

UNIVERSITÀ DEGLI STUDI DELLA CALABRIA

---

Dipartimento di Ingegneria Meccanica Energetica e  
Gestionale

Tesi di Dottorato di Ricerca  
in Ingegneria Meccanica

*Development of a High-Order  
Discontinuous Galerkin Solver for  
Internal Combustion Engine Flows*

Ing. Vanessa Covello



Supervisore

Prof. Gaetano Florio



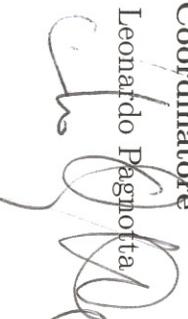
Co-Supervisore

Ing. Carmine De Battolo



Coordinatore

Prof. Leonardo Pagnotta



Novembre 2014

---

DOTTORATO DI RICERCA XXVII Ciclo

SSD ING-IND/08 MACCHINE A FLUIDO

# Development of a High-Order Discontinuous Galerkin Solver for Internal Combustion Engine Flows

by

Eng. Vanessa Covello

A thesis submitted in conformity with the requirements

for the degree of Doctor of Philosophy

Department of Mechanical Energetics,

and Management Engineering,

November 2014.

## Abstract

The aim of this work is to contribute to the development of an unstructured flow solver able to match the increasing demand of the automotive industrial sector to advance CFD-aided design and analysis procedure. The method here presented is designed to ensure high-order of accuracy even in complex geometries using both explicit and implicit schemes for the temporal discretization of the compressible Reynolds Averaged Navier-Stokes (RANS)  $k$ - $\omega$  equations. The algorithm is based on the Discontinuous Galerkin (DG) finite element method, one of the most promising high-order methods, that combines excellent dispersion and dissipation properties with high geometrical flexibility. The DG solver is based on different multi-stage explicit or many implicit or semi-implicit schemes for achieving high order accuracy in time. Here we focus on an implicit multi-stage multi-step method, known in the literature as Two Implicit Advanced Step-point (TIAS) method, analyzing the performance of the sixth-order accurate TIAS scheme for long time simulations of stiff and non stiff unsteady problems.

The second objective of this work is to demonstrate the applicability and reliability of optimization algorithms to control spurious numerical oscillations in simulation of transonic flows. The proposed optimization strategy relies on the gradient based optimization approach employing an Automatic Differen-

tiation (AD) tool for the evaluation of the sensitivities. The optimization process acts directly on the shock capturing technique, seeking for the optimal values of the shock capturing parameters.

The performance of the solver is demonstrated by solving several test-cases of direct relevance in the context of automotive and aerodynamic applications. The comparison between experimental/analytical and numerical results allowed the validation and/or revision of physical and numerical models implemented in the code. Finally, we remark that this work is the starting point of a larger investigation that aims to deal with ICE flow conditions that are poorly predicted by RANS approaches, such as flow separation and reattachment in a highly three-dimensional configuration, by using time-accurate integration of the DG space-discretized ILES and hybrid RANS-LES models.

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	Background . . . . .	10
1.2.1	transonic flows . . . . .	10
1.2.2	turbulent flows . . . . .	12
1.2.3	Optimization method for computational fluid dynamics	14
1.2.4	High order methods . . . . .	16
1.2.5	Goals of research: a high order flow solver for ICE applications based on DG method . . . . .	18
1.3	Outline of Thesis . . . . .	21
<b>2</b>	<b>Physical models</b>	<b>22</b>
2.1	RANS and $k-\omega$ equations . . . . .	23
2.1.1	Three dimensional RANS and $k-\omega$ equations in cylindrical coordinate system . . . . .	23
2.1.2	Two dimensional RANS and $k-\omega$ equations for axisymmetric flows . . . . .	26
2.2	Euler equation . . . . .	27
2.2.1	Two dimensional Euler equation . . . . .	27
2.3	Non-dimensionalization . . . . .	28
<b>3</b>	<b>Development of a high order DG flow solver for ICE applications</b>	<b>30</b>

3.1	Space discretization . . . . .	32
3.1.1	The DG discretization of RANS $k - \omega$ equations . . . . .	32
3.1.2	The DG discretization of the Euler equations . . . . .	34
3.2	Numerical Fluxes . . . . .	34
3.2.1	Godunov flux . . . . .	35
3.2.2	Roe flux . . . . .	35
3.2.3	van Leer/Hanel flux . . . . .	36
3.3	Boundary Conditions . . . . .	37
3.4	Shock capturing . . . . .	38
3.5	Time integration . . . . .	40
3.5.1	TIAS Scheme . . . . .	41
3.5.2	SSPRK Scheme . . . . .	43
3.5.3	Implicit Backward Euler Scheme . . . . .	45
3.6	Gradient based design optimization . . . . .	46
3.6.1	Overview of the optimization strategy . . . . .	46
3.6.2	Evaluation of the Gradient . . . . .	49
3.7	Optimization of shock capturing parameters . . . . .	51
<b>4</b>	<b>Results</b>	<b>54</b>
4.1	Internal turbulent flows . . . . .	55
4.1.1	Axisymmetric sudden expansion . . . . .	55
4.1.2	Axisymmetric sudden expansion with a valve . . . . .	61
4.2	Convection of an isentropic vortex by uniform flow . . . . .	70
4.2.1	Description of the test cases . . . . .	70
4.2.2	Fast vortex . . . . .	72
4.3	Shocked flows . . . . .	75
4.3.1	Two-dimensional supersonic wedge . . . . .	75
	<b>Conclusions</b>	<b>90</b>
	<i>Lists of figures</i> . . . . .	92

*Lists of tables* . . . . . 96

# Chapter 1

## Introduction

### 1.1 Motivation

Nowadays the design and research activities in the field of internal combustion engines are currently mainly focused on achieving high level of environmental compatibility, both in terms of reduced air pollutant emissions and energy consumptions. In this context the considerable advances in algorithms development and the huge increase of the computer power have made Computational Fluid Dynamics (CFD) a key discipline for industrial growth during the last two decades.

The flow fields inside an Internal Combustion Engine (ICE) system greatly affect the performance and the level of exhaust emissions, then a more accurate prediction of their complex physics phenomena could provide considerable benefits in order to reduce time and costs of the industrial production cycles. Currently, the prediction of complex turbulent flows by standard industrial codes is mainly based on the numerical solution of Reynolds Averaged Navier Stokes Equations (RANS) by means of formally second-order accurate finite volume schemes, due to their robustness and their favourable computational cost-accuracy ratio [1–5]. On the other hand the numerical accuracy provided by these low-order schemes is often inadequate to meet the increasing demand of advanced industrial sectors to improve CFD-aided design and

analysis procedures. Low-order schemes, in fact, fail to properly reproduce the fluid dynamic behaviour of complex turbulent flows, especially in the presence of non-equilibrium phenomena, high streamlines curvature or strong three-dimensional effects. Moreover, for complex applications, very fine meshes are required to obtain accurate solutions with second-order methods, leading to enormous computing times. Conversely, even on coarse meshes, high-order discretization methods allow the computation of accurate solutions with a significant reduction of the computational costs.

Several high-order methods have been emerging recently as powerful tools to go beyond the standard accuracy of finite volume discretizations and different types of high-order methods have been developed in the CFD community to deal with a wide range of problems [6–13]. One of the methods that is gaining more interest in CFD in the last years due to its many attractive features is the Discontinuous Galerkin (DG) finite element method [14–16].

The DG method is an innovative strategy that combines two key ideas which are at the basis of the finite volume (FVM) and finite element methods (FEM), approximating the solution in each element by piecewise polynomial functions with no global continuity requirement at element interfaces. Like continuous finite element methods, the DG method can increase the accuracy by raising the degree of polynomial approximation inside each element, whereas the discontinuous approximation at element interfaces allows the method to employ upwind discretizations of interface fluxes, like in high-resolution finite volume methods. The lack of global continuity allows the treatment of each element as a separate entity that communicates with the adjacent elements only through the numerical fluxes. This feature gives a good capability of parallelization. Furthermore, its compact formulation can be applied close to the boundary without any special treatment, thus increasing the robustness and the accuracy of every boundary conditions. Finally, the dispersion and dissipative properties of the method are excellent.

These latter aspects (boundary treatment, dispersion and dissipation prop-

erties) are crucial in order to deal with turbulent flows in complex geometries typical of automotive applications. Often ICE flows are dominated by large scale turbulent structures that occur at low to moderate Reynolds numbers or due to non-aerodynamic geometry conditions. For these flows the standard Reynolds-Averaged Navier-Stokes (RANS) approach is not able to predict time-averaged quantities accurately, and unsteady approaches are required to predict at least part of the unsteady turbulent structures directly. In order of complexity from most to least complex we distinguish Direct Numerical Simulation (DNS) (no turbulence modeling at all), the Large Eddy Simulation (LES) that is based on a SubGrid Scale (SGS) model for the non-resolved turbulent scales, the Implicit Large Eddy Simulation (ILES), where the spatial discretization itself acts like a SGS model, and hybrid RANS-LES (RANS near the solid body, LES elsewhere). All of these approaches require the coupling of high space accuracy with high temporal accuracy to convect all resolved turbulent structures at the right speed with minimal dissipation and dispersion. In particular, in the context of high-order methods, high accurate time integration schemes are mandatory to capture the significant flow features of transient problems and to perform accurate and efficient long time simulations of turbulent flows. The time discretization can be performed also by a discontinuous approximation [17], but the most usual approach is the application of the Method of Lines (MOL) strategy using high-order explicit Runge-Kutta methods to advance the solution in time. In general, Runge-Kutta schemes are easy to implement and parallelize, and require only limited memory storage. However, due to the strong restriction to the size of the time step for large-scale simulations and especially for high-order solutions, the convergence speed slows down dramatically, resulting in inefficient time integration technique. Furthermore, often the ordinary differential equations (ODEs) system coming from the discretization of the governing equations is stiff and fully implicit methods must be used to efficiently integrate these systems. A high-order implicit time integration approach seems well suited to be

coupled with high-order DG space discretizations of turbulence models equations. In fact, implicit methods can be very efficient in near-wall regions, that are characterized by highly-stretched grids, and, if accurate enough, they are also able to capture fine details of unsteady motions in regions farther from the walls, even using large time steps.

All the aforementioned capabilities suggest the potential of high-order approaches for the improvement of technology of automotive industries.

Please refer to Background section for a review of the topics and the main numerical techniques that have been adopted for the development of a high-order, in space and time, flow solver for ICE applications.

*Remark:* The numerical methods described in this thesis have been implemented and tested on the Discontinuous Galerkin flow solver `MIGALE` code of Bassi *et al.* (University of Bergamo) [18].

## 1.2 Background

### 1.2.1 transonic flows

Transonic flows play a important role in different area of science and engineering. These include, for instance, fluid dynamics, magnetohydrodynamics, aerospace, turbomachinery, and automotive research fields. Transonic flows may occur within internal combustion engines, through the intake and exhaust valves, and in both compressor and turbine of the turbocharger.

A flow is defined transonic if both subsonic and supersonic regions are present in the same flow field. Transonic flows are very complex and often they can develop strongly unsteady phenomena such as discontinuities, shock-boundary layer interaction and flow separation. Furthermore, they are in general sensitive to geometric boundary conditions. This sensitivity can be more significant for internal flows because of the unsteady interaction between the boundary layer generated behind the shock wave and the reflection of the

shock from the wall. Accurate and efficient simulations of these nonlinear unsteady phenomena are one of the great challenges of the computational fluid dynamics. Transonic problems are thus ideal to evaluate accuracy and robustness properties of numerical methods.

It's well known that numerical methods usually require the introduction of some form of dissipation or damping to prevent the occurrence of instabilities and large non-physical oscillations in non-linear problems with large gradients. In particular, in presence of shock, the numerical dissipation introduced by high-order discretizations is not sufficient to stabilize the solution, using an order of approximation higher than one. Several shock-capturing strategies inspired by finite volume schemes have been proposed for high-order method in order to accurately represent shocks. A straightforward approach consists on reducing the polynomial degree in those elements lying in the shock region, increasing the numerical dissipation added by the scheme. Then an adaptive mesh refinement is used in order to alleviate the problem of lack of accuracy near to the shock.

One of the most used strategies consists in adding to the DG discretized equations an artificial viscosity term that aims at controlling the high-order modes of the numerical solution within elements preserving as much as possible the spatial resolution of the discontinuities. The shock capturing term is local and acts in every element of the computational domain, adding an amount of artificial viscosity that is almost negligible in smooth region, and large in non-smooth one.

Unlike in finite volume schemes, where the shock is spread over several elements in the mesh, in DG schemes the artificial viscosity method has the capabilities to solve the shock typically in only one element (sub-cell resolution) [19–24].

### 1.2.2 turbulent flows

Most flows encountered in industrial application, are turbulent. In contrast to laminar flow, the turbulent flows are characterized by a strong chaotic molecular motion along complex irregular three-dimensional paths, and therefore by an intense mixing of the various layers of the fluid. Turbulent flows are highly unsteady, and fluctuate on a broad range of length and time scales. Turbulence leads to higher skin friction and heat transfer with respect to laminar flows due to an increased momentum and energy exchange between the molecules and the solid wall. [25]

All these characteristics play a key role in ICE systems and greatly affect the performance of internal combustion engines, in terms of energy consumption and pollutant emissions.

The most accurate approach to solve turbulent flows is the direct numerical simulation (DNS) of the Navier-Stokes equations, that allows the resolution of all the turbulent scales. Up to now, despite the continuous increase of the computer power, DNS computations can be performed only for low Reynolds number and for simple geometries. Therefore turbulent phenomena need to be modeled.

A large variety of turbulence models has been developed during the years. The closest approaches to DNS are the Large Eddy Simulations (LES) [26] and the Detached Eddy Simulations (DES) [27], in which large turbulent scales are resolved and only the small ones are modelled. These models turn out to be too expensive in terms of computing time for industrial applications, owing to the more complex nature of both the flow and the geometry.

For these reasons, three-dimensional fluid dynamics in ICEs are usually based on the Reynolds Averaged Navier-Stokes equations (RANS) approach where all the turbulent scales are modeled. Despite no detailed information can be obtained about turbulence scales using RANS approaches, their higher robustness and dramatic reduction in computational effort has greatly promoted

their use in engineering applications. The shortcomings of RANS models for ICE flows have been widely discussed in the literature [28–32].

Specifically, Hanjalic [29], showed that in a reciprocating engine the second order closure models (Reynolds Stress Models (RSM)) with appropriate modifications have the potential to solve ICE flows. The Reynolds Stress Model [33] solves modelled transport equations for the six Reynolds stress components. To close the partial differential equations for the Reynolds stress tensor an equation for the dissipation rate of the turbulence kinetic energy is usually employed. Closely related to the RSM model are the Algebraic Reynolds Stress Models (ARSM) [34, 35], in which the system of differential transport equations is reduced from seven to only two transport equations. They can be considered as a combination of first-order closure models and the RSM approach. ARSM models guarantee the anisotropic eddy diffusivities by relating the components of the Reynolds stress tensor to the transport quantities by non-linear algebraic equations, thus overcoming the high computational costs associated with second-order modelling. As part of ARSM models, the Explicit Algebraic Reynolds Stress Models (EARSM) [36–38] are recently gaining increasing attention. In EARSM models an explicit non-linear constitutive relation of the Reynolds Stress Tensor is defined in terms of the mean flow stream-rate and the rate-of-rotation tensors. Because of numerical problems mainly caused by the stiffness of RSM and the non-linearity of ARSM models, first-order closures are widely used for industrial applications with large and complex computational domains due to their higher robustness and lower computational cost. These models are based on the eddy viscosity hypothesis of Boussinesq [39], which imposes a linear relationship between turbulent shear stress and main strain rate. Despite their possible lower accuracy, for steady ICE flow analysis Bianchi and Fontanesi [40] and Auriemma et al. [41] highlighted that the two-equation turbulence models well reproduce the in-cylinder flow with a good agreement between experimental and numerical results.

Among the two-equations turbulence approaches, the  $k - \omega$  model is well

suitable for the resolution of turbulent ICE flows [42], allowing the resolution of the flow near the wall without the use of semi-empiric functions (wall-functions approach) [43,44] or viscous damping function (two-layer approach) as occurs in the  $k - \epsilon$  model.

### 1.2.3 Optimization method for computational fluid dynamics

In the recent decades, the employment of optimization method has become a powerful tool in order to improve the industrial design process and to reduce production costs. Historically, the existence of optimization methods can be traced to the period of Newton, Lagrange, and Cauchy, being related to the foundation of calculus of variations, which deals with the minimization of functionals. Following these early efforts, only the huge increase of the computer power occurred from the middle of the twentieth century has made possible the implementation of more advanced optimization procedures, encouraging further research, the development of new methods and the consequent emergence of several well-defined new areas in optimization theory [45–49].

Currently, optimization is applied to solve a wide range of engineering problems, and a wide range of optimization algorithms and methodologies are available also in the automotive sector [50–57]. For instance, for the shape optimization of the intake or exhaust ducts in internal combustion engines, to minimize the fuel consumption of gasoline engines, or for the vehicle drag reduction. Especially, in CFD context, the use of optimization algorithms are historically mainly related to i) optimization of the shape of the computational domain, in order to obtain the maximization or minimization of a particular functional, see for example the determination of the optimal shape for airfoils in order to reduce the drag or maximize the lift, ii) search for the optimal grid configuration in order to reduce the numerical error (grid-adaptivity). [58–69].

In general, optimization algorithms can be classified according to several

principles, but we can roughly divide them into indirect search methods, which require the computation of the gradient of the objective function, and direct method, depending on the objective function only through a set of function evaluations. The optimization algorithms that are not gradient-based mainly follow a stochastic approach for the resolution of the optimization problem. Stochastic optimization includes algorithms with a certain randomness in the search procedure. Some examples of stochastic algorithms are the game-theory based algorithms, in which the optimization strategy aims at emulating the evolution of a game in which different player try to fulfil their objectives, the evolutionary algorithms that emulating the evolution of specie according to Darwin's theory [70], or the genetic algorithms, belonging to the class of evolutionary algorithms but in which the input variables are discretized and stored into a binary string. Currently, stochastic optimization methods are the most advanced approach to optimization, and are gaining an increasing interest during the last decades also because they are particularly suited for the implementation on parallel computing environments.

The gradient-based approaches are the most widely used, also for hystorical reasons. The basic philosophy of these methods is to evaluate the derivative of the objective function (e.g. the lift, for aerodynamics applications) with respect to the design parameters to find the optimal values of the parameters. Since the gradient-based algorithms use gradient information, they converge much faster to a local optimum than the direct methods, especially when a large number of design parameter are involved in the optimization process. During the last decades, a large variety of gradient-based algorithms has been developed for the resolution of different kinds of problem, and a complete review of these can be found in Nocedal and Wright [71].

It's well known that the main difficulty in gradient-based optimization is the computation of the gradient vector. The gradient of the object function can be calculated in different ways. The simplest and also the oldest way is the finite difference method(FD). In fact, FD approach can be easily employed

with any flow solver, but require a large amount of computations and suffers from subtractive cancellation or round-off errors.

A more efficient and accurate way to determine the gradient is the Automatic or Algorithmic Differentiation (AD). The Automatic Differentiation (AD) is an innovative technique that applies symbolic differentiation to computer programs by means of the chain rule [72]. AD is based on the fact that any numerical code is a concatenation of various elementary operations whose differentiation rules are well known. By applying the chain rule to this concatenation an automatic differentiation of the numerical code is obtained.

Using AD, an augmented numerical code is generated from a given one, which calculates the derivatives of the functional with respect to the design parameters. Since the result is also a computer code that generates numbers rather than symbolic expressions, the technique is a numerical differentiation, although based on symbolic differentiation rules. Therefore AD is a very efficient tool for the evaluation of the gradient of the functional. Moreover, in contrast to FD, AD does not incur any truncation error, so the result is exact to machine accuracy.

The principles of AD are explained in detail e.g. in the work of Griewank [73].

#### **1.2.4 High order methods**

When the order of accuracy of a numerical method is at least greater than two, the numerical method is called high-order method.

In the last few decades, high-order methods have gained great attention in the CFD community due to their potential in providing higher accuracy solutions with lower cost than standard low-order schemes. The great potentiality of high-order schemes is related to their more efficient prediction capabilities with respect to low-order schemes. High order schemes, in fact, can achieve a prescribed error threshold on a much coarser mesh. Despite

second order scheme have been widely used in CFD during the years, many complex flow problems are still too expensive in terms of computational cost or cannot properly be solved using these schemes. Conversely, high-order methods have been shown good potential in solving complex flows such as those containing discontinuity, shocks, vortex structures or in direct simulation of turbulent flows. The most popular high-order methods are the spectral ones, whose basis functions are chosen as sums of sinusoids [74]. Although spectral methods are computationally less expensive than finite element methods, they become less accurate to face problems in complex geometries, or in presence of discontinuities [75].

In order to overcome the limitations of the spectral scheme, in the early 1980s the research moved towards the  $p$ -type finite element method.  $P$ -type FEM allows, for a given grid spacing  $h$ , a decrease of the error, by means of the increasing of the polynomial degree  $p$ . In 1981, Babuska et al. [76] applied this method to elasticity problems, concluding that the rate of convergence of the  $p$ -type method cannot be slower than that of the  $h$ -type and, in cases with singularity problem, the convergence rate of  $p$ -type is two times faster. Starting from these first studies, over the years, a significant research effort has been aimed at developing high-order accurate methods [77–80]. More recently, the computational mathematics program of the Air Force Office of Scientific Research (AFOSR) in the United States and the projects ADIGMA (Adaptive Higher-order Variational Methods for Aerodynamic Applications in Industry) [81] and IDIHOM (Industrialisation of High-Order Methods A Top-Down Approach) [82] in Europe, have devoted significant efforts for the development and assessment of high-order methods for academic and industrial applications.

Among the high-order accurate methods, Discontinuous Galerkin (DG) finite element method is now one of the most popular, due to its many attractive features, as reported previously in the section 1.1 Motivation.

The Discontinuous Galerkin Method was originally introduced by Reed and Hill [83] in 1973 for neutron transport problems. The properties of the

method in terms of stability and accuracy have been rigorously proven by Johnson and Pitkarata [84], Cockburn and Shu [12], Cockburn *et al.* [85] and Jiang and Shu [86] for arbitrary element shapes, for any number of spatial dimensions, and even for non-linear problems. During the years, implicit and explicit high order DG flow solvers have been developed for the resolution of compressible and incompressible Euler and Navier-Stokes equations, even for complex 3D applications and are now rather well-established. An implicit DG method for the coupled RANS (Reynolds-Averaged Navier-Stokes) and  $k - \omega$  turbulence model equations was developed by e.g. Bassi *et al.* [18, 87] and Hartmann [21].

### **1.2.5 Goals of research: a high order flow solver for ICE applications based on DG method**

The aim of this work is to contribute to the development of an advanced computational code dedicated to the simulation of turbulent ICE flows. The term advanced highlights the substantial progress compared to the CFD codes currently in use in the automotive industrial field, because the solver developed will be characterized by high order of accuracy even in the case of complex geometries.

In this research project the interest will be focused on the Discontinuous Galerkin (DG) finite element method, one of the most promising high-order methods for the simulation of practical turbulent flows. From the numerical point of view, the solution of the governing equations for turbulent flows is a difficult task, as it requires the use of very accurate schemes. The ideal candidates for this purpose are the spectral methods, which, however, are difficult to use in the case of complex geometries and for compressible flows in the presence of discontinuities. Therefore, DG methods, combining excellent dispersion and dissipation properties with geometric flexibility, are expected to be more appropriate to ensure high-order of accuracy, even for turbulent

flows of industrial interest.

A high-order implicit time integration approach seems well suited to be coupled with high-order DG space discretizations of turbulence model equations. Here we focus on accurate time integration by means of high-order Two Implicit Advanced Step point (TIAS) schemes [88, 89], recently applied by Nigro et al. [90] to the Discontinuous Galerkin discretized Navier-Stokes equations, to evaluate the potential of such very high-order accurate A-stable integrators for long time simulations of stiff and non-stiff unsteady flow problems. The main advantages of TIAS schemes are that they have excellent stability properties being A-stables up to the sixth-order of accuracy. Other approaches to high-order implicit time integration have been implemented in the present DG solver and presented elsewhere, [91, 92].

The second objective of this work is to demonstrate the applicability and reliability of optimization algorithms to control spurious numerical oscillations in simulation of transonic flows. The proposed optimization strategy follows the gradient based optimization approach employing an Automatic Differentiation (AD) tool [93, 94] for the evaluation of the sensitivities. The optimization process acts directly on the shock capturing technique, seeking for the optimal values of the shock capturing parameters.

As intermediate results, which are fundamental to achieve the aims of the project, several test-cases of direct relevance in the context of automotive and aerodynamic applications will be performed. The comparison with both experimental/analytical and numerical results will allow the critical validation and/or revision of physical and numerical models introduced in the computational code.

This work is the starting point of a project that aims to deal with the class of problems for which the classical RANS approaches show clear / known limitations. Flows characterized by Reynolds number not too high or by large separation zones are classic examples. These fluid dynamic problems are of great practical importance in many other research areas. A typical example is

that of the renewable energy with the development / study of wind turbines. In this particular application the physical problems particularly difficult to model with the RANS approaches are the formation of laminar separation bubbles, with turbulent reattachment, or the flow separation downstream of a profile, in a rotating system, with a very high angle of incidence. Further potential applications include the analysis of unsteady flows in axial turbomachinery because the vanes of compressors and turbines are affected by complex turbulent flows with or without flow separation, especially in off-design operating conditions.

## 1.3 Outline of Thesis

This thesis deals with the development of an high-order accurate flow solver for ICE applications based on the discontinuous Galerkin (DG) finite element method.

The outline of this work is as follows:

- In Chapter 2 we present the physical models adopted: the two-dimensional RANS and  $k - \omega$  equations for axisymmetric flows and the two dimensional Euler equations in cartesian coordinate system.
- In Chapter 3 we describe the DG flow solver. We start by presenting the high-order DG spatial discretization of RANS  $k - \omega$  and Euler equations. Then we describe the time integration schemes adopted: the high-order TIAS implicit scheme, the explicit SSPRK and the implicit backward Euler. Finally, we present gradient based design optimization and automatic differentiation (AD) strategies.
- In Chapter 4 we give the results of several test-cases. Two internal turbulent flows and the long time simulation of an isentropic vortex advection are computed to validate the proposed high-order space-accurate DG method and to evaluate the performance of a high-order time-accurate implicit time scheme, respectively. Finally, an inviscid transonic flow is analysed to evaluate the feasibility of gradient based design optimization for the optimal control of spurious numerical oscillations.

Finally, we give conclusions at the end of this work.

# Chapter 2

## Physical models

In this chapter we present the compressible governing equations used in this work. We first introduce the fully coupled Three-dimensional Reynolds Average Navier Stokes equations (RANS) and  $k$ - $\omega$  turbulence model equations in cylindrical coordinate system  $(z, r, \theta)$ , written in conservative form. Then by neglecting all the tangential terms in  $\theta$  direction, the two-dimensional form of the equations for axisymmetric flows without swirl have been derived. Finally, we introduce the two-dimensional Euler equations in cartesian coordinate system. The two-dimensional RANS and  $k$ - $\omega$  equations for axisymmetric flows without swirl, and the two-dimensional Euler equations are used to evaluate the effectiveness of algorithms and strategies proposed in this work through the solution of different test-cases. In addition, some considerations concerning the non-dimensionalization of the equations are given.

## 2.1 RANS and $k$ - $\omega$ equations

### 2.1.1 Three dimensional RANS and $k - \omega$ equations in cylindrical coordinate system

The three-dimensional RANS and  $k - \omega$  equations for compressible flows in cylindrical coordinate system  $(z, r, \theta)$  can be written in conservative form as

$$\frac{\partial}{\partial t}(\rho r) + \frac{\partial}{\partial z}(\rho r v_z) + \frac{\partial}{\partial r}(\rho r v_r) + \frac{\partial}{\partial \theta}(\rho r v_\theta) = 0, \quad (2.1)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r e_t) + \frac{\partial}{\partial z}(\rho r h_t v_z) + \frac{\partial}{\partial r}(\rho r h_t v_r) + \frac{\partial}{\partial \theta}(\rho r h_t v_\theta) \\ = A + B - rP + \beta^* \rho r \bar{k} e^{\tilde{\omega}_r}, \end{aligned} \quad (2.2)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r v_z) + \frac{\partial}{\partial z}[r(p + \rho v_z^2)] + \frac{\partial}{\partial r}(\rho r v_z v_r) + \frac{\partial}{\partial \theta}(\rho v_z v_\theta) \\ = \frac{\partial}{\partial z}(r \hat{\tau}_{zz}) + \frac{\partial}{\partial r}(r \hat{\tau}_{zr}) + \frac{\partial}{\partial \theta}(r \hat{\tau}_{z\theta}), \end{aligned} \quad (2.3)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r v_r) + \frac{\partial}{\partial z}(\rho r v_z v_r) + \frac{\partial}{\partial r}[r(p + \rho v_r^2)] + \frac{\partial}{\partial \theta}(\rho v_r v_\theta) \\ - p - \rho v_\theta^2 = \frac{\partial}{\partial z}(r \hat{\tau}_{zr}) + \frac{\partial}{\partial r}(r \hat{\tau}_{rr}) + \frac{\partial \hat{\tau}_{r\theta}}{\partial \theta} - \hat{\tau}_{\theta\theta}, \end{aligned} \quad (2.4)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r v_\theta) + \frac{\partial}{\partial z}(\rho r v_z v_\theta) + \frac{\partial}{\partial r}[r(p + \rho v_\theta^2)] + \frac{\partial}{\partial \theta}(p + \rho v_\theta^2) \\ = \frac{\partial}{\partial z}(r \hat{\tau}_{z\theta}) + \frac{\partial}{\partial r}(r \hat{\tau}_{r\theta}) + \frac{\partial \hat{\tau}_{\theta\theta}}{\partial \theta} - \hat{\tau}_{r\theta}, \end{aligned} \quad (2.5)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r k) + \frac{\partial}{\partial z}(\rho r v_z k) + \frac{\partial}{\partial r}(\rho r v_r k) + \frac{\partial}{\partial \theta}(\rho v_\theta k) \\ = \frac{\partial}{\partial z}\left(\bar{\mu}_k r \frac{\partial k}{\partial z}\right) + \frac{\partial}{\partial r}\left(\bar{\mu}_k r \frac{\partial k}{\partial r}\right) + \frac{\partial}{\partial \theta}\left(\bar{\mu}_k r \frac{\partial k}{\partial \theta}\right) \\ + rP - \beta^* \rho r \bar{k} e^{\tilde{\omega}_r}, \end{aligned} \quad (2.6)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho r \tilde{\omega}) + \frac{\partial}{\partial z}(\rho r v_z \tilde{\omega}) + \frac{\partial}{\partial r}(\rho r v_r \tilde{\omega}) + \frac{\partial}{\partial \theta}(\rho v_\theta \tilde{\omega}) \\ = \frac{\partial}{\partial z}\left(\bar{\mu}_\omega r \frac{\partial \tilde{\omega}}{\partial z}\right) + \frac{\partial}{\partial r}\left(\bar{\mu}_\omega r \frac{\partial \tilde{\omega}}{\partial r}\right) + \frac{\partial}{\partial \theta}\left(\bar{\mu}_\omega r \frac{\partial \tilde{\omega}}{\partial \theta}\right) \\ + \frac{\alpha r \tilde{\omega}}{k} P - \beta \rho r e^{\tilde{\omega}_r}. \end{aligned} \quad (2.7)$$

where  $\rho$ ,  $p$ ,  $\mathbf{v} = (v_z, v_r, v_\theta)$ ,  $e_t$ ,  $h_t = e_t + p/\rho$ ,  $k$ ,  $\omega$ , denote the density, pressure, velocity, total internal energy, total enthalpy of the mean motion, turbulent kinetic energy and specific dissipation rate, respectively. For ideal gases the

pressure is given by

$$p = (\gamma - 1) \rho \left( e_t - \frac{v_z^2}{2} - \frac{v_r^2}{2} - \frac{v_\theta^2}{2} \right),$$

where  $\gamma$  is the constant ratio of specific heats. The term  $A$  on the right-hand side of Eq. (2.2) denotes the divergence of the work of total stresses  $\nabla \cdot (r\hat{\boldsymbol{\tau}}\mathbf{v})$  and  $B$  the divergence of total heat flux vector  $\nabla \cdot (r\mathbf{q})$ . Specifically,

$$A = \frac{\partial}{\partial z} (r (\hat{\tau}_{zz}v_z + \hat{\tau}_{zr}v_r + \hat{\tau}_{z\theta}v_\theta)) + \frac{\partial}{\partial r} (r (\hat{\tau}_{zr}v_z + \hat{\tau}_{rr}v_r + \hat{\tau}_{r\theta}v_\theta)) + \frac{\partial}{\partial \theta} (\hat{\tau}_{z\theta}v_z + \hat{\tau}_{r\theta}v_r + \hat{\tau}_{\theta\theta}v_\theta),$$

$$B = -r \frac{\partial q_z}{\partial z} - \frac{\partial (rq_r)}{\partial r} - \frac{\partial q_\theta}{\partial \theta},$$

where the turbulent stress tensor components are defined as:

$$\tau_{zz} = 2\bar{\mu}_t \left( \frac{\partial v_z}{\partial z} - \frac{1}{3} \nabla \cdot \mathbf{v} \right) - \frac{2}{3} \rho k,$$

$$\tau_{rr} = 2\bar{\mu}_t \left( \frac{\partial v_r}{\partial r} - \frac{1}{3} \nabla \cdot \mathbf{v} \right) - \frac{2}{3} \rho k,$$

$$\tau_{\theta\theta} = 2\bar{\mu}_t \left( \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} - \frac{1}{3} \nabla \cdot \mathbf{v} \right) - \frac{2}{3} \rho k,$$

$$\tau_{zr} = \bar{\mu}_t \left( \frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z} \right),$$

$$\tau_{z\theta} = \bar{\mu}_t \left( \frac{\partial v_\theta}{\partial z} + \frac{1}{r} \frac{\partial v_z}{\partial \theta} \right),$$

$$\tau_{r\theta} = \bar{\mu}_t \left( \frac{\partial v_\theta}{\partial r} + \frac{1}{r} \frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r} \right).$$

and the total stresses  $\hat{\boldsymbol{\tau}}$  are given by the sum of turbulent and viscous stresses:

$$\hat{\boldsymbol{\tau}} = \boldsymbol{\tau} + \frac{\mu}{\mu_t} \left( \boldsymbol{\tau} + \frac{2}{3} \rho k \mathbf{I} \right).$$

Moreover the heat flux components are given by

$$q_z = \lambda \frac{\partial h}{\partial z}, \quad q_r = \lambda \frac{\partial h}{\partial r}, \quad q_\theta = \frac{\lambda}{r} \frac{\partial h}{\partial \theta},$$

in which  $\lambda$  indicates the gas conductivity. It is worth pointing out the presence of a source term in Eq.(2.2) related to the fact that the turbulent kinetic energy is not included in total internal energy and total enthalpy. The production term that models the energy exchange between mean and turbulent field can be written as

$$P = \bar{\mu}_t \left\{ 2 \left[ \left( \frac{\partial v_z}{\partial z} \right)^2 + \left( \frac{\partial v_r}{\partial r} \right)^2 + \left( \frac{v_r}{r} \right)^2 + \left( \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} \right)^2 + \right. \right. \\ \left. \left( \frac{1}{r} \frac{\partial v_z}{\partial \theta} + \frac{\partial v_\theta}{\partial z} \right)^2 + \left( \frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r} \right)^2 + \left( \frac{1}{r} \frac{\partial v_r}{\partial \theta} + \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \right)^2 - \right. \\ \left. \left. \frac{2}{3} (\nabla \cdot \mathbf{v})^2 \right\} - \frac{2}{3} \rho k \nabla \cdot \mathbf{v},$$

in which:

$$\nabla \cdot \mathbf{v} = \frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}.$$

The turbulence model adopted is the  $k - \omega$  Wilcox high-Reynolds number turbulence model [95], where the turbulent eddy viscosity and the effective viscosity coefficients of the turbulence equations are expressed as follows:

$$\mu_t = \frac{\alpha^* \rho k}{\omega}, \quad \mu_k = \mu + \sigma^* \mu_t, \quad \mu_\omega = \mu + \sigma \mu_t, \quad (2.8)$$

and  $\alpha$ ,  $\alpha^*$ ,  $\beta$ ,  $\beta^*$ ,  $\sigma$ , and  $\sigma^*$  are the model closure coefficients. Following the approach described in [96, 97], the model employs the variable  $\tilde{\omega} = \ln(\omega)$  instead of  $\omega$  to guarantee the positivity of  $\omega$  and to obtain a smoother near wall distribution. Moreover, the variables  $k$  and  $\tilde{\omega}$  are limited from below by

$$\bar{k} = \max(k, 0), \quad \tilde{\omega}_r = \max(\tilde{\omega}_{r0}, \tilde{\omega}), \quad (2.9)$$

where  $\tilde{\omega}_{r0}$  defines the lower bound on  $\tilde{\omega}$  that ensures the positivity of normal turbulent stresses and the fulfillment of the Schwarz inequality for shear turbulent stresses. The turbulent eddy viscosity and effective eddy viscosity are then computed by the following expressions:

$$\bar{\mu}_t = \alpha^* \rho \bar{k} e^{-\tilde{\omega}_r}, \quad \bar{\mu}_k = \mu + \sigma^* \bar{\mu}_t, \quad \bar{\mu}_\omega = \mu + \sigma \bar{\mu}_t. \quad (2.10)$$

Note that a simplification of the 3D system of governing Eqs. (2.1)-(2.7) enable to obtain a reduced system corresponding to a 2D system in axisymmetric form in the two independent variables  $(z, r)$ . This system takes into account a possible tangential component of velocity varying along  $z$  and  $r$ , therefore may be adopted instead of the full 3D system for axisymmetric flows in presence of swirl.

### 2.1.2 Two dimensional RANS and $k - \omega$ equations for axisymmetric flows

Here we consider the coupled set of RANS and  $k - \omega$  equations in axisymmetric coordinates [98] without swirl. These equations have been used to perform the simulations of two different test-cases concerning turbulent internal flows, to assess the predictive capabilities of the DG spatial discretization, as will be explained in the next chapter.

The two-dimensional RANS and  $k - \omega$  equations for compressible flows with no-swirl component in conservative form can be derived from the equations (2.1)-(2.7) by neglecting the derivatives and the component of the momentum equations in  $\theta$  direction:

$$\frac{\partial}{\partial t}(\rho r) + \frac{\partial}{\partial z}(\rho r v_z) + \frac{\partial}{\partial r}(\rho r v_r) = 0, \quad (2.11)$$

$$\frac{\partial}{\partial t}(\rho r e_t) + \frac{\partial}{\partial z}(\rho r h_t v_z) + \frac{\partial}{\partial r}(\rho r h_t v_r) = A + B - rP + \beta^* \rho \bar{k} e^{\tilde{\omega} r}, \quad (2.12)$$

$$\frac{\partial}{\partial t}(\rho r v_z) + \frac{\partial}{\partial z}[r(p + \rho v_z^2)] + \frac{\partial}{\partial r}(\rho r v_z v_r) = \frac{\partial}{\partial z}(r \hat{\tau}_{zz}) + \frac{\partial}{\partial r}(r \hat{\tau}_{zr}), \quad (2.13)$$

$$\frac{\partial}{\partial t}(\rho r v_r) + \frac{\partial}{\partial z}(\rho r v_z v_r) + \frac{\partial}{\partial r}[r(p + \rho v_r^2)] - p = \frac{\partial}{\partial z}(r \hat{\tau}_{zr}) + \frac{\partial}{\partial r}(r \hat{\tau}_{rr}), \quad (2.14)$$

$$\frac{\partial}{\partial t}(\rho r k) + \frac{\partial}{\partial z}(\rho r v_z k) + \frac{\partial}{\partial r}(\rho r v_r k) = \frac{\partial}{\partial z}\left(\bar{\mu}_k r \frac{\partial k}{\partial z}\right) + \frac{\partial}{\partial r}\left(\bar{\mu}_k r \frac{\partial k}{\partial r}\right) + rP - \beta^* \rho \bar{k} e^{\tilde{\omega} r}, \quad (2.15)$$

$$\frac{\partial}{\partial t}(\rho r \tilde{\omega}) + \frac{\partial}{\partial z}(\rho r v_z \tilde{\omega}) + \frac{\partial}{\partial r}(\rho r v_r \tilde{\omega}) = \frac{\partial}{\partial z}\left(\bar{\mu}_\omega r \frac{\partial \tilde{\omega}}{\partial z}\right) + \frac{\partial}{\partial r}\left(\bar{\mu}_\omega r \frac{\partial \tilde{\omega}}{\partial r}\right) + \frac{\alpha r \tilde{\omega}}{\bar{k}} P - \beta \rho r e^{\tilde{\omega} r}. \quad (2.16)$$

in which  $\rho$ ,  $p$ ,  $\mathbf{v} = (v_z, v_r)$ ,  $e_t$ ,  $h_t = e_t + p/\rho$  denote the density, pressure, velocity, total internal energy and total enthalpy of the mean motion, respectively, as seen in the previous paragraph. The pressure is given by

$$p = (\gamma - 1) \rho \left( e_t - \frac{v_z^2}{2} - \frac{v_r^2}{2} \right),$$

where  $\gamma$  is the constant ratio of specific heats. The terms  $A$  and  $B$  on the right-hand side of Eq. (2.12) become:

$$A = r \frac{\partial}{\partial z} (\hat{\tau}_{zz} v_z + \hat{\tau}_{zr} v_r) + \frac{\partial}{\partial r} (r \hat{\tau}_{zr} v_z + r \hat{\tau}_{rr} v_r),$$

$$B = -r \frac{\partial q_z}{\partial z} - \frac{\partial (r q_r)}{\partial r},$$

Whereas the production term can be written as:

$$P = \bar{\mu}_t \left\{ 2 \left[ \left( \frac{\partial v_z}{\partial z} \right)^2 + \left( \frac{\partial v_r}{\partial r} \right)^2 + \left( \frac{v_r}{r} \right)^2 \right] + \left( \frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r} \right)^2 - \frac{2}{3} (\nabla \cdot \mathbf{v})^2 \right\} - \frac{2}{3} \rho k \nabla \cdot \mathbf{v},$$

with:

$$\nabla \cdot \mathbf{v} = \frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r} + \frac{v_r}{r}.$$

## 2.2 Euler equation

### 2.2.1 Two dimensional Euler equation

The compressible Euler equations describe the pure convection of flow quantities in an inviscid fluid. In this work two dimensional Euler equation have been used to perform two different test cases, the first one concern the convection of an isentropic vortex to assess the performance of the high-order time integration TIAS scheme, whereas the second one is a supersonic wedge, used in this work to optimize the shock capturing strategy by the employ of two optimization algorithms based on the gradient based approach coupled to Automatic Differentiation.

The two-dimensional Euler equations in cartesian coordinates are given by:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_x}{\partial x} + \frac{\partial \rho v_y}{\partial y} = 0, \quad (2.17)$$

$$\frac{\partial \rho v_x}{\partial t} + \frac{\partial (\rho v_x^2 + p)}{\partial x} + \frac{\partial \rho v_x v_y}{\partial y} = 0, \quad (2.18)$$

$$\frac{\partial \rho v_y}{\partial t} + \frac{\partial \rho v_x v_y}{\partial x} + \frac{\partial (\rho v_y^2 + p)}{\partial y} = 0, \quad (2.19)$$

$$\frac{\partial \rho e_t}{\partial t} + \frac{\partial \rho v_x h_t}{\partial x} + \frac{\partial \rho v_y h_t}{\partial y} = 0. \quad (2.20)$$

Here,  $v_x$  and  $v_y$  are the velocity components, in the x and y directions respectively. We assume that the fluid satisfies the equation of state of a perfect gas, then the pressure is given by  $p = (\gamma - 1) \rho [e - (v_x^2 + v_y^2) / 2]$ .

## 2.3 Non-dimensionalization

In this work we consider the governing equations in non-dimensionalized variables based on the reference length  $l_r$ , density  $\rho_r$ , pressure  $p_r$  and the gas constant  $R_r$ . Therefore, the specific heat capacities at constant volume  $c_v$ , constant pressure  $c_p$  and the molecular viscosity  $\mu$  are given by

$$c_v = \frac{1}{\gamma - 1}, \quad c_p = \frac{\gamma}{\gamma - 1},$$

$$\mu = \frac{\sqrt{\gamma} M_o T^\alpha}{Re_o},$$

where  $M_o$  and  $Re_o$  are the Mach and Reynolds numbers at the reference conditions, respectively,  $T$  is the non-dimensional temperature and  $\alpha = 3/4$ . Reference values for the other quantities are derived from these by functional relationships. With this choice of non-dimensionalized variables, all the governing equations remain unchanged, except that the variables are now understood to be non-dimensionalized. Note that the reference conditions used in the numerical computations of external flows are the freestream conditions, whereas, all the ICE simulations are performed based on the inlet free stream conditions. We choose the length of the flow domain and the size of the inlet diameter as

reference lengths for the external and internal flow computations, respectively. Therefore, since the reference velocity is of the same order of the speed of sound,  $v_r = \sqrt{\frac{p_r}{\rho_r}}$ , the reference time is of the order of the time required for the acoustic wave to travel over the reference length,  $t_r = l_r/\rho_r$ .

## Chapter 3

# Development of a high order DG flow solver for ICE applications

In this chapter we present aspects of the high-order DG flow solver developed in this work.

We begin by presenting the DG discretization of the Euler and RANS  $k-\omega$  equations for two-dimensional and axisymmetric compressible flows. Subsequently, we describe the different ways of computing the numerical fluxes and present the boundary conditions used in this work. Finally we describe the algorithm for control of oscillations of high-order solutions around shocks. The shock capturing strategy has evolved from the approach presented in [22]. A gradient based optimization algorithm have been applied to control spurious oscillations. The proposed optimization strategy and the Automatic Differentiation (AD) tool for the evaluation of the sensitivities are discussed at the end of this chapter within the sections "Gradient based optimization" and "Evaluation of the gradient". The system of ordinary differential equations arising from the DG space discretization can be solved using different integrators. Here we focus our attention on the high-order implicit multi-stage, multi-step

TIAS method for use in long time simulations of stiff and non-stiff unsteady problems. Furthermore, the high-order accurate strong stability preserving Runge Kutta schemes (SSPRK4,5), and the first-order accurate backward Euler scheme are presented, which have been used to compare the performance of the sixth-order accurate TIAS scheme to the performance of a widely used high-order explicit scheme and to compute steady state solutions of internal ICE flows, respectively.

## 3.1 Space discretization

### 3.1.1 The DG discretization of RANS $k - \omega$ equations

The RANS and  $k - \omega$  turbulence model expressed in Eqs.(2.11 – 2.16) can be written in compact form as

$$\frac{\partial(r\mathbf{u})}{\partial t} + \nabla \cdot (r\mathbf{F}_c(\mathbf{u})) + \nabla \cdot (r\mathbf{F}_v(\mathbf{u}, \nabla\mathbf{u})) + r\mathbf{s}(\mathbf{u}, \nabla\mathbf{u}) = 0, \quad (3.1)$$

where  $\mathbf{u} \in \mathbb{R}^M$  denotes the vector of the  $M$  conservative variables,  $\mathbf{s} \in \mathbb{R}^M$  the sum of turbulent and axisymmetric source term vectors,  $\mathbf{F}_c, \mathbf{F}_v \in \mathbb{R}^M \otimes \mathbb{R}^N$  denote the inviscid and viscous flux functions, respectively, and  $N$  is the space dimension.

In order to construct the DG space discretization of the coupled set of RANS and  $k - \omega$  equations, we define  $\mathbf{V}_h$  to be the space of discontinuous vector-valued polynomials of degree  $n$ , on a subdivision  $\tau_h$  of the domain  $\Omega$  into non-overlapping elements  $K$  such as  $\Omega_h = \bigcup_{K \in \tau_h} K$ . Thus, the solution and test functions space is defined by

$$\mathbf{V}_h = \{\mathbf{v}_h \in L^2(\Omega_h) : \mathbf{v}_h|_K \in P_n(K) \forall K \in \tau_h\}, \quad (3.2)$$

where  $P_n(K)$  is the space of polynomial functions of degree at most  $n$  in the element  $K$ .  $\Omega_h$  is the discrete approximation of the domain,  $\partial\Omega_h$  the boundary,  $\Gamma_h^0$  the set of internal edges and  $\Gamma_h$  denotes the union of  $\Gamma_h^0$  and  $\partial\Omega_h$  element edges,  $E$ , such that  $\Gamma_h = \Gamma_h^0 \cup \partial\Omega_h$ .

The DG formulation of the compressible RANS and  $k - \omega$  Eq. (3.1) is seeking for  $\mathbf{u}_h \in \mathbf{V}_h$  such as, for an arbitrary test function  $\mathbf{v}_h \in \mathbf{V}_h$ , the following equation is satisfied:

$$\begin{aligned} & \int_{\Omega_h} \mathbf{v}_h \cdot \frac{\partial \mathbf{u}_h}{\partial t} r d\mathbf{x} - \int_{\Omega_h} \nabla \mathbf{v}_h : (\mathbf{F}_c(\mathbf{u}_h) - \mathbf{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h + \mathcal{R}(\llbracket \mathbf{u}_h \rrbracket_0))) r d\mathbf{x} \\ & + \int_{\Gamma_h^0} (\mathbf{v}_h^- - \mathbf{v}_h^+) \cdot \mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) r d\sigma - \int_{\Gamma_h^0} \llbracket \mathbf{v}_h \rrbracket : \{\mathbf{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h + \eta_e \mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket_0))\} r d\sigma \end{aligned}$$

$$\begin{aligned}
& + \int_{\partial\Omega_h} (\mathbf{v}_h \otimes \mathbf{n}) : (\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^b, \mathbf{n}) - \mathbf{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h + \eta_e \mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket_0)))_b) r d\sigma = 0 \\
& + \int_{\Omega_h} \mathbf{v}_h \cdot \mathbf{s}(\mathbf{u}_h, \nabla \mathbf{u}_h + \mathcal{R}(\llbracket \mathbf{u}_h \rrbracket_0)) r d\mathbf{x} = 0 \tag{3.3}
\end{aligned}$$

Where  $(\cdot)^+$  and  $(\cdot)^-$  denote the values of any quantity evaluated from inside and outside faces of an element  $K$ , and  $\mathbf{n}^\pm$  are the unit normal vectors to the shared edge  $E$ , such that  $\mathbf{n}^- = -\mathbf{n}^+$ , as shown in Fig. 3.1.

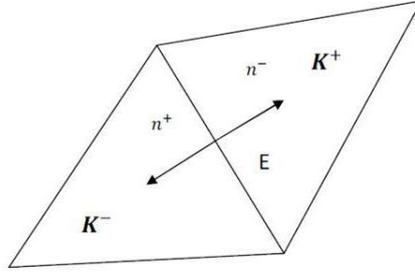


Figure 3.1: Two neighboring elements  $K^-$  and  $K^+$  sharing edge  $E$ , including the two opposite normal vectors  $\mathbf{n}^\pm$  to  $E$ .

The discretization in Eq.(3.3) is based on the *BR2* scheme [96, 99] for the approximation of the viscous part. According to *BR2* the viscous flux  $\mathbf{F}_v$  is evaluated as

$$\mathbf{F}_v|_{\Omega_h} = \mathbf{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h + \mathcal{R}(\llbracket \mathbf{u}_h \rrbracket_0)),$$

$$\mathbf{F}_v|_{\Gamma_h} = \mathbf{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h + \eta_e \mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket)),$$

in which  $\eta_e$  is called "penalty" parameter, and  $\mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket)$  and  $\mathcal{R}(\llbracket \mathbf{u}_h \rrbracket_0)$  stand for the local and global lifting operators, respectively. The lower bound of the  $\eta_e$  parameter correspond to the number of neighbours of the generic element  $K$ , to guarantee the stability of the method. The local and global lifting operators are given by:

$$\begin{aligned}
\int_{\Omega_h} \mathbf{v}_h \cdot \mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket) d\mathbf{x} &= - \int_E \{\mathbf{v}_h^T\} \cdot \llbracket \mathbf{u}_h \rrbracket d\sigma, \\
\mathcal{R}(\llbracket \mathbf{u}_h \rrbracket_0) &= \sum_E \mathcal{R}_e(\llbracket \mathbf{u}_h \rrbracket),
\end{aligned}$$

with the jump  $[[\cdot]]$  operator and the trace operator  $\{(\cdot)\}$  defined as:

$$[[\mathbf{u}_h]] = \mathbf{u}_h^+ \otimes \mathbf{n}^+ + \mathbf{u}_h^- \otimes \mathbf{n}^-, \quad \{\mathbf{v}_h\} = \frac{v_h^+ + v_h^-}{2},$$

and acting component-wise when applied to a vector. Furthermore,  $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$  and  $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^b, \mathbf{n})$  are the numerical flux functions at the interior and boundary faces, respectively. In this work all the computations have been performed using the Godunov flux.

### 3.1.2 The DG discretization of the Euler equations

The DG formulation of the two-dimensional Euler equations (2.17 – 2.20) can be derived from Eq.(3.1) by canceling  $r$  from all integral contributions, neglecting the viscous terms and considering the  $\mathbf{u}$  and  $\mathbf{F}_c$  vector components directed along the coordinate axes  $(x, y)$ . It becomes:

find  $\mathbf{u}_h \in \mathbf{V}_h$  such that

$$\begin{aligned} \int_{\Omega_h} \mathbf{v}_h \cdot \frac{\partial \mathbf{u}_h}{\partial t} d\mathbf{x} - \int_{\Omega_h} \nabla \mathbf{v}_h : \mathbf{F}_c(\mathbf{u}_h) d\mathbf{x} + \int_{\Gamma_h^o} (\mathbf{v}_h^- - \mathbf{v}_h^+) \cdot \mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) d\sigma \\ + \int_{\partial\Omega_h} (\mathbf{v}_h \otimes \mathbf{n}) : \mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^b, \mathbf{n}) d\sigma = 0 \end{aligned} \quad (3.4)$$

for all  $\mathbf{v}_h \in \mathbf{V}_h$ .

## 3.2 Numerical Fluxes

Due to the discontinuous function approximation, the convective flux  $\mathbf{F}_c(\mathbf{u}_h)$  is not uniquely defined at element interfaces. Therefore  $\mathbf{F}_c(\mathbf{u}_h) \cdot \mathbf{n}$  is replaced in the third and fourth term of Eq.(3.3) by a numerical flux function  $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$ , which depends on the internal interface state  $\mathbf{u}_h^+$ , on the neighbouring element interface state  $\mathbf{u}_h^-$  and on the unit normal vector to the element interface,  $\mathbf{n}$ . In order to guarantee the conservativeness property of

the scheme  $\mathbf{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$  must be defined in a consistent manner:

$$\mathbf{H}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{n}) = \mathbf{F}_c(\mathbf{u}_h) \cdot \mathbf{n}, \quad \mathbf{H}(\mathbf{u}_h, \mathbf{v}_h, \mathbf{n}) = -\mathbf{H}(\mathbf{v}_h, \mathbf{u}_h, \mathbf{n})$$

Many numerical flux functions satisfy the above criteria, such as the Godunov, Roe, Lax-Friedrichs, Engquist-Osher or Harlen-Lax-van Leer (HYLE) flux, for example. The following numerical flux functions are present in the code.

### 3.2.1 Godunov flux

The Godunov flux  $\mathbf{H}_G(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$  is defined by,

$$\mathbf{H}_G(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})|_{\partial K} = \mathbf{F}_c(\mathbf{u}_h^*) \cdot \mathbf{n} \quad \forall K \in \tau_h,$$

where  $(\mathbf{u}_h^*)$  is the exact solution of a planar Riemann problem in the direction normal to the interface [100]. The Godunov flux is very accurate but computationally expensive. It can be used for all flow regimes.

### 3.2.2 Roe flux

The Roe flux  $\mathbf{H}_R(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$  corresponds to the approximate solution of the Riemann problem. The Roe's method is widely employed due to its high accuracy in solving both boundary layers and shocks. It is defined by,

$$\mathbf{H}_R(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})|_{\partial K} = \frac{1}{2} (\mathbf{F}_c(\mathbf{u}^+) + \mathbf{F}_c(\mathbf{u}^-)) \cdot \mathbf{n} - \mathbf{A}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) \Delta \mathbf{u} \quad \forall K \in \tau_h,$$

where  $\mathbf{F}_c(\mathbf{u}^+)$  and  $\mathbf{F}_c(\mathbf{u}^-)$  are the convective fluxes computed on each (the interior and the exterior) side of the interface and  $\mathbf{A}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) \Delta \mathbf{u}$  is the dissipation term with  $\Delta \mathbf{u} = \mathbf{u}^- - \mathbf{u}^+$ . The matrix  $|\tilde{\mathbf{A}}|$  denotes the Roe matrix or dissipation matrix and it is computed through the so-called Roe-average variables [3] as

$$\mathbf{A}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}) = \left| \left( \frac{\partial \mathbf{F}_c}{\partial \mathbf{u}} \cdot \mathbf{n} \right) \right| = (\mathbf{T} |\Lambda| \mathbf{T}^{-1}),$$

where  $\mathbf{\Lambda} = \text{diag}\{\lambda_i\}$  is the matrix of eigenvalues, with  $\lambda_i = \{\mathbf{v} \cdot \mathbf{n}, \mathbf{v} \cdot \mathbf{n}, \mathbf{v} \cdot \mathbf{n}, \mathbf{n} \cdot \mathbf{v} \cdot \mathbf{n} + c_s, \mathbf{v} \cdot \mathbf{n} - c_s\}$ , where  $c_s$  stand for the speed of sound. The eigenvalues are evaluated using Roe's averaging, as well as the matrix of left,  $\mathbf{T}^{-1}$ , and right,  $\mathbf{T}$ , eigenvectors, respectively. We point out that  $\mathbf{T}$  is the modal matrix that diagonalizes the matrix  $\mathbf{A}$ .

### 3.2.3 van Leer/Hanel flux

The van Leer/Hanel flux  $\mathbf{H}_H(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$  is of the flux splitting type. It is given by,

$$\mathbf{H}_H(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n})|_{\partial K} = \mathbf{F}_c^+(\mathbf{u}^+) + \mathbf{F}_c^-(\mathbf{u}^-) \quad \forall K \in \tau_h.$$

According to the modified form of the van Leer scheme [101] proposed by Hanel et al. [102] for multidimensional problems [103], the flux function  $\mathbf{F}_c^\pm(\mathbf{u})$  is defined by:

$$\mathbf{F}_c^\pm(\mathbf{u}) = \rho u_n^\pm \begin{pmatrix} 1 \\ H \\ u \\ v \end{pmatrix} + p^\pm \begin{pmatrix} 0 \\ 0 \\ n_x \\ n_y \end{pmatrix}$$

with

$$u_n^\pm = \begin{cases} \pm \frac{1}{4} c_s (M_n \pm 1)^2, & |M_n| \leq 1 \\ \frac{1}{2} (u_n + |u_n|), & |M_n| > 1, \end{cases}$$

$$p^\pm = \begin{cases} \pm \frac{1}{4} p (M_n \pm 1)^2 (2 \pm M_n), & |M_n| \leq 1 \\ \frac{p}{2u_n} (u_n + |u_n|), & |M_n| > 1, \end{cases}$$

where  $u_n = \mathbf{v} \cdot \mathbf{n}$ , with  $\mathbf{n}^T = (n_x, n_y)$ ,  $c_s$  denotes the speed of sound,  $M_n = u_n/c_s$ , is the Mach number normal to the interface, and  $H$  is the total enthalpy. The van Leer/Hanel numerical flux is diffusive but stable. It can be used for subsonic, transonic and supersonic flows.

### 3.3 Boundary Conditions

When  $\partial K$  belongs to  $\partial\Omega$ , boundary conditions are weakly prescribed through the numerical fluxes. This can be easily achieved by properly defining boundary states and their derivatives that enforce the appropriate boundary types and conditions. Many boundary conditions are implemented in the solver. For the RANS and Euler computations reported here, the following approaches are used.

- **Wall**

In the case of a viscous flow, the relative velocity between the surface and the fluid is assumed to be zero,  $\mathbf{v}^b = 0$ . A set of adiabatic boundary condition is employed and the state  $\mathbf{u}^b$  has the same density and internal energy of  $\mathbf{u}_h^+$ :

$$\mathbf{u}^b = (\rho^+, 0, 0, \rho E^+, 0, \rho^+ \omega_w)^T.$$

The prescribed smooth wall value  $\omega_w$ , is computed as proposed by Menter in [104], and it is related to the first cell-height  $2y_1$  according to the relation

$$\omega_w = \frac{6\nu}{\beta (\alpha_M y_1)^2}$$

where  $\nu = \mu/\rho$  is the kinematic viscosity,  $\alpha_M = \frac{1}{\sqrt{10}}$  and  $\beta$  is a real number depending on the polynomial degree.  $\nabla \mathbf{u}^b$  is computed by prescribing the internal gradient of all the primitive variables, except for the gradient of the internal energy, which is set to zero normal to the wall,  $\nabla e \cdot \mathbf{n} = 0$ . In the case of inviscid flows, there is no friction force and the velocity vector,  $\mathbf{v}^b = \mathbf{v}^+ - (\mathbf{v} \cdot \mathbf{n})^+ \mathbf{n}$ , ensures that the normal velocity component is zero on the boundary,  $(\mathbf{v} \cdot \mathbf{n})^b = 0$ .

- **Symmetry**

The symmetry boundary condition guarantee that the discretization on

the computational domain, called half domain, resembles the discretization on the full domain, defined as the half domain plus its mirror with respect to the symmetry boundary. The state  $\mathbf{u}^b$  has the opposite normal velocity component  $(\mathbf{v} \cdot \mathbf{n})^b = (\mathbf{v} \cdot \mathbf{n})^+$  and the other variables are prescribed based on the interior. In this way the non-permeability condition is satisfied and the mass flux computed by the Riemann solver is null. Also the gradient  $\nabla \mathbf{u}^b$  is modified.

- **Inflow / Outflow**

At subsonic inflow boundaries  $\mathbf{u}^b$  is computed by imposing total enthalpy, entropy, flow angles and the Riemann invariant associated to the outgoing characteristic. At subsonic outflow boundaries the pressure is taken from the outflow and  $\mathbf{u}^b$  is computed by imposing the outgoing Riemann invariants. At supersonic inflow  $\mathbf{u}^b$  is set to freestream values,  $\mathbf{u}^b = \mathbf{u}_\infty$  (Dirichlet boundary conditions), whereas for the supersonic outflow conditions, all the quantities are taken from the flow field,  $\mathbf{u}^b = \mathbf{u}^+$  (Neumann boundary conditions). For inflow/outflow boundaries the viscous boundary flux is computed by setting  $\nabla \mathbf{u}^b$  equal to the internal value,  $\nabla \mathbf{u}^b = \nabla \mathbf{u}^+$ .

### 3.4 Shock capturing

The key idea of the shock-capturing strategy adopted in this work consists of adding to the DG discretized equations an artificial viscosity term in order to control the high-order modes of the numerical solution within the elements, preserving as much as possible the spatial resolution of discontinuities. The shock-capturing term is local and applied element by element, adding an amount of artificial viscosity proportional to the inviscid residual. Thus, it is almost negligible except that near flow discontinuities. The shock-capturing

term added to Eqs.(3.3) or (3.4), reads:

$$\sum_{K \in T_h} \int_K \epsilon_p(\mathbf{u}_h^\pm, \mathbf{u}_h) (\nabla_h \mathbf{u}_h \cdot \mathbf{b}) (\nabla_h \mathbf{v}_h \cdot \mathbf{b}) dx$$

where the artificial viscosity coefficient  $\epsilon_p$  and the pressure gradient vector  $\mathbf{b}(\mathbf{u}_h)$  are given by

$$\epsilon_p(\mathbf{u}_h^\pm, \mathbf{u}_h) = Ch_K^2 \frac{|s_p(\mathbf{u}_h^\pm, \mathbf{u}_h)| + |d_p(\mathbf{u}_h)|}{p(\mathbf{u}_h)} f_p(\mathbf{u}_h), \quad (3.5)$$

$$\mathbf{b}(\mathbf{u}_h) = \frac{\nabla_h p(\mathbf{u}_h)}{|\nabla_h p(\mathbf{u}_h)| + \epsilon},$$

respectively, and

$$s_p(\mathbf{u}_h^\pm, \mathbf{u}_h) = \sum_i \frac{\partial p(\mathbf{u}_h)}{\partial \mathbf{u}_h^i} s_i(\mathbf{u}_h^\pm)$$

,

$$d_p(\mathbf{u}_h) = \sum_i \frac{\partial p(\mathbf{u}_h)}{\partial \mathbf{u}_h^i} (\nabla_h \cdot \mathbf{F}_c(\mathbf{u}_h))_i$$

.

The components  $s_i$  of the function  $\mathbf{s}$  defined by the solution of the problem

$$\int_K v_h^i s^i(\mathbf{u}_h^\pm) dx = \int_{\partial K} v_h^i (\mathbf{H}_i(\mathbf{u}_h^i, \mathbf{u}_h^-, \mathbf{n}) - \mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n}^-)_i dx$$

are actually the jump of the interface in normal direction between the numerical and internal inviscid flux components. The scalar term  $d_p(\mathbf{u}_h)$  is computed as sum of products of the convective flux divergence of cell for the component  $i$  and the partial derivative of the pressure with respect to the conservative variable  $\mathbf{u}_h^i$ . The pressure sensor  $f_p(\mathbf{u}_h)$  defined as

$$f_p(\mathbf{u}_h) = \frac{|\nabla_h p(\mathbf{u}_h)|}{p(\mathbf{u}_h)} \left( \frac{h_K}{k} \right),$$

preserves the accuracy of the solution within smooth regions and allows to use the same value of the user-defined parameter  $C$  for different degree

of polynomial approximation. Finally,  $k$  is the polynomial degree and  $h_K$  represents the reference dimension of the generic element  $K$ , given by

$$h_K = N \frac{\Omega_K}{S_K}$$

where  $\Omega_K$  and  $S_K$  denote the volume and the surface of  $K$ .

### 3.5 Time integration

The DG space discretized equations (3.3) and (3.4), both result in the following system of ordinary differential equations in time:

$$\mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = 0, \quad (3.6)$$

where  $\mathbf{M}$  denotes the global mass matrix, and  $\mathbf{U}$  and  $\mathbf{R}$  are the global vectors of degrees of freedom (dofs) and of residuals, respectively. Moreover, using a set of orthonormal basis functions, the  $\mathbf{M}$  matrix was reduced to the identity matrix.

Explicit and accurate time integration of Eq. (3.6) are usually efficiently performed by means of multi-stage schemes. It's well known that, the traditional Runge-Kutta schemes are easy to implement and parallelize but, due to the strong restriction to the size of the time step, for large-scale simulations and especially for high-order solutions the convergence speed slows down dramatically, resulting in inefficient time integration technique. Time-step restrictions can be avoided by using implicit schemes.

Many implicit or semi-implicit multistage (Rosenbrock and ESDIRK), multi-step (BDF) and multistage-multi-step (MEBDF) integration schemes have been implemented in the code and presented elsewhere [91, 92]. Here, we focus our attention on the five-step, fourth-stage, sixth-order accurate Two implicit Advanced Step Point (TIAS) scheme that is in the process of being thoroughly tested and validated. Further two different time-integration schemes are presented: SSPRK(5,4) and Backward-Euler. The former is the

standard five-stage, fourth-order accurate Strong Stability Preserving Runge-Kutta scheme [105]. The latter, used for steady state computations, is the single-stage first-order accurate implicit Euler method [106].

### 3.5.1 TIAS Scheme

The Two Implicit Advanced Step Point (TIAS) method belongs to the class of multi-step and multi-stage schemes, but with better stability properties than EBDF and MEBDF [107, 108] methods. In fact, TIAS are A-stable up to the sixth-order of accuracy, whereas EBDF or MEBDF are A-stable up to order 4. The TIAS method was originally presented in [88, 89] and its stability properties were investigated in detail in [109]. A first example of TIAS schemes applied to the discontinuous Galerkin discretized Euler and Navier-Stokes equations was presented by Nigro et al. in [90]. TIAS schemes involve four stages: the first three are predictor stages based on a standard  $k - step$  implicit Backward Differentiation Formulae (BDF) [110–112], the last one is a corrector stage that uses an advanced implicit  $k - step$  formula of order  $k + 1$ . We point out that since four non-linear stages have to be solved per time step the use of an efficient algorithm is required to make TIAS schemes competitive respect to explicit and other implicit schemes. In this respect, the multi-stage predictor-corrector structure of the method allows to obtain accurate starting solutions of the Newton iterations and to efficiently implement the schemes in a variable order/variable step algorithm, making TIAS of interest for complex applications where residual and Jacobian evaluations usually dominated the total computational cost. Assuming that approximate solutions  $\mathbf{U}_{n+1}$  have been calculated at  $t_{n+j}$  with  $0 \leq j \leq k - 1$ , the  $k - step$  TIAS algorithm of order  $k + 1$ , advances the solution in time successively solving the following four stages:

- Stage 1. Computation of the first predictor  $\bar{\mathbf{U}}_{n+k}$  of order  $k$  with a  $k - step$  BDF:

$$\mathbf{M}(\bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j \mathbf{U}_{n+j}) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}_{n+k}) = 0$$

- Stage 2. Computation of the second predictor  $\bar{\mathbf{U}}_{n+k+1}$  of order  $k$  with a  $k$ -step BDF:

$$\mathbf{M}(\bar{\mathbf{U}}_{n+k+1} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-2} \hat{\alpha}_j \mathbf{U}_{n+j+1}) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}_{n+k+1}) = 0$$

- Stage 3. Computation of the third predictor  $\bar{\mathbf{U}}_{n+k+2}$  of order  $k$  with a  $k$ -step BDF:

$$\mathbf{M}(\bar{\mathbf{U}}_{n+k+2} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}_{n+k+1} + \hat{\alpha}_{k-2} \bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-3} \hat{\alpha}_j \mathbf{U}_{n+j+2}) + \Delta t \hat{\beta}_k \mathbf{R}(\bar{\mathbf{U}}_{n+k+2}) = 0$$

- Stage 4. Computation of the corrected solution  $\mathbf{U}_{n+k}$  of order  $k+1$  using:

$$\mathbf{M}(\mathbf{U}_{n+k} + \sum_{j=0}^{k-1} \tilde{\alpha}_j \mathbf{U}_{n+j}) + \Delta t \left[ \tilde{\beta}_{k+2} \mathbf{R}(\bar{\mathbf{U}}_{n+k+2}) + \tilde{\beta}_{k+1} \mathbf{R}(\bar{\mathbf{U}}_{n+k+1}) + \beta_k \mathbf{R}(\bar{\mathbf{U}}_{n+k}) + (\tilde{\beta}_k - \beta_k) \mathbf{R}(\mathbf{U}_{n+k}) \right] = 0.$$

$\hat{\alpha}_j$  and  $\hat{\beta}_k$  are the BDF coefficients of the first three stages and  $\tilde{\alpha}_j$ ,  $\tilde{\beta}_{k+2}$ ,  $\tilde{\beta}_{k+1}$ ,  $\tilde{\beta}_k$  and  $\beta_k$  are the TIAS coefficients in the last stage. We remark that  $\tilde{\beta}_{k+2}$  and  $\beta_k$  are free coefficients determining the stability properties of the scheme, while the other coefficients, expressed in terms of  $\tilde{\beta}_{k+2}$ , are determined so that the scheme has order  $k+1$ . The residuals in stage 4 are computed once that each of the previous three stages has been solved:

$$\mathbf{R}(\bar{\mathbf{U}}_{n+k}) = -\frac{\mathbf{M}}{\Delta t \hat{\beta}_k} (\bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j \mathbf{U}_{n+j})$$

$$\mathbf{R}(\bar{\mathbf{U}}_{n+k+1}) = -\frac{\mathbf{M}}{\Delta t \hat{\beta}_k} (\bar{\mathbf{U}}_{n+k+1} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-2} \hat{\alpha}_j \mathbf{U}_{n+j+1})$$

$$\mathbf{R}(\bar{\mathbf{U}}_{n+k+2}) = -\frac{\mathbf{M}}{\Delta t \hat{\beta}_k} (\bar{\mathbf{U}}_{n+k+2} + \hat{\alpha}_{k-1} \bar{\mathbf{U}}_{n+k+1} + \hat{\alpha}_{k-2} \bar{\mathbf{U}}_{n+k} + \sum_{j=0}^{k-3} \hat{\alpha}_j \mathbf{U}_{n+j+2}).$$

For further details concerning TIAS coefficients and the numerical techniques used to improve the efficiency of the scheme please refer to [90].

### 3.5.2 SSPRK Scheme

Explicit and accurate solutions of the hyperbolic equations are also performed by means of high-order accurate multi-stage Runge Kutta schemes. The DG space discretized Euler equations (3.4) including the shock capturing terms, results

$$\mathbf{M}(\mathbf{U})\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) + \mathbf{D}(\mathbf{U})(\mathbf{U}) = 0, \quad (3.7)$$

where the discretized shock-capturing term has been written as the product of the block diagonal matrix  $\mathbf{D}(\mathbf{U})$  times the vector of solution *DOFs*. Each block of  $\mathbf{D}$  couples the elemental  $\mathbf{U}$  components and within the element such blocks are equal to each other. The shock-capturing term is a diffusion-like term, typically subject to wild variations even within one element and that can attain large values in very few elements of the computational grid. Seeking to avoid any further time step restriction, it is advisable to employ some degree of implicitness for the time integration of this term. Therefore Eq.(3.7) is rewritten as

$$\mathbf{M}(\mathbf{U})\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) + \mathbf{D}(\mathbf{U})(\mathbf{U} + d\mathbf{U}) = 0 \quad (3.8)$$

and this leads to

$$\frac{d\mathbf{U}}{dt} + \tilde{\mathbf{R}}(\mathbf{U}) = 0 \quad (3.9)$$

where

$$\tilde{\mathbf{R}}(\mathbf{U}) = [\mathbf{M}(\mathbf{U}) + \mathbf{D}(\mathbf{U})dt]^{-1}[\mathbf{R}(\mathbf{U}) + \mathbf{D}(\mathbf{U})\mathbf{U}].$$

In multi-stage Runge-Kutta schemes the solution is advanced in time in several stages and the residual is evaluated at intermediate states.

The solution of the semi-discrete system in Eq.(3.9) is advanced from time

$t^n$  to time  $t^{n+1}$  applying the following expression:

$$\begin{aligned} \mathbf{U}^0 &= \mathbf{U}^n, \\ \mathbf{U}^i &= \mathbf{U}^0 - \Delta t \sum_{k=0}^{i-1} c_{i,k} \tilde{\mathbf{R}}(\mathbf{U}^k), \quad i = 1, 2, \dots, s, \\ \mathbf{U}^{n+1} &= \mathbf{U}^s, \end{aligned} \quad (3.10)$$

where  $i$  is the stage counter and  $c_{i,k}$  are the coefficients of the  $i$ th-stage. The  $c_{i,k}$  coefficients are related to the usual  $a_{i,k}$  and  $b_k$  Butcher coefficients by the following expressions:

$$\begin{aligned} a_{i,k} &= c_{i-1,k-1} \quad k = 1, \dots, i-1; \quad i = 2, \dots, s, \\ b_k &= c_{s,k-1} \quad k = 1, \dots, s. \end{aligned}$$

The Runge-Kutta scheme employed in this work is the five stage, fourth-order accurate Strong Stability Preserving Runge Kutta scheme: SSPRK(5,4). For further details about the coefficients of the optimal SSPRK(5,4) scheme refer to [105].

The local time step  $\Delta t$  on each element  $K$  is computed taking into account the *CFL* stability condition:

$$\Delta t = \frac{\sigma}{2p+1} \cdot \frac{|K|}{\Lambda_c^x + \Lambda_c^y}, \quad (3.11)$$

where  $p$  denotes the polynomial degree of the spatial discretization [113]. Moreover  $\sigma$  is a factor introduced to take into account that SSP can be more efficient than TVD Runge-Kutta schemes. Finally, the convective spectral radii are defined as:

$$\begin{aligned} \Lambda_c^x &= (|\bar{u}| + \bar{c}_x) \Delta S^x, \\ \Lambda_c^y &= (|\bar{v}| + \bar{c}_y) \Delta S^y. \end{aligned}$$

The variables  $\Delta S^x$  and  $\Delta S^y$  are the projections of the elemental volume,  $|K|$ , onto the  $x$  and  $y$  axis, respectively, whereas  $|\bar{u}|$ ,  $\bar{c}_x$  and  $|\bar{v}|$ ,  $\bar{c}_y$  are obtained using the mean values of the flow quantities on each element  $K$ .

### 3.5.3 Implicit Backward Euler Scheme

An implicit time discretization is used to compute steady state solutions of the RANS  $k - \omega$  equations. The backward Euler time discretization of Eq.(3.6) can be written as

$$\left[ \frac{\mathbf{M}}{\Delta t} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}} \right] \Delta \mathbf{U}^n = -\mathbf{R}^n, \quad (3.12)$$

where  $\Delta \mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n$ ,  $\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}}$  is the Jacobian matrix of the DG space discretization and  $\left[ \frac{\mathbf{M}}{\Delta t} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}} \right]$  denotes the global system matrix. The matrix  $\left[ \frac{\mathbf{M}}{\Delta t} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}} \right]$  can be regarded as an  $N_K \times N_K$  block sparse matrix where  $N_K$  is the number of elements in  $\tau_h$  and the rank of each block is  $M \times N_{dof}^K$ , where  $N_{dof}^K$  is the number of dofs for each of the  $M$  conservative variables in the generic element  $K$ . The Jacobian matrix of the DG discretization has been computed analytically without any approximation and, using very large time steps, the method can therefore achieve quadratic convergence in the computation of steady state solutions. In the limit  $\Delta t \rightarrow \infty$  Eq. (3.12) is in reality identical to one iteration of the Newton method applied to the steady discrete problem.

Finally, we underline that GMRES (Generalized Minimal RESidual) algorithm [114] has been used to solve the linear systems resulting from the implicit time integrations at each time step. For this purpose, we usually employ the block Jacobi method with one block per process, each of which is solved with ILU(0), or the Additive Schwarz Method (ASM). In the code we rely on PETSc, [115], for the linear solvers. PETSc is the software upon which the code relies for the purpose of parallelization. The SPDM (Single Process Multiple Data) strategy is based on grid partitioning accomplished by means of the METIS package [116].

## 3.6 Gradient based design optimization

Gradient based design optimization is the most widely used approach to optimization, and also the oldest. This method is based on the evaluation of the gradient of the objective function with respect to the design parameters. For complex models and discretizations, gradient computation is often the bottleneck in the optimization process. Different adjoint approaches have been developed in order to compute the gradients.

In this work we focus our attention on the Automatic Differentiation (AD) technique. In the next sections we describe the optimization algorithms used in this work and we present AD more in details.

### 3.6.1 Overview of the optimization strategy

Mathematically speaking, an optimization problem consists in determining the minimum or maximum value of a function subject to constraints on its variables [71]. We use the following notation:

- $\boldsymbol{\alpha}$  is the vector of variables, also called unknowns or design parameters;
- $J(\boldsymbol{\alpha})$  is the objective function that we want to minimize;

Beginning from the starting vector  $\boldsymbol{\alpha}_0$ , the iterative optimization algorithm generates a sequence of solutions  $\boldsymbol{\alpha}_k$ , where  $k$  denotes the number of optimization iterations, and terminate when either no more progress can be made or when it find the minimum/maximum value of the objective function approximated with a sufficient level of accuracy. In this work, we employ the *line search* method for moving from the current vector  $\boldsymbol{\alpha}_k$  to a new one  $\boldsymbol{\alpha}_{k+1}$ .

#### Line search method

Using the line search approach the optimization algorithm searches the new values  $\boldsymbol{\alpha}_{k+1}$  with a lower value of  $J$  moving from  $\boldsymbol{\alpha}_k$  along a chosen direction  $\mathbf{p}_k$ . Each iteration of a line search method can be expressed as follows:

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \text{step}_k \mathbf{p}_k,$$

where the positive scalar  $\text{step}_k$  is the distance from one point to another along the search direction  $\mathbf{p}_k$ .

The step length can be calculated by solving, exactly (exact line search) or approximately (inexact line search), the one-dimensional problem:

$$\min J(\boldsymbol{\alpha}_k + \text{step}_k \mathbf{p}_k) \text{ with } \text{step}_k > 0.$$

The descent direction  $\mathbf{p}_k$  can be computed using various strategies, and can be generally expressed as:

$$\mathbf{p}_k = B_k^{-1} \nabla J_k$$

in which  $B_k$  is a symmetric and non-singular matrix. Depending on the type of the  $B_k$  matrix, we can distinguish different kind of methods. In the steepest descent method  $B_k$  is the identity matrix  $I$ , in the Newton's method  $B_k$  is the Hessian  $\nabla^2 J(\boldsymbol{\alpha}_k)$ , whereas in Quasi Newton methods  $B_k$  is an appropriate approximation of the Hessian.

In this work we focus our attention on the inexact line search method for the computation of the step length  $\text{step}_k$  and we use two different approaches to compute the line search direction  $\mathbf{p}_k$ , the Steepest Descent method and the BFGS Quasi-Newton method.

- **Inexact line search**

Inexact line search algorithms try out a sequence of candidate values for  $\text{step}_k$ , accepting one of these values when certain conditions are satisfied. Several sophisticated line search algorithms are available, but the simplest one is obtained imposing a reduction in  $J$  such that,

$$J(\boldsymbol{\alpha}_k + \text{step}_k \mathbf{p}_k) < J(\boldsymbol{\alpha}_k).$$

We use this simple criterion coupled to the steepest descent method, and a more sophisticate criterion coupled with the BFGS. It is based on the fulfillment of the following sufficient decrease condition:

$$J(\boldsymbol{\alpha}_k + \text{step}_k \mathbf{p}_k) \leq J(\boldsymbol{\alpha}_k) + c_1 \text{step}_k \nabla J_k^T \mathbf{p}_k$$

in which  $c_1$  stands for a constant  $c_1 \in (0, 1)$ , usually chosen to be quite small, a typical value is  $c_1 = 10^{-4}$ . In practice, enforcing the sufficient decrease conditions, the reduction in  $J$  should be proportional to both the step length  $\text{step}_k$  and the directional derivative  $\nabla J_k^T \mathbf{p}_k$ .

- **Steepest descent method**

The key idea of steepest descent is moving from  $\boldsymbol{\alpha}_k$  to  $\boldsymbol{\alpha}_{k+1}$ , following the direction for which  $J$  decreases most rapidly. In particular, steepest descent method moves along:

$$\mathbf{p}_k = -\nabla J_k$$

The steepest descent direction  $-\nabla J_k$  is the most basic approach, and the algorithm will eventually converge where the gradient is zero. The main advantage of the steepest descent method is obviously its simplicity, related to the fact that it requires only the calculation of the gradient  $\nabla J_k$ , but not of the second derivatives. The main drawback is that a dramatic slowdown of convergence may occur, especially in presence of complex problems. For a more detailed discussion about the property of the steepest descent method, concerning its rate of convergence and so on, please refer to [71].

- **Quasi Newton method - BFGS method**

Quasi Newton methods, like steepest descent method, only require the gradient of the objective function to be supplied at each iteration. Evaluating the changes in gradients step by step, they construct a model based on an approximation  $B_k$  of the Hessian  $\nabla^2 J(\boldsymbol{\alpha})$ , that is good enough to produce significant efficiency improvement with respect to the steepest descent method. The most popular quasi-Newton algorithm is the BFGS method, developed by Broyden [117], Fletcher [45], Goldfarb [118], and Shanno [119] in 1970, and well known for its robustness, self-correcting

properties, as well as for its superlinear rate of convergence. For a complete discussion about the BFGS method please refer to [71]. The search direction with BFGS approach is computed as follows:

$$\mathbf{p}_k = H_k^{-1} \nabla J_k ,$$

Here  $H_k^{-1}$  is a symmetric positive definite matrix that approximate the inverse of the Hessian.  $H_k^{-1}$  is updated at every iteration by means of:

$$H_{k+1} = (I - \rho_k \mathbf{s}_k \mathbf{y}_k^T) H_k (I - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T ,$$

$$\text{with } \mathbf{s}_k = \boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k, \quad \mathbf{y}_k = \nabla J_{k+1} - \nabla J_k ,$$

$$\text{and } \rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} .$$

The inverse of the Hessian at the first iteration,  $H_0$ , is approximated by setting it to be the identity matrix,  $I$ .

### 3.6.2 Evaluation of the Gradient

The evaluation of the gradient of the objective function plays a key role in order to achieve an efficient and accurate gradient based optimization. An attractive alternative to finite difference (FD) method to perform differentiation, is Automatic Differentiation AD [72, 73]. AD is an upcoming technology which provides software for automatic computation of derivatives of a general function provided by the user. In this work we use AD through the Tapenade code [93, 94], while the FD approach is used to carry out the validation of the sensitivity computations computed by AD.

#### Automatic Differentiation

The FD methods are very time-consuming and the choice of the perturbation step greatly affect the accuracy in the calculation of derivatives. A very small step leads to the problem of round-off errors, and a too large step will gives an erroneous value of the derivative [120, 121]. Moreover, the grid topology and a grid refinement could greatly affect the accuracy of the results [122].

AD offers a way to overcome the drawbacks of the FD approach, allowing to obtain accurate sensitivity computations in a more efficient way, especially in presence of several design variables.

More specifically, the Automatic Differentiation is a technique that applies symbolic differentiation to computer programs using the chain rule. Using AD, an augmented computer code is generated from a given computer code, which calculates the desired sensitivity derivatives. As a result, AD does not incur in any truncation error, and the results are exact to machine accuracy. The principles of AD are explained in detail in the work of Griewank [73].

There are two basic modes of automatic differentiation: the *forward* and *reverse* modes. In this work we apply AD in *forward* mode, also known as *tangent* mode. If we suppose that we want to differentiate a function  $J$  with the input vector  $\boldsymbol{\alpha}$  and the output vector  $\mathbf{Y}$  such that  $J(\boldsymbol{\alpha}) = \mathbf{Y}$  with  $\boldsymbol{\alpha} \in R^n$  and  $\mathbf{Y} \in R^m$ , the forward mode gives the Jacobian vector product, computing the directional derivative  $\dot{\mathbf{Y}} = \mathbf{D} \cdot \dot{\boldsymbol{\alpha}}$  for each direction  $\dot{\boldsymbol{\alpha}}$ , that is:

$$\dot{\mathbf{Y}} = \begin{pmatrix} \dot{Y}_1 \\ \dot{Y}_2 \\ \vdots \\ \dot{Y}_m \end{pmatrix} = \begin{pmatrix} \frac{\partial Y_1}{\partial \alpha_1} & \frac{\partial Y_1}{\partial \alpha_2} & \cdots & \frac{\partial Y_1}{\partial \alpha_n} \\ \frac{\partial Y_2}{\partial \alpha_1} & \frac{\partial Y_2}{\partial \alpha_2} & \cdots & \frac{\partial Y_2}{\partial \alpha_n} \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial Y_m}{\partial \alpha_1} & \frac{\partial Y_m}{\partial \alpha_2} & \cdots & \frac{\partial Y_m}{\partial \alpha_n} \end{pmatrix} \cdot \begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \vdots \\ \dot{\alpha}_n \end{pmatrix} = \mathbf{D} \cdot \dot{\boldsymbol{\alpha}}$$

It is evident that, the forward mode gives the derivatives only in one direction  $\dot{\alpha}_i$ , then we may compute the sensitivities of the cost function with respect to a single design variable at once.

$$\dot{\mathbf{Y}} = \left( \frac{\partial Y_1}{\partial \alpha_i} \quad \frac{\partial Y_2}{\partial \alpha_i} \quad \cdots \quad \frac{\partial Y_m}{\partial \alpha_i} \right)^T \dot{\alpha}_i$$

In fact, in case of  $m = 1$ , i.e. only scalar output, the forward mode returns

only one single value  $\dot{Y}_1 = \frac{\partial Y_1}{\partial \alpha_i}$  which is the derivative with respect to only one independent variable, or design variable,  $\alpha_i$  if one sets  $\dot{\alpha}_i = 1$ . In presence of  $n$  different design variables, the forward mode has to be executed  $n$  times in order to compute all the  $n$  elements of the complete gradient vector. Therefore the forward mode is suitable to evaluate sensitivities only for problems that are characterized by a few number of design variables.

In this work differentiation of the compressible DG code has been carried out by the AD tool TAPENADE 3.6. We point out that the differentiation of the MPI calls involved in the evaluation of the sensitivities, has been carried out by hand and that AD results have been validated by the finite difference method.

### 3.7 Optimization of shock capturing parameters

The gradient based algorithm applied to control spurious oscillations impacts on the amount of artificial viscosity through two design parameters,  $\alpha_1$  and  $\alpha_2$ , locally adjusting the dissipative contributions based on i) the interface jumps between the numerical and internal inviscid fluxes,  $s_p$ , and ii) the convective flux divergence within the elements,  $d_p$ . The artificial viscosity coefficient  $\epsilon_p$  of Eq.(3.5) has been then modified as follows,

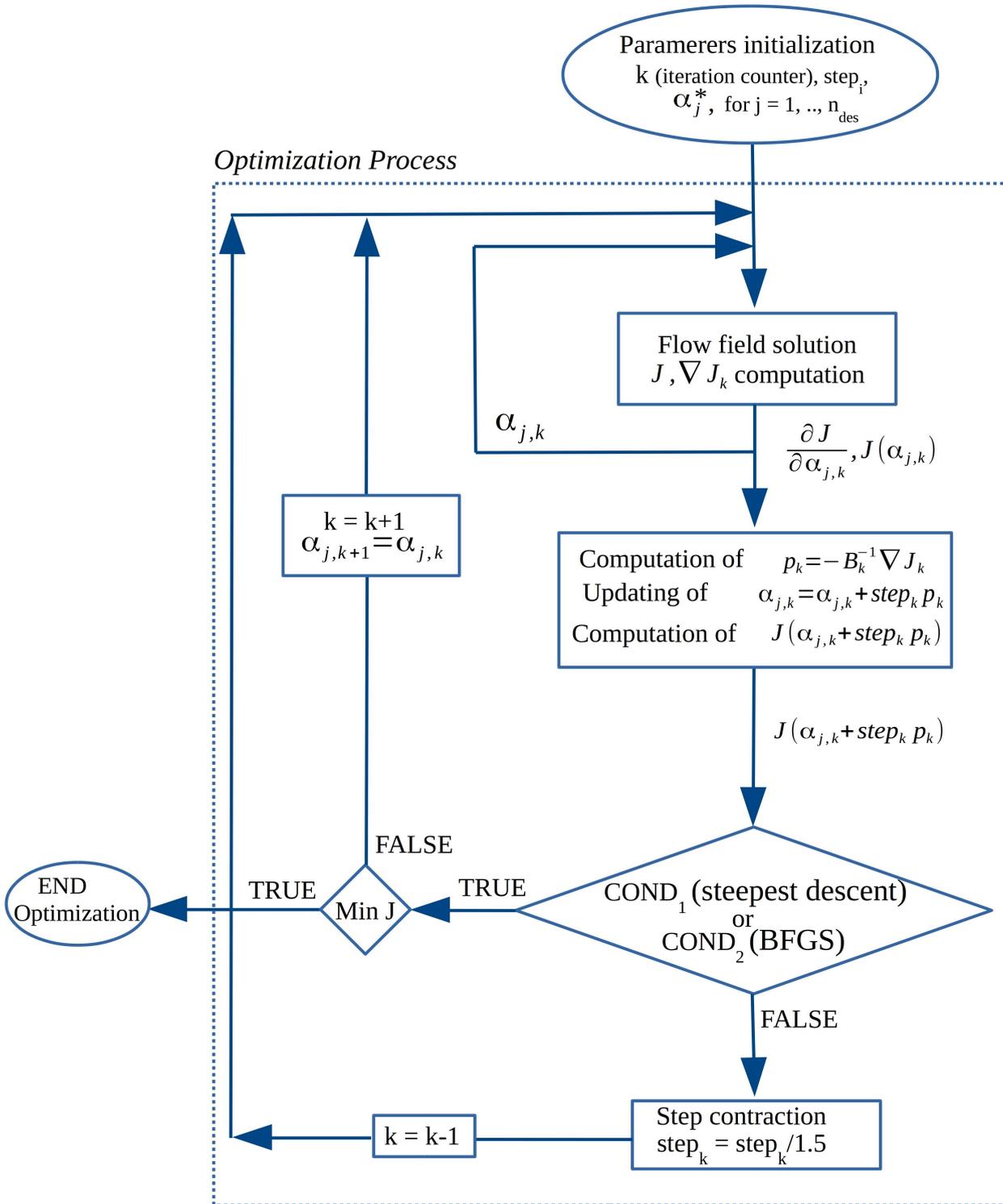
$$\epsilon_p(\mathbf{u}_h^\pm, \mathbf{u}_h) = Ch_k^2 \frac{\alpha_1 |s_p(\mathbf{u}_h^\pm, \mathbf{u}_h)| + \alpha_2 |d_p(\mathbf{u}_h)|}{p(\mathbf{u}_h)} f_p(\mathbf{u}_h).$$

where the design parameters  $\alpha_1$  and  $\alpha_2$  are the weighting factors of the  $|s_p(\mathbf{u}_h^\pm, \mathbf{u}_h)|$  and  $|d_p(\mathbf{u}_h)|$  terms, respectively.

The goal of the optimization procedure is to find the optimal values of  $\alpha_1$  and  $\alpha_2$  parameters, in order to minimize the scalar function  $J$ . For the computation presented in this work  $J$  has been set equal to the  $L_2$  norm of the

error between the exact and the numerical pressure field. The implemented procedure employs two different algorithms, steepest descent method and the BFGS method, for the computation of the line search  $\mathbf{p}_k$ , and the Automatic Differentiation in *forward mode*, for the computation of the gradients. The optimization algorithm is iterative. The sequence of operations in the optimization procedure is given in Fig. 3.2.

To start the iteration process initial guesses for the design parameters,  $\alpha_j^*$ , and an initial step size,  $step_i$ , are given. Then the flow field and the cost function with its gradient are computed. The gradient vector is evaluated each component at time, by calling the AD in forward mode according to the number of design variables,  $n_{des}$ . The values of  $\frac{\partial J}{\partial \alpha_{j,k}}$  are then used to compute the search direction  $\mathbf{p}_k$  and to update  $\alpha_{j,k}$  and  $J$ . The next operation is to verify that  $J(\alpha_{j,k} + step_k \mathbf{p}_k)$  satisfies the steepest descent/BFGS decrease condition. If this condition is fulfilled the updated values of  $\alpha_{j,k}$  are used in the next iteration step. Otherwise, a contraction of the step size,  $step_k$ , is imposed, and the procedure restarts from the iteration  $k - 1$ . The process continues until convergence of the minimum of  $J$ .



$$\text{COND}_1 \rightarrow J(\alpha_k + \text{step}_k p_k) < J(\alpha_k)$$

$$\text{COND}_2 \rightarrow J(\alpha_k + \text{step}_k p_k) \leq J(\alpha_k) + c_1 \text{step}_k \nabla J_k^T p_k$$

Figure 3.2: scheme of the optimization process

# Chapter 4

## Results

The purpose of this section is to present the results of two internal turbulent flow computations, to evaluate the performance of the TIAS-DG scheme considered in Section 3.5.1 and to validate the proposed high-order shock-capturing scheme presented in Section 3.7.

The turbulent DG computations are compared to experiments in two steady state flow rigs. The first configuration, called Dellenbacks configuration, is solved to reproduce the first difficulty occurring in internal combustion engines: the strong separation due to the increase of the diameter between the intake ports and the cylinder. The second more complex configuration, a simple engine head geometry with one valve creating strong separations and a strong pressure gradient, has been computed to simulate the flow entering the cylinder.

A canonical test case for the evaluation of LES/DES capabilities of high-order methods has been used to compare the efficiency of the sixth-order accurate TIAS scheme with that of the fourth-order accurate SSPRK explicit scheme.

Finally, a supersonic flow past a wedge has been computed to evaluate the feasibility of the proposed gradient based design optimization approach for the optimal control of spurious numerical oscillations.

## 4.1 Internal turbulent flows

### 4.1.1 Axisymmetric sudden expansion

#### Description of the test-case

The axisymmetric sudden expansion analyzed [123] has an expansion ratio approximately of 0.5. The flow conditions are without swirl with a Reynolds number of  $Re = 30000$ , based on the diameter and on the flow quantities at the inlet section of the duct.

This test-case aims at evaluating the potential of the DG approach for internal turbulent flows in the presence of flow separation that occurs in many industrial applications such as the intake systems of ICE, through nacelle in a crosswind and in many others applications.

Although the experiments have been carried out using water, the computations have been performed with a compressible DG solver at the Mach number of 0.05 and the same Reynolds number of the experiment. The low compressibility of the present flow conditions has a very small impact on the accuracy of the numerical results. The computations have been performed, starting from an uniform flow field, with a sequence of polynomial approximations up to the sixth order of accuracy. The computational domain starts two diameters  $D$  upstream the abrupt expansion and it ends fifteen diameters downstream. Owing to the axisymmetric nature of the problem, we use a two-dimensional grid composed by 2400 quadrangular cells (Fig.4.1). The origin of the  $z$  axis ( $r = 0$ ) is located at the inflow of the duct and the change of section is placed at  $z = 0.1$ . The diameters upstream and downstream of the expansion are  $D = 0.05078\text{m}$  and  $D_2 = 0.0985\text{m}$ , respectively. In Fig.4.1 the considered measurement stations based on the ratio  $z/D$  are shown.

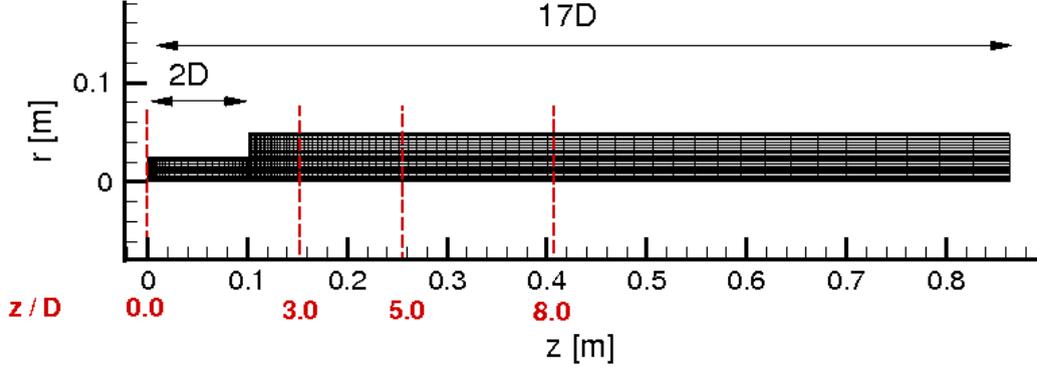


Figure 4.1: Computational grid for the abrupt axisymmetric expansion test-case, including measurement stations.

Concerning the boundary conditions, at the inlet, the velocity profile has been taken from the experimental measurements and then converted into an entropy profile by assuming a constant static pressure and with a given stagnation enthalpy so that the implementation of the inlet boundary condition has been carried out in the manner described in [124]. Moreover, the values of turbulence intensity,  $I_t$ , and turbulent viscosity ratio,  $\frac{\mu_t}{\mu}$ , assigned at the inlet are  $I_t = 4\%$  and  $\frac{\mu_t}{\mu} = 50$ , respectively. Conversely, at the outlet boundary, only the pressure has been specified in order to obtain a Mach number of 0.05 at the inlet, whereas entropy, total enthalpy and flow angles have been extrapolated from the interior. At the wall a zero heat flux (adiabatic) no-slip boundary condition has been imposed. The wall boundary condition  $\hat{\omega}_w$  has been set as proposed by Menter in [104]. The prescribed smooth wall value  $\omega_w$  is related to the first cell-height  $2y_1$  according to the relation

$$\omega_w = \frac{6\nu}{\beta(\alpha_M y_1)},$$

where  $\alpha_M = \frac{1}{\sqrt{10}}$  and  $\beta$  is a real number depending on the polynomial degree.

To assess the performances of the proposed DG method in terms of accuracy we perform both a qualitative and quantitative analysis of the numerical results. For the former analysis we present the contour plots of pressure in the channel and the streamtraces at the recirculation region close to the abrupt

expansion, for the latter we compare the numerical axial velocity profiles with the Dellenbacks experimental data. Finally, the skin friction  $C_f$  distribution along the wall is shown to determine the separation point.

## Results

In this section we present the results obtained by the DG code and their validation against the experimental data. Fig. 4.2 shows the behaviour of the  $y^+$  for the first cell-centroid  $y_1$  at the sections given in Fig. 4.9 for different polynomial degrees. The distance of the first grid line parallel to the wall of the duct is equal to  $2y_1 = 0.0015$ . As expected the highest  $y^+$  values have been computed at the inlet where a polynomial approximation dependency of  $y^+$  is evident. Conversely, after the sudden expansion, the  $y^+$  values reduce almost independently of the spatial discretization.

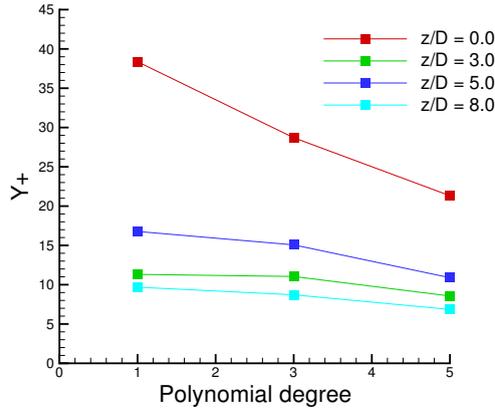


Figure 4.2:  $y^+$  values for the first cell-centroid  $y_1$  as a function of different polynomial degrees at the considered measurement stations.

The plots in Fig. 4.3 show the pressure contours for  $P_1$ ,  $P_3$  and  $P_5$  elements. Such plots point out qualitatively the prediction capability of the DG method for different polynomial approximations and how an increase in the polynomial degree leads to a significant improvement in the characterization of the pressure field.

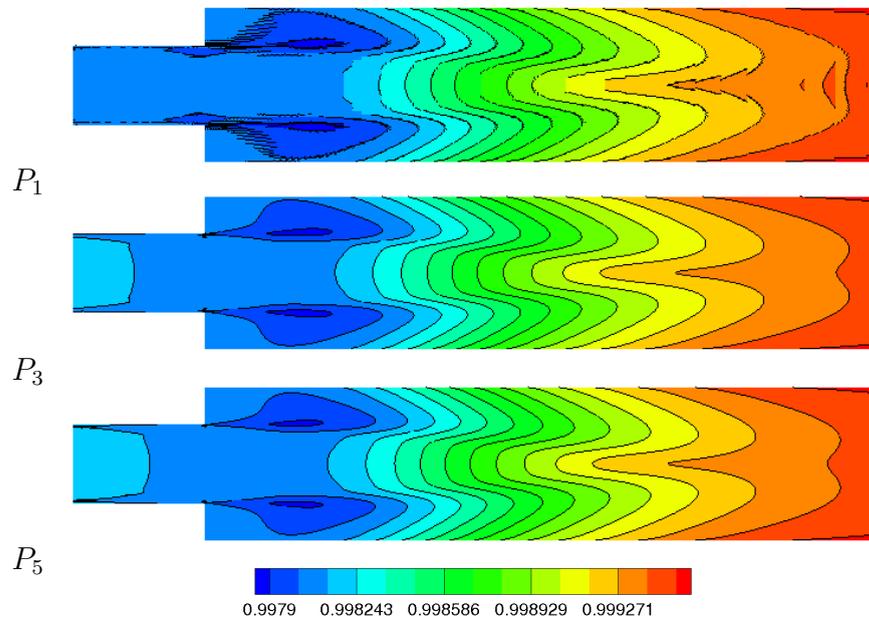


Figure 4.3: Contours of pressure obtained using  $P_1$  (top row),  $P_3$  (middle row) and  $P_5$  (bottom row) elements.

The effects of a high-order discretization on the solution accuracy are more evident in Fig. 4.4 that shows the streamlines close to the corner for different polynomial degrees. Notice that the lower order  $P_1$  solution fails to capture the presence of recirculating regions close to the corner of the wall, whereas the  $P_3$  solution predicts the presence of two contra-rotating vortices. Moreover, a further increase in the polynomial degree, from  $P_3$  to  $P_5$ , only slightly affects the streamlines pattern.

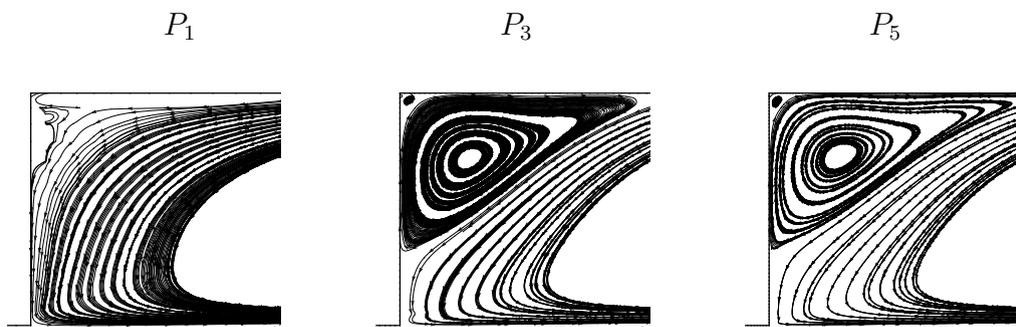


Figure 4.4: Snapshots of streamtraces close to the corner of the wall, for  $P_1$  (left column),  $P_3$  (middle column) and  $P_5$  (right column) elements.

The quantitative assessment of the numerical results has been carried out by comparing the axial velocities measured inside the duct along the sections indicated in Fig. 4.9. The numerical and experimental comparison is reported in Fig. 4.5 as a function of the radial distance from the axis, for different polynomial degrees.

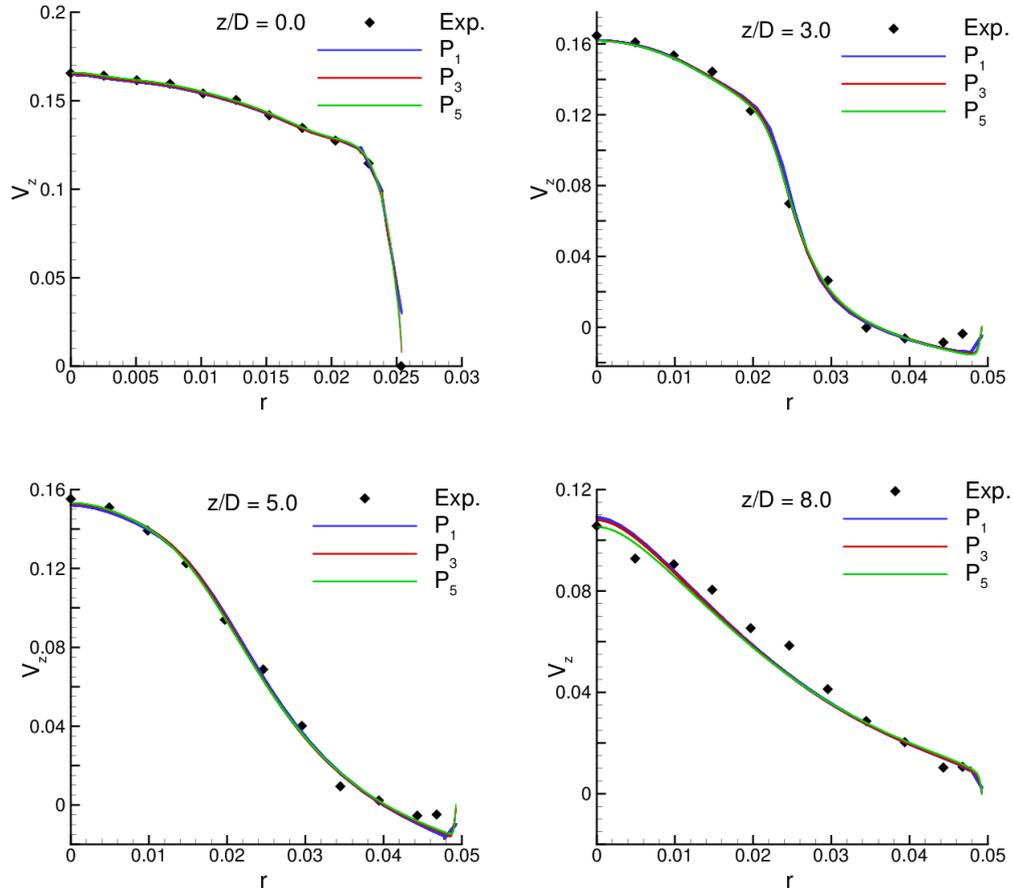


Figure 4.5: DG axial velocity profiles compared with the experimental data at the considered measurement stations:  $z/D = 0.0$  and  $z/D = 3.0$  (top row),  $z/D = 5.0$  and  $z/D = 8.0$  (bottom row).

We observe that even using  $P_1$  elements the numerical profiles are in good agreement with the experimental ones. Nevertheless, while the higher accuracy only marginally affects the computation of the axial velocity at certain distance from the axis, the higher degrees of approximation allow a more accurate prediction of the flow close to the axis and to the wall, as shown in Fig. 4.6.

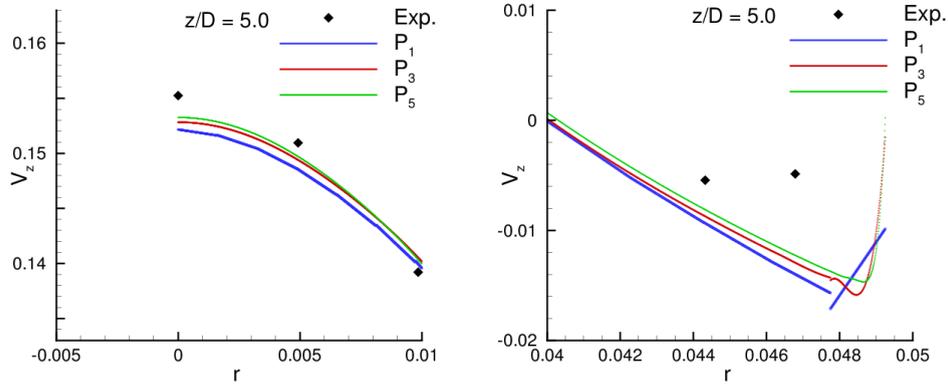


Figure 4.6: DG axial velocity profiles compared with the experimental data at  $z/D = 5.0$ , close-up near the axis (left column) and the wall (right column).

In Fig. 4.7 the skin friction  $C_f$  distribution along the wall of the duct is shown for elements ranging from  $P_1$  to  $P_5$ . The plot shows that  $P_1$  and  $P_2$  solutions suffer from a lack of accuracy, whereas the higher order approximations converge towards the same solution. This result suggests that the higher order solutions resolve boundary layers and recirculating regions more accurately. Finally, the reattachment length, approximately equal to  $0.36m$ , is almost independent on the order of accuracy and slightly overestimated respect to the experimental value of  $0.3218m$ .

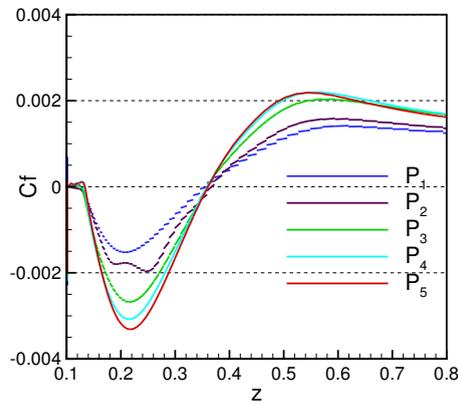


Figure 4.7: Skin friction coefficient  $C_f$  along the wall of the duct, for  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  and  $P_5$  elements.

## 4.1.2 Axisymmetric sudden expansion with a valve

### Description of the test-case

This test-case aims at evaluating the potential of the DG approach for the resolution of internal turbulent flows in the presence of a more complex geometry, closer to an ICE configuration, through the comparison with experimental data. The experimental analysis described in [125] was carried out by means of a steady-flow rig, enabling fluid flow to be forced through the engine head, while the valve lift was fixed to the selected values of 10 mm. Fig.4.8 shows the engine head while Table 4.1 lists its main characteristics.

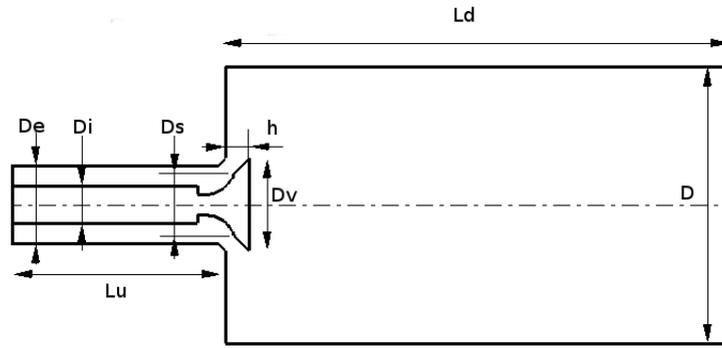


Figure 4.8: Detailed view of the geometry of the engine head

Description	[mm]
<i>Internal diameter, <math>D_i</math></i>	17
<i>External diameter, <math>D_e</math></i>	34
<i>Cylinder diameter, <math>D</math></i>	120
<i>Seat diameter, <math>D_s</math></i>	27.6
<i>Downstream pipe length, <math>L_d</math></i>	960
<i>Upstream pipe length, <math>L_u</math></i>	132
<i>Valve lift, <math>h</math></i>	10
<i>Inner seat diameter, <math>D_v</math></i>	40

Table 4.1: Engine head geometry

The axisymmetric sudden expansion with a valve has an expansion ratio of 3.5 and it is the same analyzed by Thobois et al. in [126]. Although the

experimental setup does not impose swirl, a small swirl appears in the velocity measurements. We remark that unpublished measures obtained in addition to those presented in the work of Graftieaux et al. [125], were graciously provided by the authors. In particular it's worth to point out that the experimental data have been registered starting from a section placed at  $90mm$  upstream the valve. As for the Dellenback's test-case, the flow conditions refer to a Reynolds number of  $Re = 30000$ , based on the diameter and the flow quantities at the outlet section of the engine head. The computational domain consists of an intake duct fourteen inlet equivalent diameter long and the engine cylinder with the valve. The cylinder has been extended up to  $8D$  to ensure uniform pressure at the exit section.

The computational grid has been obtained by Gmesh [127], and is composed by 6309 quadrangular elements, with a local grid refinement close to the valve and to the wall. The distance of the first grid line parallel to the wall of the duct is equal to  $2y_1 = 0.0005mm$ . In the exhaust duct the continuous coarsening of the mesh significantly reduces the impact of spurious reflections at the outlet section on the solution accuracy. Figure 4.9 shows the computational grid and the four measurement stations used for the accuracy analysis located at  $-90mm$ ,  $-27mm$ ,  $20mm$ ,  $70mm$  from the cylinder head, respectively.

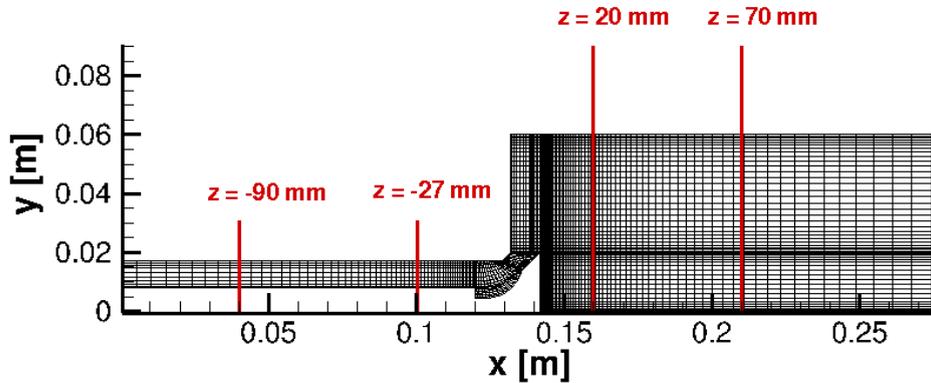


Figure 4.9: Computational grid upstream and downstream the valve, including measurement stations.

The axisymmetric computations have been performed at the inlet Mach-

number of 0.2 and at the same Reynolds number and valve lift of the rig experiments. At the inlet uniform profiles of total enthalpy and total entropy have been specified, with the values of turbulence intensity,  $I_t$ , and turbulent viscosity ratio,  $\frac{\mu_t}{\mu}$ , of  $I_t = 0.5\%$  and  $\frac{\mu_t}{\mu} = 1$ , respectively. At the outlet boundary only the pressure has been specified, to obtain a Mach number of about 0.2 at the inlet. At the wall a zero heat flux (adiabatic) no-slip boundary condition has been imposed, following the approach proposed by Menter in [104].

## Results

In this section we present some results used to evaluate the capability of the proposed high-order DG solver in predicting the steady flow through an intake valve of a reciprocating engine and the resulting flow structures within the cylinder. The accuracy of the converged solutions is analyzed both qualitatively and quantitatively. First, mean flow contours are shown for a qualitative comparison. Then, for the quantitative analysis, the non-dimensional velocity and turbulent kinetic energy distributions along different sections of the engine head are compared with the corresponding experimental data.

The flow structure inside the engine head is shown in Fig.4.10 in terms of the streamlines of the  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$  solutions. In all cases the speed flow around the intake valve forms a wall jet along the cylinder surface moving towards the exit section. Major recirculation regions occur behind the valve and in the corner; a small separation forms at the valve throat. While these results seem to indicate that accuracy higher than second-order would not be necessary, as very similar streamlines are obtained, a closer examination of local features of the flow field suggests the opposite conclusion. In fact, Fig.4.11, shows that using the higher order accurate solutions the small corner vortex and the small separation regions that occur at the valve throat and close to the engine head, respectively, are significantly better resolved than those of lower-order.

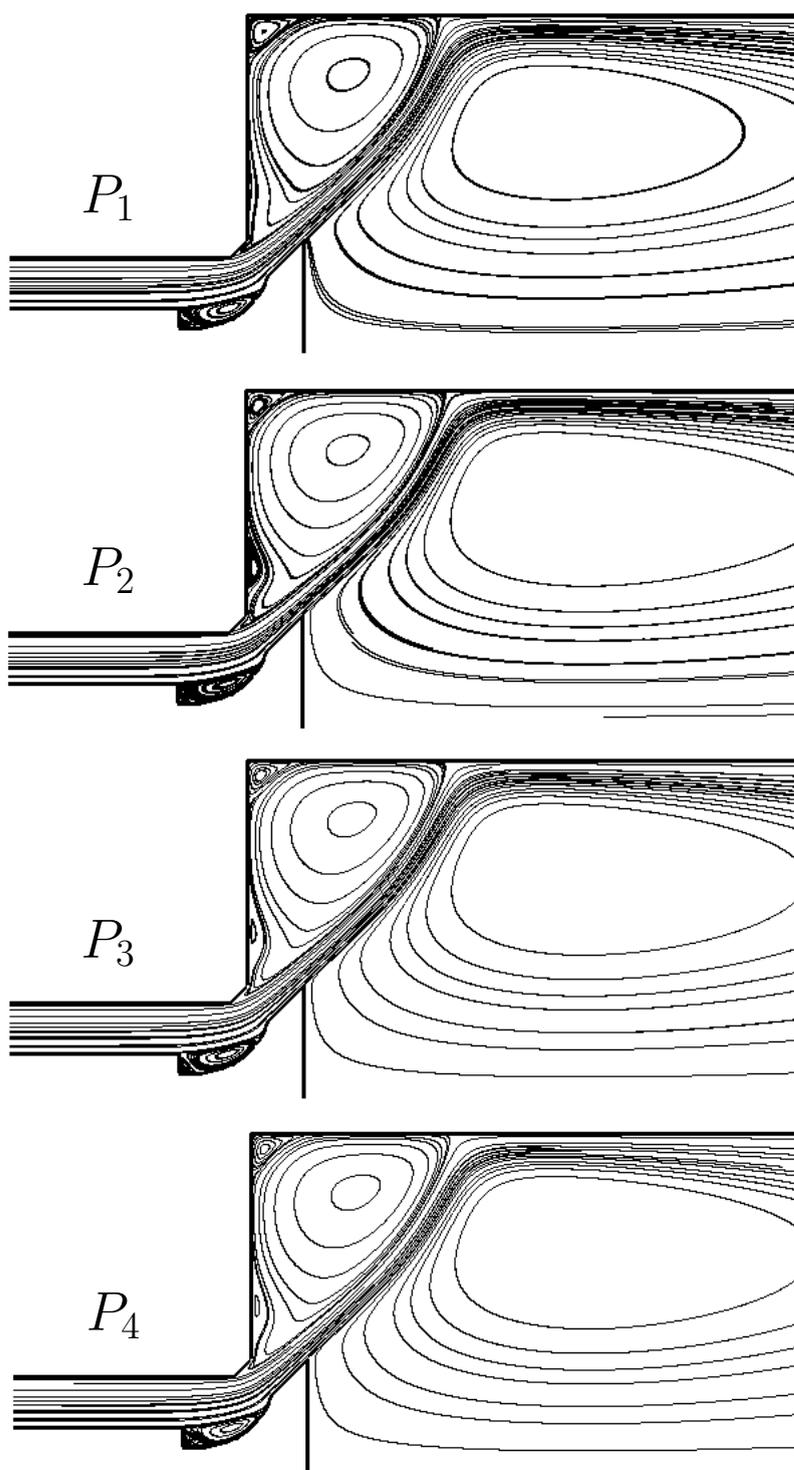


Figure 4.10: Streamlines along the intake valve and inside the cylinder, for  $P_1$  (first row),  $P_2$  (second row),  $P_3$  (third row) and  $P_4$  (fourth row) elements.

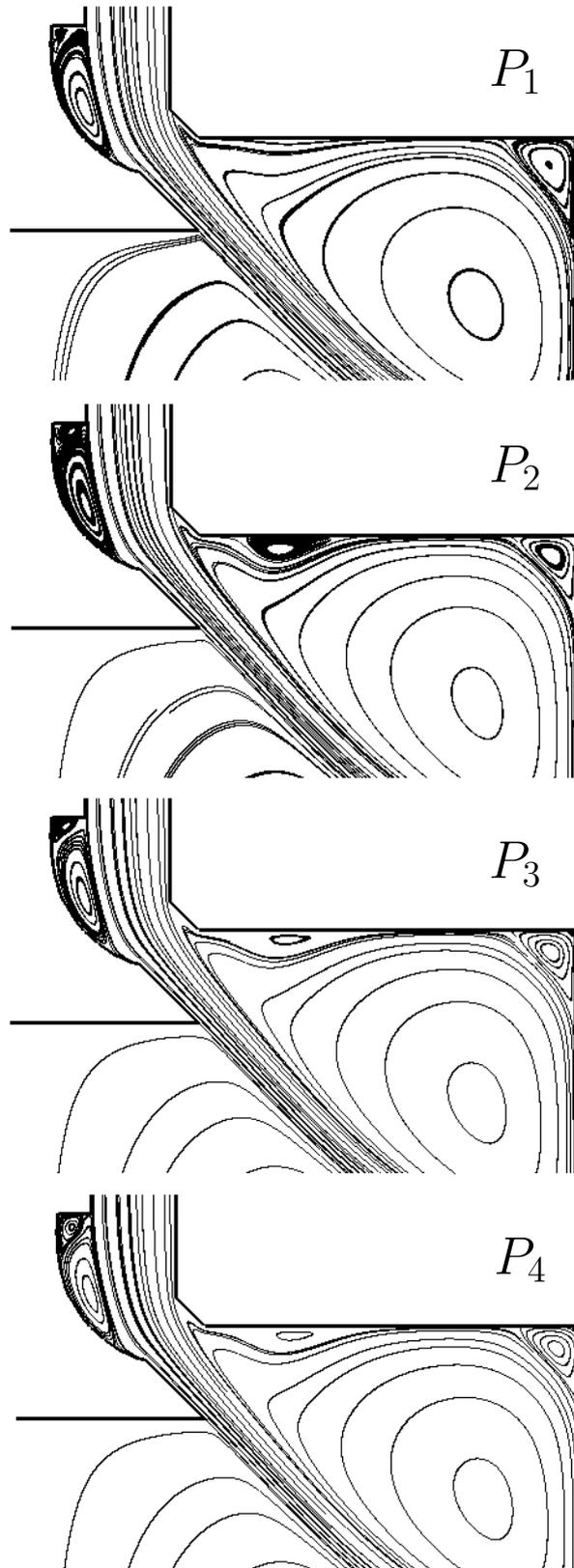


Figure 4.11: Magnified view of the streamlines close to the intake valve, for  $P_1$  (first row),  $P_2$  (second row),  $P_3$  (third row) and  $P_4$  (fourth row) elements.

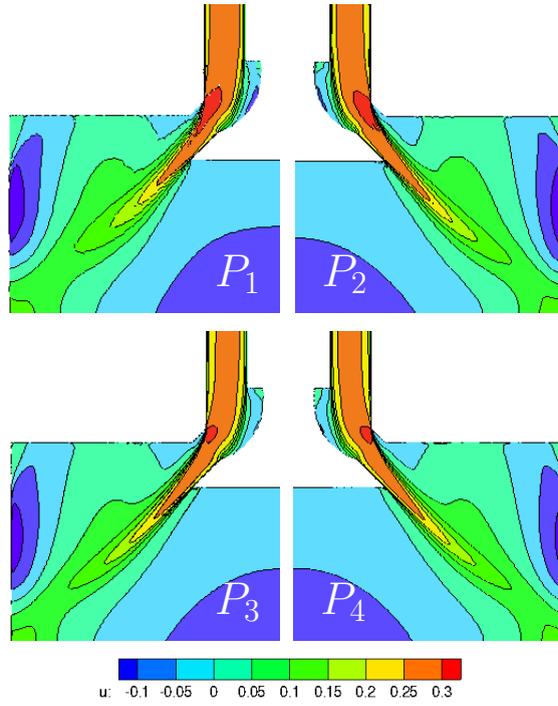


Figure 4.12: contours of the axial velocity,  $u$ , close to the valve, for  $P_1$  (left column, first row),  $P_2$  (right column, first row),  $P_3$  (left column, bottom row) and  $P_4$  (right column, bottom row) elements.

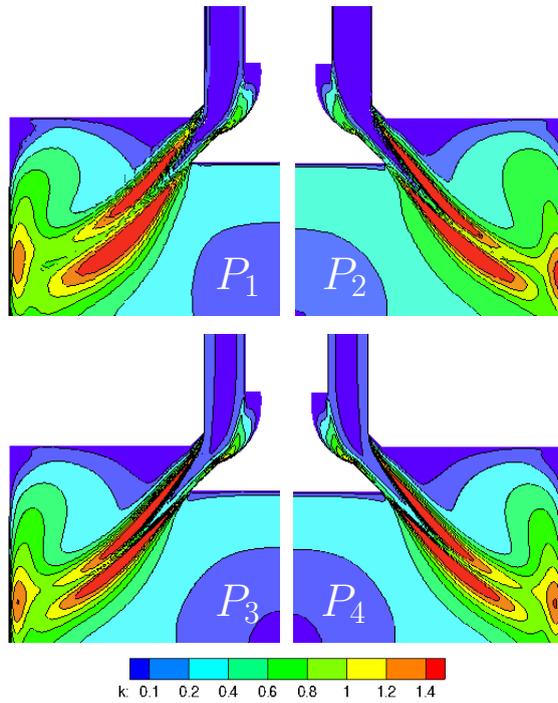


Figure 4.13: contours of the turbulent kinetic energy,  $k$ , close to the valve, for  $P_1$  (left column, first row),  $P_2$  (right column, first row),  $P_3$  (left column, bottom row) and  $P_4$  (right column, bottom row) elements.

In addition, as pointed out in Figs.4.12 and 4.13,  $P_3$  and  $P_4$  approximations achieve a better resolution of the axial velocity and turbulent kinetic energy contours.

In order to assess the accuracy of the numerical results, we compare the non-dimensional mean, and fluctuating axial velocities,  $u$  and  $u' = \sqrt{\frac{2}{3}k}$ , respectively, with the corresponding experimental values measured along the sections reported in Fig.4.9.

The numerical and experimental velocity ratio  $u/u_{bulk}$  are shown in Fig.4.14 as a function of the non-dimensional radial distance  $r/R$  for the four measurement sections considered:  $z$  equal to  $-90$ ,  $-27$ ,  $20$  and  $70$  mm from the engine head. We remark that the reference parameters  $R$  and  $u_{bulk}$  are the radius of the intake duct (or cylinder) and the ratio of the mass flow rate to the inlet area, respectively. The figure shows the velocity profiles for second, third- and fourth- order accurate solutions. The numerical results are generally in good agreement with the experimental data regardless of the spatial discretization. In particular, the annular flow near the exit of the intake duct and the jet valve breakdown are correctly predicted. Furthermore, the wide recirculation region behind the valve seems also to be well reproduced. However, there also exist some discrepancies between numerical and experimental data. Figure 4.15 shows that, although the higher order solutions lead to better predictions of both location and intensity of the peak in the valve jet, all computations under-predict the mean velocity at  $z = 70$ mm close to the cylinder axis. This means that the recirculation zone behind the valve is larger in the simulations than that in the experiments. This discrepancy might be attributed to the  $k - \omega$  model that suffers from a lack of information on the turbulence anisotropy, very important in predicting the correct mixing efficiency.

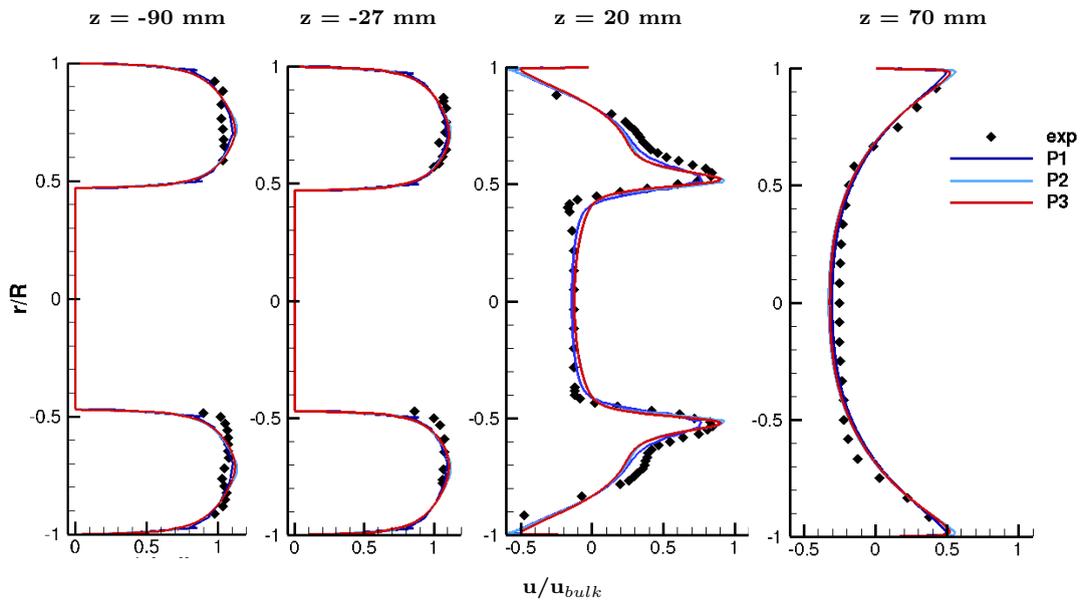


Figure 4.14: Comparison of numerical and experimental axial velocity at -90mm, -27mm, 20mm, 70mm from the cylinder head using  $P_1$ ,  $P_2$ ,  $P_3$  solutions.

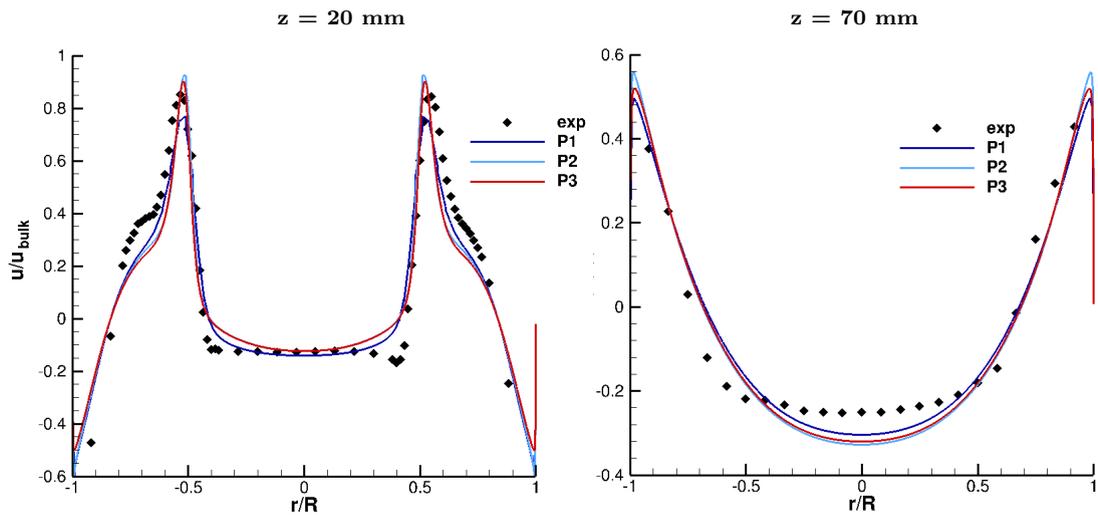


Figure 4.15: Comparison of numerical and experimental axial velocity inside the combustion chamber, at 20mm (right plot) and 70mm (left plot) from the cylinder head, using  $P_1$ ,  $P_2$  and  $P_3$  solutions.

Figure 4.16 shows that the axial velocity fluctuations accurate up to order 4 are also in reasonable agreement with the experimental data. Furthermore, it can be observed that the highest degrees of approximation allows to obtain fluctuation distributions that better match the measurements. In the jet region

it appears that the  $k - \omega$  model under-predicts the turbulent kinetic energy, as more clearly illustrated in Fig.4.17.

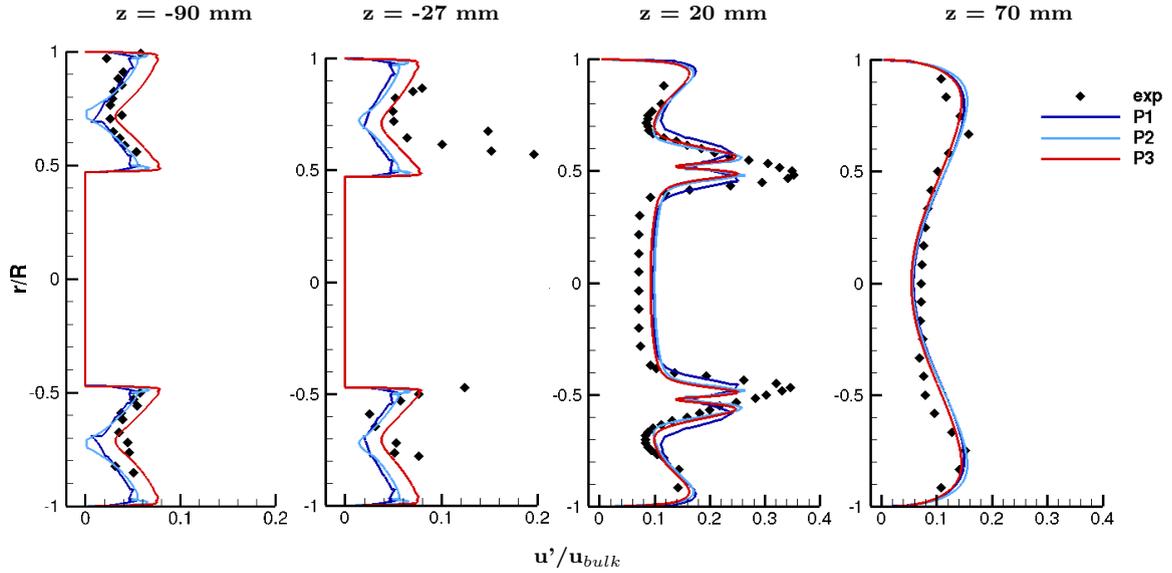


Figure 4.16: Comparison of numerical and experimental axial fluctuation velocity at -90mm, -27mm, 20mm, 70mm from the cylinder head, using  $P_1$ ,  $P_2$  and  $P_3$  solutions.

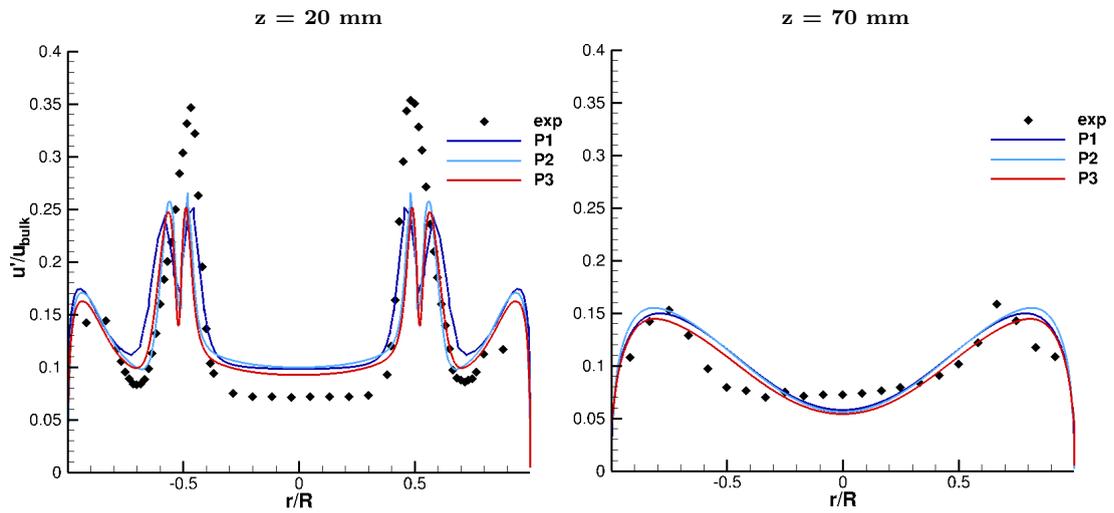


Figure 4.17: Comparison of numerical and experimental axial fluctuation velocity inside the combustion chamber, at 20mm (right plot) and 70mm (left plot) from the cylinder head, using  $P_1$ ,  $P_2$  and  $P_3$  solutions.

As pointed out by Parpais et al. [128] a possible explanation is the interaction between the small scales generated by the jet and the large scales

of the neighboring recirculation zones, which leads to non-equilibrium conditions. Finally, at  $z = 70mm$  numerical results are in better agreement with LDA data.

## 4.2 Convection of an isentropic vortex by uniform flow

### 4.2.1 Description of the test cases

The aim of this test-case is to assess the performance of the Two Implicit Advanced Step Point (TIAS) method for long time simulations of a two dimensional isentropic vortex convected by uniform flow [129]. For the case of the free stream Mach number  $M_\infty = 0.5$ , the results here presented extend to the non-stiff case the analysis carried out by Nigro et al. [90], where  $M_\infty = 0.05$ . The performance of the four-stage, sixth-order accurate implicit TIAS scheme, TIAS(4,6), has been compared with that of the SSPRK(5,4) explicit scheme.

The nonlinear system of equations at each stage of TIAS scheme has been solved by using the Newton scheme. The resulting linear systems have been iteratively solved using the restarted generalized minimum residual (GMRES) method with the block Jacobi preconditioning available in the PETSc library [115]. A preliminary analysis of the results has shown that a linear-solver normalized-residual tolerance of  $10^{-2}$  allows efficient computations while a Newton tolerance around one order less than the solution error does not affect the accuracy of the results.

The efficiency of both the time integration schemes is presented in terms of the L2 norm error of the u -velocity component versus the computational cost expressed in work unit. The work unit is defined as the ratio of the wall clock time of the computation to the corresponding time obtained using TauBench, an unstructured grid benchmark whose kernel is derived from Tau code [130].

All the computations have been performed in parallel, using from 16 to 144 cores, on the Eurora cluster at CINECA (Intel Xeon Eight-Core Sandy-Bridge E5-2658 2.10 GHz processors). TAUBench on 1 core was 8.99433 s.

### Initial solution

The test case analyzed is an isentropic convecting vortex for which the exact solution is known. The vortex initially centered in  $(x_0, y_0)$  is convected by a uniform flow with a freestream Mach number  $M_\infty$ . The resulting initial flow variables are:

$$\begin{aligned} u &= u_\infty - \frac{(u_\infty \beta)(y - y_0)}{R} e^{-r^2/2}, \\ v &= \frac{(u_\infty \beta)(x - x_0)}{R} e^{-r^2/2}, \\ T &= T_\infty - \frac{1}{2} \frac{(\gamma - 1)}{\gamma} (u_\infty \beta)^2 e^{-r^2}, \end{aligned} \quad (4.1)$$

with  $T_\infty = 1$ ,  $u_\infty = M_\infty * \sqrt{\gamma}$ , where  $\gamma = 1.4$  is the ratio of specific heats of the fluid,  $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$  is the distance from the vortex center, and  $R$  and  $\beta$  denotes the characteristic radius and the vortex strength, respectively. The test-case is solved on a quadrangular domain  $(x, y) = [0, \dots, L_x = 0.1] \times [0, \dots, L_y = 0.1]$  and the vortex is initially placed at  $(x_0, y_0) = (0.05, 0.05)$ .

Another issue to be addressed using multi-step methods concerns the unknown initial solutions needed to start the temporal integration. As the vortex is convected without distortion by the mean flow, the additional starting solutions have been obtained by translating the initial solution at the velocity set by the freestream.

### Test-case setup

The domain is discretized by three successively refined uniform cartesian grids, composed by  $16 \times 16$  (coarse),  $32 \times 32$  (medium) and  $64 \times 64$  (fine) elements, respectively. Periodic boundary conditions are imposed at top and bottom, and at left and right boundaries. Figure 4.18 shows the  $16 \times 16$  computational grid.

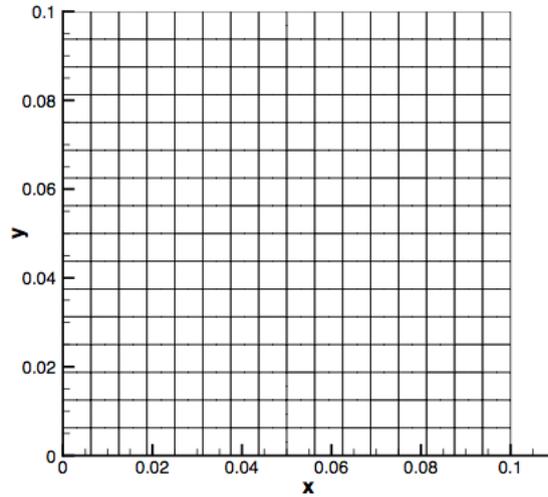


Figure 4.18: computational domain discretized with  $16 \times 16$  elements.

The unsteady simulations have been performed for different polynomial degrees up to a final time corresponding to 50 period  $T$  of vortex revolution, where the time period is defined as  $T = L_x/u_\infty$ . Concerning the computations performed with SSPRK(5,4), due to the superior stability properties of this scheme with respect to other Runge-Kutta schemes, the CFL number has computed according to,

$$\text{CFL}_{RK-45} = 2 \cdot \frac{1}{2p + 1}.$$

where  $p$  denotes the degree of polynomial approximation.

### 4.2.2 Fast vortex

The "fast" problem ( $M_\infty = 0.5$ ,  $\beta = 1/5$ ,  $R = 0.005$ ) has been computed on the quadrangular grids by using polynomial approximations from  $P_2$  to  $P_5$ . For each polynomial degree a temporal refinement study has been performed in order to determine the largest time-step value that ensures that the temporal discretization error does not affect the solution accuracy, after 50 vortex period, on a given mesh.

Grid	TIAS (4,6)				SSPRK (5,4)			
	P <sup>2</sup>	P <sup>3</sup>	P <sup>4</sup>	P <sup>5</sup>	P <sup>2</sup>	P <sup>3</sup>	P <sup>4</sup>	P <sup>5</sup>
Coarse	$T/80$	$T/125$	$T/125$	$T/160$	$T/1680$	$T/2400$	$T/3055$	$T/3735$
Medium	$T/80$	$T/200$	$T/250$	$T/320$	$T/3360$	$T/4800$	$T/6110$	$T/7470$
Fine	$T/200$	$T/320$	$T/500$	$T/800$	$T/6720$	$T/9600$	$T/12220$	–

Table 4.2: Time-step size for different discretization levels as a function of the vortex period  $T$ .

The performance of the TIAS-DG and SSPRK-DG schemes are reported in Fig.4.19, by plotting the  $L_2$ -norm errors of the  $u$ -velocity component versus  $1/\sqrt{nDOFs}$  (left plot), with  $nDOFs$  equal to the total number of degrees of freedom per equation for the different spatial discretizations, and work units (right plot).

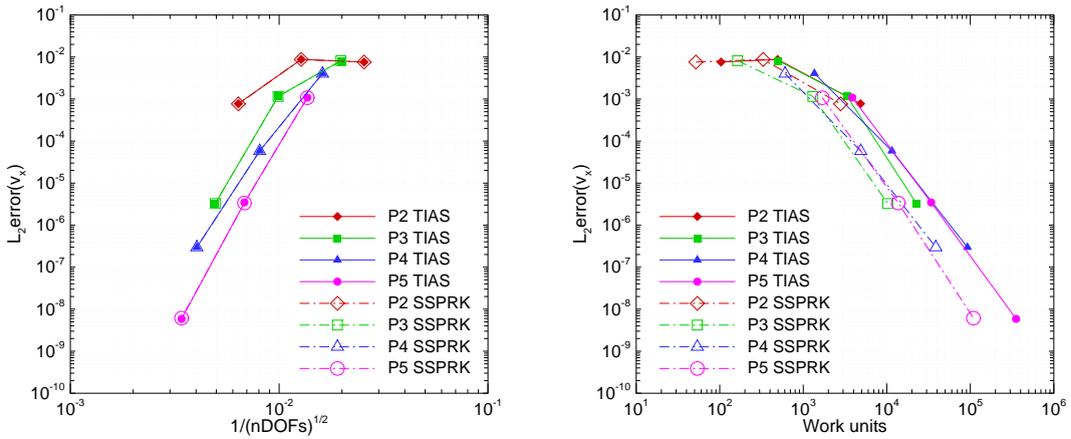


Figure 4.19:  $L_2$  error( $u$ ) as a function of  $1/\sqrt{nDOFs}$  (left) and work units (right).

The left plot highlights that the convergence histories corresponding to the SSPRK-DG and TIAS-DG solutions are almost indistinguishable, with  $P_5$  solutions achieving very low error levels. However a significant discrepancy between the time step sizes of the TIAS-DG scheme and the explicit scheme can be observed in Table 4.2. For instance, to reach an accuracy level of order of  $10^{-2}$ , the implicit time-step size was about 20 times bigger than the explicit one, using  $P_2$  elements, while for an accuracy level of order of  $10^{-6}$

this ratio increased to 30 with  $P_3$  computations. However, since the TIAS scheme consists of a 4 non-linear systems at each time-step, the computational efficiency of the schemes needs to be discussed. The right plot of Fig.4.19 shows that the schemes have a similar asymptotic behaviour, but SSPRK-DG outperforms TIAS-DG. In fact, to achieve an accuracy level of order  $10^{-2}$  the SSPRK-DG scheme is about 2 times faster than TIAS-DG, whereas for an accuracy level of  $10^{-6}$  this ratio increases to 2.2. These results allow to draw the opposite conclusion to that of Nigro et al. [90], who found that TIAS-DG outperforms SSPRK-DG when  $M_\infty = 0.05$ .

## 4.3 Shocked flows

### 4.3.1 Two-dimensional supersonic wedge

#### Description of the test-case

This test case aims at evaluating the feasibility of gradient-based methods for optimal control of oscillations to a low dissipative high-order DG scheme. The exact weak solution to the two-dimensional supersonic wedge has been computed using the shock relations presented in the text written by Shapiro [131]. Columns State 1 and State 2 of Table 4.3 show the known and computed flow conditions upstream and downstream the oblique shock, respectively, for a wedge angle of 15 degrees, that corresponds to the incident shock angle of 32.24 degrees.

	State 1	State 2
$M$	3.0	2.255
$\rho$	1.0	2.032
$p$	1.0	2.821

Table 4.3: Flow conditions upstream (state 1) and downstream (state 2) the oblique shock

The computational grid is composed by 1178 quadrangular elements, with the grid lines aligned with the wedge, see Fig. 4.20.

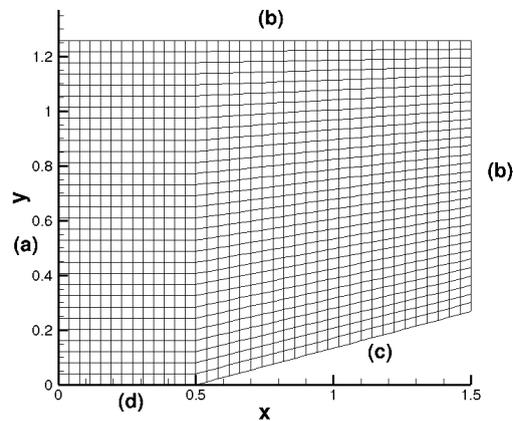


Figure 4.20: Computational domain for the two dimensional supersonic wedge.

On the wall surface (c) of the wedge a zero heat flux slip boundary conditions is imposed. For the other boundaries entropy and stagnation enthalpy are specified at inflow (a), pressure at outflow (b) and symmetry at the  $y = 0$  boundary (d).

### Effects of polynomial degree and shock capturing parameter on the solution accuracy

Figure 4.21 shows the pressure contours of  $P_1$ ,  $P_3$  and  $P_5$  solutions, obtained using a shock capturing coefficient  $csc = 0.1$ . The figure shows that the higher the order of accuracy, the more the shock front is better resolved and less smeared.

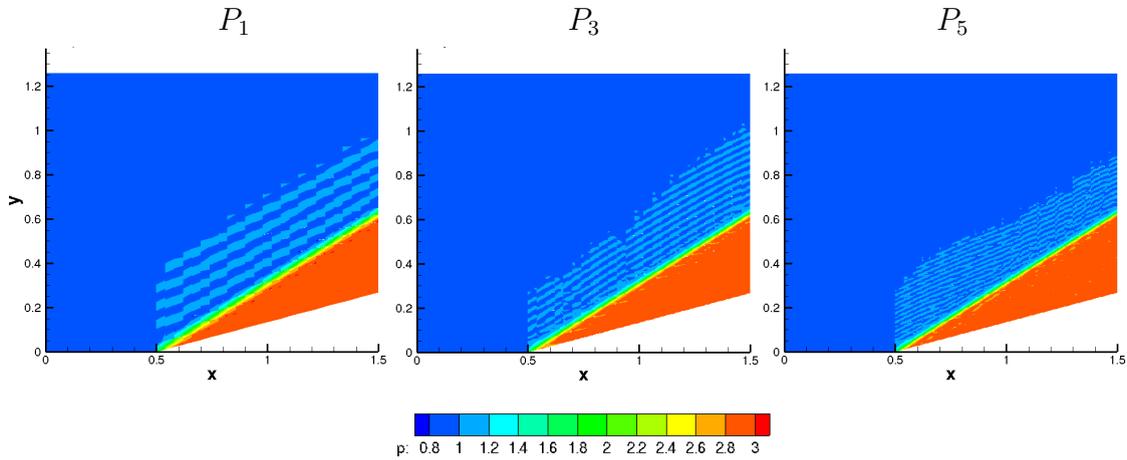


Figure 4.21: Pressure contours for  $P_1$  (left plot),  $P_3$  (middle plot) and  $P_5$  elements (right plot), using  $csc = 0.1$ .

A more quantitative comparison is reported in Fig. 4.22, where the pressure distributions along the  $y = 0.3$  line and exact solutions are compared. It can be noted that an increase of the polynomial degree strongly affects the accuracy of the solution close to the shock location. As can be appreciated from Fig. 4.23 pre- and post-oscillations are relatively small for  $P_3$  and  $P_5$  solutions, whilst the  $P_1$  solution exhibits significant undershoots and overshoots.

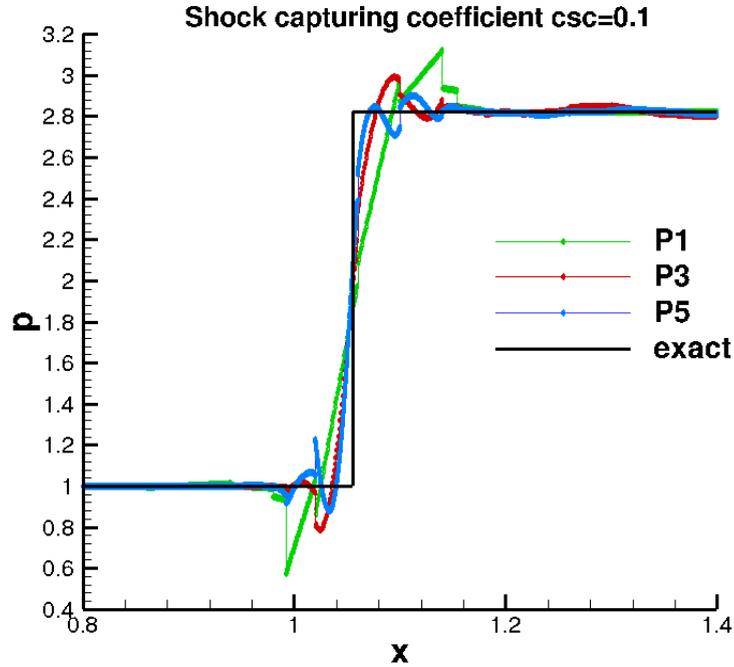


Figure 4.22: Comparison between exact and  $P_1$ ,  $P_3$  and  $P_5$  solutions, along the  $y = 0.3$  line, using  $csc = 0.1$ .

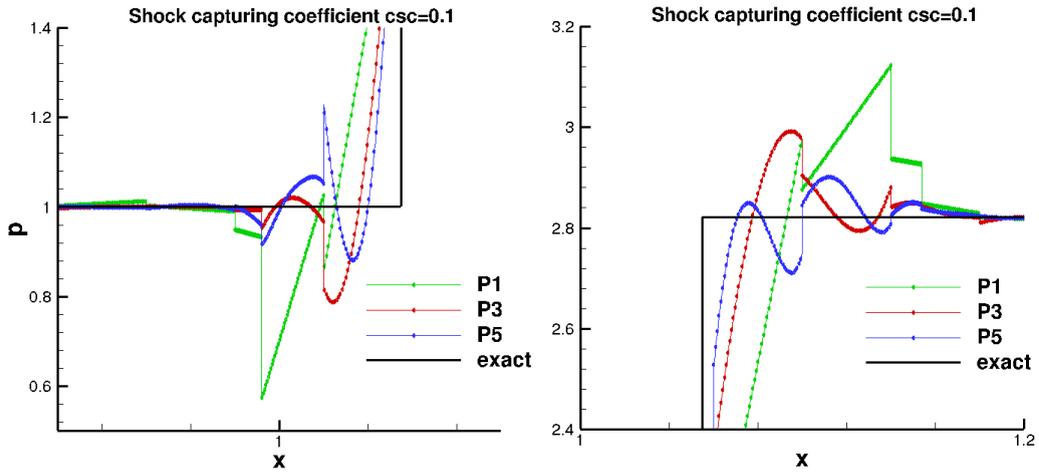


Figure 4.23: Undershoots and overshoots of the pressure upstream (left plot) and downstream (right plot) the shock. Exact solution versus approximate solutions computed along the  $y = 0.3$  line using  $P_1$ ,  $P_3$  and  $P_5$  elements, with  $csc = 0.1$ .

The next two figures compare  $P_5$  solutions for increasingly higher shock capturing coefficients. Figures 4.24 and 4.25 show that oscillations, but also the resolution of the discontinuity, reduce increasing the  $csc$  values. As a result, in the choice of the more appropriate value of the shock capturing coefficient we

face a trade-off between sub-cell resolution of the shock and magnitude of pre- and post-shock oscillations. This motivated the effort to develop numerical strategies in order to automatically optimize the shock capturing parameters.

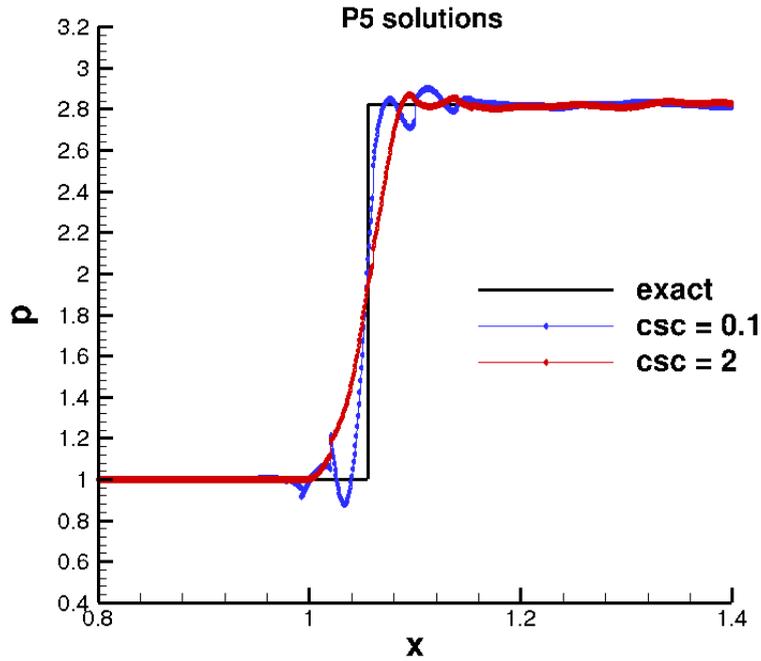


Figure 4.24: Comparison between exact and  $P_5$  solution along the  $y = 0.3$  line, for  $csc = 0.1$  and 2.

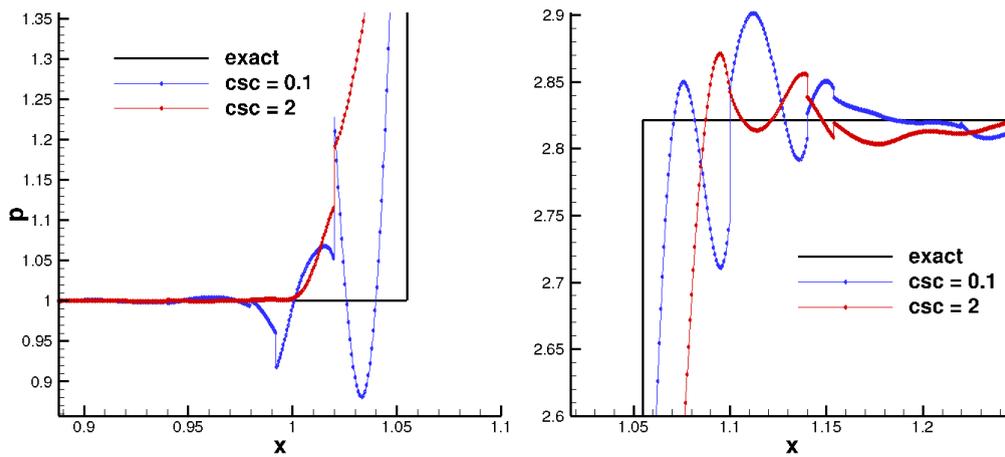


Figure 4.25: Undershoot and overshoot of pressure upstream (left plot) and downstream the shock (right plot). Exact solution versus  $P_5$  solutions computed along the  $y = 0.3$  line using  $csc$  0.1 and 2.

## Assessment of the optimization strategy

The optimization procedure implemented in this work aims to find the optimal value of some shock capturing parameters by minimizing a specific cost function. More specifically, the design parameters  $\alpha_1$  and  $\alpha_2$  have been introduced in the shock capturing term as follows,

$$\epsilon_p(\mathbf{u}_h^\pm, \mathbf{u}_h) = Ch_k^2 \frac{\alpha_1 |s_p(\mathbf{u}_h^\pm, \mathbf{u}_h)| + \alpha_2 |d_p(\mathbf{u}_h)|}{p(\mathbf{u}_h)} f_p(\mathbf{u}_h).$$

The cost function  $J$  is the  $L_2$  norm of the error between the exact and the numerical pressure field,

$$J = \|Error(p)\|_{L_2(\Omega)} = \left[ \frac{\int_{\Omega} (p - p^{exact})^2 d\Omega}{\int_{\Omega} d\Omega} \right]^{1/2} = \left[ \frac{\sum_{i=1}^{N_e} \int_{K_i} (p - p^{exact})_i^2 d\mathbf{x}}{\sum_{i=1}^{N_e} |K_i|} \right]^{1/2},$$

where the sum is performed on all elements and the element integral is computed with Gauss quadrature formula of suitable orders of accuracy. Concerning the algorithms employed to solve the proposed optimization model, the steepest descent method and the BFGS method are used. DG computations starting from different initial step size are performed for different  $csc$  values and using linear ( $P_1$ ), quadratic ( $P_2$ ) and cubic elements ( $P_3$ ). For all the simulations the optimization process starts with the initial values of the design parameters,  $\alpha_1$  and  $\alpha_2$ , equal to 1.

### Verification of the inviscid gradients

The optimization strategy is based on the evaluation of the gradient by means of Automatic Differentiation. The derivation has been carried out in *forward – mode* using the Tapenade code [93,94]. The AD generated sensitivities computations have been validated against finite differences, using the following FD model form:

$$\left( \frac{\partial J}{\partial \alpha_i} \right)_{FD} = \frac{J(\alpha_i - \delta \alpha_i) - J(\alpha_i)}{\delta \alpha_i}.$$

Tables 4.4 and 4.5 compare FD and AD sensitivities for different polynomial degrees. Both the table show that AD sensitivity values are in perfect agreement with the ones computed using finite difference. We remark that FD results have been obtained with a perturbation step of  $\delta\alpha = 10^{-5}$ .

Polynomial degree	$\frac{\partial J}{\partial \alpha_1} AD$	$\frac{\partial J}{\partial \alpha_1} FD$
1	7.8145451725E-004	7.8145324101E-004
2	5.5691441904E-004	5.5691448259E-004
3	1.0050376617E-003	1.0050398201E-003
4	1.0676063008E-003	1.0676088666E-003
5	1.5449287081E-003	1.5449299361E-003

Table 4.4: Sensitivity of the cost function with respect to  $\alpha_1$ , computed using finite difference and automatic differentiation methods as a function of the polynomial degree.

Polynomial degree	$\frac{\partial J}{\partial \alpha_2} AD$	$\frac{\partial J}{\partial \alpha_2} FD$
1	1.2913726697E-003	1.2913716124E-003
2	1.7145549053E-003	1.7145525846E-003
3	1.6175297560E-003	1.6175293591E-003
4	1.1714025128E-003	1.1714045921E-003
5	1.6829608295E-003	1.6829623237E-003

Table 4.5: Sensitivity of the cost function with respect to  $\alpha_2$ , computed using finite difference and automatic differentiation methods as a function of the polynomial degree.

The selection of the perturbation step size is one of the problems associated with finite differencing as it is difficult to make this choice a priori. Therefore, the perturbation step size effect has been first investigated. Table 4.6 and Fig. 4.26 show the difference between AD and FD sensitivities

$$Err_i = \left| \left( \frac{\partial J}{\partial \alpha_i} \right)_{FD} - \left( \frac{\partial J}{\partial \alpha_i} \right)_{AD} \right|$$

for different perturbation step sizes, using  $P_1$  elements. It can be seen that accurate finite difference approximations are computed for  $\delta\alpha = 10^{-5}$ , whereas large  $Err_i$  values occur for both smaller and larger perturbation magnitudes.

For small perturbation sizes cancellation error is dominant whereas truncation error becomes dominant at large  $\delta\alpha$ .

$\delta\alpha$	Err <sub>1</sub>	Err <sub>2</sub>
1.00E-002	7.7940745778E-001	1.2890216182E+000
1.00E-003	7.7351319236E-002	1.2783531067E-001
1.00E-004	7.0329638410E-003	1.1622248178E-002
1.00E-005	1.2762475210E-009	1.0573054700E-009
1.00E-006	7.0330907991E-004	1.1622354105E-003
1.00E-007	7.7363996408E-004	1.2784589371E-003

Table 4.6: Differences between FD and AD sensitivities for several perturbation steps, using  $P_1$  elements and  $csc = 0.1$ .

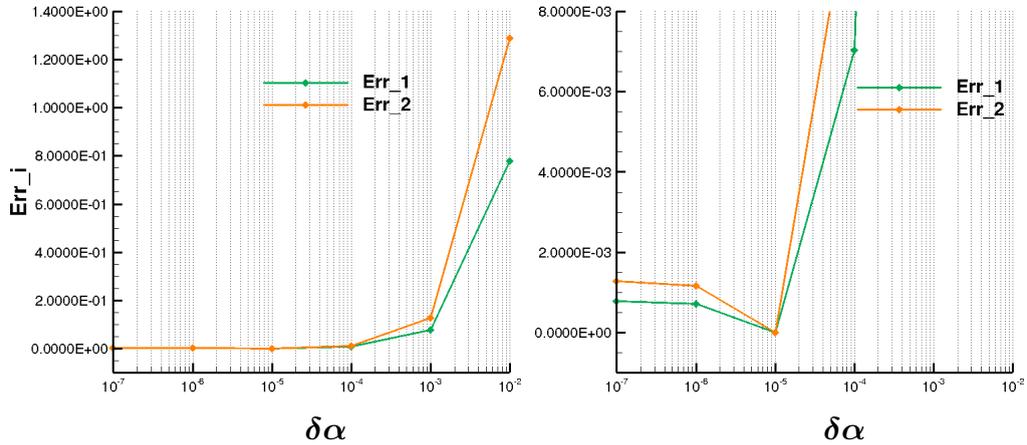


Figure 4.26: Behaviour of the differences between FD and AD sensitivities as a function of the perturbation step  $\delta\alpha$ , for  $csc = 0.1$  and  $P_1$  elements (left plot) ,magnified view (right plot)

### Steepest descent method computations

Here we present the results obtained using the steepest descent method. The updating of each design variable during the iterative optimization process is based on the following algorithm:

$$\begin{aligned} & \text{if } (J_{i_{opt}+1} < J_{i_{opt}}) \\ & \quad \alpha_{i,i_{opt}+1} = \alpha_{i,i_{opt}} - step_{i_{opt}} \left( \frac{\partial J}{\partial \alpha_i} \right)_{i_{opt}}, \quad i = 1, 2 \\ & \text{else} \\ & \quad step_{i_{opt}} = step_{i_{opt}}/1.5, \\ & \quad i_{opt} = i_{opt} - 1 \quad \text{and recompute all.} \end{aligned}$$

We note that the variation of the step length,  $step_{i_{opt}}$ , during an iteration is based on the successive reduction method, with a contraction coefficient of 1.5.

- *P1 Solutions*

Figure 4.27 compares the histories of the design variables, cost function and sensitivities versus the number of iteration cycles  $i_{opt}$ , for different  $csc$  values and using the initial step size of 100. The plots show that the shock capturing coefficient  $csc$  strongly affects the convergence of the optimization cycles. It can be noticed that cost functions and sensitivities converge to their minimum values for  $csc \leq 0.05$ , whereas the behaviours of all the optimization quantities stagnate after 8 iteration cycles when  $csc = 0.1$ . There is evidence also that the convergence slow-down as  $csc$  reduces. Finally, we see that the design parameters do not achieve their optimal values, even for the lower  $csc$  coefficients.

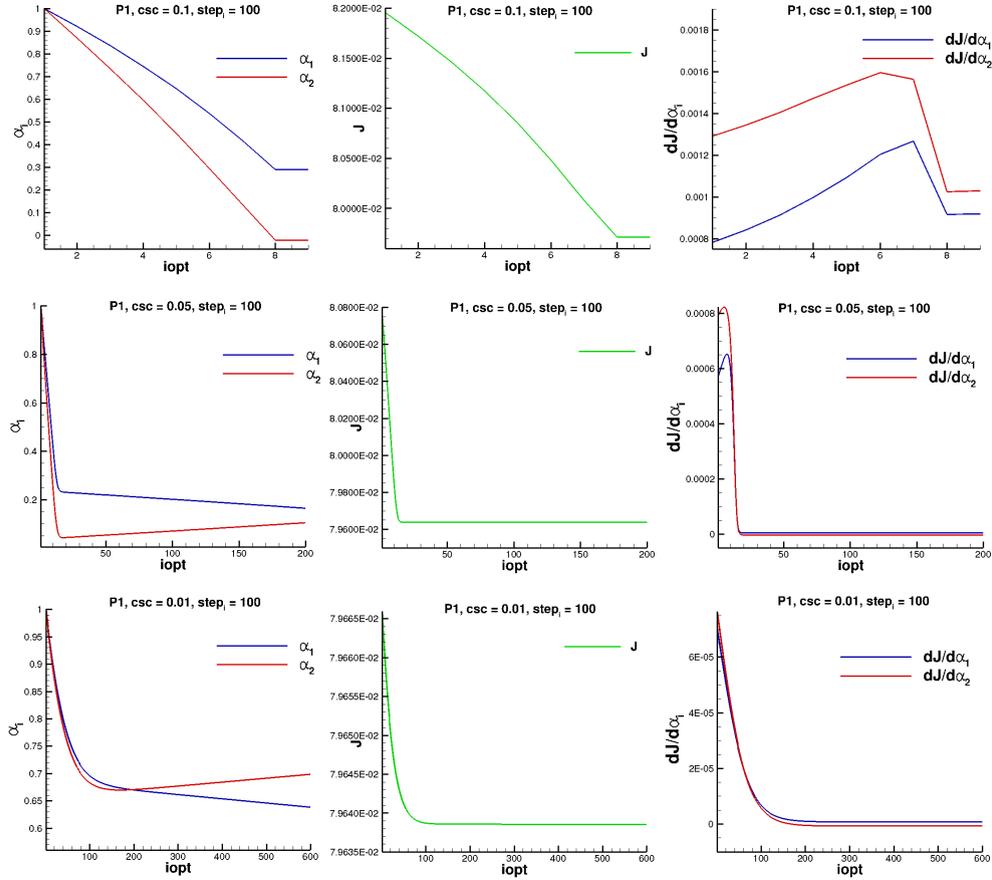


Figure 4.27: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_1$  elements and an initial step size of 100.  $csc = 0.1$  (top row),  $csc = 0.05$  (middle row) and  $csc = 0.01$  (bottom row).

- $P_2$  Solutions

Although several tests on different combinations of the optimization parameters ( $csc$ , initial step size) have been carried out, we observed that  $P_2$  computations blow-up during the optimization cycles. In Fig. 4.28 we present the results for  $csc = 0.1, 0.05$  and using an initial step size of 100, before the blow-up of the solutions occurred.

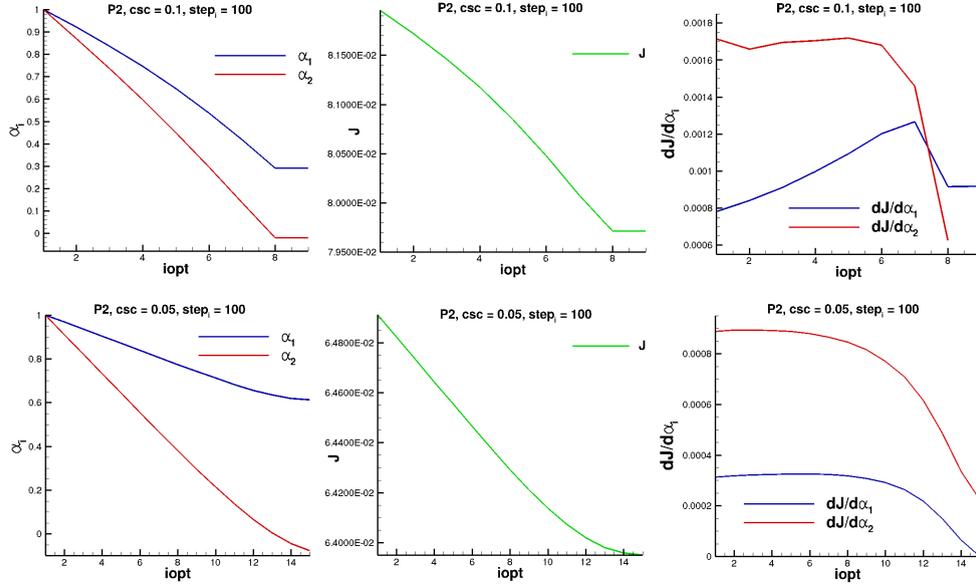


Figure 4.28: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_2$  elements and an initial step size of 100.  $csc = 0.1$  (top row) and  $csc = 0.05$  (bottom row).

The plots confirm the effect of the shock capturing coefficient on the convergence process shown above for  $P_1$  computations. We observe the stagnation of the optimization quantities for  $csc = 0.1$  and a more regular convergence behaviour, without stagnation, when  $csc = 0.05$ . Finally, differently from  $P_1$  results, we remark that the  $\alpha_2$  design variables achieves unphysical negative values in both testing conditions.

- *P3 Solutions*

In Fig.4.29 we present the results obtained under the same testing conditions as shown in Fig.4.28. The graphs in the figure shows a lack of convergence and more irregular behaviours of cost function and sensitivities during the optimization process, as compared with  $P_1$  and  $P_2$  results. We remark that in both cases the iterative process completed without the occurrence of blow-up, achieving very small step sizes.

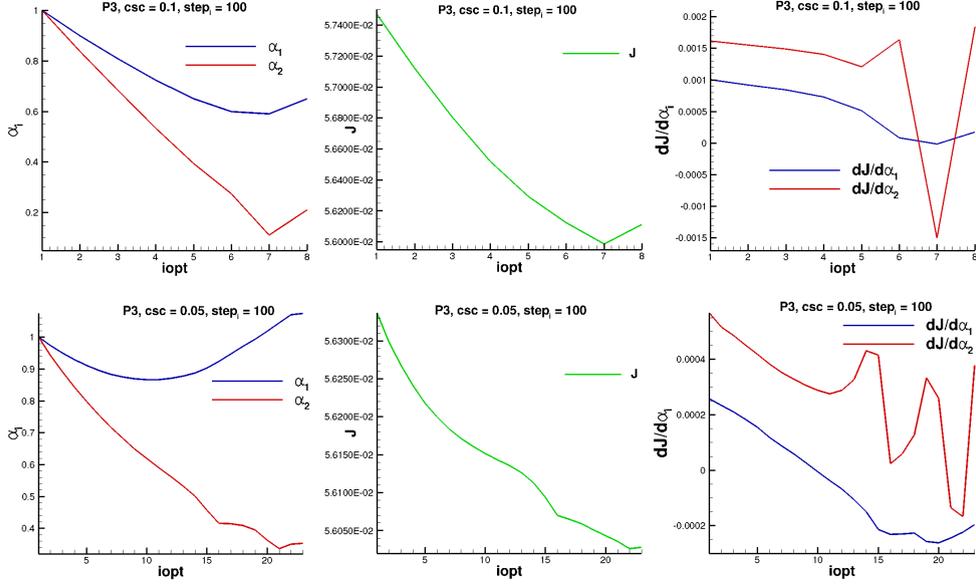


Figure 4.29: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_3$  elements and an initial step size of 100.  $csc = 0.1$  (top row) and  $csc = 0.05$  (bottom row).

In order to overcome the convergence problems during the iterative optimization process, the Quasi-Newton BFGS method has been applied, and the results are discussed below.

### BFGS method computations

Here the results obtained by applying the BFGS Quasi-Newton method to the proposed optimization model are given. The updating of each design variables during the optimization process is based on the fulfillment of the following condition:

$$\text{if } J_{iopt+1} \leq J_{iopt} + c_1 \text{step}_{iopt} \nabla J_{iopt}^T p_{iopt}$$

$$\alpha_{i,iopt+1} = \alpha_{i,iopt} + \text{step}_{iopt} p_{iopt}, \quad i = 1, 2$$

else

$$\text{step}_{iopt} = \text{step}_{iopt} / 1.5,$$

$$iopt = iopt - 1 \quad \text{and recompute all.}$$

As before, we employ the successive reduction method for the updating of the step size during the iterations, with a contraction coefficient of 1.5. Concerning the evaluation of the search directions  $p_{i_{opt}}$  and for further details about the BFGS method please refer to the section 3.6.1 of the previous chapter. We remark that using the BFGS method the iterative optimization process required initial step size much smaller than those that can be imposed by using the steepest descent method.

- *P1 Solutions*

The Figs. 4.30 shows results of the simulations carried out for  $csc = 0.01$  and using initial step sizes of 1 and 0.01, respectively.

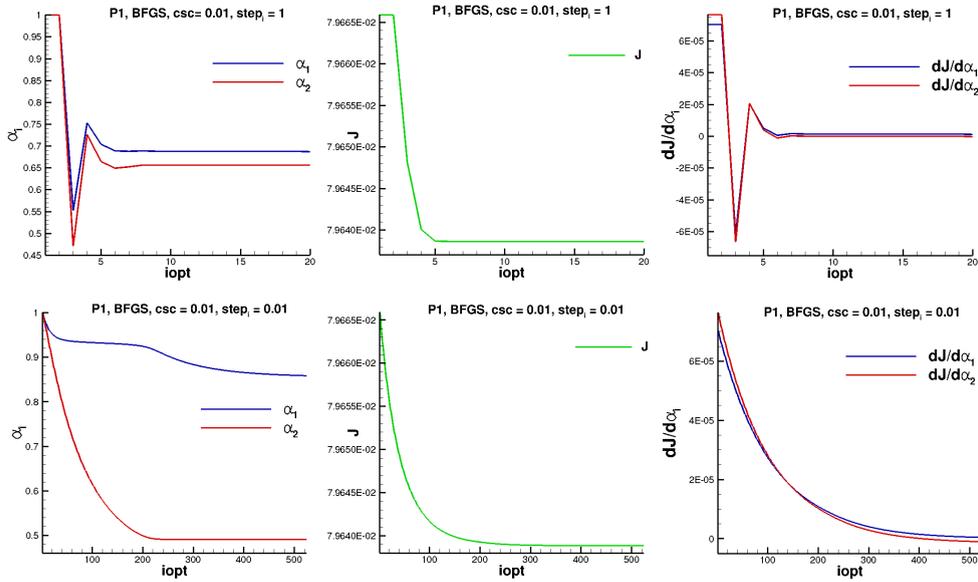


Figure 4.30: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_1$  elements and  $csc = 0.01$ .  $step_i = 1$  (top row),  $step_i = 0.01$  (bottom row).

The plots show that cost function and sensitivities converges after 5 optimization cycles, when the initial step size is taken equal to 1. Furthermore, in contrast to the corresponding second-order accurate steepest

descent computations, also the design variables achieve their optimal values:  $\alpha_{1,opt} = 0.687$  and  $\alpha_{2,opt} = 0.656$ . We remark that these values are in good agreement with those computed using the steepest descent method when  $csc = 0.01$ . The oscillations in the convergence history of sensitivities disappear when the initial step size is taken equal to 0.01. Furthermore, reducing the initial step size, a higher number of iteration cycles is required for achieving the convergence of all the optimization quantities. Finally, we observe that whereas the optimal value of  $J$  is in perfect agreement with the one obtained for  $step_i = 1$ , there is a significant discrepancy between the optimal values of the design variables. This mismatch suggests the presence of a local minimum.

- *P2 Solutions*

Overall the plots of Fig. 4.31 display the lack of convergence shown by the corresponding steepest descent  $P_2$  computations. In particular, simulation diverged after 7 iteration cycles when  $csc = 0.008$  and  $step_i = 1$ . Furthermore, even for  $csc = 0.01$  and  $step_i = 0.01$ , the optimization quantities do not achieve their optimal values, although during the iterative process no blow-up has occurred.

Finally, we note that, unlike the steepest descent computations, in the BFGS  $P_2$  simulations, the  $\alpha_2$  design variable is positive during all the iterations.

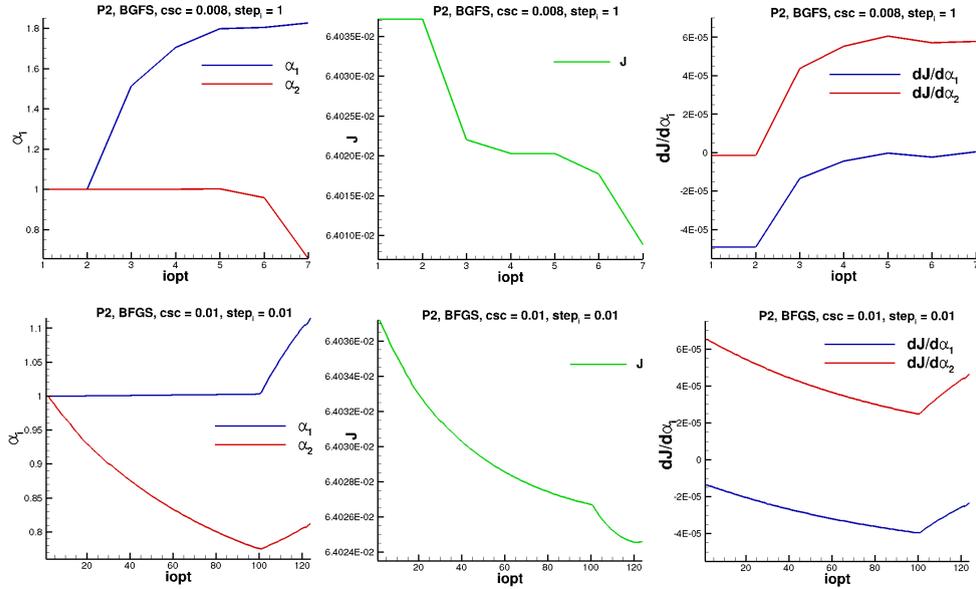


Figure 4.31: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_2$  elements.  $step_i = 1$  and  $csc = 0.008$  (top row),  $step_i = 0.01$  and  $csc = 0.01$  (bottom row).

- *P3 Solutions*

A dramatic improvement with respect to the steepest descent results is evident from the plots of Fig. 4.32.

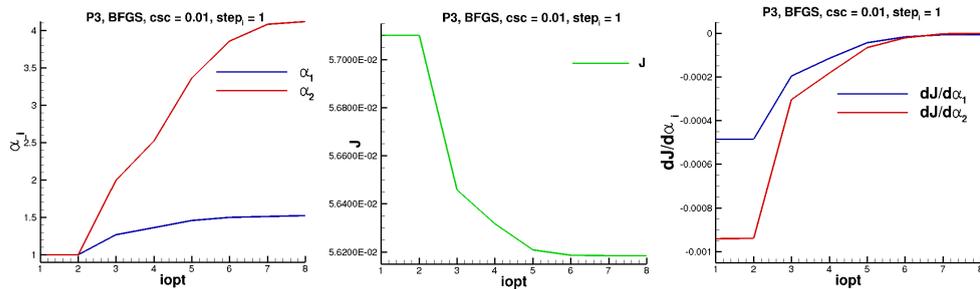


Figure 4.32: Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using  $P_3$  elements,  $step_i = 1$ ,  $csc = 0.01$ .

The BFGS method was successful in minimizing the cost function. The optimization is converged in about 7 iterations for  $step_i = 1$  and  $csc =$

0.01. Furthermore, it can be seen, from the plot, that optimal values of the design variables have been determined ( $\alpha_{1,opt} = 1.521$ ,  $\alpha_{2,opt} = 4.115$ ). Finally, unlike the steepest descent method where very irregular sensitivity behaviours were observed, in the BFGS simulations the gradients of the design variables converge without oscillations.

# Conclusions

In this work a DG discretization, based on the method proposed by Bassi et al. [96], has been proposed for the numerical solution of internal combustion engine (ICE) flows. The capabilities and robustness of an implicit DG method have been assessed by solving the RANS and  $k-\omega$  turbulence model equations on classical ICE test cases. This approach has proved to be very well suited to deal with the stiffness induced by the stretched grids, low Mach number flow regions and by highly nonlinear governing equations, typical of turbulent flow computations. Afterwards, an implicit approach to higher-order time integration, coupled with high-order DG discretizations, has been evaluated further for efficient long time simulations of unsteady flows. In this work we focused on the high-order two implicit advanced step-point (TIAS) method. The performance of the sixth-order accurate TIAS scheme was investigated on a classical test-case for the evaluation of LES/DES capabilities of high-order methods, with the result of an higher efficiency of the five-stage, fourth-order accurate SSPRK explicit scheme with respect to the TIAS scheme for non-stiff testing conditions, whereas TIAS scheme was proved to outperform SSPRK in the stiff case by Nigro et al. [90]. As a second contribution to a robust and accurate flow solver for ICE applications a non-standard optimization approach to control spurious oscillations in simulation of transonic flows has been presented and its feasibility demonstrated by means of different strategies and numerical examples.

The main drawback of our axisymmetric ICE flow computations is related to the isotropic, scalar description of the turbulence. This motivates the future

effort at implementing advanced turbulence modeling capabilities, such as the hybrid RANS-LES approach. Furthermore, for turbulent flows at not-too-high Reynolds number, typical of ICE applications, the ILES approach, which looks quite simple and suitable for high-order DG discretization, will be numerically evaluated using high-order accurate implicit schemes to advance in time the solution of the DG space-discretized ILES system. On the side of shock capturing, the proposed gradient based approach, using automatic differentiation for the accurate evaluation of sensitivity, causes iterative convergence to the minimum of the cost function to stall/blow-up using  $P_2$  elements. The reasons for such convergence deterioration are currently under investigation. Ongoing work is devoted to a thorough validation of the proposed high-order accurate shock-capturing DG scheme for inviscid computations of reference test cases available in the literature.

# List of Figures

3.1	Two neighboring elements $K^-$ and $K^+$ sharing edge $E$ , including the two opposite normal vectors $\mathbf{n}^\pm$ to $E$ . . . . .	33
3.2	scheme of the optimization process . . . . .	53
4.1	Computational grid for the abrupt axisymmetric expansion test-case, including measurement stations. . . . .	56
4.2	$y^+$ values for the first cell-centroid $y_1$ as a function of different polynomial degrees at the considered measurement stations. . . . .	57
4.3	Contours of pressure obtained using $P_1$ (top row), $P_3$ (middle row) and $P_5$ (bottom row) elements. . . . .	58
4.4	Snapshots of streamtraces close to the corner of the wall, for $P_1$ (left column), $P_3$ (middle column) and $P_5$ (right column) elements. . . . .	58
4.5	DG axial velocity profiles compared with the experimental data at the considered measurement stations: $z/D = 0.0$ and $z/D = 3.0$ (top row), $z/D = 5.0$ and $z/D = 8.0$ (bottom row). . . . .	59
4.6	DG axial velocity profiles compared with the experimental data at $z/D = 5.0$ , close-up near the axis (left column) and the wall (right column). . . . .	60
4.7	Skin friction coefficient $C_f$ along the wall of the duct, for $P_1$ , $P_2$ , $P_3$ , $P_4$ and $P_5$ elements. . . . .	60
4.8	Detailed view of the geometry of the engine head . . . . .	61

4.9	Computational grid upstream and downstream the valve, including measurement stations. . . . .	62
4.10	Streamlines along the intake valve and inside the cylinder, for $P_1$ (first row), $P_2$ (second row), $P_3$ (third row) and $P_4$ (fourth row) elements. . . . .	64
4.11	Magnified view of the streamlines close to the intake valve, for $P_1$ (first row), $P_2$ (second row), $P_3$ (third row) and $P_4$ (fourth row) elements. . . . .	65
4.12	contours of the axial velocity, $u$ , close to the valve, for $P_1$ (left column, first row), $P_2$ (right column, first row), $P_3$ (left column, bottom row) and $P_4$ (right column, bottom row) elements. . .	66
4.13	contours of the turbulent kinetic energy, $k$ , close to the valve, for $P_1$ (left column, first row), $P_2$ (right column, first row), $P_3$ (left column, bottom row) and $P_4$ (right column, bottom row) elements. . . . .	66
4.14	Comparison of numerical and experimental axial velocity at -90mm, -27mm, 20mm, 70mm from the cylinder head using $P_1$ , $P_2$ , $P_3$ solutions. . . . .	68
4.15	Comparison of numerical and experimental axial velocity inside the combustion chamber, at 20mm (right plot) and 70mm (left plot) from the cylinder head, using $P_1$ , $P_2$ and $P_3$ solutions. . .	68
4.16	Comparison of numerical and experimental axial fluctuation velocity at -90mm, -27mm, 20mm, 70mm from the cylinder head, using $P_1$ , $P_2$ and $P_3$ solutions. . . . .	69
4.17	Comparison of numerical and experimental axial fluctuation velocity inside the combustion chamber, at 20mm (right plot) and 70mm (left plot) from the cylinder head, using $P_1$ , $P_2$ and $P_3$ solutions. . . . .	69
4.18	computational domain discretized with $16 \times 16$ elements. . . .	72

4.19	$L_2$ error( $u$ ) as a function of $1/\sqrt{nDOFs}$ (left) and work units (right). . . . .	73
4.20	Computational domain for the two dimensional supersonic wedge.	75
4.21	Pressure contours for $P_1$ (left plot), $P_3$ (middle plot) and $P_5$ elements (right plot), using $csc = 0.1$ . . . . .	76
4.22	Comparison between exact and $P_1$ , $P_3$ and $P_5$ solutions, along the $y = 0.3$ line, using $csc = 0.1$ . . . . .	77
4.23	Undershoots and overshoots of the pressure upstream (left plot) and downstream (right plot) the shock. Exact solution versus approximate solutions computed along the $y = 0.3$ line using $P_1$ , $P_3$ and $P_5$ elements, with $csc = 0.1$ . . . . .	77
4.24	Comparison between exact and $P_5$ solution along the $y = 0.3$ line, for $csc = 0.1$ and 2. . . . .	78
4.25	Undershoot and overshoot of pressure upstream (left plot) and downstream the shock (right plot). Exact solution versus $P_5$ solutions computed along the $y = 0.3$ line using $csc$ 0.1 and 2.	78
4.26	Behaviour of the differences between FD and AD sensitivities as a function of the perturbation step $\delta\alpha$ , for $csc = 0.1$ and $P_1$ elements (left plot) ,magnified view (right plot) . . . . .	81
4.27	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_1$ elements and an initial step size of 100. $csc = 0.1$ (top row), $csc = 0.05$ (middle row) and $csc = 0.01$ (bottom row). . . . .	83
4.28	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_2$ elements and an initial step size of 100. $csc = 0.1$ (top row) and $csc = 0.05$ (bottom row). . . . .	84

4.29	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_3$ elements and an initial step size of 100. $csc = 0.1$ (top row) and $csc = 0.05$ (bottom row). . . .	85
4.30	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_1$ elements and $csc = 0.01$ . $step_i = 1$ (top row), $step_i = 0.01$ (bottom row). . . . .	86
4.31	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_2$ elements. $step_i = 1$ and $csc = 0.008$ (top row), $step_i = 0.01$ and $csc = 0.01$ (bottom row).	88
4.32	Behaviour of design variables (left column), cost function (middle column), and sensitivities (right column) versus the number of optimization cycles, using $P_3$ elements, $step_i = 1$ , $csc = 0.01$ .	88

# List of Tables

4.1	Engine head geometry . . . . .	61
4.2	Time-step size for different discretization levels as a function of the vortex period $T$ . . . . .	73
4.3	Flow conditions upstream (state 1) and downstream (state 2) the oblique shock . . . . .	75
4.4	Sensitivity of the cost function with respect to $\alpha_1$ , computed using finite difference and automatic differentiation methods as a function of the polynomial degree. . . . .	80
4.5	Sensitivity of the cost function with respect to $\alpha_2$ , computed using finite difference and automatic differentiation methods as a function of the polynomial degree. . . . .	80
4.6	Differences between FD and AD sensitivities for several perturbation steps, using $P_1$ elements and $csc = 0.1$ . . . . .	81

# Bibliography

- [1] B. Van Leer. Flux-vector splitting for the Euler equations. *Technical Report* 81-11, ICASE, 1981.
- [2] B. Van Leer. Upwind-difference methods for Aerodynamic problems governed by the Euler equations. *Lectures in Applied Mathematics*, 22, 1985.
- [3] P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, *Journal of Computational Physics*, 43(2):357-372, 1981.
- [4] P. L. Roe, Characteristic-based schemes for the Euler equations, *Annual Review of Fluid Mechanics*, 18:337-365, 1986.
- [5] B. Van Leer, J. Thomas, P. Roe, R. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations. *AIAA Paper*, 87-1104, 1987.
- [6] A. Harten, B. Engquist, S. Osher, S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes III. *Journal of Computational Physics*, 71:231, 1987.
- [7] S. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16-42, 1992.
- [8] B. V. Leer. Towards the ultimate conservative difference scheme V. A second order sequel to Godunovs method. *Journal of Computational Physics*, 32:101-136, 1979.

- [9] M. Visbal, D. Gaitonde. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics*, 181(1):155-185, 2002.
- [10] R. Abgrall, A. Larat, M. Ricchiuto. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *Journal of Computational Physics*, 230(11):4103-4136, 2011.
- [11] T. Barth, P. Frederickson. High-order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper*, 90-0013, 1990.
- [12] B. Cockburn, C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws ii: general framework. *Mathematics of Computation*, 52:411-435, 1989.
- [13] F. Bassi, S. Rebay. 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.
- [14] B. Cockburn, S. Hou, C. W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*, 54:454-581, 1990.
- [15] R. Hartmann, L. Held, T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the rans and  $k-\omega$  turbulence model equations. *Journal of Computational Physics*, 230(11):4268-4284, 2011.
- [16] C. S. Chou, C. W. Shu. High order residual distribution conservative finite difference WENO schemes for convection-diffusion steady state problems on non-smooth meshes. *Journal of Computational Physics*, 224:992-1020, 2006.
- [17] F. Lorcher, G. Gassner, C. D. Munz, A Discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension, *Journal of Scientific Computing*, 32:175-199, 2007.

- [18] F. Bassi, A. Crivellini, D. A. Di Pietro, S. Rebay. A high-order Discontinuous Galerkin solver for 3D aerodynamic turbulent flows. *Proceedings of the Conference ECCOMAS CFD*, 2006.
- [19] C.W. Shu, S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439-471, 1988.
- [20] A. Burbeau, A. Sagaut, C.H. Bruneau. A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 169(1):111-150, 2001.
- [21] R. Hartmann. Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. *International Journal for Numerical Methods in Fluids*, 72:883894, 2013.
- [22] F. Bassi, L. Botti, A. Colombo, A. Ghidoni, S. Rebay. Discontinuous Galerkin for Turbulent Flows. *Adaptive High-Order Methods in Computational Fluid Dynamics, Advances in Computational Fluid Dynamics, World Science Books*. Ed. Z. J. Wang, 2011.
- [23] C. W. Shu, S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. II. *Journal of Computational Physics*, 83(1):32-78, 1989.
- [24] P. Persson, J. Peraire. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. *American Institute of Aeronautics and Astronautics*, 10.2514/6.2006-112, 2006.
- [25] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*, Elsevier Science, 460, Hardbound, 2001.
- [26] M. Germano, U. Piomelli, P. Moin, W. H. Cabot. A dynamic sub-grid scale eddy viscosity model. *Physics of Fluids*, A(3):17601765, 1991.

- [27] P. R. Spalart. Comments on the feasibility of LES for wing and on a hybrid RANS/LES approach, *1st ASOSR CONFERENCE on DNS/LES*. Arlington, TX, 1997.
- [28] T. Gatski, J. Bonnet. Compressibility, Turbulence and High Speed Flow. Elsevier, Amsterdam, 2009.
- [29] K. Hanjalic. Will RANS Survive LES? A View of Perspectives. *Journal of Fluids Engineering*, 127:831-839, 2005.
- [30] C. H. Son, T. A. Shethaji, C. J. Rutland, H. Barths, A. Lippert, S.H. El Tahry. Application of nonlinear turbulence models in an engine-type flow configuration. *International Journal of Engine Research*, 8:449-464, 2007.
- [31] S. H. El Tahry, D. C. Haworth, Directions in turbulence modeling for in-cylinder Flows in reciprocating IC engines. *AIAA Journal of Propulsion and Power*, 8:1040-1048, 1992.
- [32] S. H. El Tahry, D. C. Haworth. A perspective on the state-of-the-art in IC engine combustion modeling. *Proc. SIAM Sixth International Conference on Combustion*, New Orleans-LA, 1996.
- [33] B. E. Launder, G. J. Reece, W. Rodi. Progress in the Development of a Reynolds-Stress Turbulence Closure, *Journal of Fluid Mechanics*, 68(3):537-566, 1975.
- [34] W. Rodi. The prediction of free turbulent boundary layers by use of a two equation model of turbulence, PhD thesis, University of London, 1972.
- [35] W. Rodi. A new algebraic relation for calculating the Reynolds stresses. *Z. Angew. Math. Mech.* 56, T219221, 1976.
- [36] S. Wallin, A. Johansson. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows, *Journal of Fluid Mechanics*, 403:89-132, 2000,

- [37] T. B. Gatski, C. G. Speziale. On explicit algebraic stress models for complex turbulent flows, *Journal of Fluid Mechanics*, 254:59-78, 1993.
- [38] S. S. Girimaji, Fully-explicit and self-consistent algebraic Reynolds stress model. *Theoretical and Computational Fluid Dynamics*, 8:387-402 (also ICASE Rep. 95-82), 1996.
- [39] J. Boussinesq. Essai sur la thorie des eaux courantes, Mmoires presents par divers savants l'Acadmie des Sciences, 23(1):1-680, 1987.
- [40] G. M. Bianchi, S. Fontanesi. On the Applications of Low-Reynolds Cubic k-epsilon Turbulence Models in 3D Simulations of ICE Intake Flows, *SAE Paper* No. 2003-01-0003, 2003.
- [41] M. Auriemma, G. Caputo, F. E. Corcione, G. Valentino. Fluid Dynamic Analysis of the Intake System for a HDDI Diesel Engine by Star-CD Code and LDA Technique, *SAE Paper* No. 2003-01-0002, 2003.
- [42] A. Algieri, S. Bova, C. De Bartolo, A. Nigro. Numerical and experimental analysis of the intake flow in a high performance four-stroke motorcycle engine: influence of the two-equation turbulence models, *Journal of engineering for gas turbines and power*, 129 (4), 1095-1105, 2007.
- [43] B. E. Launder, D. B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3:269-289, 1974.
- [44] D. B. Spalding. A Single Formula for the Law of the Wall. *ASME J. Applied Mechanics*, 28:455-458, 1961.
- [45] R. Fletcher. A new approach to variable metric algorithms, *Computer Journal*, 13:317-322, 1970.
- [46] S. S. Rao. Game theory approach for multiobjective structural optimization. *Computers & Structures*, 25(1):119-127, 1987.

- [47] I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
- [48] H. P. Schwefel. *Numerical optimization for computer models*. John Wiley, Chichester, 1981.
- [49] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan, Ann Arbor, 1975.
- [50] J. Macek, O. Vtek, M. Polsek, M. Valsek, Z. Sika. Global and local optimization of transient diesel engine operations by combined one and three dimensional simulations. *Proc. THIESEL 2008 Conference on Thermo-and Fluid Dynamic Processes in Diesel Engines*, Valencia, 2008.
- [51] E. Rask, M. Sellnau. Simulation-based engine calibration: tools, techniques, and applications. *SAE Technical Paper*, 2004-01-1264, 2004.
- [52] Y. Shi, R. D. Reitz. Optimization study of the effects of bowl geometry, spray targeting and swirl ratio for a heavy-duty diesel engine operated at low-and high-load, *International Journal of Engine Research*, 9(4):325-46, 2008.
- [53] T. Ganapathy, K. Murugesan, R. P. Gakkhar. Performance optimization of Jatropha biodiesel engine model using Taguchi approach. *Applied Energy Journal*, 86:2476-86, 2009.
- [54] B. Saerens, J. Vandersteen, T. Persoons, J. Swevers, M. Diehl, E. Van den Bulck. Minimization of the fuel consumption of a gasoline engine using dynamic optimization. *Applied Energy Journal*, 86:1582-8, 2009.
- [55] M. Gumus, M. Atmaca, T. Yilmaz. Efficiency of an Otto engine under alternative power optimizations. *International Journal of Energy Research*, 33(8):74552, 2009.

- [56] K. Atashkari, N. Nariman-Zadeh, M. Golcu, A. Khalkhali, A. Jamali. Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms. *Energy Conversion and Management Journal*, 48(3):1029-41, 2007.
- [57] N. Fateh, S. Parashar, J. Silvestri. Game theory approach to engine performance optimization, *SAE Technical Paper*, 2008-01-0871, 2008.
- [58] G. Carpentieri, B. Koren, M. J. L. van Tooren. Adjoint-based aerodynamic shape optimization on unstructured meshes, *Journal of Computational Physics*, 224(1):267-287, 2007.
- [59] S. B. Hazra, V. Schulz, J. Brezillon, N. R. Gauger. Aerodynamic shape optimization using simultaneous pseudo-timestepping, *Journal of Computational Physics*, 204:46-64, 2005.
- [60] J. Brezillon, N. R. Gauger. 2D and 3D aerodynamic shape optimisation using the adjoint approach, *Aerospace Science and Technology*, 8 (8):715-727, 2004.
- [61] A. Fazzolari, N. R. Gauger, J. Brezillon. Efficient aerodynamic shape optimization in MDO context, *Journal of Computational and Applied Mathematics*, 203(2):548-560, 2007.
- [62] Li Wang, W. Kyle Anderson. Shape Sensitivity Analysis for the Compressible Navier-Stokes Equations Via Discontinuous Galerkin Methods, *Computers & Fluids*, 69:93-107, 2012.
- [63] Li Wang, D. J. Mavriplis, W. Kyle Anderson. Adjoint Sensitivity Formulation for Discontinuous Galerkin Discretizations in Unsteady Inviscid Flow Problems, *AIAA Journal*, 48(12):2867-2883, 2010.
- [64] M. B. Giles, N. A. Pierce. An Introduction to the Adjoint Approach to Design, *Flow, Turbulence and Combustion*, 65:393-415, 2000.

- [65] F. Beux, A. Dervieux. Exact-gradient shape optimization of a 2-D Euler flow, *Finite Elements in Analysis and Design*, Elsevier, 12:281-302, 1992.
- [66] I. Babuvka, W. C. Rheinboldt. Error estimates for adaptive finite element computations, *SIAM Journal on Numerical Analysis*, 1978.
- [67] B. Mohamadi, O. Pironneau. Applied shape optimization for fluids, Oxford University Press, 2000.
- [68] O. Pironneau. On optimum design in fluid mechanics, *Journal of Fluid Mechanics*, 64 (01):97-110, 1974.
- [69] L. Wang, D. J. Mavriplis, Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations, *Journal of Computational Physics* 228(20):76437661, 2009.
- [70] C. Darwin. On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life. John Murray, London, 1859.
- [71] J. Nocedal, S. J. Wright. Numerical Optimization, Springer Series in Operational Research, Springer, 1999.
- [72] L. B. Rall. Automatic Differentiation - Techniques and Applications, Springer-Lecture Notes in Computer Science, 120, 1981.
- [73] A. Griewank. Evaluating Derivatives: Principles and techniques of Algorithmic Differentiation, Society for Industrial and Applied Mathematics, Philadelphia-USA, 2000.
- [74] S. Gottlieb, A. Orszag. Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, Philadelphia, 1977.
- [75] S. Gottlieb, J. S. Hesthaven. Spectral Methods for Hyperbolic Problems, *Journal of Computational and Applied Mathematics*, 128(1-2):83-131,2001.

- [76] I. Babuska, B.A. Szabo, and I.N. Katz. The p-Version of the Finite Element Method. *SIAM Journal on Numerical Analysis*, 18(3):515-545, 1981.
- [77] J. Ekaterinaris. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41:192-300, 2005.
- [78] P. Houston, E. Suli. A Note on the Design of hp-Adaptive Finite Element Methods for Elliptic Partial Differential Equations. *Computer Methods in Applied Mechanics and Engineering* 194(2-5):229-243, 2005.
- [79] T. Leicht, R. Hartmann. Error Estimation and hp-Adaptive Mesh Refinement for Discontinuous Galerkin Methods. Adaptive High-Order Methods in Computational Fluid Dynamics, *World Science Books*, Ed. Z. J. Wang, 2(3):6794, 2011.
- [80] Z. J. Wang. High-order methods for the Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43:1-41, 2007.
- [81] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, K. Sorensen. ADIGMA - A European Initiative on the Development of Adaptive Higher-order Variational Methods for Aerospace Applications. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Springer, 113, 2010.
- [82] N. Kroll. European project on Industrialization of High-Order Methods for Aeronautical Applications. *Proceedings of the ECCOMAS 2012*, September 10-14, Vienna, Austria, 2012.
- [83] W. H. Reed, T. R. Hill. Triangular mesh methods for the neutron transport equation. *1em Technical Report LA-UR-73-479*, Los Alamos Scientific Laboratory, 1973.
- [84] C. Johnson, J. Pitkarata. An Analysis of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation, *Mathematics of Computation*, 46:1-26, 1986.

- [85] B. Cockburn, S. Y. Lin, C. W. Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One Dimensional Systems. *Journal of Computational Physics*, 84:90-113, 1989.
- [86] G. Jiang, C. W. Shu. On Cell Entropy Inequality for Discontinuous Galerkin Method for Hyperbolic Equations. *AIAA Journal*, 36:775782, 1998.
- [87] F. Bassi, A. Crivellini, S. Rebay, M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged NavierStokes and  $k - \omega$  turbulence model equations, *Computers & Fluids*, 34:507-540, 2005.
- [88] G. Y. Psihoyios. Advanced step-point methods for the solution of initial value problems, PhD thesis, University of London Imperial College of Science and Technology, 29/12/1995.
- [89] G. Y. Psihoyios, J. R. Cash. A stability result for general linear methods with characteristic function having real poles only, *BIT Numerical Mathematics*, (ISSN 00063835), 38(3):612617, 1998.
- [90] A. Nigro, C. De Bartolo, F. Bassi, A. Ghidoni. Up to sixth-order accurate A-stable implicit schemes applied to the discontinuous Galerkin discretized Navier-Stokes equations, *Journal of Computational Physics*, 276:136162, 2014.
- [91] A. Nigro, A. Ghidoni, S. Rebay, F. Bassi. Modified Extended BDF scheme for the discontinuous Galerkin solution of unsteady compressible flows, *International Journal for Numerical Methods in Fluids*, 76(9):549-574, 2014.
- [92] A. Nigro, S. Renda, C. De Bartolo, R. Hartmann, F. Bassi. A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows, *International Journal for Numerical Methods in Fluids* 72(1):43-68, 2013.

- [93] L. Hasco, V. Pascual. The Tapenade Automatic Differentiation tool: principles, model, and specifications, *ACM Transaction on mathematical software*, 39(3), 2013.
- [94] L. Hasco, V. Pascual. Tapenade 2.1 users guide, *Technical Report 0300*, INRIA, 2004. <http://www-sop.inria.fr/tropics/>.
- [95] D. C. Wilcox. Turbulence Modeling for CFD. 3rd edition, DCW Industries, Inc., La Canada CA, 2006.
- [96] F. Bassi, A. Crivellini, S. Rebay, M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and  $k - \omega$  turbulence model equations. *Computers & Fluids*, 34:507-540, 2005.
- [97] F. Bassi, L. Botti, A. Colombo, A. Crivellini, N. Franchina, A. Ghidoni, et al. Very high-order accurate discontinuous Galerkin computation of transonic turbulent flows on aeronautical configurations. In: ADIGMA a European initiative on the development of adaptive higher-order variational methods for aerospace applications”. *Notes on numerical fluid mechanics and multidisciplinary design*, Springer, 113:25-38, 2010.
- [98] F. Bassi, F. Cecchi, N. Franchina, S. Rebay, M. Savini. High-order discontinuous Galerkin computation of axisymmetric transonic flows in safety relief valves. *Computers & Fluids*, 49:203-213, 2011.
- [99] F. Bassi, S. Rebay. High-order accurate discontinuous Galerkin methods in computational fluid dynamics: from model problems to complex turbulent flows-Part 1 35Th CFD/Adigma Course on Very high order discretization methods, edited by H. Deconinck; 2008.
- [100] E. Toro. Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, 2009.
- [101] B. van Leer. Flux vector splitting for the Euler equations, *Lecture notes in physics*, 170:507-512, 1982.

- [102] D. Hanel, R. Schwane, G. Seider. On the accuracy of upwind schemes for the resolution of the Navier-Stokes equations, *AIAA Paper*, 87-1005, 1987.
- [103] K. L. Bibb, J. Peraire, C. J. Riley. Hypersonic flow computations on unstructured meshes, *AIAA Paper*, 97-0625, 1997.
- [104] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications, *AIAA Journal*, 32(8):1598-1605, 1994.
- [105] R. J. Spiteri, S. J. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods, *Technical Report CS-2001-01*, Acadia University, Wolfville, Nova Scotia, Canada, 2001.
- [106] J. C. Butcher. Numerical Methods for Ordinary Differential Equations, John Wiley & Sons, New York, 2003.
- [107] J. R. Cash. The integration of stiff initial value problems in ODEs using Modified Extended Backward Differentiation Formulae. *Computers & Mathematics with Applications*, 9(5):645-657, 1983.
- [108] J. R. Cash. On the Integration of Stiff Systems of O.D.E.s Using Extended Backward Differentiation Formulae. *Numerische Mathematik*, 34:235-246, 1980.
- [109] G. Psihoyios. A general formula for the stability functions of a group of Implicit Advanced Step-point (IAS) methods. *Mathematical and Computer Modelling*, 46(1-2):214-224, 2007.
- [110] C. F. Curtiss, J. O. Hirschfelder. Integration of stiff equation. *Proceedings of the National Academy of Sciences, USA*, 38:235-243, 1952.
- [111] W. Gear. Simultaneous Numerical Solution of Differential Algebraic Equation. *IEEE Transaction on Circuit Theory*, 18:89-95, 1971.
- [112] U. Ascher, L. Petzold. Computer Method for Ordinary Differential Equations and Differential Algebraic Equations. *Siam*, 1998.

- [113] R. Courant, K. O. Friederichs, H. Lewy. Uber die partiellen Differenzengleichungen der Physik. *Mathematische Annalen*, 100:32-74, 1928.
- [114] Y. Saad, M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific Computing*, 7:856-869, 1986.
- [115] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, H. Zhang. PETSc Web page <http://www.mcs.anl.gov/Petsc>; 2001.
- [116] G. Karypis, V. Kumar. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs, *SIAM Journal on Scientific Computing*, 20(1):359-392, 1999.
- [117] C. G. Broyden. The convergence of a class of double rank minimization algorithms:2:The new algorithm, *Journal of the Institute of Mathematics and its Applications*, 6:222-231, 1970.
- [118] D. Goldfarb. A family of variable metric methods derived by variational means, *Mathematics of Computation*, 24:23-26, 1970.
- [119] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation*, 24:647-650, 1970.
- [120] F. Beux F, A. Dervieux. Exact-gradient shape optimization of a 2-D Euler flow. *Finite Elements in Analysis and Design*, 12:281-302, 1992.
- [121] H. Hu. Development of a sensitivity derivative version of the FPX CFD code via automatic differentiation. *Advances in Engineering Software*, 30(4):273-279, 1999.
- [122] S. Kim, J. J. Alonso, A. Jameson. A gradient accuracy study for the adjoint-based Navier-Stokes design method. *AIAA Paper 99-0299*, 1999.

- [123] P. A. Dellenback, D. E. Metzger, G. P. Neitzel. Measurements in Turbulent Swirling Flow Through an Abrupt Axisymmetric Expansion. *AIAA journal*, 26(6), 1988.
- [124] D. L. Darmofal, P. Moinier, M. B. Giles. Eigenmode analysis of boundary conditions for the one-dimensional preconditioned Euler equations. *Journal of Computational Physics*, 160:369-384, 2000.
- [125] L. Graftieaux, M. Michard, N. Grosjean. Combining PIV, POD and vortex identification algorithms for the study of unsteady turbulent swirling flows. *Measurement Science & Technology*, 12, 2001.
- [126] L. Thobois, G. Rymer, T. Souleres, T. Poinso. Large-Eddy Simulation IC Engines Geometries, *SAE Technical Paper*, 2004-01-1854, 2004.
- [127] C. Geuzaine, J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309-1331, 2009.
- [128] S. Parpais, H. Touil, J. P. Bertoglio, M. Michard. A spectral closure for inhomogeneous turbulence applied to the computation of an engine related flow, *Engineering Turbulence Modelling and Experiments-4*, W. Rodi and D. Laurence, 1999.
- [129] Z.J. Wang et. al. High-Order CFD Methods: Current Status and Perspective, *International Journal for Numerical Methods in Fluids*, 00:1-42, 2012. Published online in Wiley InterScience (www.interscience.wiley.com).
- [130] web page: [www.ipacs-benchmark.org](http://www.ipacs-benchmark.org).
- [131] A. H. Shapiro. The Dynamics and Thermodynamics of Compressible Fluid Flow, The Ronald Press Company, New York, 1954.