on the initial and the adapted meshes is reported in Figure 7.12 and Figure 7.13, respectively. A comparison between Figure 7.13 and Figure 7.11 shows that the error indicator identifies correctly the main sources of error in the flow: the leading edge, the trailing edge, the shocks and the contact surface downstream of the profile.

The performances of the algorithm are evaluated for $p = 1$ and $p = 2$ by computing the errors on the lift (C_L) and drag (C_D) coefficients. The errors are evaluated by taking the RWTH values from the second edition of the Workshop [11] ($Cl = 3.52914010^{-1}$;$Cd = 2.27463610^{-2}$) as reference. In Figure 7.16 and

Figure 7.15 the errors on the lift and drag coefficients are reported as a function of the equivalent length scale. It can be seen that the convergence behavior is quite irregular. This is in line with the results obtained by other groups and is due to the presence of singularities in the solution. Furthermore, the results show that, with the setup chosen for this test, there is no advantage in using uniform high-order reconstructions in the presence of discontinuities. Indeed, the results for $p = 1$ and $p = 2$ are quite close in Figure 7.15 while in Figure 7.16 the $p = 1$ reconstruction performs better than the $p = 2$ scheme. A similar behavior can be seen in Figure 7.15 and Figure 7.16 where the errors are plotted as a function of the work units. These results suggest the introduction of an hybrid hp-adaptive strategy, which will be described in the following Section.

Further investigations should be performed in order to clearly understand why the h-adaptive $p = 1$ scheme performs better than the h-adaptive $p = 2$ scheme in this test case. A possible answer is related to the fact that the extra degrees of freedom introduced by the higher-order reconstructions increase significantly the cost of the computation but they introduce a small benefit. Indeed, the global error is dominated by the presence of the singularities on which the high-order schemes show an accuracy degradation. Starting from this point, it could be useful to introduce a very aggressive h-refinement strategy in order to keep the error related to the singularities sufficiently low: in this way the high-order schemes would be able to show their accuracy which would not be hided by the errors on the singularities. If this hypothesis is correct, than the h-adaptive strategy should become more aggressive when high-order reconstructions are chosen. This would explain why the proposed h-adaptive strategy (which is very "soft" because at each refinement the size of the elements is only halved) gives good results for $p = 1$ but it is not convenient for $p = 2$. In order to adopt more aggressive h-refinement strategies it would be possible to remesh the domain after each adaptation instead of splitting the elements. However, this last approach would be very expensive in unsteady problems.

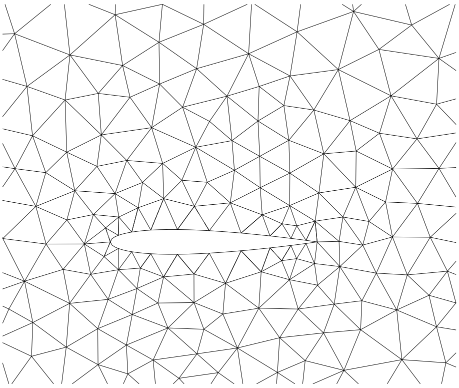

Figure 7.10. Initial mesh (1060 elements).

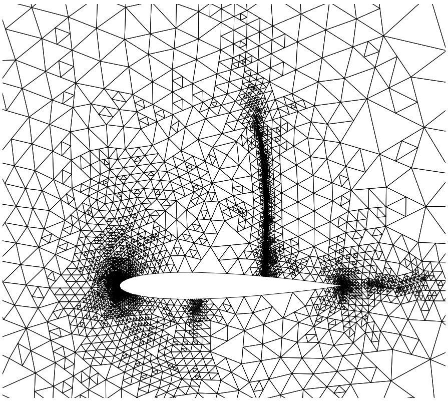

Figure 7.11. Adapted mesh after 5 h-refinements (11806 elements).

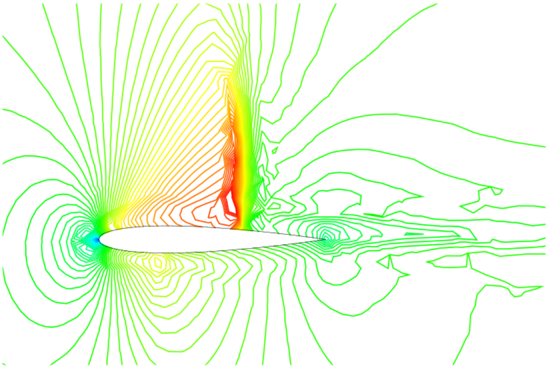

Figure 7.12. Transonic NACA0012: Mach field on the initial mesh (p=1).

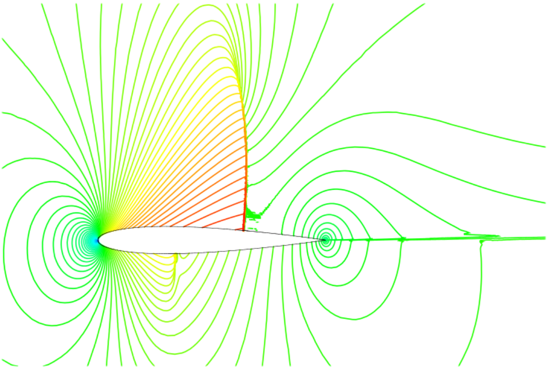

Figure 7.13. Transonic NACA0012: Mach field on the adapted mesh after 5 h-refinements (p=1).

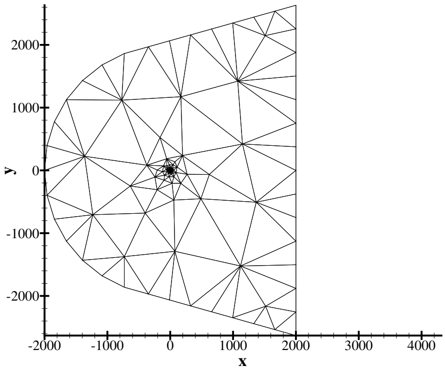

Figure 7.14. Computational domain for NACA0012 problem.

7.4 hp-adaptive algorithm

In the previous Sections of this Chapter both p-adaptive and h-adaptive algorithms are described. Both algorithms are more efficient than uniform (order and mesh refinement) discretization in smooth and discontinuous problems, respectively. However, real problems are characterized by a mix of smooth and singular regions. For this reason, it is natural to introduce an hybrid approach in which p-refinement is used in smooth regions and h-refinement is used on singularities.

The previously described test case on the NACA0012 airfoil is considered here as example. Since the flow is steady and the initial mesh is very coarse then the algorithm performs only element splitting and there is no need to agglomerate elements.
The approach chosen for this work can be summarized by the following steps:

- Evaluate the error indicator (Eq. 7.1) in each element
- Select a fraction of the total elements (e.g. 20%) with the largest error
- Perform a smoothness test on the selected elements and choose between h or p-refinement

The smoothness test is performed by the discontinuity indicator proposed by Persson and Peraire [86] applied to pressure. The indicator is the same which is used for the feedback filtering algorithm (see Section 6.1). The threshold chosen for this test case is:

$$SR = -2.5 - 4\log_{10}(p_e) \tag{7.6}$$

where p_e is the reconstruction order inside the element. If $SI > SR$ then h-refinement is chosen. Otherwise, the order of the element in increased to the level $p_e + 1$.

The results reported in Figure 7.15, Figure 7.16, Figure 7.17 and Figure 7.18 show clearly the advantages of the hp-adaptive algorithm over the h-adaptive algorithm, both in terms of equivalent length scale and work units.

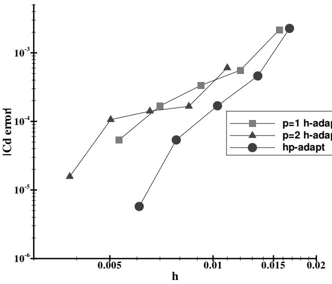

Figure 7.15. Transonic NACA0012: Cd error vs length scale

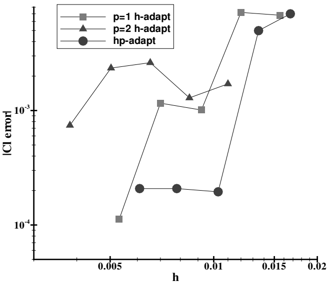

Figure 7.16. Transonic NACA0012: Cl error vs length scale

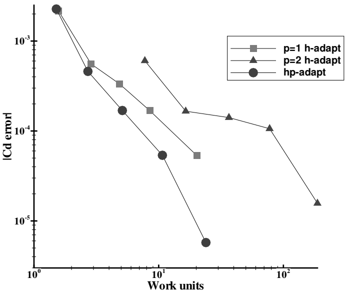

Figure 7.17. Transonic NACA0012: Cd error vs work units

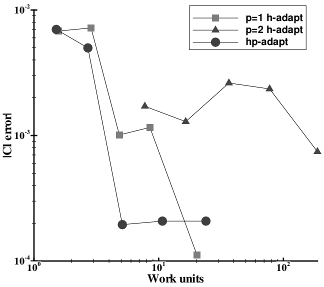

Figure 7.18. Transonic NACA0012: Cl error vs work units

Part III

Numerical results

Chapter 8

Performances

This Chapter is devoted to the performances of the proposed numerical discretization. It is subdivided in two Sections. The first Section is devoted to the techniques that have been implemented in this work in order to accelerate the convergence for steady problems. In the second Section there are the results for some test cases from three Workshop on High-Order CFD Methods ([68], [11], [69]).

8.1 Acceleration for steady problems

One of the lessons learned from CFD Workshops [68],[11],[69] is that the computational time required to obtain a given error level is strongly affected by both the chosen numerical methods and the implementation. Indeed, it is quite common to see different groups which obtain the same error level with very different computational times (for example by a factor 10^2 or 10^3). These differences become even more evident when steady test cases are considered. Indeed, the total computational time required to obtain the converged steady-state solution can be dramatically reduced by the use of some acceleration techniques.

First of all, implicit time integration schemes can be quite effective in damping the transient. This is particularly true when the physical model contains equations with stiff terms, which can represent a severe bottle-neck for explicit time integration schemes. For example, in Section 4.5 the advantages of the implicit approach over the explicit RK4 scheme are clearly shown in a heat diffusion problem, as can be seen in Figure 4.8. Similar advantages can be found in more complex system of equations, like for example Navier-Stokes equations or RANS equations. However, in this case the benefits are reduced by the presence of non-linear terms which can introduce a limit on the maximum allowable CFL number, even with implicit time

integration schemes. For example, the presence of moving shock waves inside the domain introduces a severe limitation on the CFL number.
In this work, the robustness of the implicit time integration is increased by the introduction of some positivity checks on pressure and density: after each time step, the sign of pressure and density in the quadrature points is checked. If a negative value is found, then the last update is removed and the solution is brought back to the previous time step. After that a smaller time step is chosen and the procedure is repeated. This approach is used together with several CFL evolution laws, which are described in Section 5.2.3. It is important to keep in mind that a small variation in the parameters which control the CFL evolution law can change the order of magnitude of the total computational time. Clearly, a compromise between speed and robustness has to be chosen.
Finally, it is necessary to remember the main drawback of implicit schemes: the large memory requirements. Indeed, the jacobian matrix which results from the implicit discretization is very large and its storage in the memory can be prohibitively expensive in complex 3D problems.

An effective technique which can reduce the computational time in steady problems is the multigrid method. This approach is commonly used in the framework of finite volume methods. The solution is advanced in time on several levels of nested grids: the idea is that the low frequency oscillations which characterized the transient on the original grid can be be damped more quickly on a coarser grids, where they appear as high frequency oscillations. The described approach can be referred to as h-multigrid, where the "h" indicates that the acceleration is obtained by exploiting different grids.
In the framework of discontinuous Galerkin methods, an alternative approach can be chosen: the p-multigrid method. The idea is to substitute the different level of grids with different reconstruction orders. Furthermore, if the solution is described by a hierarchical basis, it is very easy to reduce and extend the solution from a level to the next. Several works have been done on this topic (see for example [111], [112], [113]). However, some authors have found that p-multigrid techniques applied to DG discretization are not very efficient if the time integration is performed by explicit schemes [112]. For this reason, p-multigrid DG methods are usually associated with implicit time integration, at least for the coarser level of the solution which involve less degrees of freedom and so less memory requirements. This behavior is in contrast to what happens in the framework of finite volume methods in which h-multigrid methods are efficiently used together with explicit RK schemes.
In this work, the multigrid approach has not been implemented yet but a simpler (but less effective) approach is used. In particular, the hierarchical nature of the employed modal bases is exploited to perform a sequence of computations in which the reconstruction order p is increased from $p = 0$ to the final p. In this way, a

sequence of progressively more accurate (and more expensive) solution is obtained, and each solution is initialized by the previous coarser solution.
In Figure 8.1 the evolution of the residuals is reported as a function of the computational time for the test case of the channel with the smooth bump (see Section 8.2.1). The computations were performed by a RK3 time integration scheme. The green curve describes the results obtained by using a $p = 2$ reconstruction from the beginning of the simulation. The blue curve refers to the results obtained by a sequence of progressively higher-order reconstructions, from $p = 0$ to $p = 2$. The plot shows that, even in this simple and smooth test case, the technique allows a reduction in the computational time.

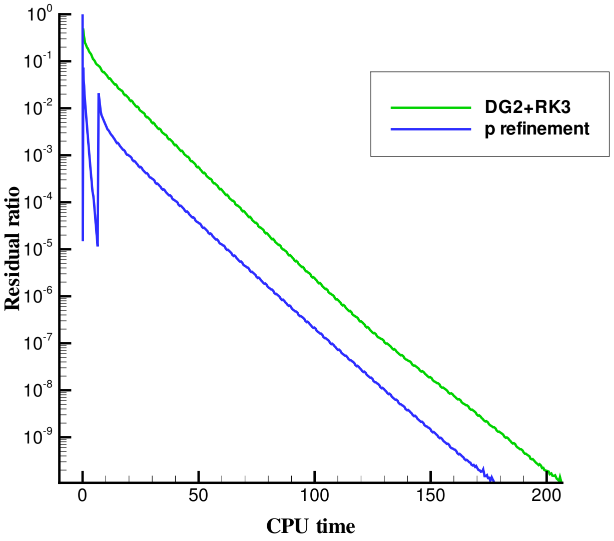

Figure 8.1. Channel with smooth bump: acceleration by p-refinement.

The approach becomes even more convenient in problems in which there are shock waves. Indeed, the initial transient is usually characterized by the presence of moving shock waves which limit the maximum CFL number. However, if the computation is started with a $p = 0$ reconstruction then an approximate solution in which the shock waves are at rest is quickly obtained. Then the expensive higher-order reconstructions are activated and they can benefit of a larger time step since the shock waves are already close to the final position.

Finally, similar benefits can be obtained by the use of h or hp adaptive algorithms. Indeed, the recursive local refinement makes it possible to obtain a sequence of progressively more accurate (and more expensive) solutions, in which each solution is initialized by the previous coarser solution.

8.2 Workshop test problems

The proposed discretization has been evaluated on several test cases from the three editions of the Workshop on High-Order CFD Methods ([68],[11],[69]). These tests are very useful because allow a direct comparison of the accuracy and the efficiency of the proposed discretization with different approaches proposed by other groups. The results for the transonic NACA0012 and the laminar flat plate test cases which were submitted to the Third Workshop [69] are not reported in this Section because they are already described in Sections 7.3.1 and 4.4.4, respectively.

8.2.1 Subsonic flow in a channel with a smooth bump

This test case was proposed at the First and Second Workshops on High-Order CFD Methods ([68],[11]). The computations performed during the research activity were submitted to the Second Edition of the Workshop.
The problem is characterized by an inviscid steady flow in a channel with a smooth bump. The flow field is subsonic. The geometry of the problem is reported in Figure 8.2.

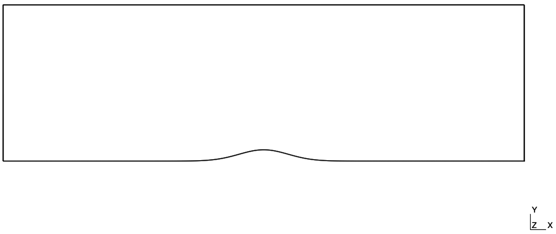

Figure 8.2. Channel with a smooth bump.

The lower and upper boundaries are described by the following laws:

$$y = 0.0625e^{-25x^2} \qquad -1.5 \leq x \leq 1.5 \tag{8.1}$$

$$y = 0.8 \qquad -1.5 \leq x \leq 1.5 \tag{8.2}$$

They are treated as solid walls. The left side is a subsonic inlet in which total temperature, total pressure and velocity angle are imposed ($T^0_{in} = 1$, $P^0_{in} = 1$ and $\alpha_{in} = 0^o$). The right side is a subsonic outlet in which the static pressure is imposed in order to obtain the prescribed exit Mach number ($M_{exit} = 0.5$).
The flow field was initialized with a uniform flow field at $M = 0.5$. The computations were carried on until the L2-norm of the density residual was reduced by 10 orders of magnitude with respect to the initial value. In particular, the residuals of the first modal coefficient (constant basis function) were monitored.
The quality of the numerical results was evaluated by the computation of the L2-norm of the entropy error as a function of the equivalent length scale. The performances of the code were evaluated by measuring the work units required to obtain the steady solution (defined as the solution obtained after a reduction of 10 orders of magnitude in the density residuals).

The results submitted by the author to the Second Edition of the Workshop were obtained by a preliminary version of the code. In particular, that version could perform only explicit time integration by means of Runge-Kutta schemes. For this reason, it was very slow in steady problems like the described one.
In order to accelerate the convergence to the steady solution, a sequence of increasing order reconstructions are employed. This approach introduces a small benefit as is shown in Figure 8.1 of Section 8.1.
These simulations were carried out by using an element basis obtained by a tensor product of Legendre polynomials. The orthonormalization of the basis was not available in that preliminary version of the code.
The domain was discretized by a sequence of structured meshes (16x4, 32x8, 64x16, 128x32). The elements at wall had curvilinear edges (up to quartic elements) while the internal elements were linear.

In Figure 8.3 the L2 entropy error is reported as a function of the equivalent length scale for several reconstruction orders (p = 1,2,3). The numerical results show a convergence rate which is very close to the theoretical value (dotted lines).
In Figure 8.4 the L2 entropy error is reported as a function of the work units. This plot shows that, in this problem, high-order methods are more convenient than low-order methods. This benefit can be significantly increased by the use of implicit time integration schemes which can dramatically reduce the convergence time of high-order discretizations in this steady problem.
The main achievement obtained by this test case is the confirm that the code is able to obtain the expected convergence rate on smooth problems with curvilinear meshes.

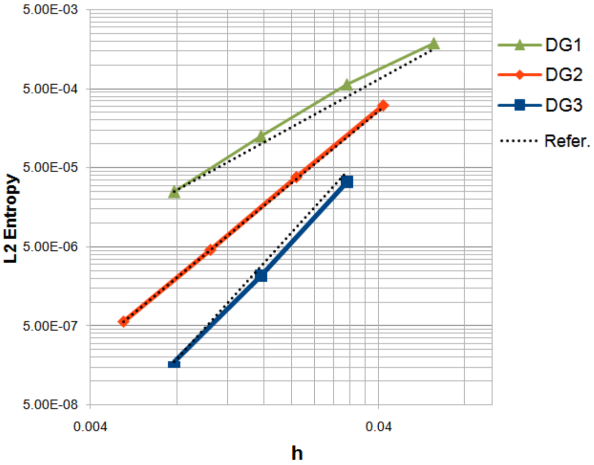

Figure 8.3. Channel with smooth bump: entropy error vs equivalent length scale.

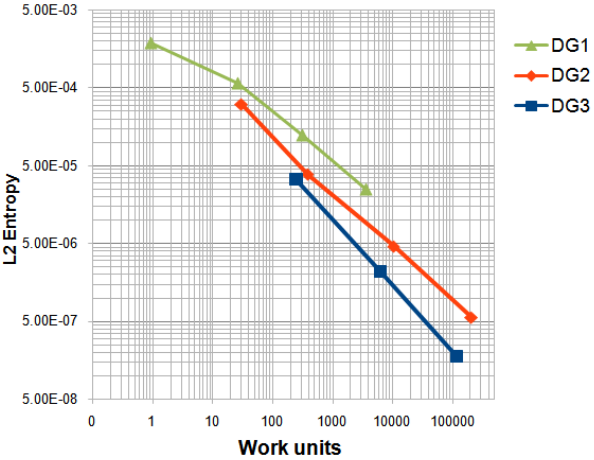

Figure 8.4. Channel with smooth bump: entropy error vs work units.

8.2.2 Transonic Ringleb flow

The Ringleb flow is a transonic inviscid smooth flow. Its analytical solution can be found by the hodograph method (see Section 6.1.6). This test case was proposed at the First, Second and Third Workshops on High-Order CFD Methods ([68], [11], [69]). The computations performed during the research activity were submitted to the Second Edition of the Workshop.

The domain is reported in Figure 8.5. It is delimited by two streamlines which are defined by the values $\kappa_{min} = 0.7$ and $\kappa_{max} = 1.5$ where κ is the streamlines function already used in Section 6.1.6. The subsonic inlet (red curve) and the subsonic outlet (blue curve) are defined by the isospeed line $\bar{q} = 0.5$ where $\bar{q}$ is defined according to Section 6.1.6. It is interesting to note the the isospeed lines are circles and so the inlet and the outlet are curvilinear boundaries. At the inlet, total temperature, total pressure and flow angle are imposed. At the outlet the static pressure is imposed in order to obtain the prescribed value for the exit velocity.

The flow field is smooth and it is characterized by a transonic acceleration region and a subsequent transonic deceleration region.

The numerical solution of this test case is difficult due to the presence of an instability in the transonic deceleration region. Indeed, if the solution is not sufficiently accurate in this zone, a spurious shock wave can appear (see Figure 8.7). It is crucial to properly describe the geometry of the boundaries. In the tests performed during this work, it was found that it is mandatory to describe the wall elements by high order curvilinear mappings. Furthermore, the coordinates of the wall boundary points have to be defined in double precision. Indeed, the small perturbations related to a single precision boundary description are sufficient to promote the development of the shock. A similar behavior has been found by other participants to the Workshops. Some groups have found that the shock can be more easily avoided if the lateral boundaries are described by an exact boundary condition instead of a solid wall boundary condition. However, in this work only the solid wall boundary condition is used. This choice is due to the fact that in general problems the exact solution is not known and so it is better to test the behavior of the discretization with a more affordable boundary condition.

The instructions for this test case require that the solution is initialized with the exact solution and then the computations have to be carried on until there is a drop of 10 orders of magnitude in the density residuals. Since the initial solution is a projection of the exact solution then the initial residuals are very low: in some cases it is impossible to reduce the density residuals by a factor equal to 10^{-10} because the residuals become too close to the machine precision. An example of the residual history for one of these cases is reported in Figure 8.9. For this reason, in some configurations a less severe convergence criteria is adopted (reduction of a factor 10^{-8}).

All the computations were performed on structured meshes (18x12, 36x24, 72x48)

generated by a free code [114] and curved by an in-house tool. The solution inside each element is described by a tensor product of Legendre polynomials.
The accuracy of the simulations was evaluated by computing the L2-norm of the entropy error on the steady solution.
In Figure 8.10 the error is reported as a function of the equivalent length scale for $p = 1,2,3$ reconstructions. In the plot there is only one point for the $p = 1$ reconstruction. This is due to the fact that a shockless solution was obtained only on the finest mesh (72x48) with $p = 1$. In contrast, the higher-order reconstructions are able to give the shockless solution even on the coarsest meshes. The observed convergence rate for $p = 2$ and $p = 3$ is very close to the theoretical value. Furthermore, the error level is good if compared to the results of the other participants.
As far as the performances are concerned, these computations were performed at the end of the first year and so only explicit time integration schemes were available in the code. In particular, explicit Runge-Kutta schemes are employed in order to obtain the same discretization order in space and time. In Figure 8.11 the error is reported as a function of the work units for the different reconstruction orders. From this plot it is not possible to understand clearly which is the most convenient reconstruction order. This irregular behavior is probably due to the fact that different convergence criterias are used when the residuals are too close to the machine precision. Furthermore, the performances of the higher-order schemes in this test case can be positively affected by the use of an implicit time integration scheme.
However, the main achievement of this test case is the confirm that the proposed discretization is able to describe the shockless transition: this is a quite severe test according to the literature [115].

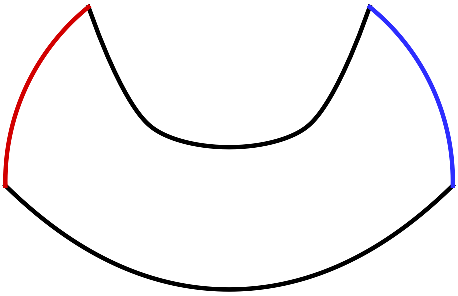

Figure 8.5. Domain for the transonic Ringleb flow.

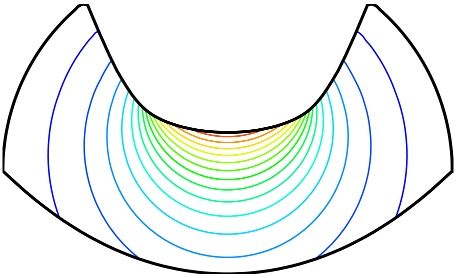

Figure 8.6. Mach field for the transonic Ringleb flow: initial solution.

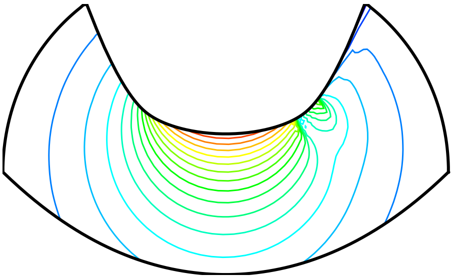

Figure 8.7. Mach field for the transonic Ringleb flow: spurious shock.

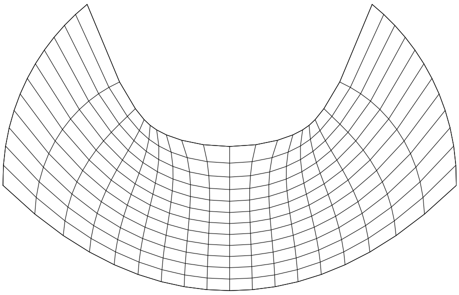

Figure 8.8. Example of structured mesh for the transonic Ringleb flow.

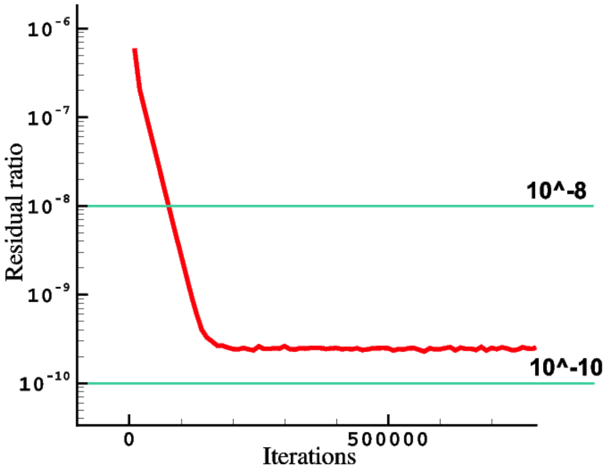

Figure 8.9. Transonic Ringleb flow: convergence history.

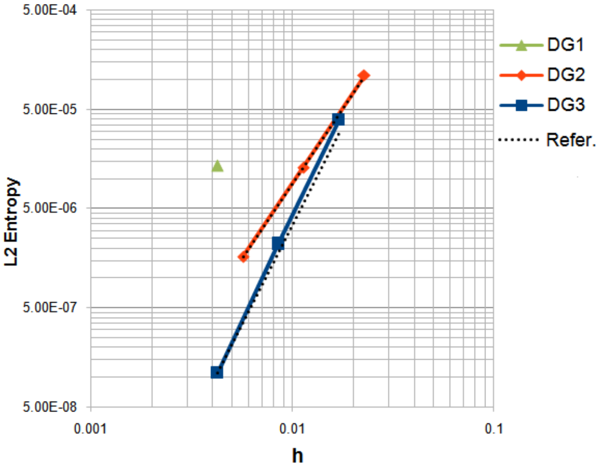

Figure 8.10. Transonic Ringleb flow: entropy error vs equivalent length scale.

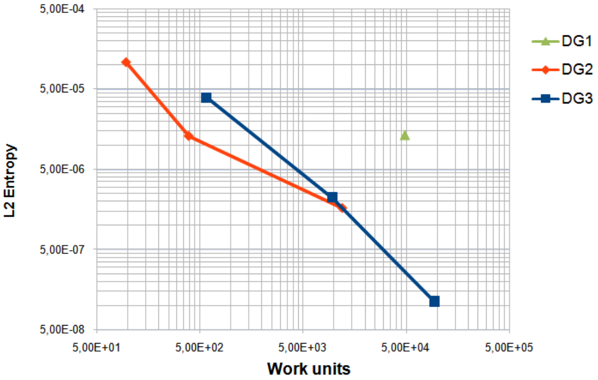

Figure 8.11. Transonic Ringleb flow: entropy error vs work units.

8.2.3 Vortex transport by uniform flow

This test case was proposed at the First, Second and Third Workshops on High-Order CFD Methods ([68], [11], [69]). The computations performed during the research activity were submitted to the Second Edition of the Workshop.
The aim of this problem is to test high-order method's capability to preserve vorticity in an unsteady inviscid flow. This capability is important in the framework of Large-Eddy and Detached-Eddy simulations and has to be preserved for a wide range of Mach number, including very low Mach numbers.
Compressible solvers can give several problems when applied to very low Mach number flows. Also this aspect is investigated in this test case.
The domain is square ([0,1] × [0,1]) and periodic conditions are imposed on all boundaries. The flow field is initialized by a uniform field ($M_\infty = 0.05$) superposed to a vortical motion described by the following perturbations on the velocity components and on the temperature:

$$\delta u = -u_\infty \bar{\beta} \frac{y - 0.5}{R} e^{-r^2/2} \tag{8.3}$$

$$\delta v = u_\infty \bar{\beta} \frac{x - 0.5}{R} e^{-r^2/2} \tag{8.4}$$

$$\delta T = 0.5(u_\infty \bar{\beta})^2 e^{-r^2} / C_p \tag{8.5}$$

$$r = \frac{\sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{R} \tag{8.6}$$

where $\bar{\beta} = 1/50$, $R = 0.05$ and C_p is the constant pressure specific heat.
In the exact solution the superposed vortex is transported without distortions by the uniform flow.
The simulation has to be carried on until 50 vortex revolutions are computed and then the L2-norm of the velocity error is computed. Since the flow is periodic the initial solution can be used as reference solution for the computation of the error.

Also for this test case, the element basis was obtained by a tensor product of Legendre polynomials without orthonormalization. Time integration is performed by means of explicit Runge-Kutta algorithms (described in Section 5.1) chosen in order to match the orders of accuracy in space and in time.
A first set of simulations was performed on cartesian (regular) structured meshes (24x24, 48x48, 96x96) for p = 1,2,3. In Figure 8.12 the distribution of the y-component of the velocity is reported at the end of the simulation. The results are reported for the $p = 1$ computation on the 64x64 mesh and for the $p = 3$ computation on the 32x32 mesh. This two computations require the same total number of degrees of freedom but the results are significantly different. Indeed, while the $p = 3$ reconstruction describes accurately the vortex, the $p = 1$ reconstruction shows a large error.
In Figure 8.13 the error is reported as a function of the equivalent length scale. It can be seen that the $p = 1$ reconstruction is very far from the asymptotic range and so its convergence rate is very low. In contrast, the $p = 2$ and $p = 3$ reconstructions reach quickly a convergence rate very close to the theoretical value. These results show that the higher-order reconstructions are significantly more convenient than the $p = 1$ scheme, for this kind of problems.
The same conclusions can be found from the plot of the work units. Indeed, Figure 8.14 shows that the work units required by the higher-order schemes to obtain a given level of error are significantly lower than the work units required by the second order scheme ($p = 1$).

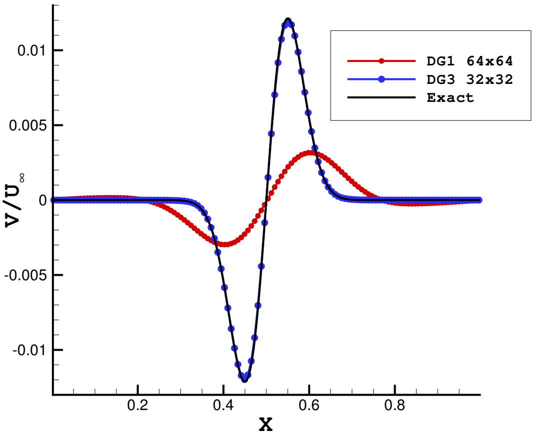

Figure 8.12. Vortex transport: final velocity distribution on regular mesh.

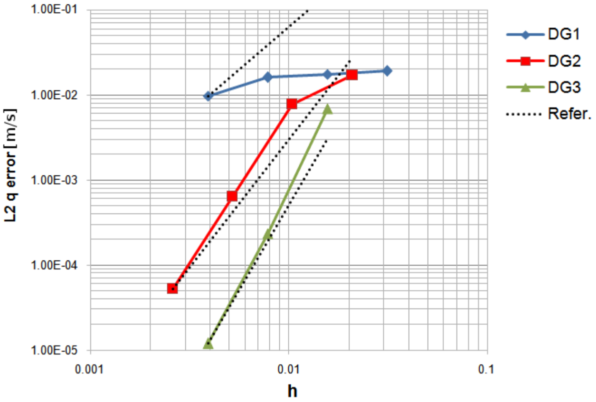

Figure 8.13. Vortex transport: error vs equivalent length scale on regular meshes.

A second set of simulations was performed on a sequence of three perturbed meshes obtained the application of random perturbartions on the coordinate of the nodes. In particular, the maximum perturbation is defined as

$$\delta_{MAX} = 0.15h \tag{8.7}$$

where h is the size of the elements. An example of this process is reported in Figure 8.15.

In Figure 8.16 and Figure 8.17 the error is reported as function of the equivalent

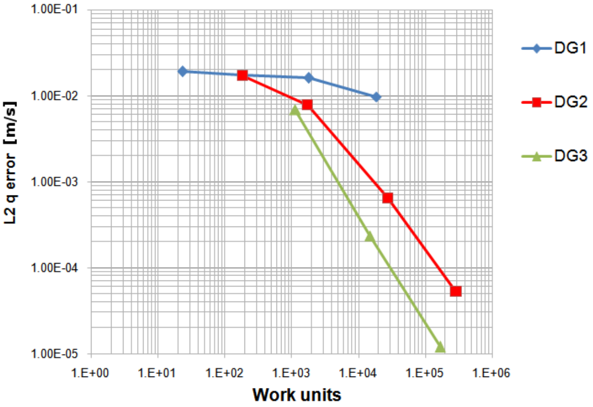

Figure 8.14. Vortex transport: error vs work units on regular meshes.

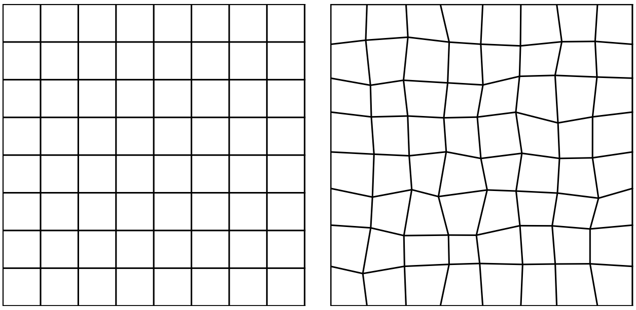

Figure 8.15. Vortex transport: example of regular and perturbed mesh.

length scale and work units for the simulations on the perturbed meshes. The results do not show significant differences with respect to the results on the regular meshes: this means that the proposed discretization is not very sensitive to the quality of the mesh and so it can be interesting for the use on irregular meshes obtained by general purpose mesh generator for complicate geometries.

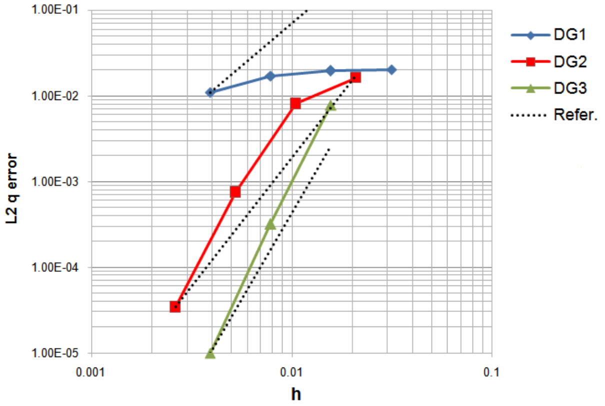

Figure 8.16. Vortex transport: error vs equivalent length scale on perturbed meshes.

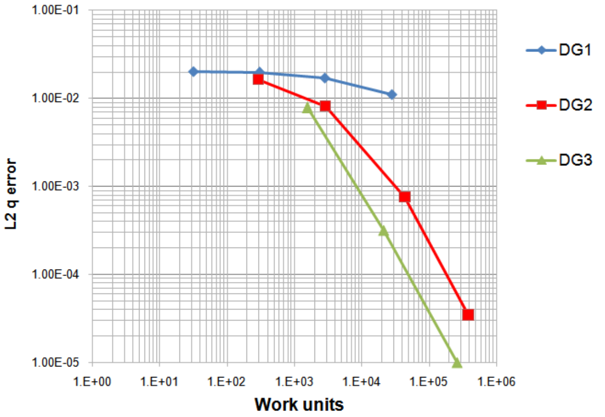

Figure 8.17. Vortex transport: error vs work units on perturbed meshes.

8.2.4 Unsteady viscous flow over tandem airfoils

This 2D test case was proposed at the First and Second edition of the International Workshop on High-Order CFD Methods ([68], [11]). It is characterized by the presence of two NACA0012 airfoils in tandem configuration. The leading airfoil is rotated by 10° about its aerodynamic center (25% of the chord) while the trailing airfoil is translated backward by 1.5 chords. The far field boundary is a circle with a radius of 100 chords centered on the leading edge of the trailing airfoil. The free-stream Mach number is $M_\infty = 0.2$ and the Reynolds number based on the chord of an airfoil is $Re_\infty = 10^4$. An initial smooth condition is prescribed according to the following relations:

$$v(x,y) = 0 \qquad P(x,y) = P_\infty \qquad \rho(x,y) = \rho_\infty$$
$$u(x,y) = \begin{cases} u_\infty & d > \delta \\ u_\infty \sin(\frac{\pi d}{2\delta}) & d \leq \delta \end{cases}, \tag{8.8}$$

where u and v are the cartesian velocity components, d is the distance to the nearest wall and δ is a constant equal to 5% of the chord length.

Two mixed structured-unstructured grids with 2101 and 8404 elements were used. Quadrilaterals were used near the airfoils and triangles in the rest of the domain. Elements with cubic edges were employed at solid walls. Here the size of the first layer of elements was equal to 0.005 chords in the direction normal to wall for the finest mesh. Figure 8.18 shows a detail of the finest mesh near the two bodies.

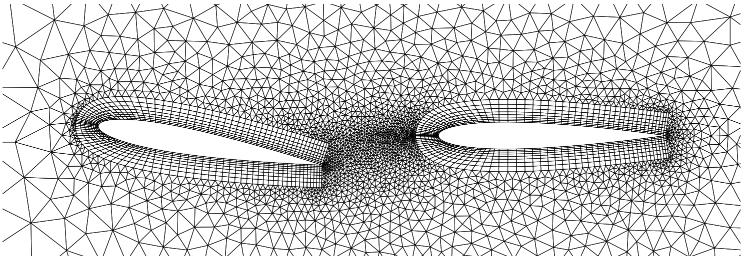

Figure 8.18. Detail of the finest mesh used for the tandem airfoils test.

The drag coefficient of the trailing airfoil was monitored during the first 20 convective times. Several simulations were performed by changing the reconstruction order from DG1 to DG5 on both meshes. Time integration was performed by RK4LS.

The flow field is characterized by the stall of the leading airfoil. The vortices which are generated in the separation region strongly interact with the trailing airfoil. A plot of the entropy field at the end of the simulation is reported in Figure 8.19 for DG5 on the finest mesh. The time evolution of the drag and lift coefficient for the trailing airfoil is reported in Figure 8.20 and Figure 8.21 for $p = 3,4,5$ on the finest mesh.

It is clear from these results that the flow is not periodic, at least in the time interval considered here. The chaotic aperiodic evolution of the vortices makes it difficult to perform a deterministic study. The results obtained by the different simulations tend to overlap for the first part of the computation but after a certain time they follow a different evolution. It can be seen that when the number of degrees of freedom is increased (with both mesh refinement or order refinement) the time interval in which the different solutions overlap becomes more extended. In particular the plots show that the DG4 method follows well the DG5 solution up to $time = 11$. In contrast, the DG3 prediction follows the higher order solutions only up to $time = 5$. A similar behavior can be seen in the results reported by other groups on the website of the First edition of the Workshop [68]. This test case showed that the proposed discretization is able to deal with strong unsteady wall-vortex interactions.

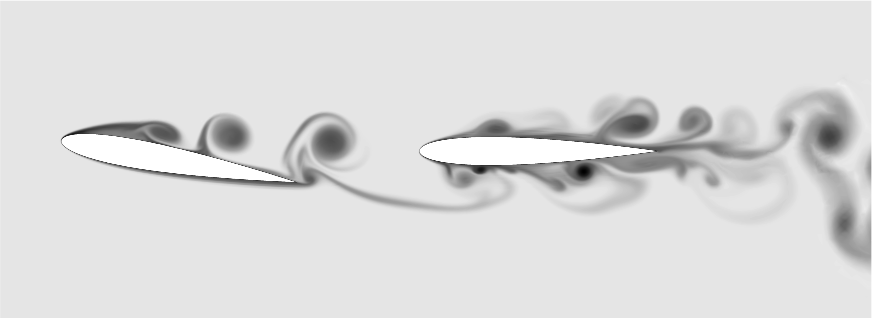

Figure 8.19. Entropy field for the tandem airfoils test at the end of the simulation (DG5RK4LS finest mesh)

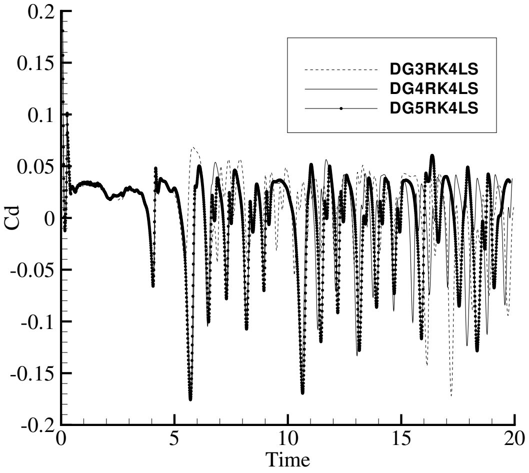

Figure 8.20. Drag coefficient for the trailing airfoil (finest mesh p=3,4,5).

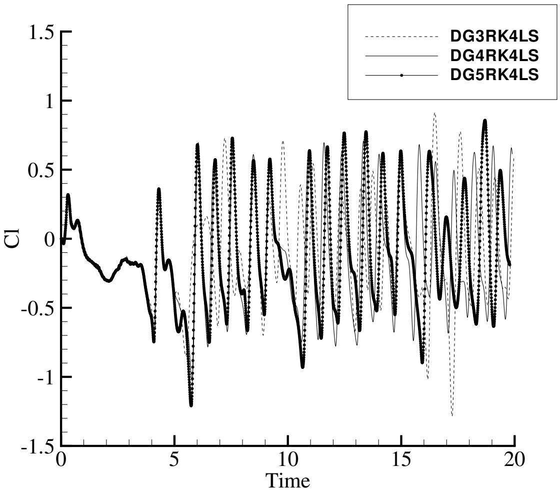

Figure 8.21. Lift coefficient for the trailing airfoil (finest mesh p=3,4,5).

Chapter 9

Aerospace propulsion examples and validation

9.1 VKI-LS59 turbine cascade

The VKI-LS59 turbine-rotor cascade was studied during the research activity in order to evaluate the behavior of the proposed discretization on high Reynolds number turbomachinery flows. The experimental data available in the literature ([116], [117]) indicate that the flow is nearly two-dimensional. Indeed, end-wall effects and aspect ratio influence are practically negligible and so it is acceptable to perform a 2D simulation.

The simulation refers to a transonic working condition, characterized by the following parameters:

$$\alpha_{in} = 30^o \qquad M_{2is} = 1.2 \qquad Re_{2is} = 6.63 \times 10^5 \tag{9.1}$$

The Reynolds number Re_{2is} is based on the blade chord c and the isentropic exit conditions.

The Spalart-Allmaras model is used in fully-turbulent mode. Indeed, the modified turbulent viscosity at the inlet is set as $\tilde{\nu} = 3\nu$, according to [17].

The computations were carried out with the basis A and a reconstruction order $p = 1$ is chosen. Shock capturing was performed by means of the feedback filtering algorithm proposed in Section 6.1. Time integration was performed by an explicit RK2 algorithm. A set of refined meshes was employed in order to evaluate the grid dependence behavior. The final mesh contains 19866 elements. It is an hybrid mesh in which there is a structured O-type zone near the blade. The surrounding region is filled with an unstructured quadrilateral mesh, as Figure 9.1 shows. Wall elements have parabolic edges. The size of the elements near the wall is $5 \times 10^{-4} c$

in the normal direction. The distance between the leading edge and the inlet is equal to one chord. The same distance is adopted between the trailing edge and the outlet.

In Figure 9.2 the computed Mach field is reported. The results show clearly the presence of the two shock waves which are generated by the trailing edge. Furthermore, it is possible to see on the suction side the interaction between the boundary layer and the shock wave which is coming from the next blade. In particular, the incoming shock wave generates two reflected shock waves due to the presence of a small separation bubble in the impinging point.

In Figure 9.3 the wall isentropic Mach number distribution is reported and compared with experimental data from [116]. The plot shows that the simulation is able to detect the main features of the distribution. In particular, the position of the shock on the suction side is correctly computed.

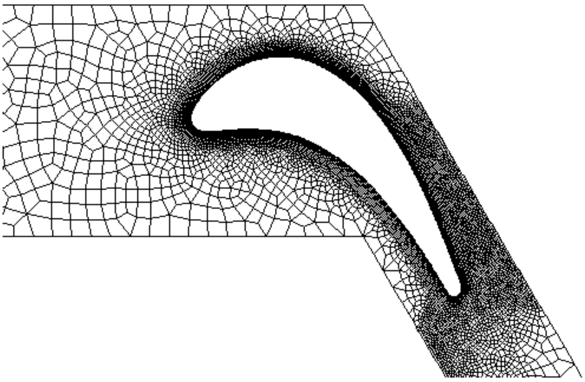

Figure 9.1. VKI-LS59 cascade: detail of the mesh.

9.2 T106c turbine cascade

Nowadays, several efforts are devoted to the improvement of low pressure turbines (LPT) in turbofan engines. One of the goals is the reduction of the weight. This leads to a reduction in the number of blades. The obvious consequence is that the remaining blades have to be more loaded. For this reason, high-lift and ultra-high-lift blades have been developed. When a blade is forced to work in such conditions, in which the turning angle is very high, it is mandatory to optimize the blade's shape in order to reduce fluid dynamics losses, otherwise the benefits obtained by the weight reduction can be obfuscated by a deterioration of the efficiency.

Hourmouziadis [118] reported the working conditions for a LPT for the entire flight

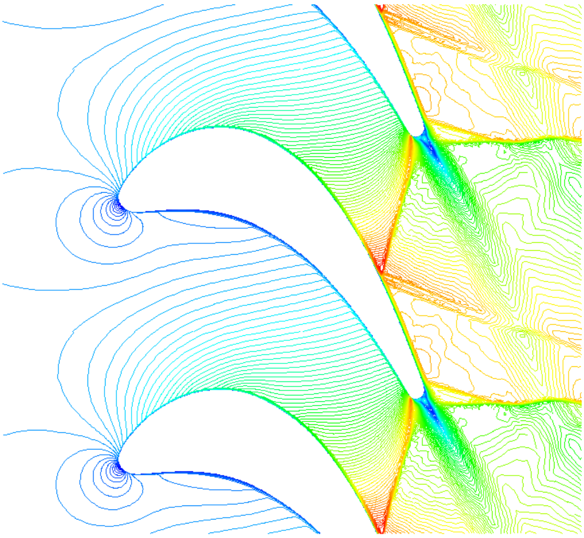

Figure 9.2. VKI-LS59 cascade: Mach field.

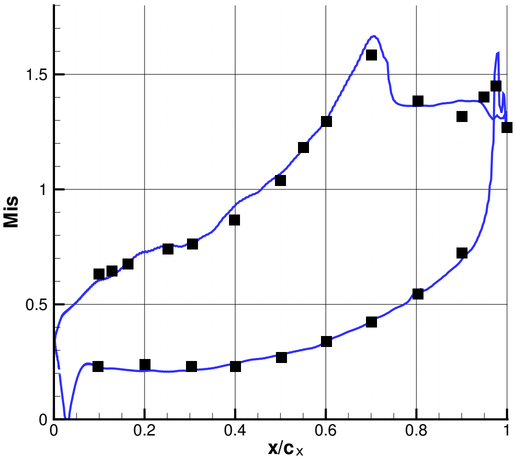

Figure 9.3. VKI-LS59 cascade: wall isentropic Mach distribution.

envelope of a typical commercial aircraft. This data, which are representative of the late 1980s design, show that at cruise condition the Reynolds number for the blades of the LPT is below 200000 while at takeoff is more than 400000. Nowadays, the LPT Reynolds number at cruise condition can be even below 100000.

This very low Reynolds number introduces several problems related to the aerodynamic behavior of the blades. At high Reynolds number the flow is usually attached and so the required turning angle is obtained. On the contrary, at low Reynolds number it is possible to observe separation which reduces the efficiency and the turning angle. The less severe separation, which can occur at intermediate Reynolds numbers, is characterized by the presence of a separation bubble after which the flow reattachs. However, for very low Reynolds number the separation bubble becomes larger and finally an open separation is obtained. This last condition is very detrimental in terms of efficiency.

For these reasons, several efforts have been done in the last years for the prediction of these phenomena. The T106c high-lift cascade is representative of this class of problems. It has been investigated both experimentally and numerically. The testing working conditions are characterized by low Reynolds numbers and several turbulence intensity. The upstream turbulence intensity can significantly affect the separation behavior.

In this work, the T106c cascade was studied by means of the developed discontinuous Galerkin approach applied to the laminar kinetic energy transition model (see Section 2.3.3). Two working conditions were considered, in which the exit isentropic Reynolds number is 185000 and 80000, respectively. The isentropic exit Mach number is 0.65 and the inlet flow angle is 32.7°, for both cases. The inlet turbulence intensity was set as $T_u = 0.8\%$, according to the experimental value which corresponds to the tests without grid in the wind tunnel. The inlet turbulence length scale was set as $l_t/c_x = 2.5 \times 10^{-3}$, where c_x is the axial chord. This value comes from a study on the decay of inlet turbulence [15]. The inlet specific dissipation ω_{in} was then obtained by k_{in} and l_t according to Eq. 2.67.

The experiments have shown that the flow is characterized by a laminar separation and a subsequent transition to turbulence. After the separation the flow can reattach o remain separated, depending on the working Reynolds number.

The simulations were performed by $p = 1$ reconstructions on hybrid meshes. The element basis A was employed. In Figure 9.4 an example of mesh is reported: it is obtained by an O-type structured mesh surrounded by an unstructured quadrilateral mesh. The wall elements have parabolic edges. The grid spacing close to wall is such that the dimensionless wall distance y^+ is less than one on all the surface with the exception of some points near the leading edge.

The extension of the structured zone in the wall normal direction has been chosen by a trial-and-error process, in order to include the shear layer inside the structured mesh. This is due to the fact that the LKE model requires the evaluation of the thickness of the shear layer (vorticity thickness) which can be evaluated more

easily on a structured mesh. However, the tests done in this work have showed that the evaluation of this parameter (which controls the production term in the LKE equation) is quite problematic. Indeed, if the topology of the grid is changed (from O to H or C) or the direction of the grid lines in the structured region is altered then the results can be affected and it is necessary to tune again the calibration constant C_1 which appears in Eq. 2.54. In this work the constant C_1 is set to 0.009 while the original value $C_1 = 0.01$ is suggested in [15].
In Figure 9.11 the computed wall isentropic Mach number distribution is reported and compared with the experimental results from [119]. The simulations are able to correctly predict how the separation region changes when the Reynolds number is reduced. In particular, the extension of the plateau in the distribution on the suction side and the reduction of the maximum Mach number are related to the extension of the separation. The Mach field, the laminar kinetic energy distribution and the turbulent kinetic energy distribution are reported in Figure 9.5, 9.6, 9.7, 9.8 , 9.9, 9.10 for $M_{2s} = 185000$ and $M_{2s} = 80000$, respectively. Again, the pictures show clearly the dramatic variation in the separation region when the Reynolds number is reduced to 80000.

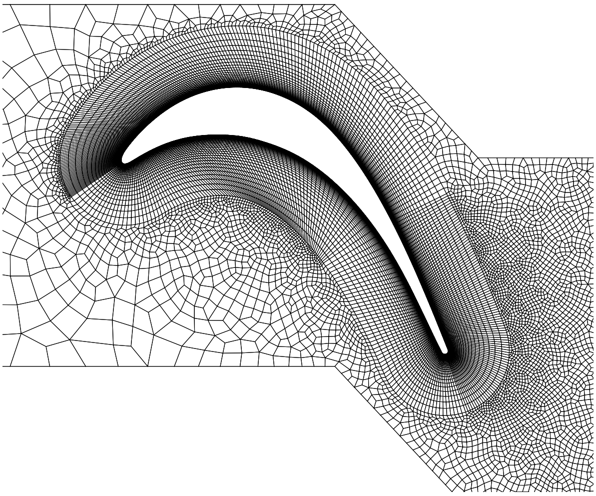

Figure 9.4. T106c cascade: example of hybrid mesh.

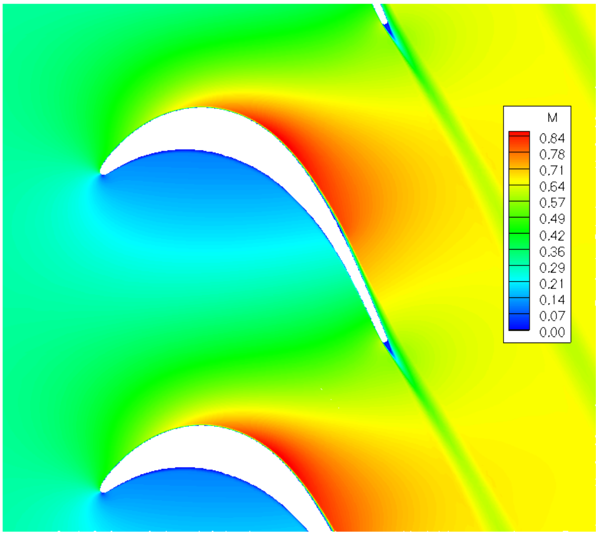

Figure 9.5. T106c cascade: Mach at $Re_{2s} = 185000$, $M_{2s} = 0.65$

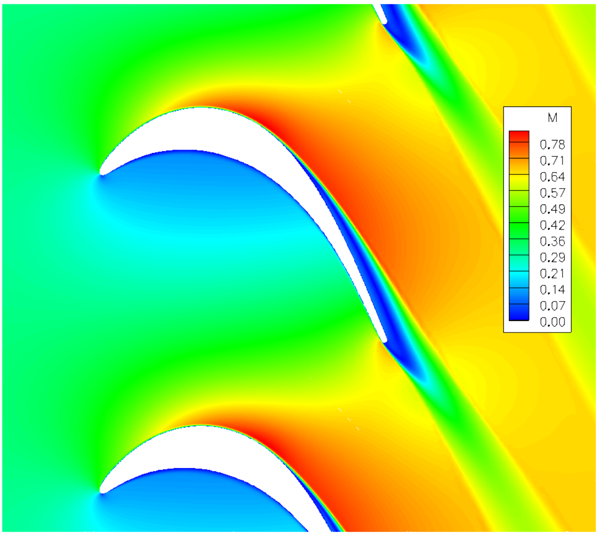

Figure 9.6. T106c cascade: Mach at $Re_{2s} = 80000$, $M_{2s} = 0.65$

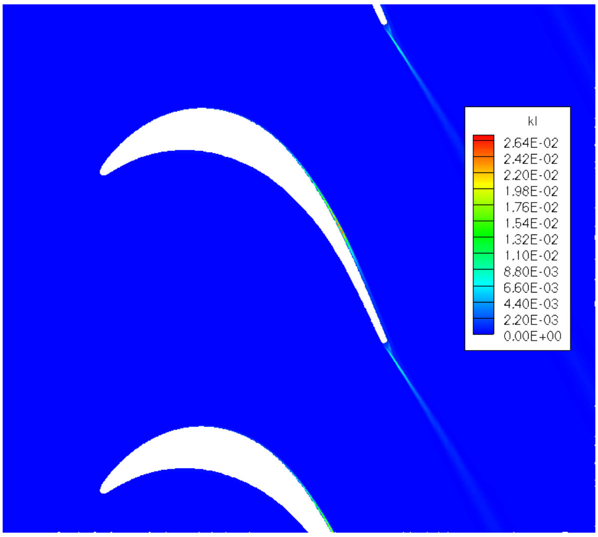

Figure 9.7. T106c cascade: laminar kinetic energy at $Re_{2s} = 185000$, $M_{2s} = 0.65$

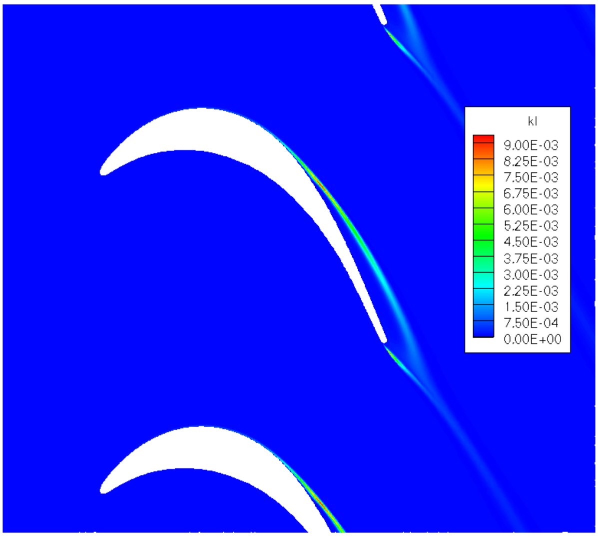

Figure 9.8. T106c cascade: laminar kinetic energy at $Re_{2s} = 80000$, $M_{2s} = 0.65$

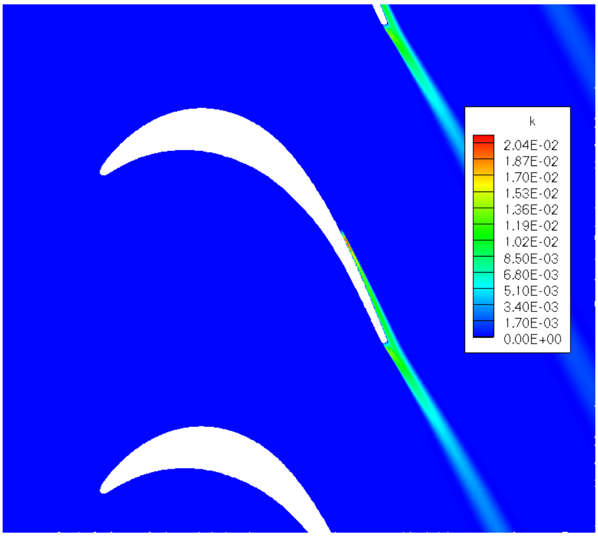

Figure 9.9. T106c cascade: turbulent kinetic energy at $Re_{2s} = 185000$, $M_{2s} = 0.65$

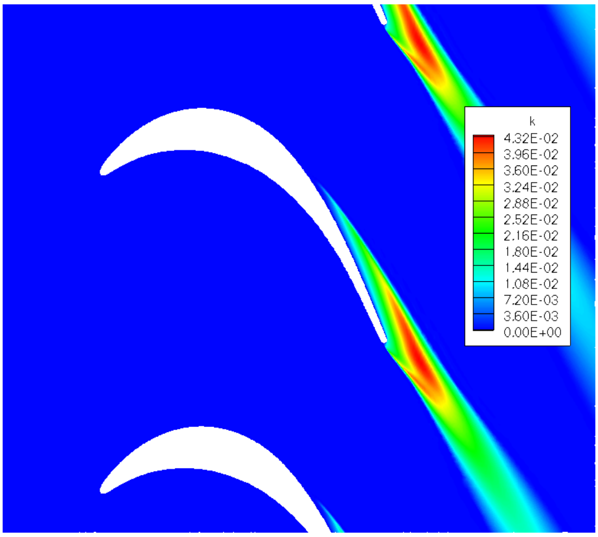

Figure 9.10. T106c cascade: turbulent kinetic energy at $Re_{2s} = 80000$, $M_{2s} = 0.65$

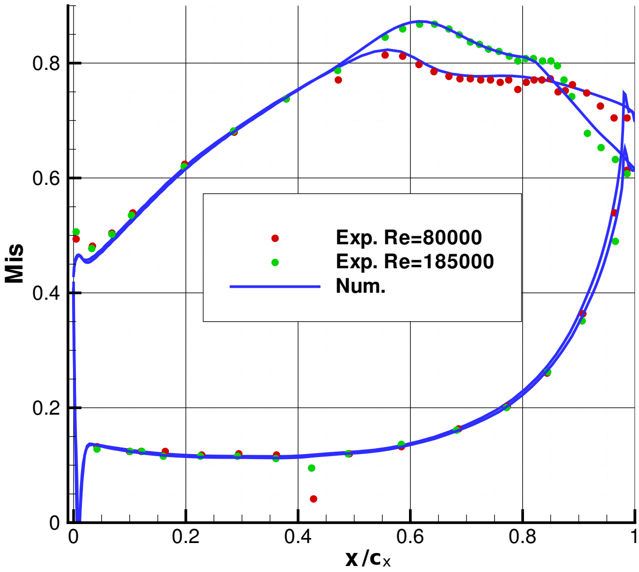

Figure 9.11. T106c cascade: wall isentropic Mach distribution ($M_{2s} = 0.65$).

Chapter 10

Conclusions

This thesis describes the main activities performed during the research project. The reader can found an evolution in the complexity of the reported test cases across the Chapters. Through all the work it is possible to find a mix between numerical problems and physical modeling issues. The numerical issues dominated the first part of the work because initially there was the need to build a solid numerical platform. Several work has been done on the evaluation of the numerical accuracy and on the performance comparison with other groups. These aspects have also driven the development of the two proposed algorithms (Feedback Filtering and Enhanced Stability Recovery) which can show both advantages and drawbacks over other existing methods, depending on the problem.

The developed numerical platform, together with the set of physical models introduced in the second part of the work, can be used to predict the behavior of industrial flows in the aerospace field.

Several aspects can be improved. First of all, the code is parallelized only for shared memory systems. The support for distributed memory systems should be introduced.

As far as the implicit integration scheme is concerned, an analytic evaluation of the jacobian should be introduced also for Navier-Stokes and RANS equations in order to improve performances and make easier the parallelization.

Since aerospace propulsion systems are characterized by compressible flows, further work should be devoted to shock capturing techniques. In particular, the implementation of the artificial viscosity schemes [82] and [90] would be interesting, like the implementation of WENO limiters[93] in the DG framework.

Finally, all the simulations performed in this work are 2D and so the next step is the extension to 3D. This would make it possible to perform LES and DNS simulations, provided that a massive parallelization is introduced.

Appendix A

Adimensionalization

All the equations reported in Chapter 2 and implemented in the code are adimensionalized with respect to the following reference values:

- LENGTH L_{ref}
- TEMPERATURE T_{ref}
- PRESSURE P_{ref}
- DENSITY ρ_{ref}
- SPEED $u_{ref} = \sqrt{RT_{ref}}$
- TIME $t_{ref} = L_{ref}/u_{ref}$
- ENERGY PER MASS UNIT $e_{ref} = u_{ref}^2$
- ENTROPY $S_{ref} = c_v$ (constant volume heat capacity)

When the previous adimensionalization is applied to equations with diffusive terms then the following reference Reynolds number appears in the equations:

$$Re_{ref} = \frac{\rho_{ref} u_{ref} L_{ref}}{\mu_{ref}} \tag{A.1}$$

Also the data which define all the studied test cases are normalized with respect to the previous value.

Appendix B

Normalization of performances

In order to compare the performances of different discretizations it is fundamental to define a set of proper cost functions. The results reported in this work are evaluated by two cost functions: the equivalent length scale and the work units.
The equivalent length scale h is defined as:

$$h = \frac{1}{\sqrt{nDOFs}} \tag{B.1}$$

where $nDOFs$ is the total number of degrees of freedom per equation, which depends on both the number of element and the order of the elements. The square root is employed because 2D problems are considered. In 3D, the cubic root has to be used.
The meaning of this cost function is directly related to the memory requirements of the discretization.

The work units (WUs) are a measure of the total computational time which is normalized in order to make possible a comparison between simulations performed on different machines:

$$WUs = \frac{\Delta t}{\tau_{ref} n_{core}} \tag{B.2}$$

where Δt is the wall clock time required by the simulation, τ_{ref} is the reference time and n_{core} is the number of employed cores. The reference time τ_{ref} is obtained by the TauBench program which can be downloaded from [68], [11] or [69]. TauBench has to be run in serial, since the number of cores is already taken in account by n_{core}.

Appendix C

High-order visualizations

The DG discretization introduces several degrees of freedom per equation in each element. This is in contrast to what is done in finite volume methods, in which only the average values of the conservative variables are known inside each control volume. For this reason, the visualizations of high-order DG solutions require particular care in order to take into account all the information memorized in the elements.

In this work a simple post-processing approach has been developed. In particular, the visualizations are done on finer meshes obtained by splitting the computational meshes. In particular, the larger the number of degrees of freedom in the original element is, the larger the number of subdivisions of the element will be. The values of the solution in the additional nodes are computed thanks to the high-order internal element reconstruction.

In this way, the output file can be read by standard visualization tools developed for finite volumes or finite elements methods.

In the following, an example of these visualization meshes is reported for both quadrilateral and triangular meshes. In order to simplify the implementation, the visualization on triangular elements is performed by handling the triangles as quadrilateral elements with a collapsed edge.

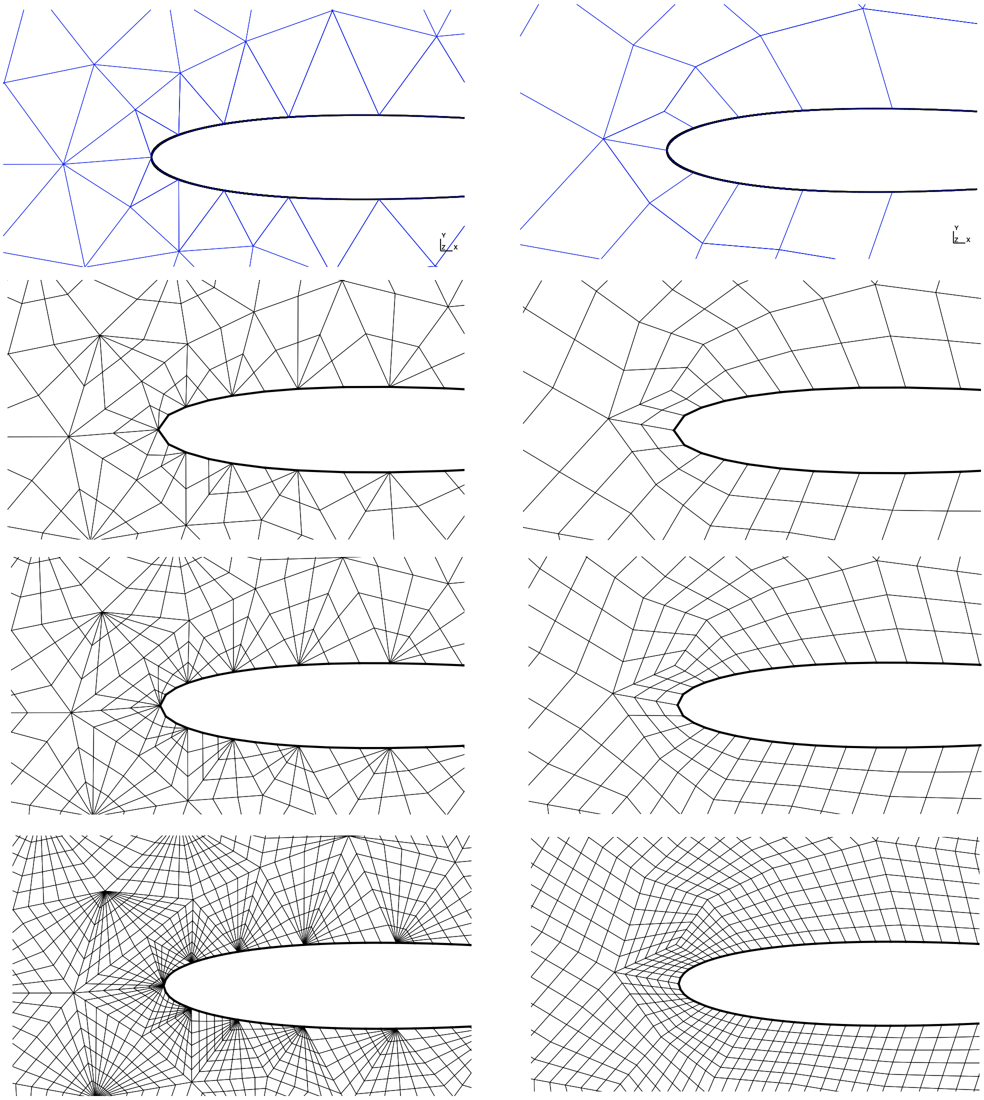

Figure C.1. Curvilinear computational meshes (blue) and visualization meshes (black) with different subdivisions for triangles and quadrilaterals.

Appendix D

Meshes

The meshes used during the research activity were generated in several ways.
An elliptic mesh generator developed by Professor Larocca at Politecnico di Torino was employed for H, C or O structured meshes. This approach makes it possible to obtain grid lines which are normal to the wall.
However, the meshes obtained by this tool are linear. In order to obtain higher-order curvilinear meshes a specific tool has been developed. The tool reads the linear meshes in PLOT3D format and then adds the points required to define the curvilinear elements. Only the elements near the boundary of the domain are curved. The tools requires a parametric representation of the solid wall. A possible alternative approach is based on agglomeration of linear meshes [32].

Unstructured meshes were obtained by Gmsh [120]. Both the Delaunay and the Delaunay for quads algorithms have been successfully employed, for both triangular and quadrilateral meshes.

Some simulations of the flow around airfoils were performed by mixed structured-unstructured meshes. In particular, it is useful to introduce a structured region near the airfoil in order to use high-aspect ratio elements in the boundary layer. Furthermore, the structured nature of the mesh in this zone is useful to simplify the evaluation of the vorticity thickness if the LKE transition model (Section 2.3.3) is chosen. On the other hand, unstructured meshes are very useful for the discretization of the space far from the airfoil because the local size of the elements can be chosen with more freedom.
In order to obtain this kind of meshes, a tool has been developed. The tool reads the geometry of the airfoil and then projects the geometry points far from the wall in order to define an external boundary for the structured zone. The curves generated in this way are used to define an input file for Gmsh, which produce the

final mixed mesh.

Finally, the optimization tool for high-order meshes included in Gmsh is used for some RANS simulations. Indeed, in high Reynolds number RANS simulation the elements in the boundary layer have a very high aspect ratio. For this reason, when the wall edge is curved the element can become singular due to overlapping edges. The optimization tool included in Gmsh extends the deformation of the element to several layers avoiding the overlap. In this way the mesh has curvilinear elements not only at wall but also far from it.

Appendix E

List of symbols

In the following the list of the main symbols used through the text is reported.

Model variables

a = Speed of sound
c = Maximum signal speed
c_f = Wall friction coefficient
c_P = Wall pressure coefficient
E = Total energy (internal+kinematic) per unit volume
F, G = Physical fluxes
k = Turbulent kinetic energy
k_L = Laminar kinetic energy
l_t = Turbulence length scale
u = Generic conservative variable
M_{is} = Isentropic Mach number
M_∞ = Freestream Mach number
P = Pressure
P° = Total pressure
Pr = Prandtl number
Pr_t = Turbulent Prandtl number
Q = Source term in the heat equation
q = Velocity magnitude
q_x, q_y = Cartesian components of heat flux
Re_{ref} = Reference Reynolds number obtained by adimensionalization
Re_∞ = Freestream Reynolds number
S = Entropy
S_{ij} = Mean strain rate tensor
T = Temperature

T° = Total temperature
T_u = Turbulence level
u, v = Cartesian component of velocity
u_τ = Wall friction velocity
α_∞ = Freestream angle of attack
γ = Specific heat ratio
μ = Dinamyc viscosity
ν = Kinematic viscosity
λ = Heat conductivity
τ_w = Wall shear stress
ω = Specific dissipation rate
ρ = Density
τ_{ij} = Laminar stress tensor
$\bar{\tau}_{ij}$ = Turbulent stress tensor
$\hat{\tau}_{ij}$ = Total (laminar+turbulent) stress tensor
$c_{b1}, \sigma, c_{b2}, \kappa, c_{w1}, c_{w2}, c_{w3}, c_{v1}, c_{t1}, c_{t2}, c_{t3}, c_{t4}, r_{lim}, c_{v2}, c_{v3}$ = Constants for the Spalart-Allmaras model
$\sigma_k, \sigma_\omega, \beta_0^*, \beta_0, \alpha_0, \alpha_0^*, R_k = 6, R_\omega$ = Constans for the low-Re Wilcox $k - \omega$ model
C_1, C_2, C_3, C_4 = Constants for the laminar kinetic energy model

Geometry

d = Distance from the closest solid wall
L_1, L_2, L_3 = Area coordinates for triangle
n_x, n_y = Cartesian components of normal unit vector
x, y = Cartesian coordinates in physical space
ξ, η = Cartesian coordinates on the reference quadrilateral and local coordinates for interface reference frame

Discretization

$\left[\hat{\mathbf{B}}\right]$ = Boundary matrix for Dirichlet BC's with the ESR method
CFL = Stability coefficient for time integration
$[\mathbf{D}]$ = Element derivative matrix
$[\mathbf{D}_r]$ = Recovery derivative matrix
F_h, G_h = Physical fluxes computed as a function of u_h
$\hat{F}_h, \hat{G}_h$ = Interface numerical fluxes
g = User defined constant in the feedback filtering procedure
H = Element characteristic size
h = Equivalent length scale which depends on $nDOFs$
$[\mathbb{J}]$ = Jacobian matrix for implicit scheme
$[\mathbf{M}]$ = Element mass matrix
$[\mathbb{M}]$ = Global mass matrix

N = Number of degrees of freedom in an element
N_F = Number of faces in an element
$nDOFs$ = Total number of degrees of freedom per equation
p = Reconstruction order in space
$\mathbf{R}$ = Element residual of the discretization
$\mathbb{R}$ = Global residual of the discretization
$[\mathbf{R}]$ = Recovery matrix
$[\tilde{\mathbf{R}}]$ = Interface recovery matrix
r = Filter order in the feedback fitlering procedure
$\mathbf{r}_\sigma, \mathbf{R}$ = Local and global lifting operators for BR2 method
S_{Σ_e} = Smoothness sensor for element e
SI = Smoothness indicator in the feedback filtering procedure
SR = Smoothness requirement in the feedback filtering procedure
s_e = Element sensor for adaptivity
u_h = Element solution for the conservative variable u
u^r = Recovery solution for the conservative variable u
$\mathbf{u}$ = Vector of degrees of freedom
$\{\mathbf{W}\}$ = Boundary vector for Dirichlet BC's with the ESR method
$\mathbf{z}_h$ = Modified gradient for BR2 method
α_{iter} = Filter intensity in the feedback filtering procedure
η_0 = Penalty parameter for BR2 method
η_e = Residual-based error indicator
Φ = Element vasis function
Ψ = Recovery vasis function

Bibliography

[1] Ferrero A., Larocca F. Feedback filtering in discontinuous Galerkin methods for Euler equations, *Progress in Computational Fluid Dynamics*, In press.

[2] Ferrero A., Larocca F., Puppo G. A robust and adaptive recovery-based discontinuous Galerkin method for the numerical solution of convection-diffusion equations, *Int. J. Numer. Meth. Fluids*, 77:63-91, 2015.

[3] van Leer B., Nomura S. Discontinuous Galerkin for Diffusion, *AIAA paper 2005-5108*, 2005.

[4] Bassi F., Rebay S., Mariotti G., Pedinotti S., Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere R, Dibelius G, editors. *2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, Technologisch Institut, Antwerpen, Belgium, 1997, 99-108.

[5] Löhner R. Improved error and work estimates for high-order elements, *Int. J. Numer. Meth. Fluids*, 72:1207-1218, 2013

[6] Wang Z.J., Fidkowski K., Abgrall R., Bassi F., Caraeni D., Cary A., Deconinck H., Hartmann R., Hillewaert K., Huynh HT., Kroll N., May G., Persson P.O., van Leer B., Visbal M. High-order CFD methods: current status and perspective *Int. J. Numer. Meth. Fluids*, 72: 811-845, 2013.

[7] Liu X.-D., Osher S., Chan T. Weighted essentially nonoscillatory scheme, *J. Compu. Phys.*, 115:200-212, 1994

[8] Levy D., Puppo G., Russo G. Compact Central WENO Schemes for Multidimensional Conservation Laws , *SIAM J. Sci. Comp.*, 22:656-672, 2000.

[9] Cada M., Torrilhon M. Compact third-order limiter functins for finite volume methods, *J. Comput. Phys.*, 228:4118-4145, 2009.

[10] Shu C.W., TVB uniformly high-order schemes for conservation laws, *Math. Comput.*, 49:105-121, 1987.

[11] Second International Workshop on High-Order CFD Methods, Cologne, May 27-28 2013: http://www.dlr.de/as/hiocfd (Accessed January 2015)

[12] Wilcox D.C. *Turbulence Modeling for CFD*, 1th edition, DCW Industries, Inc., La Canada CA, 1993.

[13] Wilcox D.C. *Turbulence Modeling for CFD*, 2nd edition, DCW Industries, Inc., La Canada CA, 1998.

[14] Wilcox D.C. *Turbulence Modeling for CFD*, 3rd edition, DCW Industries, Inc., La Canada CA, 2006.

[15] Pacciani R., Marconcini M., Arnone A., Bertini F. An assessment of the laminar kinetic energy concept for the prediction of high-lift, low-Reynolds number cascade flows, *Proc. IMechE Part A: J. of Power and Energy*, Vol. 225 pp. 995-1003, 2011.

[16] Spalart P.R., Allmaras S.R. A One-Equation Turbulence Model for Aerodynamic Flows, *Recherche Aerospatiale*, No. 1, pp. 5-21, 1994.

[17] NASA Turbulence Modeling Resource, http://turbmodels.larc.nasa.gov/spalart.html (Accessed January 2015)

[18] Allmaras S.R., Johnson F.T.,and Spalart P.R. Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model, Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, 9-13 July 2012.

[19] Rumsey C.L., Spalart P.R. Turbulence Model Behavior in Low Reynolds Number Regions of Aerodynamic Flowfields, 38th AIAA Fluid Dynamics Conference and Exhibit, June 23 - 26, 2008, Seattle, WA, *AIAA Paper 2008-4403*

[20] Bassi F., Crivellini A., Rebay S., Savini M. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-omega turbulence model equations, *Computers & Fluids*, 34: 507-540, 2005.

[21] Langtry R.B., Menter F.R. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes, *AIAA Journal*, 47(12), pp. 2894-2906, 2009.

[22] Babajee J. *Detailed numerical characterization of the separation-induced transition, in- cluding bursting, in a low-pressure turbine environment*, Ecole Centrale de Lyon; Institut von Karman de dynamique des uides (Rhode-Saint-Genese, Belgique), PhD Thesis, 2013.

[23] Mayle R.E., Schulz A., The Path to Predicting Bypass Transition, *ASME J. Turbomach.*, 119(3), pp. 405-411, 1997.

[24] Walters D.K., Leylek J.H., A New Model for Boundary Layer Transition Using a Single-Point RANS Approach, *ASME J. Turbomach.*, 126(1), pp. 193-202, 2004,

[25] Lardeau S., Leschziner M.A., Li N., Modelling Bypass Transition With Low-Reynolds-Number Non-Linear Eddy-Viscosity Closure, *Flow,Turbul. Combust.*, 73, pp. 49-76, 2004.

[26] Pacciani R., Marconcini M., Fadai-Ghotbi A., Lardeau S., Leschziner M.A. Calculation of High-Lift Cascades in Low Pressure Turbine Conditions Using a Three-Equation Model, *Journal of Turbomachinery*, Vol. 133, 2011.

[27] Menter F.R. Two-equation eddy-viscosity turbulence models for engineering applications, *AIAA Journal*, 32(8):1598-605, 1994.

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

[29] Bassi F., Botti L., Colombo N., Ghidoni A., 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.

[30] Hill T.T. and Reed W.H. Triangular mesh methods for neutron transport equation, *Tech. Rep. LA-UR-73-479*, Los Alamos Scientific Laboratory, 1973.

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

[32] Bassi F., Botti L., Colombo A., Di Pietro D.A., Tesini P. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations, *Journal of Computational Physics* 2012; **231** : 45-65.

[33] Solin P., Segeth K., Dolezel I. *Higher-Order Finite Element Methods*, Chapman and Hall/CRC Press, 2003.

[34] Oñate E. *Structural Analysis with the Finite Element Method: Linear Statics, Volume 1: Basis and Solids*. Springer, 2009.

[35] Osher S., Solomon F. Upwind schemes for hyperbolic systems of conservation laws, *Mathematics of Computation* 1982, 38:339.

[36] Pandolfi M. A contribution to the numerical prediction of unsteady flows, *AIAA Journal*, 22: 602-610, 1984

[37] Liou MS. A sequel to AUSM: AUSM+, *Journal of Computational Physics*, Vol. 129, pp. 364, (1996).

[38] Liou MS. On a new class of flux splitting, *Lecture Notes in Physics*, Springer-Verlag, 414, (1993).

[39] Roe P.L. Approximate solvers, parameter vectors and difference schemes, J. Comput. Phys., 43, 357, (1981).

[40] Harten A., Lax P.D., van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Reviews* 25 (1): 35-61, (1983).

[41] Nishikawa H., Kitamura K. Very Simple, Carbuncle-Free, Boundary-Layer Resolving, Rotated-Hybrid Riemann Solvers, *Journal of Computational Physics*, 227, pp. 2560-2581, (2008).

[42] Pandolfi M. and D'Ambrosio D. Numerical instabilities in upwind methods: analysis and cures for the "Carbuncle Phenomenon". In: *Journal of Computational Physics*, vol. 166, pp. 271-301, (2001).

[43] Lax P.D. Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Comm. Pure Appl. Math.*, 7, 159-193, (1954).

[44] Leveque R.J., *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, (2002).

[45] Wheatley V., Kumar H., Huguenot P., On the role of Riemann solvers in

Discontinuous Galerkin methods for magnetohydrodynamics, Journal of Computational Physics, Vol. 229, pp. 660-680, (2010).

[46] Arnold D.N., Brezzi F., Cockburn B., Marini D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39, 5: 1749-1779, (2002).

[47] Lions J.L. Problemes aux limites non homogenes a donees irregulieres: une methode d'approximation, *Numerical Analysis of Partial Differential Equations (C.I.M.E. 2 Ciclo, Ispra, 1967).* Edizioni Cremonese: Rome, 1968; 283-292.

[48] Nitsche J.A. Uber ein Variationsprinzip zur Losung Dirichlet-Problemen bei Verwendung von Teilraumen, die keinen Randbedingungen unteworfen sind. *Abh. Math. Sem. Univ. Hamburg* 36: 9-5, (1971).

[49] Babuska I., Zlamal M. Nonconforming elements in the finite element method with penalty, *SIAM Journal on Numerical Analysis*, 10: 863-875, (1973).

[50] Bassi F., Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *Journal of Computational Physics*, 131: 267-279, (1997).

[51] Cockburn B., Shu C.W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM Journal on Numerical Analysis*, 35: 2440-2463, (1998).

[52] Baumann C.E., Oden J.T. A discontinuous hp finite element method for convection-diffusion problems. *Comp. Meth. Appl. Mech. Eng.*, 175: 311-341, (1991).

[53] Gassner G., Lorcher F., Munz C.D., A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes, *Journal of Computational Physics*, 224: 1049-1063, (2007).

[54] Dumbser M., Zanotti O. Very High Order PNPM Schemes on Unstructured Meshes for the Resistive Relativistic MHD Equations, *Journal of Computational Physics*, 228: 6991-7006, (2009).

[55] Luo H., Luo L., Nourgaliev R., Mousseau V.A. A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations on Arbitrary Grids, *Journal of Computational Physics* 229: 6961-6978, (2010).

[56] Dumbser M., Balsara D.S., Toro E.F., Munz C.D.. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227: 8209-8253, (2008).

[57] Huynh H.T. A reconstruction approach to high-order schemes including Discontinuous Galerkin for diffusion, *AIAA Paper 2009-0403*, (2009).

[58] Borrel M., Ryan J. The Elastoplast Discontinuous Galerkin (EDG) method for the Navier-Stokes equations, *Journal of Computational Physics*, 231: 1-22, (2012).

[59] French D.A., Galbraithy M.C., Osorio M. Error Analysis of a Modified Discontinuous Galerkin Recovery Scheme for Diffusion Problems, *AIAA Paper*

2010-1071, (2010).

[60] Lo M., van Leer B. Analysis and Implementation of Recovery-Based Discontinuous Galerkin for Diffusion, *AIAA Paper 2009-3786*, (2009).

[61] van Leer B., Nomura S., van Raalte M. A discontinuous Galerkin Method for Diffusion Based on Recovery, *AIAA Paper 2007-4083*, (2007).

[62] van Leer B., Lo M. Unification of Discontinuous Galerkin Methods for Advection and Diffusion, *AIAA Paper 2009-400*, (2009).

[63] Nourgaliev R., Theofanous T., Park H., Mousseau V., Knoll D. Direct Numerical simulation of interfacial flows, *AIAA Paper 2008-1453*, (2008).

[64] van Leer B., Lo M., Gitik R., Nomura S. *Adaptive High-order Methods in Computational Fluid Dynamics.* World Scientific: Singapore; 185-201.

[65] Eaton J.W., Bateman D., Hauberg S. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations.* CreateSpace Independent Publishing Platform, (2009).

[66] Riviére B. *Discontinuous Galerkin methods for solving elliptic and parabolic equations : theory and implementation.* SIAM: Philadelphia, 66, (2008).

[67] Bazilevs Y., Hughes T.J.R. Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Computers & Fluids*, 36: 12-26, (2007).

[68] First International Workshop on High-Order CFD Methods, Nashville, January 7-8, 2012: http://zjwang.com/hiocfd.html (Accessed January 2015)

[69] Third International Workshop on High-Order CFD Methods, Orlando, January 3-4, 2015: https://www.grc.nasa.gov/hiocfd/ (Accessed January 2015)

[70] Quarteroni A., Sacco R., Salieri F. *Matematica numerica*, Springer, Milan, (2008).

[71] Schwartzkopff T., Munz C.D., Toro E.F. ADER: High-order approach for linear hyperbolic systems in 2D. *Journal of Scientific Computing.* Vol. 17, pages 231-240, (2002).

[72] Nigro A., Ghidoni A., Rebay S., Bassi F. Modified extended BDF scheme for the discontinuous Galerkin solution of unsteady compressible flows, *Int. J. Numer. Meth. Fluids*, 76:549-574, (2014).

[73] Cockburn B. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, An Introduction to the Discontinuous Galerkin spatial discretization Galerkin Method for Convection-dominated Problems*, Lecture Notes in Mathematics, Springer, pp. 151-268, (1998).

[74] Gottlieb S., Shu C-W. Total variation diminishing Runge-Kutta schemes, *Math. Comput.* 67 (221), 73-85, (1998).

[75] Kaw A., Kalu E., Numerical Methods with Applications (1st ed.), (2008)

[76] Carpenter M.H., Kennedy C. Fourth-order 2N-storage Runge-Kutta schemes, *NASA Report TM 109112*, NASA Langley Research Center, (1994).

[77] Spiteri R., Ruuth S., A new class of optimal high-order strong stability preserving time discretization methods, *SIAM Journal on Numerical Analysis*, Vol. 40, pp. 469-491, (2002).

[78] CSR sparse matrix format. http://netlib.org./linalg/html_templates/node91.html (Accessed December 2014)

[79] Brown P.N., Saad Y. Hybrid Krylov methods for nonlinear systems od equations, *SIAM J. Sci. and Stat. Comput.*, 11(3):450-481, (1990).

[80] Saad Y., Schultz M.H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 7:856-869, (1986).

[81] Lukarski D., Trost N. PARALUTION Project, http://www.paralution.com/ (Accessed December 2014)

[82] Bassi F., Botti L., Colombo A., Crivellini A., Franchina N., Ghidoni A., Rebay S. (2010) Very high-order accurate discontinuous Galerkin computation of transonic turbulent flows on aeronautical configurations, in A*DIGMA - 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. Sorensen, Eds., Vol. 113 of Notes of Numerical Fluid Mechanics and Multidisciplinary Design, Springer Berlin, pp. 25-38.

[83] Mulder W.A., van Leer B. Experiments with implicit upwind methods for the Euler equations, *Journal of Computational Physics* Volume 59, Issue 2, Pages 232-246, (1985).

[84] Bücker H.M., Pollul B., Rasch A. On CFL evolution strategies for implicit upwind methods in linearized Euler equations, *International Journal for Numerical Methods in Fluids*, Volume 59, Issue 1, pages 1-18, (2009).

[85] Taube A., 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. Sorensen, Eds., Vol. 113 of Notes of Numerical Fluid Mechanics and Multidisciplinary Design, Springer Berlin, pp. 195-207, (2010).

[86] Persson P.-O., Peraire J. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

[87] Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, Berlin, (2009).

[88] Shapiro, A. *The Dynamics and Thermodynamics of Compressible Fluid Flow*, The Ronald Press Company, New York, (1953).

[89] Nguyen N.C., Peraire J. An Adaptive Shock-Capturing HDG Method for Compressible Flows, *AIAA Paper* 2011-3060, (2011).

[90] Hartmann R. Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow, *Int. J. Numer. Meth. Fluids* 72, 8, pages 883-894, (2013).

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

[92] Biswas R., Devine K., Flaherty J.E. Parallel adaptive finite element methods for conservation laws, *Applied Numerical Mathematics* 14: 255-284, (1994).

[93] Li W., Ren Y-X. The multi-dimensional limiters for discontinuous Galerkin method on unstructure grids, *Seventh International Conference on Computational Fluid Dynamics*, Hawaii, July 9-13, 2012.

[94] Hesthaven J.S., Kirby R.M. Filtering in Legendre spectral methods, *Math. Comput.* 77(263):1425-1452, (2008).

[95] Allaneau Y., Jameson A. Connections between the filtered discontinuous Galerkin method and the flux reconstruction approach to high-order discretizations, *Comput. Methods Appl. Mech. Engrg.* 200: 3628-3636, (2011).

[96] Gottlieb D., Hesthaven J.S. Spectral Methods for Hyperbolic Problems, *J. Comput. Appl. Math.*, 128, 83-131 (2001).

[97] Hesthaven J.S., Warburton T. *Nodal Discontinuous Galerkin Methods*, Texts in Applied Mathematics, Springer (2008).

[98] Boyd J.P. The Erfc-Log Filter and the Asymptotics of the Euler and Vandeven Sequence Accelerations, *Houston Journal of Mathematics*, (1996).

[99] Ueckermann M.P., Lermusiaux P.F.J. High-order schemes for 2D unsteady biogeochemical ocean models, *Ocean Dynamics*, 60, 6, pp 1415-1445, (2010).

[100] Kanevsky A., Carpenter M.H., Hesthaven J.S. Idempotent filtering in spectral and spectral element methods, *J. Comput. Phys.*, 220: 41-58, (2006).

[101] Chiocchia G. Exact solutions to transonic and supersonic flows, *Technical Report AR-211*, AGARD, (1985).

[102] Barter G.E. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order Discontinuous Galerkin Finite Element Method*, PhD Thesis, Massachusetts Institute of Technology, Massachusetts, (2008).

[103] Vassberg J.C., Jameson A. In Pursuit of Grid Convergence for Two-Dimensional Euler Solutions, *Journal of Aircraft*, Vol. 47, pp. 1152-1166, (2010).

[104] Kroll N., Bieler H., Deconinck H., Couaillier V., van der Ven H., Sorensen K. ADIGMA - a European initiative on the development of adaptive higher-order variational methods for aerospace applications. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 113. Springer, 2010.

[105] Hartmann R., Held J., Leicht T. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations, *J. Comput. Phys.*, Vol. 230, No. 11, pp. 4268-4284, (2011).

[106] Leicht T., Hartmann R. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations, *Journal of Computational Physics* 229: 7344-7360, (2010).

[107] Norberg C. Fluctuating lift on a circular cylinder: review and new measurements, *Journal of Fluids and Structures*, 17: 57-96, (2003).

[108] Tritton DJ. Experiments on the flow past a circular cylinder at low reynolds number, *Journal of Fluid Mechanics*, 6: 547-567, (1959).

[109] von Wieselsberger C. Neuere Festellungen Ã¼ber die Gesetze des FlÃ¼ssigkeits un Luftwiderstands, *Phys. Z.* 22: 321-328, (1921).

[110] Williamson C.H.K. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low reynolds number, *Journal of Fluid Mechanics*, 206 : 579-627, (1989).

[111] Bassi F., Colombo A., Franchina N., Ghidoni A., Rebay S. High-order accurate p-multigrid discontinuous Galerkin solution of the RANS and $k - \omega$ turbulence model equations, *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, J.C.F. Pereira and A. Sequeira (Eds), Lisbon, Portugal, 14-17 June 2010.

[112] Luo H., Baum J.D., Löhner R. A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, *Journal of Computational Physics* 211(2):767 (2006)

[113] Fidkowski K.J., Oliver T.A., Lu J., Darmofal D.L. p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, *Journal of Computational Physics*, 207: 92-113, (2005).

[114] Masatsuka K. Mesh generator for Ringleb flow, http://www.cfdbooks.com (Accessed January 2015).

[115] Wang Z.J., Liu Y. Extension of the spectral volume method to high-order boundary representation, *J. Comput. Phys.* 211: 154-178, (2006).

[116] Sieverding C.H. Experimantal data on two transonic turbine blade sections anc comparisons with various theoretical methods, *VKI Report No. LS59*, von Karman Institute, Belgium, (1973).

[117] Kiock R., Lethaus F., Baines N.C., Sieverding C.F. The transonic flow through a plane turbine cascade as measured in four European wind tunnels, *ASME J. Eng. Gas Turbines Power*, 108, 277-285, (1986).

[118] Hourmouziadis J. *Aerodynamic Design of Low Pressure Turbines*, AGARD Lecture Series, 167, (1989).

[119] Michalek J., Monaldi M., Arts T. Aerodynamic performance of a very high lift low pressure turbine airfoil (T106C) at low Reynolds number and high Mach numer with effect of free stream turbulence intensity, *ASME Paper GT2010-22884*, Glasgow, UK, 14-18 June 2010.

[120] Geuzaine C., Remacle J.F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79: 1309-1331, (2009).