Three-dimensional discontinuous spectral/hp element methods for compressible flows

by

Daniele De Grazia

Department of Aeronautics
Imperial College London

This thesis is submitted for the degree of Doctor of Philosophy of Imperial College London

2016

“Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza.”

Dante Alighieri,
Inferno, Canto XXVI.

Declaration

This is to certify that the work presented in this thesis has been carried out at Imperial College London and has not been previously submitted to any other university or technical institution for a degree or award. The thesis comprises only my original work, except where due acknowledgement is made in the text.

Daniele De Grazia

Acknowledgements

I would like to thank all the people who have made this journey towards the PhD a memorable experience.

First, I would like to thank my supervisor, Professor Spencer Sherwin, who gave me this great opportunity. Professor Sherwin introduced me to the fascinating world of CFD and patiently guided throughout my research path making this thesis possible.

I am also extremely grateful to my co-supervisor Dr Peter Vincent for all the support he provided during my PhD years. The many discussions we had about high-order methods were extremely fruitful and I have really enjoyed collaborating with him.

I would also like to express my sincere gratitude to Dr. David Moxey who has never withdrawn his human and technical support during these years. I wish him all the best for his career.

I would also like to thank Professor Anatoly Ruban for all the help and assistance and Dr Marina Kravtsova for all the fruitful discussions and suggestions.

I have greatly enjoyed the company of my colleagues and friends in Mathematics: Arshad, Calum Curtis, David, Emma, Hui, Jamie, Joe, Liam, Mat, Maximilian, Nikos, Radu, Raquel, Ruben, Thomas and Tomass as well as the time spent with my colleagues and friends in Aeronautics: Alessandro, Bastien, Chris, Dirk, Douglas, Gabriele, Justin, Laurent, Lisa, Olya, Rodrigo, Thibault.

A special thanks goes to my friend Gianmarco who was the first to convince me to undertake this PhD challenge which has been a great experience and with whom I shared a lot of unforgettable moments. I will always remember the good times we had in the lab and in London.

I am also extremely grateful to my friends Jean and Mario who helped me to keep my sanity during these research years and made the PhD more enjoyable.

I would also like to thank all the friends and relatives in Italy which have always made me feel close to home during these years.

I am extremely grateful to my parents and my brother for their continuous support. All of this would have never been possible without all their love and support.

Above all, I thank Fabiana. She always encouraged me to pursue my passions and has always been supportive during these years convincing me to not give up, so I would like to dedicate this thesis to her.

Abstract

In this thesis we analyse and develop two high-order schemes which belong to the class of discontinuous spectral/hp element methods focusing on compressible aerodynamic studies and, more specifically, on boundary-layer flows. We investigate the discontinuous Galerkin method and the flux reconstruction approach providing a detailed analysis of the connections between these methods. The connections found enable a better understanding of the broader class of discontinuous spectral/hp element methods.

From this perspective it was evident that some of the issues of the discontinuous Galerkin method are also encountered in the flux reconstruction approach, and in particular, the aliasing errors of the two schemes are identical. The techniques applied in the more famous discontinuous Galerkin method for tackling these errors can be also extended to the flux reconstruction approach. We present two dealiasing strategies based on the concept of consistent integration of the nonlinear terms. The first is a localised approach which targets in each element the nonlinearities arising in the problem, while the second is a more global approach which involves a higher quadrature of the overall right-hand side of the discretised equation(s). Both the strategies have been observed to be effective in enhancing the robustness of the schemes considered.

We finally present the direct numerical simulation of a high-speed subsonic boundary-layer flow past a three-dimensional roughness element, achieved by means of the compressible aerodynamic solver developed. This type of analyses have been widely performed in the past with approximated theories. Only recently, has DNS been used due to the improvement of numerical techniques and an increase in computational resources for similar studies in low-speed subsonic, supersonic and hypersonic regimes. This thesis takes a first step to close the gap between the results for a high-speed subsonic regime and the results in supersonic and hypersonic regimes.

Contents

List of Figures

List of Tables

Introduction

In this chapter we present an overview of the underlying foundations of this thesis. We successively detail the motivations and the objectives of this work as well as its main scientific and computational contributions. Finally we provide the outline of the manuscript.

Overview

In the past five decades computational fluid dynamics (CFD) has become a very powerful and versatile tool for the analysis and solution of problems related to different fields, such as engineering, physics, chemistry and medicine (biomedical flows). Since CFD began, one of the major challenges has been to increase the limit of CFD in terms of accuracy while reducing its computational cost. This will result in the ability to study more complex flows in a greater detail, as also stated in the seminal paper by Orszag and Israeli (1974).

Most commercial CFD software adopt low-order methods (methods for which the spatial order of accuracy is at most two) to simulate fluid flows over complex geometries. These have become extremely robust and efficient thanks to the enormous research effort spent in their development. However, especially for problems requiring a high level of accuracy, the low-order methods are inadequate.
In recent years there has been much interest in high-order methods as they offer the potential to obtain more accurate approximations with less computational cost compared to lower order methods. Several high-order methods have been developed for structured grids. These include finite difference schemes with extended stencils, compact finite difference schemes (Lele (1992), Kim and Lee (1996)), ENO and WENO schemes (Harten et al. (1987), Shu and Osher (1988, 1989), Liu et al. (1994)). These schemes have been adopted successfully to solve very challenging fluid flow problems (Nagarajan et al. (2003), Lele (1989, 1992), Visbal and Gaitonde (2002)). However, since they can only be used on structured grids, they are not suited for obtaining accurate solutions over complex geometries, which are extremely difficult to mesh with structured grids. Nowadays, a significant portion of CFD analysis time is spent generating grids, hence, the development of high-

order methods for unstructured grids is desirable, because the mesh generation process is easier and often automated for those grids. A great variety of high-order methods for unstructured grids have been developed in the last decades. Among them are methods in which solution reconstruction within an element is done using information from neighbouring elements. The idea behind those approaches is to use an extended stencil, as in high-order finite difference schemes on a structured mesh, to achieve high-order accuracy. Examples of such methods are k-exact finite-volume methods and finite-volume type ENO and WENO schemes (Barth and Frederickson (1990), Delanaye and Liu (1999), Abgrall (1994) Ollivier-Gooch (1997), Barth and Deconinck (1999)). These methods tend to be complex to implement, because in contrast to structured grids, the neighbouring cell data is not easily accessible. In addition since they use the extended stencils the parallelisation of these methods to run on multiple cores can be quite difficult.
Because of the aforementioned issues associated with high-order methods for unstructured grids that use extended stencils, in recent years, a new class of high-order numerical methods for the solution of partial differential problems which make use of a tessellation of the domain in separate elements (h-type refinement), with spectral-like resolution properties (p-type refinement) in each element, namely spectral/hp element methods, has gained significant attention, both in academia and industry. The reason for this increased interest is manifold. On one hand, as all the high order methods, spectral/hp element methods offer an arbitrary order of spatial accuracy, whilst maintaining a relatively lower computational cost (for a given error threshold one aims to achieve) than traditional low-order finite element schemes. On the other hand, they are well-suited for applications involving complex geometries, which can be a challenge for other high order methods. The combination of these two factors makes spectral/hp element methods competitive in various fields, including computational fluid dynamics (CFD), computational aero-acoustics (CAA) and magnetohydrodynamics (MHD), to cite just some of them (see Cockburn et al. (2000)). Specifically, the use of spectral/hp element methods is particularly tailored for CFD applications which involve bluff bodies, vortical flows and unsteady aerodynamics because of their low level of numerical diffusion. However, the use of these methods can also be advantageous for problems where a high spatial resolution is required, such as direct numerical simulation (DNS) or the large-eddy simulation (LES) of boundary-layer flows at high-Reynolds numbers. A spectral/hp element method traditionally involves the variational form of the governing equations and, inside each element of the tessellated domain, the definition of an arbitrary degree 'P' polynomial describing the solution onto a set of 'Q' solution points. The elements are then connected by means of some elemental coupling. This coupling can be applied in two different ways which result in two different families of spectral/hp element methods: 'continuous' and 'discontinuous' methods. The first family enforces C^0-continuity of

the solution between adjacent elements, while the second allows the solution to be discontinuous yet enforcing continuity on the fluxes by calculating a common interface flux commonly called numerical flux.

In this context the aim of this work is to develop and improve a comprehensive numerical tool to solve compressible aerodynamic problems (either inviscid or viscous) inside the spectral/hp element library Nektar++. From this perspective one of the objectives was to gain a deeper insight into the features of this class of methods in order to exploit all possible connections among the different schemes. The compressible aerodynamic solver developed should be capable to solve challenging complex fluid flow problems on complex unstructured grids and the method adopted must be *accurate* and *efficient*.
From the first perspective the order of accuracy indicates how rapidly errors decrease in the limit as space discretization (space accuracy) and time discretizazion (time accuracy) tend to zero. A scheme is said to be nth-order accurate in space if the associated solution error E is proportional to the mesh discretizaztion size Δx as:

$$E(h) = C_s \Delta x^n \, .$$

A scheme is said to be nth-order accurate in space if the associated solution error E is proportional to the step size Δt as:

$$E(h) = C_t \Delta t^n \, .$$

Even if in the most of thesis we refer to space accuracy, in time-dependent problems like the ones we are interested in, the time accuracy is as significant as the other (Chang et al. (2013), Chen et al. (2003) Cockburn and Shu (1998a), Cockburn and Shu (1998b)).
From the efficiency point of view we want to guarantee both parallel and numerical efficiency. The first tells us how fast our parallel implementation is close to the optimal speedup $S_C(P)$:

$$E_{C,\,par} = \frac{S_C(P)}{P} \, .$$

The desirable efficiency would be 1 (= 100%), i.e. the use of P processors accelerates the work by a factor P. The second compares the fastest sequential algorithm with the fastest parallel algorithm implemented on one processor via the relationship:

$$E_{C,\,num} = \frac{t_{parallel}}{t_{serial}} \, .$$

The scaled efficiency of a parallel algorithm can then be computed as:

$$E = E_{C,\,par} \cdot E_{C,\,num} \, .$$

In addition the algorithms of the solver must be: *robust, reliable and stable* where we use the following definitions:

- robust: the numerical method does not produce a widly different result for very small changes in the input data;
- reliable: when validated through comparison with other existing results (analytical/numerical/experimental) the results of the method have to be consistent;
- stable: numerical errors which are generated during the solution of discretized equations should not be magnified.

Specifically there are different definitions for stability and, for the one applied in this work, a numerical method is stable in a stability region Λ if, for any positive time T there is a constant $C_T > 0$ such that:

$$||u^n||_{\mathrm{L}_2} \leq C_T ||u^0||_{\mathrm{L}_2} \,,$$

for $0 \leq n\Delta t \leq T$, with $(\Delta t, \Delta x) \in \Lambda$ being Δt, Δx the temporal and spatial discretizations, u^0 the initial solution and $||\cdot||_{\mathrm{L}_2}$ the L_2 norm vastly used in the remainder of this thesis.
Therefore the objective of this work is to develop an effective compressible aerodynamic solver with all the aforementioned characteristics. This tool must be capable to solve the Euler and Navier-Stokes equation on unstructured complex grids. This thesis is concerned with 'discontinuous' spectral/hp element methods and their applications to compressible boundary-layer flows. Specifically we aim to use the tool developed to study a boundary layer past a three-dimensional roughness element.

In the next section we will provide a brief account of the history of the principal discontinuous spectral/hp element methods which have constituted the foundations of this work.

Discontinuous spectral/hp element methods

There are three main classes of discontinuous spectral/hp element methods, namely the discontinuous Galerkin method, the spectral difference scheme and the flux reconstruction approach. In this thesis the first and the third method have been explored in detail, but a brief introduction of the spectral difference scheme is also provided since it can be recast within the broader class of discontinuous spectral/hp element methods and it can be included in the flux reconstruction framework as we shall see in chapter 1.

One of the most widely adopted discontinuous spectral/hp element methods is the so called *discontinuous Galerkin* (DG) method originally proposed by Reed and Hill (1973) to solve the neutron transport equation. The DG scheme has undergone significant development in various fields during the years. An important

milestone is the work by Cockburn et al (1989b) who, for the first time, successfully applied the Runge-Kutta schemes to the DG method. The same authors have later provided a more general framework (Cockburn and Shu (1989); Cockburn et al. (1990)) and they applied the concept to more complex hyperbolic system of conservation laws including the compressible Euler equations (Cockburn and Shu (1998b)). The DG method has been also applied to elliptic problems; the work by Bassi and Rebay (1997) represents the first of a series of papers where a second order equation is split into a system of two first order equations. In this way the compact and discontinuous nature of the DG scheme is maintained also for second order problems. Based on this concept, other works include (Cockburn and Shu (1998a)), where the well-known local discontinuous Galerkin (LDG) approach is presented for the first time and (Arnold et al. (2001)), where the authors provide a unified analysis of the discontinuous Galerkin method applied to elliptic equations. Classically, the DG method starts with an expression of the underlying problem in a variational form, in which the governing system of equations is multiplied by a test function and integrated over the domain, which is itself partitioned into non-overlapping elements. The DG discretisation then allows a selection of both a basis of polynomials that represent the solution locally on each element and a set of quadrature points on which the inner products arising in the variational formulation can be calculated. In the DG method, the integration of the non-linear terms can be numerically exact (this usually produces a computationally costly implementation) or can be inexact; this is the case of collocation based DG methods, where the use of a collocation projection of the flux function allows a reduction of the computational costs. Examples of collocation-based DG methods are the so-called nodal DG schemes (Hesthaven and Warburton (2008)), whereby Lagrange interpolants are combined with a set of nodal solution points on a given element, producing an efficient implementation of the DG method.

While the class of DG methods is probably the most well-known and widely adopted, more recently, another discontinuous spectral/hp element method is becoming increasingly popular. This relatively recent approach, namely the *spectral difference* (SD) scheme, was first introduced by Kopriva and Kolias (1996) under the name of 'staggered grid Chebyshev multi domain' method, and has been further developed some years later by Liu et al. (2006), who applied the method to both triangular and quadrilateral elements and named it the SD scheme. The SD scheme is conceptually similar to the DG method but is based on the differential form of the problem. This aspect has offered an alternative route that avoids the need for quadrature rules, making these schemes easier and potentially more efficient to implement.

The most recent discontinuous spectral/hp element method is however the *flux reconstruction* (FR) approach, first proposed by Huynh (2007). This approach is also based on the differential form of the governing equations and shares the

same benefits as the SD scheme, potentially more efficient than variational-based methods, where there is the necessity of calculating integrals. As shown by Huynh, the FR approach can recover a wide range of numerical schemes including the SD and the nodal DG methods (at least for linear problems). This approach has been significantly developed in the past few years. Vincent et al. (2011a), for instance, have identified a new class of energy-stable FR schemes, referred to as Vincent-Castonguay-Jameson-Huynh (VCJH) schemes on quadrilateral tensor product meshes. Here, a single real-valued parameter 'c' dictates the scheme that is recovered and enables one to recover differential and variational based schemes, such as a particular SD method or a nodal DG scheme. The same authors, have extended the VCJH schemes to triangular elements (Castonguay et al. (2011b)) and applied this new class of energy-stable schemes to the compressible Euler equations (Castonguay et al. (2011a)). They have also gained some insights into the dispersion and dissipation properties of the VCJH schemes (Vincent et al. (2011b)) as well as into the nonlinear stability properties of these schemes (Jameson et al. (2012)). Some extensions to elliptic problems have also been presented in the literature, see for example (Williams et al. (2011)), where the FR approach has been applied to the compressible Navier-Stokes equations.

In the next section we will provide a brief literature review of the main approaches applied for boundary-layer flow studies and, specifically, for the analysis of flows past roughness elements which is the final goal of this thesis work.

Applications to boundary-layer flows

As briefly mentioned above, discontinuous spectral/hp element methods are particularly suited for CFD applications where a high spatial resolution and favourable dispersion and dissipation properties are required. Some examples of recent applications of this class of methods in CFD are (Landmann et al. (2008); Uranga et al (2011); Bolemann et al. (2015)). Recently, with the increase of computational resources, compressible boundary-layer flows at high-Reynolds numbers have started to represent a potentially interesting area of applicability of discontinuous spectral/hp element methods. In fact, the correct analysis of this type of flows usually requires a very high spatial resolution and computationally efficient algorithms in order to provide an accurate solution in a reasonable amount of time.

Particularly interesting from this point of view is the analysis of the laminar-turbulent transition scenario. Currently a large number of tools to investigate the laminar-turbulent transition exists. The most significant methods which deal with this type of problems are related to the linear stability theory Reed et al. (1996). It involves the decomposition of the primitive variables of the Navier-Stokes equations into a mean and a perturbation part (small compared to the mean quantity). This allows the linearisation of the Navier-Stokes equations with different approx-

imations depending on the flow configuration. In this framework the local and non-local stability theories are the two main approaches. The first approach is used in presence of locally quasi-parallel flows, while the second is typically used to analyse non-parallel flows. The Parabolized Stability Equations (PSE) (Thorwald (1997)) is the principal tool in the context of non-local stability theory.
Specifically, in this thesis, the focus is on the study of flows past roughness elements, steps and gaps. These features are very common in aircraft wings and their implications in the onset mechanisms of laminar-turbulent transition are crucial for reducing the skin-friction drag - one of the largest targets of many aircraft manufacturers, including Airbus, Boeing and Embraer. This class of problems have been widely studied in the past by means of various methodologies, such as the asymptotic triple-deck theory (Smith (1973); Rizzetta et al. (1978)), experiments (Klebanoff and Tidstrom (1972)) and DNS (Rizzetta and Visbal. (2007); Rizzetta et al. (2010); Rizzetta and Visbal. (2014)) just to cite a few. As we can see, only recently, with the increase of computational resources the DNS has started to be used and can become a powerful tool to study the boundary-layer scenario and to validate previous models that are used for predicting the separation mechanism and location. Triple-deck theory is more suitable to study the transition mechanism when non-linear effects are small and therefore its accuracy decreases as the non-linearity in the flow increases. Hence, in three-dimensional boundary layers, the DNS of the equations governing the flow is the most powerful tool to investigate the transition scenario since DNS can capture all the aspects of the transition process without using any approximation models. Most of the DNS studies of boundary-layer flows over roughness element performed in the past are concerned with low-speed subsonic, supersonic and hypersonic regime. In these studies the high-speed subsonic regime which is typical of civil aeronautical industry has not yet been considered. A more detailed review of the studies of flow past roughness element that have been performed in the past can be found in chapter 5.

In the next section we will describe the main motivations and objectives behind this work which is part of the 'Laminar Flow Control UK' (LFC-UK) project (http://www.imperial.ac.uk/lfc-uk) at Imperial College London (ICL).

Motivations and objectives

The primary aim of 'LFC-UK' project is the development of underpinning technology for laminar flow control in a compressible regime, the technology of drag reduction on aircraft. The programme is based around a team of researchers covering all theoretical, experimental and computational aspects of the problem. From the last perspective an efficient numerical software for DNS and LES simulations is unavoidable and, at the beginning of this thesis work, the Nektar++

open source library developed in the SherwinLab (http://sherwinlab.ae.ic.ac.uk) at Imperial College, was considered as the possible tool for such simulations. In the SherwinLab, of which the author is a member, a robust effort has been made in the past few years to develop a comprehensive numerical tool in order to solve a wide range of partial differential problems. These efforts have come together in the creation of the C++ object-oriented spectral/hp element library Nektar++ (http://www.nektar.info). The Nektar++ project began ten years ago with the main scope of developing a flexible and efficient spectral element framework to solve various differential problems, with a special focus on fluid dynamics. From a DNS/LES perspective it was crucial to develop the core building blocks to produce a reliable, accurate and efficient numerical tool for compressible aerodynamics, using cutting-edge techniques and developing novel numerical technologies. The library, at the beginning of this work, included a well-developed implementation of the continuous Galerkin (CG) method applied to the incompressible Navier-Stokes equations, and a relatively well-developed DG method applied to the compressible Euler equations. The relatively new discontinuous spectral/hp element methods based on the differential form of the governing equations, namely the SD and FR approaches, had been consistently promoted, especially at the beginning of this work, as potentially competitive compared to the well established DG schemes for compressible flow applications. Given the vast variety of discontinuous spectral/hp element methods and their development in different time frames, with the DG method being the oldest and probably the most established, it was crucial to understand whether there are some connections between these relatively similar schemes and, if this is the case, how we can use these connections to attain some advantages. The FR approach in particular, given its ability of recovering a broad range of existing schemes, including the SD and DG methods, was a very attractive option and its integration within the Nektar++ library provided a solid starting point from which we could compare the two schemes within the same numerical environment and possibly produce new strategies for further developing and improving these methods. From the last perspective, an interesting topic of research was studying what connections exist between the DG method and the FR approach. Although some literature on this topic already existed, this was confined to linear problems and the connections between the two numerical frameworks for nonlinear equations constituted an important gap which prevented to fully understand which the implications of adopting one or the other approach were for more complex applications.

From an engineering perspective, another important topic of research of this thesis, was constituted by the numerical stability of discontinuous spectral/hp element methods and the development of effective strategies for improving it. In fact, one of the aims was to construct a reliable numerical framework based on this class of methods for the DNS/LES of compressible flows. Therefore, understanding the

connections between the two classes of schemes was a promising candidate to gain further insights into the underlying numerics and into the strategies for improving the numerical stability, yet maintaining the advantages in terms of efficiency of these schemes. The overarching goal of the research project was to understand the state-of-the-art of the current numerical tools in the context of discontinuous spectral/hp element methods used in compressible aerodynamics, analyse their intrinsic connections, investigate strategies for solving the existing shortcomings, with the ultimate goal of deploying a robust, reliable and efficient discontinuous spectral/hp element numerical framework for compressible aerodynamics. To prove the effectiveness of the new numerical tool developed, the additional, yet no less important, goal regarded providing some DNS results (relevant to the LFC-UK group) which resulted in the DNS of a flow past a roughness element in transonic conditions. This work starts to address the lack of DNS data of boundary-layer flows past roughness elements in a high-speed subsonic regime which is typical of civil aeronautical applications.

Current study: scientific and numerical contributions

This research has contributed to the development of the Nektar++ project in the SherwinLab at ICL and, as part of this work, different research topics were investigated. Specifically, considerable research effort was made in exploring the connections between existing and novel discontinuous spectral/hp element methods. As mentioned above, one the principal issue of this class of schemes is the lack of robustness and stability which is one of the reason preventing their use in industrial environments so far. Different strategies to improve the numerical stability of this class of schemes have been studied. In addition, to prove the effectiveness of the CFD tool developed some aerodynamic problems concerning separated flows were explored with the final aim of a direct numerical simulation analysis of a boundary-layer flow over a three-dimensional hump. These efforts have resulted in a series of journal publications and conference presentations. In the following, we highlight the main contributions of the thesis to the existing literature and we briefly present the additional features which have been integrated into the Nektar++ library as part of this thesis.

Connections between DG method and FR schemes

All the numerical schemes presented above are similar, although the discontinuous Galerkin method is based on the variational form of the equations while the spectral difference and flux reconstruction are based on the differential form. All these

schemes constitute a wider class of discontinuous spectral/hp element methods and therefore it seems reasonable to ask what connections exist between these schemes and how we can exploit them. Regarding this topic some publications can be found in the literature; as already mentioned, Huynh (2007) showed that the FR approach can recover a particular SD or DG scheme for one-dimensional problems. Allaneau and Jameson (2011) have gone further, showing that, the VCJH schemes can be seen as special DG schemes, where a linear filter is applied to the right-hand side of the DG method. However, these works are restricted to linear problems and the underlying connections between the FR approach and the DG method for nonlinear problems on generic tensor product irregular meshes (i.e. meshes formed by deformed or curvilinear elements) have so far remained unexplored.

In this work we have throughly investigated the connections between the DG and the FR schemes. The original idea came from a gap in the literature regarding the connections between the DG method and the FR schemes for multi-dimensional nonlinear problems. This resulted in a first paper published in 'International Journal for Numerical Methods in Fluids', (De Grazia et al. (2014)) by the author and his co-workers, where they explored the connections between three nodal versions of tensor product discontinuous Galerkin spectral element approximations and two types of fux reconstruction schemes for solving systems of conservation laws on quadrilateral meshes. The different types of discontinuous Galerkin approximations arose from the choice of the solution points of the Lagrange basis representing the solution and from the quadrature approximation used to integrate the mass matrix and the other terms of the discretisation. We considered both a linear and a nonlinear advection equation on a regular mesh, examining the mathematical properties which connect these discretisations. These arguments were further confirmed by the results of an empirical numerical study. In this work we additionally made some considerations concerning the aliasing sources due the the nonlinearity of the flux functions and their intrinsic implications on the connections between the two class of schemes and we presented a comparative analysis of the performance of the FR and DG schemes investigated. This study is presented in chapter 2.

In a second paper (Mengaldo et al. (2015c)) the author and co-workers extended the work in (De Grazia et al. (2014)) by considering irregular/curvilinear meshes and therefore non-constant geometric terms arising in the discretised equations due to the mapping between the reference and the physical space. In this second paper, the author and co-workers also made some additional considerations in terms of aliasing issues due to the irregularity of the mesh and explored their implications.

Dealiasing techniques for discontinuous spectral/hp element methods

In the previous section, we have mentioned that, for the DG method, it is possible to achieve different numerical properties depending on the choice of the quadrature and the solution points. For instance, one may choose to use a collocation projection, leading to the class of collocated DG schemes, or use a higher quadrature, leading to a better integration of the nonlinear terms (when present). This is a particularly relevant point in the context of discontinuous spectral/hp element methods. In fact the use of collocation-based methods, such as the nodal DG schemes as well as the FR approach and the SD scheme, as they have been originally presented, under-integrate the nonlinear terms of the discretised equations, and therefore introduce errors. These errors yield to the so-called polynomial aliasing (or simply aliasing) which has been investigated in a number of papers, especially in the spectral method and spectral element method communities. These works include (Orszag (1971); Kirby and Karniadakis (2003); Kirby and Sherwin (2006b)), where the authors argue that the non-exact integration of the nonlinear terms also referred to as 'insufficient quadrature' leads to a degradation of the accuracy. The accuracy degradation can be quantified by theoretical estimates (Canuto and Quarteroni (1982)) and might be negligible for well-resolved problems. However, when dealing with badly- or marginally-resolved simulations (which is often the case for high-Reynolds number flows in complex geometries), these errors besides degrading the accuracy, also affect the numerical stability and therefore the reliability of discontinuous spectral/hp element methods, leading to the so-called aliasing-driven instabilities. These aliasing-driven instabilities, in turn, affect the numerical stability of this class of schemes. A lack of numerical stability is one of the most important shortcoming of discontinuous spectral/hp element methods (along with the mesh generation and the algorithmic complexity) and it is crucial to develop new or optimise existing strategies to address this issue. To tackle this lack of stability different techniques have been applied in the past. These include the adoption of spectral vanishing viscosity (SVV) (Kirby and Karniadakis (2002); Pasquetti (2006); Kirby and Sherwin (2006a)), polynomial filtering (Fischer and Mullen (2001); Fischer et al. (2002)), more stable formulations of the nonlinear terms (Blaisdell et al. (1996)) and over-integration (i.e. consistent integration of the nonlinear terms) (Kirby and Karniadakis (2003)). In particular this last technique, also referred to as over-integration, seemed particularly attractive and its deployment to the FR approach, given the connections found in the previous works by the author and co-workers, was a natural step. These considerations resulted in a paper published in 'Journal of Computational Physics' (Mengaldo et al. (2015a)), where we detailed two dealiasing strategies based on the concept of over-integration (of the linear terms) or consistent integration (of the nonlin-

ear terms), the first of which uses a localised approach that is useful when the nonlinearities only arise in parts of the problem and the second is based on the more traditional approach of using a higher quadrature. The main goal of both dealiasing techniques was to improve the numerical stability in the discontinuous spectral/hp element methods explored, thereby reducing aliasing-driven instabilities. We also addressed the role of the volumetric and boundary term contribution naturally arising in the formulation of discontinuous spectral/hp element methods. This work is presented in chapter 4, where we show the main results of the paper.

Guide to the robust implementation of boundary conditions

Given the objectives of this thesis, the implementation of boundary conditions has represented a crucial aspect. In fact, the nature of boundary conditions, and how they are implemented, can have a significant impact on the stability and accuracy of a CFD solver. A practical guideline for the robust implementation of boundary conditions in compact high-order methods for compressible aerodynamics has been published by the author and his co-workers in a conference paper accepted at the 2014 AIAA conference in Atlanta (Mengaldo et al. (2014)). The aim of this paper was to assess how different boundary condition implementations impact the performance of discontinuous spectral/hp element methods, when these schemes are used to solve the Euler and compressible Navier- Stokes equations on irregular grids. Specifically, the paper investigated inflow/outflow and wall boundary conditions enforced by modifying the boundary flux - therefore in a weak manner. In this paper we provided implementation details as well as guidelines for the successful integration of weak boundary conditions into a discontinuous spectral/hp element framework. Part of the contents of this paper is reported in chapter 3.

Boundary-layer flow studies

The numerical framework developed and analysed during this thesis has been thoroughly tested, since the ultimate goal was to produce a DNS/LES compressible solver for the LFC-UK centre at ICL. The tests have been carried out in several two- and three-dimensional geometries and the numerical tool has proven successful being applied to a problem relevant to the LFC-UK group. Specifically the problem considered implied the investigations of the boundary-layer separation in transonic flows caused by a three-dimensional isolated wall roughness. The process of the separation was analysed by means of a DNS of the compressible Navier-Stokes equations (through the discontinuous spectral/hp element methods developed throughout the thesis). The calculations were conducted by using a free-stream Mach number $Ma_\infty = 0.87$ and the Reynolds number adopted was $Re = 4 \times 10^5$ based on the streamwise position of the roughness element. We used

different roughness element heights, some of which were large enough to cause a well-developed separation region behind the roughness. The requirement in terms of spatial resolution were quite severe and the DNS performed by using the discontinuous spectral/hp element methods developed have proven particularly reliable, accurate and ultimately robust in terms of numerical stability. In particular, in order to achieve stable simulations, when required by the flow characteristics, we needed to adopt an appropriate dealiasing strategy.

This study is presented in chapter 5 together with a literature review of previous work on this topic. A paper with the content of this chapter is currently in preparation.

Nektar++ project

Nektar++ is an open-source software framework designed to support the development of high-performance scalable solvers for partial differential equations using the spectral/hp element method. The Nektar++ project has been designed to make this class of numerical approaches accessible to the broader scientific and industrial communities within an efficient object-oriented C++ framework. The software supports a variety of functionalities, including advanced pre- and post-processing tools as well as pre-written solvers relying on the underlying spectral/hp element method numerics. Since Nektar++ is an ongoing project, not all the required features for our study were available in the library when the research presented in this thesis started. Therefore, time has been dedicated to introduce all the algorithms and C++ classes necessary for our investigations.

Specifically the author and coworkers were responsible for:

- debugging and validation of DG advection class and development, debugging and validation of DG diffusion class;
- development, debugging and validation of FR advection and diffusion classes;
- development, debugging and validation of localised dealiasing strategy for improving the numerical stability of both the DG and FR approaches
- implementation of the compressible flow solver.

In particular, from the latter perspective, the author played an essential role in debugging and validating the viscous term of the Navier-Stokes equations and in developing, debugging and validating the boundary conditions. Several post-processing routines have also been implemented to obtain the results of this thesis. For further details on the Nektar++ project one can refer to the paper 'Nektar++: An open-source spectral/hp element framework' where the author and co-developers have presented the Nektar++ functionalities (Cantwell et al. (2015)).

List of publications

Parts of the work presented in this thesis have been disseminated through a number of written publications and oral communications; these are listed below together with an explanation of the author's contribution:

Journal papers

- DE GRAZIA, D., MENGALDO, G., MOXEY, D., VINCENT, P.E. & SHERWIN, S.J 2014. *Connections between the discontinuous Galerkin method and high-order flux reconstruction schemes.* International Journal for Numerical Methods in Fluids **75** (12), 860-877.

  The author of this thesis wrote most of the paper, mathematically demonstrated the connections between the different schemes and performed the simulations.

- MENGALDO, G., DE GRAZIA, D., MOXEY, D., VINCENT, P.E. & SHERWIN, S.J 2015. *Dealiasing techniques for high-order spectral element methods on regular and irregular grids.* Journal of Computational Physics **299**, 56-81.

  The author of this thesis implemented one of the dealiasing strategy, wrote part of the paper regarding the numerical experiments and performed part of the simulations.

- CANTWELL, C.D., MOXEY, D., COMERFORD, A., BOLIS, A., ROCCO, G., MENGALDO,G., DE GRAZIA, D., YAKOLEV, S., LOMBARD, J.-E., EKELSHOT, D. et al. 2015. *Nektar++: An open-source spectral/hp element framework.* Computer Physics Communications **192**, 205-219

  The author of this thesis performed the simulation of the compressible flow past a cylinder and helped in writing the description of the compressible flow solver.

- DE GRAZIA, D., MOXEY, D., KRAVTSOVA, M.A., RUBAN, A.I. & SHERWIN, S.J. *DNS of a compressible boundary-layer flow past an isolated three-dimensional hump in a high-speed subsonic regime.* Under review. Physical Review Fluids.

  The author of this thesis wrote the paper and performed the simulations of the flow past the roughness element.

Conference papers

- MENGALDO, G., DE GRAZIA, D., PEIRO, J., FARRINGTON, A., WITHERDEN, F., VINCENT, P.E. & SHERWIN, S.J. 2014. *A guide to the*

implementation of boundary conditions in compact high-order methods for compressible aerodynamics. 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation, Atlanta, GA.

The author of this thesis implemented part of the boundary conditions. He performed some of the simulations of the paper, especially those regarding the Euler BCs and wrote the part of the paper related to these numerical experiments

Technical notes

- MENGALDO, G., DE GRAZIA, D., VINCENT, P.E. & SHERWIN, S.J 2015. *On the Connections Between Discontinuous Galerkin and Flux Reconstruction Schemes: Extension to Curvilinear Meshes*. Technical Note, Journal of Scientific Computing, 1-21.

  The author of this thesis mathematically demonstrated the connections starting from the work done on regular grids. He performed part of the simulation presented in the paper.

Oral presentations

- DE GRAZIA, D., MENGALDO, G., MOXEY, D., VINCENT, P.E. & SHERWIN, S.J 2014. *Equivlences between specific discontinuous Galerkin methods and high-order flux reconstruction schemes*. International Conference on Spectral and High-Order Methods, Salt Lake City, UT.

Outline of the thesis

This thesis is organised as follows. In chapter 1, after a brief description of the most common spatial discretisations one can adopt for a partial differential equation, we introduce the foundations of spectral/hp element methods, highlighting the main operations required, such as numerical integration and differentiation. We also present the nomenclature used throughout this thesis and we finally detail the two discontinuous spatial discretisations analysed in this work, the DG method and the FR approach, for both first and second order problems. We then present some validation results for both the advection and the diffusion operators. In chapter 2, we detail the connections between the DG method and the FR approach for multidimensional systems of conservation laws taking in consideration both linear and nonlinear problems on tensor product grids. In chapter 3 we present the compressible Euler and Navier-Stokes equations and their discretisations by means of the DG and FR approaches. In this chapter we also show the implementation of the boundary conditions and some results to verify the implementation of the

compressible flow solver. In chapter 4 we describe the dealiasing strategies applied to both the DG and FR approaches and how they can help enhancing the numerical stability of these methods. In chapter 5, we present applications to high-speed subsonic flows past a three-dimensional roughness element. In the last chapter we finally draw the conclusions. Finally, we note that, although part of the relevant literature has been presented in this introductory chapter, in the rest of the thesis, at the beginning of each chapter, we will provide a detailed literature review concerning the specific topic explored in the chapter.

Chapter 1

Numerical discretisation of PDEs

In this chapter the fundamental concepts behind the design and implementation of a high order spectral/hp element method are described. First an overview of the spatial discretisation for solving PDEs is given in order to provide a brief theoretical background. Then the focus is moved to the spectral/hp element technique illustrating the main steps of the implementation. This is the spatial discretisation used for all the numerical experiments performed in this thesis. We detail the main operations required to construct the spatial discretisation describing the time discretisation techniques usually adopted. Subsequently, we introduce the two discontinuous spectral/hp element methods developed and analysed in this thesis, namely the discontinuous Galerkin (DG) method and the flux reconstruction (FR) approach. We finally present some results on the unsteady advection and unsteady diffusion equations, which demonstrate the correct implementation of the two methods and highlight their main features.

1.1 Numerical methods

Before describing in detail the spectral/hp element method we briefly present the Finite Element and the Spectral method. In the spectral/hp element method the properties of the aforementioned are combined and this makes the spectral/hp element discretisation particularly suitable for solving PDEs on complex geometries which occur in a wide range of fluid dynamic applications.

1.1.1 The finite element method

The finite element method was first proposed in the field of structural engineering. After approximately one decade the method was recognised as a form of the Rayleigh-Ritz problem. The relation between the two techniques derives from considering the variational form.

For example, given the sufficiently smooth real functions $p(x)$, $u(x)$ and $f(x)$, the quadratic functional

$$\mathcal{F}(u) = \int_0^1 \left[p(x)(\frac{du}{dx}(x))^2 + q(x)(u(x))^2 - 2f(x)u(x) \right] dx \tag{1.1}$$

has a minimum with respect to a variation in $u(x)$ which is given by the following Euler-Lagrange equation

$$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u(x) = f(x)\,. \tag{1.2}$$

Therefore, finding a solution of the differential equation 1.2 is equivalent to determining the value of $u(x)$ which minimises the functional of Eq. 1.1.

In the Rayleigh-Ritz approach the solution is approximated by a finite number of functions $u(x) = \sum_i^N q_i \Phi_i(x)$ to determine the unknown coefficients q_i, which minimise the functional. In the finite element method the solution is also approximated by a finite number of functions, which are typically local in nature opposed to the global functions of the Rayleigh-Ritz approach. The finite element method starts from the differential equation (1.2) written into an integral form (Eq. 1.1) which gives the Galerkin formulation of the problem. This can be reduced to an algebraic system to be solved numerically.

In the finite element method the domain Ω is split into N subdomains Ω_n whose characteristic size is h usually called elements. Inside each element the solution is approximated by a piecewise interpolation function. By reducing the element size (*h-type* refinement) the error in numerical solution decays and algebraic convergence is achieved. This method allows for the use of unstructured meshes which makes it attractive for studies which involve geometric complexity where finite difference methods are less well suited.

1.1.2 Spectral methods

The formulation of modern spectral methods was first introduced by Gottlieb and Orszag (1977). Canuto et al. (1987) focused on the fluid dynamics algorithms and application of global spectral methods.

The spectral methods perform a global discretisation of the domain using a single representation of a function $u(x)$ throughout the domain by a linear combination of continuous functions via a truncated series expansion, for instance,

$$u(x) \approx u_N(x) = \sum_{n=0}^{N} \widehat{u}_n \phi_n, \tag{1.3}$$

where $\phi_n(x)$ are the basis functions. This series is then substituted into a differential (or integral) equation and the unknown coefficients $\widehat{u}_n$ are computed through

the minimisation of the residual function. The expansion basis is constructed by high-order orthogonal functions which may be the Chebyshev polynomials, the Legendre polynomials, or another member of the family of the Jacobi polynomials. The exponential convergence is achieved by increasing the polynomial order P or the basis functions (*p-type* refinement). Unlike the finite element method the order of convergence is not fixed and is related to the regularity of the solution. For a sufficiently smooth solution they show better properties of accuracy and convergence than the finite element methods. These properties are easily lost if the solution has finite regularity or if the domain is irregular. The difficulty in finding a continuous expansion throughout complex domains limits the use of these methods to simple geometries.

1.1.3 Spectral/hp element methods

Spectral/hp element methods was first proposed by Patera (1984). They can be considered an extension of the well known finite element methods. They combine the high geometrical flexibility of the finite element discretisation with the superior convergence and accuracy of the spectral methods. In the spectral/hp element discretisation the computational domain is split into a subset of elemental subdomains. In each element the solution is represented by a series of high-order polynomials (*hp-type* discretisation). Convergence can be achieved by both refining the mesh or increasing the polynomial order of the approximation.
In the following section the method of weighted residuals will be presented which leads to the standard Galerkin formulation typical of the spectral/hp element methods.

1.2 The method of weighted residuals

When solving a differential equation we approximate the solution numerically with a finite representation replacing an otherwise infinite expansion. In this way the differential equation cannot be satisfied everywhere in our domain but we require it is able to satisfy a finite number of conditions. The choice of the conditions which have to be satisfied define the numerical method or the projection operator of our scheme. The method of the weighted residuals describes how it is possible to construct many of the common numerical methods by choosing different weight (test) functions and starting from an integral or weak form of the equations.

For describing this method we consider a linear differential equation defined in a domain Ω

$$\mathcal{L}(u) = 0, \tag{1.4}$$

with an appropriate set of boundary and initial conditions. We assume that it is possible to accurately represent the solution with the following approximation:

$$u^{\delta}(\boldsymbol{x},t) = u_0(\boldsymbol{x},t) + \sum_{i=1}^{N_{dof}} \widehat{u}_i(t)\Phi_i(\boldsymbol{x}) \tag{1.5}$$

where $\phi_i(\boldsymbol{x})$ are the trial (or expansion) functions, $\widehat{u}_i(t)$ are the N_{dof} unknown coefficients and $u_0(\boldsymbol{x},t)$ is selected to satisfy the initial and boundary conditions of the problem. Substituting Eq. 1.5 into Eq. 1.4 we obtain a nonzero residual R denoted by:

$$\mathcal{L}(u^{\delta}) = R(u^{\delta}). \tag{1.6}$$

In order to solve Eq. 1.6 we can place a restriction on the residual R which will reduce Eq. 1.6 to a system of ordinary differential equations in $\widehat{u}_i(t)$. If the original Eq. 1.4 is independent of time the coefficients $\widehat{u}_i$ can be directly calculated from the solution of a system of algebraic equations. There is no unique way to set a restriction, each way will lead to a different numerical scheme. To define the restriction to be applied we first introduce the Legendre inner product f, g over the domain Ω, defined as

$$(f,g) = \int_{\Omega} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}, \tag{1.7}$$

where f and g are two general functions depending on space only. Specifically, through the restriction, we require that the inner product of the residual with respect to a *weight* (or *test*) function $v_j(\boldsymbol{x})$ is equal to zero, that is

$$(v_j(\boldsymbol{x}), R) = 0, \qquad\qquad j = 1, \ldots, N_{dof}. \tag{1.8}$$

The weighted residual is then said to be zero and the name of the technique derives from this expression. We note that when $N_{dof} \to \infty$ the residual $R(\boldsymbol{x})$ tends to zero since the approximate solution $u^{\delta}(\boldsymbol{x},t)$ asymptotically approaches the exact solution $u(\boldsymbol{x},t)$. The scheme obtained depends on the choice of the trial function $\phi_i(\boldsymbol{x})$ and of the test function $v_j(\boldsymbol{x})$. In the following we report the most popular numerical approaches and the associated trial and test functions.

1. Collocation method:
   In the collocation method the test function is the Dirac delta function

   $$v_j(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_j), \tag{1.9}$$

   where $\boldsymbol{x}_j$ denote a set of collocation points. At each collocation point the residual is set to zero ($R(\boldsymbol{x}_j) = 0$) and the differential equation is exactly satisfied. This method is usually employed in spectral methods (Gottlieb and Orszag (1977); Canuto et al. (1987)) and in finite-difference-method (LeVeque (2007); Lele (1992)).

2. Finite volume method:
   In the finite volume (FV) method, the domain is split into N_{dof} non-overlapping subdomains Ω_j and the test function is of the form

$$v_j(\boldsymbol{x}) = \begin{cases} 1, & inside\ \Omega_j, \\ 0, & outside\ \Omega_j. \end{cases} \tag{1.10}$$

   This method is one of the most popular discretisation in computational fluid dynamics. There are different versions of FV method, both low and high-order. The interested reader can find a broad overview of finite volume methods for hyperbolic problems in (LeVeque (2002)). Specific references for high-order FV schemes can be found in Barth and Frederickson (1990) for $k-exact$ methods, Harten et al. (1987), Abgrall (1994) for FV type essentially nonoscillatory (ENO) methods and Liu et al. (1994), Hu and Shu (1999) for FV type weighted essentially non-oscillatory (WENO) methods.

3. Least squares method:
   In the least square method the test function is set to $v_j = \partial R/\partial u_j$. With this choice it is possible to find the coefficients $\widehat{u}_i(t)$ which minimise the quadratic functional (R, R). This formulation has been used in traditional finite element methods (Bochev and Gunzburger (1998); Jiang (1998)) and, more recently, has been extended to spectral/hp element methods (Proot and Gerrtisma (2002)).

4. Galerkin method:
   In this method (also known as Bubnov-Galerkin method), the test functions are selected to be identical to the trial functions such that $v_j(\boldsymbol{x}) = \Phi_j(\boldsymbol{x})$. This type of projection is a minimisation of the residual over the domain Ω. Most of the finite element and spectral element methods use this formulation (Karniadakis and Sherwin (2005)). A broader class of the Galerkin method known as the Petrov-Galerkin method, or sometimes the generalised Galerkin method, adopts tests functions which can be similar, but not identical to the expansion functions $v_j(\boldsymbol{x}) = \phi_j(\boldsymbol{x})$. In particular, these test functions are based upon a perturbation of the trial functions which improves the numerical stability property of the schemes (Heinrich and Zienkiewicz (1977); Brooks and Hughes (1982); Hughes et al. (1982)). Other references on the continuous Galerkin (CG) methods are Hughes and Brooks (1979, 1982). For additional details on high-order CG methods see articles by Sherwin and Karniadakis (1995), Sherwin et al. (1998) and the textbook by Karniadakis and Sherwin (2005). The discontinuous form of the Galerkin method (DG) was first proposed by Reed and Hill (1973). Numerous variants of the DG approach exist and, it can be noted that many have been specifically developed

to treat elliptic viscous terms such as Arnold et al. (2001), Cockburn and Shu (1998a), Bassi and Rebay (1997). Finally, it is pertinent to highlight a relatively new class of so called hybridized DG (HDG) methods, which were recently proposed by Cockburn and Shu (2001) and further developed by Nguyen et al. (2009a) and Nguyen et al. (2009b, 2011). For more information about DG schemes see the textbook by Cockburn et al. (2000) and the textbook by Hesthaven and Warburton (2008).
For further details on the background of the methods of weighted residual, one can refer to Finlayson (1972) and Fletcher (1984).

The method of weighted residuals shows how to construct different numerical formulation by defining the type of projection operator. These formulations can be applied to any partial differential problem. However, the method does not define the trial functions and the approximation space. These can be expansion functions which are non-zero throughout the solution domain (also referred to as global expansion functions) like those of the spectral methods or expansion functions defined in a local finite region of the domain like those of the finite element methods. After having defined both the projection operator and the approximation space, the spatial discretisation of the problem is complete and the PDE(s) has become an algebraic system of equations (or a system of ordinary differential equations (ODE)s if the original problem depends on time). In this work, we focus on spectral/hp element methods, which combine the local nature of the expansion functions typical of finite element methods and the arbitrary expansion functions common in spectral methods.

Even if in this thesis we mostly deal with smooth solutions of infinite regularity for which exponential convergence rates are realised, the author wants to remark that if the solution contains discontinuities, such as shock waves, problems arise. In fact, representing a discontinuous solution with a high-order polynomial can lead to the formation of spurious oscillations. These oscillations that develop close to the discontinuity are the so called Gibbs phenomena and may eventually cause the solution to blow up. This type of instabilities must not be confused with the aliasing-driven instabilities caused by the collocation projection of the flux and described in the future chapters even if some of the techniques to tackle them are the same. The Gibbs oscillations can be explained with the Godunov's theorem (Godunov (1959)):

"There are no monotone, linear schemes for the linear advection equation of second or higher order of accuracy".

In other words, high order accuracy and monotonicity are contradictory requirements. The only way to avoid oscillation and maintain at least a second

order of accuracy is to use nonlinear schemes (see Sweby (1984) and Van Leer (1974)). One of the objectives of the Nektar++ project is to realise "high-resolution" methods. This term applies to methods that are at least second-order accurate on smooth solutions and yet give well resolved, nonoscillatory discontinuities. The main idea of an high resolution method is to attempt to use a high order method, but to modify the method increasing the amount of numerical dissipation in the proximity of a discontinuity as attempted by the ENO and WENO schemes mentioned before.

1.3 The spectral/hp element method

The spectral/*hp* element methods combine the geometrical flexibility of finite element methods, and the superior spatial accuracy properties of spectral method as already mentioned in 1.1.3. The first step of a spectral/*hp* element method involves a decomposition into subdomains or elements. Each element is then mapped into a reference (standard) element trough a parametric map between the original and the reference space. In each standard element we represent the solution through an arbitrary-degree expansion basis. In these elements there are operations, such as multiplications, integrations and differentiations, which need to be performed in order to solve the local discretised problems. After having solved these problems inside the elements, the global problem is finalised by using specific connectivity rules at the interfaces between adjacent elements. The original PDE problem has now been transformed into a system of ODEs which needs to be advanced in time trough a suitable time-integration scheme. If the original PDE does not depend on time we obtain a system of algebraic equations.

In the next subsections we describe the main steps for implementing the spectral/hp element method and we review the most common time-integration schemes adopted for solving time-dependent problems.

1.3.1 Spatial discretisation

As just mentioned there are five main building blocks constituting the spectral/*hp* element methods. The first is the domain decomposition by which the computational domain is split into non-overlapping elements. Then there is the mapping between the local and the standard element trough which each arbitrary-shaped element is transformed into a reference element. After that the approximation of the solution is obtained through an expansion basis inside each standard element. In the fourth block the numerical integration and differentiation are performed and then the connectivity rules are applied. Figure 1.1 shows an illustrative diagram with the main operations involved in the construction of the spectral/*hp* element method.

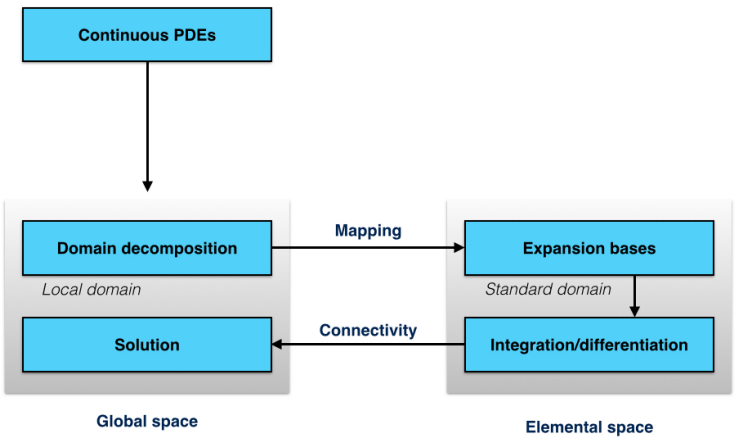


Figure 1.1: Illustrative diagram of the main building blocks constituting the spectral/hp element method.

In the following we introduce these operations for one-dimensional, quadrilateral and hexahedral tensor product elements. We do not present these concepts for other element shapes such as triangles because the main focus of the thesis has been on quadrilateral and hexahedral meshes.

Domain decomposition

Consider a computational domain Ω which can be split into N non-overlapping subdomains (elements) Ω_n as follows

$$\Omega = \bigcup_{n=1}^{N} \Omega_n, \qquad \bigcap_{n=1}^{N} \Omega_n = 0. \tag{1.11}$$

There are various element shapes which can be used in the spectral/*hp* methods. These are: segments in one-dimensional problems, quadrilaterals and triangles in two-dimensional problems and hexahedra, tetrahedra, prisms and pyramids in three-dimensional problems. To facilitate the implementations of the main operations required, in spectral/*hp* element methods we map each shape into a reference element usually called standard domain/element and denoted by Ω_s (see Figure 1.2).

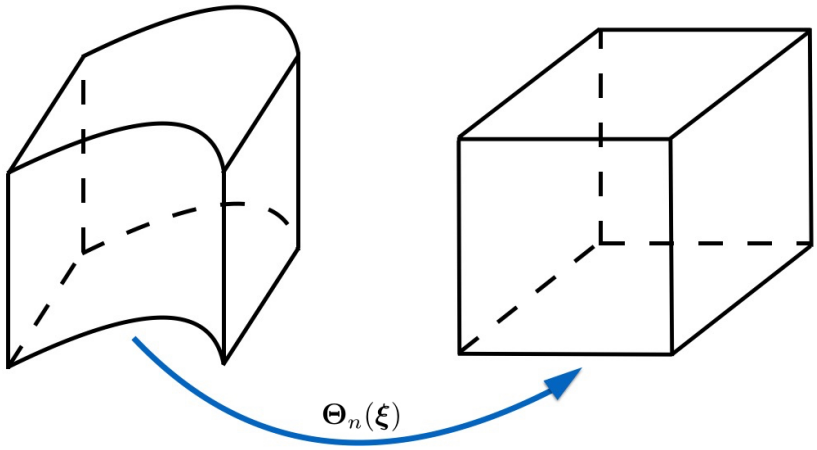


Figure 1.2: Transformation from a physical element to a standard element in the case of a hexahedral shape.

For segments the standard element is defined as

$$\Omega_s = (\xi) \in [-1, 1], \tag{1.12}$$

with ξ being the one-dimensional coordinate associated to Ω_s. The standard quadrilateral element is instead defined as

$$\Omega_s = (\xi_1, \xi_2) \in [-1, 1] \times [-1, 1], \tag{1.13}$$

with ξ_1 and ξ_2 being the two-dimensional orthogonal coordinates associated with Ω_s. The standard hexahedral element is instead defined as

$$\Omega_s = (\xi_1, \xi_2, \xi_3) \in [-1, 1] \times [-1, 1] \times [-1, 1], \tag{1.14}$$

with ξ_1, ξ_2, ξ_3 being the three-dimensional orthogonal coordinates associated with Ω_s.

In the case of a segment we will indicate with the letter "L" and "R" the left and right corner. In the case of a quadrilateral element the letter "B" and the letter "T" indicate the bottom and top edges, respectively, while, the letter "L" and the letter "R", the left and right edges. In the case of a hexahedral element the letter "FW" and the letter "BW" indicate the forward and backward faces, the letter "B"

and the letter "T" the bottom and top faces, while, the letter "L" and the letter "R", the left and right faces. Note that with $\boldsymbol{\Theta}(\boldsymbol{\xi})$ we denote the mapping between the local and the standard element described in the next paragraph.

Map between local and standard element

Having split the computational domain in N elements we need to map each element into a reference element. For one-dimensional elements we can write the map as

$$x = \Theta_n(\xi) = \phi_0(\xi)x_n + \phi_1(\xi)x_{n+1} \qquad \xi \in \Omega_s, \tag{1.15}$$

where ϕ_0 and ϕ_1 are the generic expansions for mapping the coordinates. Eq. 1.15 is known as *parametric mapping* and, if the order of ϕ_0 and ϕ_1 is the same as the dependent variables, then the mapping is defined as iso-parameteric, while if the order is lower or higher, the mapping is defined as sub- or super-parametric, respectively. In case of a straight segment Eq. 1.15 becomes

$$x = \Theta_n(\xi) = x_n\frac{1-\xi}{2} + x_{n+1}\frac{1+\xi}{2} \qquad \xi \in \Omega_s, \tag{1.16}$$

All the concepts expressed in the one-dimensional case can be applied for multi-dimensional tensor product elements. The general map in the multi-dimensional case reads as

$$(\boldsymbol{x}) = \boldsymbol{\Theta}_n(\boldsymbol{\xi}). \tag{1.17}$$

For an arbitrary-shaped straight-sided quadrilateral with vertices A, B, C and D the mapping onto the standard region is

$$\begin{aligned}\boldsymbol{x} = (x_1, x_2)^T = \boldsymbol{\Theta}_n(\xi_1, \xi_2) = \boldsymbol{x}^A\frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + \boldsymbol{x}^B\frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + \\ + \boldsymbol{x}^C\frac{1-\xi_1}{2}\frac{1+\xi_2}{2} + \boldsymbol{x}^D\frac{1+\xi_1}{2}\frac{1+\xi_2}{2}\,.\end{aligned} \tag{1.18}$$

An analogous map can be constructed for three-dimensional straight-sided elements. In particular for an arbitrary-shaped straight-sided hexahedron with vertices A, B, C, D, E, F, G and H the mapping onto the standard region is

$$\begin{aligned}&\boldsymbol{x} = (x_1, x_2, x_3)^T = \boldsymbol{\Theta}_n(\xi_1, \xi_2, \xi_3) = \\ &\boldsymbol{x}^A\frac{1-\xi_1}{2}\frac{1-\xi_2}{2}\frac{1-\xi_3}{2} + \boldsymbol{x}^B\frac{1+\xi_1}{2}\frac{1-\xi_2}{2}\frac{1-\xi_3}{2} + \\ &\boldsymbol{x}^C\frac{1-\xi_1}{2}\frac{1+\xi_2}{2}\frac{1-\xi_3}{2} + \boldsymbol{x}^D\frac{1+\xi_1}{2}\frac{1+\xi_2}{2}\frac{1-\xi_3}{2} + \\ &\boldsymbol{x}^E\frac{1-\xi_1}{2}\frac{1-\xi_2}{2}\frac{1+\xi_3}{2} + \boldsymbol{x}^F\frac{1+\xi_1}{2}\frac{1-\xi_2}{2}\frac{1+\xi_3}{2} + \\ &\boldsymbol{x}^G\frac{1-\xi_1}{2}\frac{1+\xi_2}{2}\frac{1+\xi_3}{2} + \boldsymbol{x}^H\frac{1+\xi_1}{2}\frac{1+\xi_2}{2}\frac{1+\xi_3}{2}\,.\end{aligned} \tag{1.19}$$

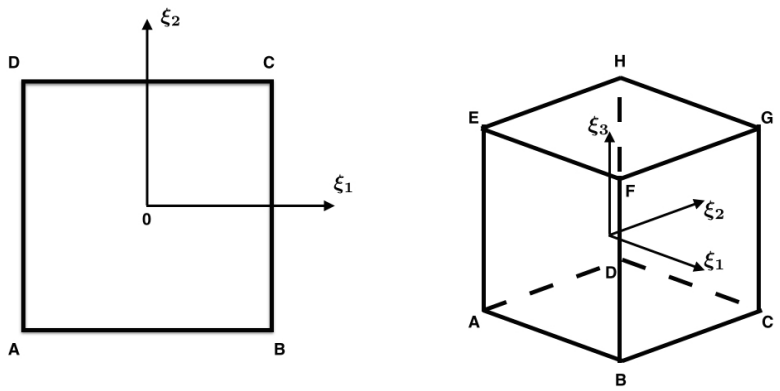
In case of curvilinear regions we can use the standard isoparametric mapping which for an arbitrary-shaped curved-sided two-dimensional element is

$$\boldsymbol{x} = (x_1, x_2)^T = \boldsymbol{\Theta}(\xi_1, \xi_2) = \sum_{p=0}^{Q_1} \sum_{q=0}^{Q_2} \widehat{\boldsymbol{x}}_{pq} \phi_p(\xi_1) \phi_q(\xi_2)\,, \tag{1.20}$$

where ϕ_p and ϕ_q are the same basis functions used for representing the solution. We note that, if $\boldsymbol{x}$ is a polynomial of order $P < Q_1, Q_2$, the mapping is subparametric, while if $P > Q_1, Q_2$ the mapping is super-parametric. An analogous mapping can be adopted for a three-dimensional curvilinear element:

$$\boldsymbol{x} = (x_1, x_2, x_3)^T = \boldsymbol{\Theta}(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{Q_1} \sum_{q=0}^{Q_2} \sum_{r=0}^{Q_3} \widehat{\boldsymbol{x}}_{pqr} \phi_p(\xi_1) \phi_q(\xi_2) \phi_q(\xi_3)\,. \tag{1.21}$$

Figure 1.3 shows the labelling used for a quadrilateral and a hexahedral shape.



(a) (b)

Figure 1.3: Vertex labelling for the quadrilateral (a) and hexahedral (b) shapes.

From the mapping defined above we can obtain the geometric terms required by the transformation from the physical to the reference space and viceversa. From this perspective, we write the deformation gradient $\mathbf{G}_{2D}$ for the two-dimensional case as

$$\mathbf{G}_{2D} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} \end{bmatrix}, \tag{1.22}$$

with the following relations:

$$\frac{\partial \xi_1}{\partial x_1} = \frac{1}{J_{2D}} \frac{\partial x_2}{\partial \xi_2}, \quad \frac{\partial \xi_1}{\partial x_2} = -\frac{1}{J_{2D}} \frac{\partial x_1}{\partial \xi_2}, \quad \frac{\partial \xi_2}{\partial x_1} = -\frac{1}{J_{2D}} \frac{\partial x_2}{\partial \xi_1}, \quad \frac{\partial \xi_2}{\partial x_2} = \frac{1}{J_{2D}} \frac{\partial x_1}{\partial \xi_1},$$

where $J_{2D} = |G_{2D}|$ is the Jacobian of the transformation and is defined as the determinant of the deformation gradient. In three-dimensional cases the deformation gradient $\mathbf{G}_{3D}$ is

$$\mathbf{G}_{3D} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_1}{\partial \xi_3} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_3} \\ \dfrac{\partial x_3}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_3} \end{bmatrix}, \tag{1.23}$$

where we can use the following relations:

$$\frac{\partial \xi_1}{\partial x_1} = \frac{1}{J_{3D}} \left(\frac{\partial x_2}{\partial \xi_2} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_2}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_2} \right), \quad \frac{\partial \xi_1}{\partial x_2} = -\frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_2} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_2} \right),$$
$$\frac{\partial \xi_1}{\partial x_3} = \frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_3} \frac{\partial x_2}{\partial \xi_2} \right),$$
$$\frac{\partial \xi_2}{\partial x_1} = -\frac{1}{J_{3D}} \left(\frac{\partial x_2}{\partial \xi_1} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_2}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_1} \right), \quad \frac{\partial \xi_2}{\partial x_2} = \frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_1} \right),$$
$$\frac{\partial \xi_2}{\partial x_3} = -\frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_3} \frac{\partial x_2}{\partial \xi_1} \right),$$
$$\frac{\partial \xi_3}{\partial x_1} = \frac{1}{J_{3D}} \left(\frac{\partial x_2}{\partial \xi_1} \frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_2}{\partial \xi_2} \frac{\partial x_3}{\partial \xi_1} \right), \quad \frac{\partial \xi_3}{\partial x_2} = -\frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_3}{\partial \xi_1} \right),$$
$$\frac{\partial \xi_3}{\partial x_3} = \frac{1}{J_{3D}} \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1} \right),$$

with $J_{3D} = |G_{3D}|$ the Jacobian of the transformation defined as the determinant of the deformation gradient. Note that the one-dimensional case simply reduces to $\mathbf{G}_{1D} = \partial x / \partial \xi = J_{1D}$. In the remainder of the thesis these geometric terms will be used extensively.

Expansion bases

After having introduced the geometric map from the local to the standard space with the geometric terms needed for the transformations of the equations, we must define the expansion bases representing the solutions and the fluxes in

the discretised equations. In spectral/hp element methods, the expansion bases employed in each standard element are usually polynomials and can be of modal or nodal type. To highlight the differences between a modal and a nodal expansion, we define the following two complete sets of polynomials up to order P:

$$\begin{aligned} \phi_p^M(\xi) &= \xi^p, \qquad p = 0, \ldots, P, \\ \phi_p^N(\xi) &= \frac{\prod_{q=0, q\neq p}^{P}(\xi - \xi_q)}{\prod_{q=0, q\neq p}^{P}(\xi_p - \xi_q)} \qquad p = 0, \ldots, P. \end{aligned} \tag{1.24}$$

The first expansion $\phi_p^M(\xi)$ is called *modal* or *hierarchical* expansion because the order $(P-1)$ expansion set is contained within the order P expansion set. This expansion therefore satisfies the following relationship:

$$\chi_{P-1}^{\delta} \subset \chi_P^{\delta}. \tag{1.25}$$

All the modes (or polynomials) of a simple modal basis usually influence the boundary points of a given element. When we need to use connectivity rules between the various elements of the spatial discretisation this can be inefficient. It is therefore common to apply a boundary-interior decomposition to the expansion, where only two modes are non-zero at the boundaries, while all the others are identically zero at the boundaries and have non-zero values at the interior points only. An example of modal basis using a boundary-interior decomposition can be written as:

$$\phi_p(\xi) = \begin{cases} \psi_0^a(\xi) = \dfrac{1-\xi}{2}, & p = 0 \\ \psi_p^a(\xi) = \left(\dfrac{1-\xi}{2}\right)\left(\dfrac{1-\xi}{2}\right) P_{p-1}^{1,1}, & 0 < p < P \\ \psi_P^a(\xi) = \dfrac{1+\xi}{2}, & p = P, \end{cases} \tag{1.26}$$

where $P_{p-1}^{1,1}$ is a Jacobi polynomial. This is the most widely adopted modal basis in spectral/hp element methods (Karniadakis and Sherwin (2005)) and was developed by Peano (1976), Szabo and Babushka (1991) and Oden (1994).
The second expansion in Eq. 1.24 $\phi_p^N(\xi)$, are Lagrange polynomials defined on a set of $(P+1)$ nodal points ξ_q. In contrast to the modal expansion set, the Lagrange polynomials have a non-hierarchical basis since it consists of $(P+1)$ polynomials of order P, thus $\chi_{P-1}^{\delta} \not\subset \chi_P^{\delta}$. The Lagrange expansion has the important property that $\phi_p^N = \delta_{pq}$ where δ_{pq} is the Kronecker delta function, which implies

$$u^{\delta}(\xi_q) = \sum_{p=0}^{P} u_p \phi_p^N(\xi_q) = \sum_{p=0}^{P} u_p \delta_{pq} = u_q. \tag{1.27}$$

The coefficients of the Lagrange expansion basis are therefore the physical values of the discrete solution u at the nodal point ξ_q. In the Lagrange basis, the boundary-interior decomposition is not required since it is already satisfied providing that the nodal points include the element boundaries. Each of the expansion basis defined, both modal and nodal, can be seen as a one-dimensional tensor. The extension to two- and three-dimensional is therefore straightforward and can be obtained by the product of the one-dimensional tensors. For quadrilateral shapes we can write:

$$\phi_{pq}(\xi_1, \xi_2) = \phi_p(\xi_1)\phi_q(\xi_2), \quad 0 \leq p, q;\ p \leq P_1,\ q \leq P_2, \tag{1.28}$$

while for hexahedral:

$$\begin{aligned} \phi_{pqr}(\xi_1, \xi_2, \xi_3) = \phi_p(\xi_1)\phi_q(\xi_2)\phi_r(\xi_3), \qquad & 0 \leq p, q, r; \\ & p \leq P_1,\ q \leq P_2,\ r \leq P_3. \end{aligned} \tag{1.29}$$

In Eq. 1.28, 1.29 we allow the polynomials to be different in each coordinate direction. Note that the modal expansion on these shapes maintains its hierarchy property while the nodal basis maintains the Kronecker delta property.

We now introduce the matrix form of the expansion bases, which will be extensively used in the next chapters of this thesis. Specifically, when considering a tensor-based expansion, we evaluate the variables of the equations on a set of nodal points. For instance, the one-dimensional function $u(\xi_q)$ on a set of Q nodal points $\boldsymbol{\xi} = [\xi_0, \xi_1, \ldots, \xi_{Q-1}]$ is:

$$\mathbf{u} = [u(\xi_0), u(\xi_1), \ldots, u(\xi_{Q-1})]^T. \tag{1.30}$$

Analogously, the two-dimensional case is denoted by:

$$\mathbf{u} = [u(\xi_{1,0}, \xi_{2,0}), \ldots, u(\xi_{1,Q-1}, \xi_{2,0}), u(\xi_{1,0}, \xi_{2,1}), \ldots, u(\xi_{1,Q_1-1}, \xi_{2,Q_2-1})]^T, \tag{1.31}$$

while the three-dimensional is:

$$\begin{aligned} \mathbf{u} = [&u(\xi_{1,0}, \xi_{2,0}, \xi_{3,0}), \ldots, u(\xi_{1,Q-1}, \xi_{2,0}, \xi_{3,0}), u(\xi_{1,0}, \xi_{2,1}, \xi_{3,0}), \ldots, \\ &u(\xi_{1,Q_1-1}, \xi_{2,Q_2-1}\xi_{3,0}), u(\xi_{1,0}, \xi_{2,0}, \xi_{3,1}), \ldots, u(\xi_{1,Q_1-1}, \xi_{2,Q_2-1}\xi_{3,Q_3-1})]^T, \end{aligned} \tag{1.32}$$

where Q_1, Q_2 and Q_3 represent the points in the ξ_1, ξ_2 and ξ_3 directions respectively.

The matrix form for the one-dimensional case of the expansion basis is written as follows:

$$\mathbf{B} = \begin{bmatrix} \phi_0(\xi_0) & \ldots & \phi_p(\xi_0) & \ldots & \phi_P(\xi_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_0(\xi_{Q-1}) & \ldots & \phi_p(\xi_{Q-1}) & \ldots & \phi_P(\xi_{Q-1}) \end{bmatrix},$$

where P is the order of the expansion and $\mathbf{B}$ s the basis matrix. In two-dimensional case the basis matrix is:

$$\mathbf{B} = \begin{bmatrix} \phi_{00}(\xi_{1,0}, \xi_{2,0}) & \dots & \phi_{pq}(\xi_{1,0}, \xi_{2,0}) & \dots & \phi_{P_1P_2}(\xi_{1,0}, \xi_{2,0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{00}(\xi_{1,\overline{Q}_1}, \xi_{2,0}) & \dots & \phi_{pq}(\xi_{1,\overline{Q}_1}, \xi_{2,0}) & \dots & \phi_{P_1P_2}(\xi_{1,\overline{Q}_1}, \xi_{2,0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{00}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}) & \dots & \phi_{pq}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}) & \dots & \phi_{P_1P_2}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}) \end{bmatrix},$$

where P_1 and P_2 are the orders of the expansion in the ξ_1 and ξ_2 directions respectively and $\overline{Q}_1 = Q_1 - 1$, $\overline{Q}_2 = Q_2 - 1$. Analogously, in the three-dimensional case we can write:

$$\mathbf{B} = \begin{bmatrix} \phi_{000}(\xi_{1,0}, \xi_{2,0}, \xi_{3,0}) & \dots & \phi_{pqr}(\xi_{1,0}, \xi_{2,0}, \xi_{3,0}) & \dots & \phi_{P_1P_2,P_3}(\xi_{1,0}, \xi_{2,0}, \xi_{3,0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{000}(\xi_{1,\overline{Q}_1}, \xi_{2,0}, \xi_{3,0}) & \dots & \phi_{pqr}(\xi_{1,\overline{Q}_1}, \xi_{2,0}, \xi_{3,0}) & \dots & \phi_{P_1P_2P_3}(\xi_{1,\overline{Q}_1}, \xi_{2,0}, \xi_{3,0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{000}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,0}) & \dots & \phi_{pqr}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,0}) & \dots & \phi_{P_1P_2P_3}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{000}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,\overline{Q}_3}) & \dots & \phi_{pqr}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,\overline{Q}_3}) & \dots & \phi_{P_1P_2P_3}(\xi_{1,\overline{Q}_1}, \xi_{2,\overline{Q}_2}, \xi_{3,\overline{Q}_3}) \end{bmatrix},$$

where P_1, P_2 and P_3 are the orders of the expansion in the ξ_1, ξ_2 and ξ_3 directions respectively and $\overline{Q}_1 = Q_1 - 1$, $\overline{Q}_2 = Q_2 - 1$, $\overline{Q}_3 = Q_3 - 1$.

In this work, we mainly used nodal bases. However, in the Nektar++ library, all the implementations adopted can also use modal bases. After having introduced the one-dimensional expansion bases it is showed how to apply them to a standard quadrilateral and hexahedral element through a tensor product approach. Therefore the conventions adopted are explained and, we can now define the main operations required in spectral/hp elements methods: numerical integration and numerical differentiation.

Numerical integration

The integrals which arise in the formulation of spectral/hp method usually need to be evaluated in the physical space and therefore we have integrals of the type:

$$\int_{x_n}^{x_{n+1}} u(x)dx = \int_{-1}^{1} J_n u(\Theta(\xi))d\xi, \tag{1.33}$$

where $u(\xi)$ is the integrand and can be a combination of polynomial bases, like $u(\xi) = \phi_p(\xi)\phi_q(\xi)$, while J_n is the one-dimensional Jacobian of the transformation from the local element to reference element. The numerical integration or quadrature of Eq. 1.33 is:

$$\int_{-1}^{1} J_n u(\Theta(\xi))d\xi \approx \sum_{i=0}^{Q-1} w_i J_{n_i} u(\xi_i), \tag{1.34}$$

where w_i are the *weights*, J_{n_i} are the Jacobians, ξ_i is the abscissa representing the Q distinct quadrature points in the interval $-1 \leq \xi_i \leq 1$. There are different types of numerical integration, however, in this work we concentrate on Gaussian quadrature rules which are used in Nektar++. For further details on other forms of numerical integration, the interested reader can refer to Appendix B in Karniadakis and Sherwin (2005). This type of quadrature is widely adopted in spectral/hp element methods and it employs a Lagrange polynomial ℓ onto a set of Q quadrature points to represent the integrand $u(\xi)$:

$$u^{\delta}(\xi) = \sum_{i=0}^{Q-1} u(\xi_i)\ell_i(\xi) + \epsilon(u), \tag{1.35}$$

where $\epsilon(u)$ is the approximation error. By substituting 1.35 into Eq. 1.33 we obtain the following summation:

$$\begin{gathered} \int_{-1}^{1} J_n u(\xi)d\xi = \\ \sum_{i=0}^{Q-1} \int_{-1}^{1} J_{n_i}[u(\xi_i)\ell_i(\xi) + \epsilon(u)]d\xi = \sum_{i=0}^{Q-1} w_i J_{n_i} u(\xi_i) + \int_{-1}^{1} J_{n_i}\epsilon(u)d\xi, \end{gathered} \tag{1.36}$$

where w_i are the weights of the quadrature. The integral associated to the Gaussian quadrature in Eq. 1.36 is known as Legendre integration. For better characterising this integration we need to define a set of quadrature points or zeros ξ_i. From this perspective we can distinguish between three quadratures which depend on the distribution of the quadrature points:

- *Gauss-Legendre*, whose points distribution is defined only inside the element;
- *Gauss-Radau-Legendre*, whose point distribution includes only one end of the element;
- *Gauss-Lobatto-Legendre*, whose points distribution include both the ends of the element.

Each of these quadratures allows the exact integration of polynomials of different orders. The Gauss-Legendre quadrature allows the exact integration of a polynomial integrand of order $(2Q-1)$, that is $\int_1^1 \epsilon(u)d\xi = 0$ if $u^\delta(\xi) \in \mathbb{P}_{2Q-1}(-1,1)$ or less. The Gauss-Radau-Legendre quadrature permits the exact integration if $u^\delta(\xi) \in \mathbb{P}_{2Q-2}(-1,1)$ or less, whereas Gauss-Lobatto-Legendre integrates exactly integrand polynomials $u^\delta(\xi) \in \mathbb{P}_{2Q-3}(-1,1)$ or less. In this work, Gauss-Legendre and Gauss-Lobatto-Legendre quadratures were considered.

In case of a quadrilateral tensor product element with $\Omega_s = (\xi_1, \xi_2) \in [1,1] \times [-1,1]$ the extension is straightforward and reads as follows:

$$\iint_{-1}^{1} J_n u(\xi_1,\xi_2) d\xi_1 d\xi_2 \approx \sum_{i=0}^{Q_1-1} w_i \left\{ \sum_{j=0}^{Q_2-1} w_j J_{n_{ij}} u(\xi_{1_i}, \xi_{2_j}) \right\}, \tag{1.37}$$

where Q_1 and Q_2 are the number of quadrature points in the ξ_1 and ξ_2 directions, respectively and $J_{n_{ij}}$ are the two-dimensional point-wise Jacobians of the transformation between the physical and the standard space.

For the hexahedral tensor product element with $\Omega_s = (\xi_1, \xi_2, \xi_3) \in [1,1] \times [-1,1] \times [-1,1]$ we can write:

$$\begin{gathered} \iint_{-1}^{1} J_n u(\xi_1,\xi_2,\xi_3) d\xi_1 d\xi_2 d\xi_3 \approx \\ \sum_{i=0}^{Q_1-1} w_i \left\{ \sum_{j=0}^{Q_2-1} w_j \left[\sum_{j=0}^{Q_3-1} w_k J_{n_{ijk}} u(\xi_{1_i}, \xi_{2_j}, \xi_{3_k}) \right] \right\}, \end{gathered} \tag{1.38}$$

where Q_1, Q_2 and Q_3 are the number of quadrature points in the ξ_1, ξ_2 and ξ_3 directions, respectively and $J_{n_{ijk}}$ are the three-dimensional point-wise Jacobians of the transformation. All the considerations concerning the exact integration are still valid for the multidimensional case.
As was done before for the expansion bases we now introduce the weight matrix. Specifically for the one-dimensional case the weight matrix $\mathbf{W}$ can be written as follows:

$$\mathbf{W} = \begin{bmatrix} w_0 J_{n_0} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & w_i J_{n_i} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & w_{Q-1} J_{n_{Q-1}} \end{bmatrix}.$$

Analogously, for the two-dimensional case we write:

$$\mathbf{W} = \begin{bmatrix} w_{00}J_{n_{00}} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & w_{ij}J_{n_{ij}} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & w_{(Q_1-1)(Q_2-1)}J_{n_{(Q_1-1)(Q_2-1)}} \end{bmatrix},$$

while for the three-dimensional case we have:

$$\mathbf{W} = \begin{bmatrix} w_{000}J_{n_{000}} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & w_{ijk}J_{n_{ijk}} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & w_{(Q_1-1)(Q_2-1)(Q_3-1)}J_{n_{(Q_1-1)(Q_2-1)(Q_3-1)}} \end{bmatrix},$$

By using the weight matrix $\mathbf{W}$, we can rewrite the one-, two- or three-dimensional case in compact form as follows:

$$\mathcal{I} = \mathbf{W}u^{\delta}, \tag{1.39}$$

where $\mathcal{I}$ is the expression in Eq. 1.36, 1.37 or 1.38 and we used the definition of Eq. 1.30, 1.31 and 1.32 for u^{δ}.

As final remark, we point out that the error associated to the inexact integration (also referred to as under-integration) can lead to numerical instabilities (also known as aliasing-driven instabilities). In fact the aliasing of the under-resolved higher modes can inject energy into the lower modes and this leads the simulations to instability. This aspect has been well explained in Karniadakis and Sherwin (2005) and the investigation of these aliasing issues is carried out in this thesis (chapter 4), where we analyse the implications of these aliasing issues in discontinuous spectral/hp element methods and we propose possible strategies to alleviate them.

Numerical differentiation

The other important operation to be taken into account when implementing spectral/hp element methods is the numerical differentiation. Let us consider a generic function $u(\xi)$ and approximate it in terms of Lagrange polynomials $l_i(\xi)$ on a set of Q nodal points ξ_i:

$$u^{\delta}(\xi) = \sum_{i=0}^{Q-1} u(\xi_i)\ell_i(\xi), \tag{1.40}$$

where $Q \geq P+1$ and where we assume $u^{\delta}(\xi) \in \mathbb{P}_P([-1,1])$. In this way the representation of the function $u^{\delta}(\xi)$ is exact and its derivative at the nodal points ξ_i can be defined as follows:

$$\frac{du^{\delta}(\xi)}{d\xi} = \sum_{j=0}^{Q-1} \left.\frac{d\ell_j(\xi)}{d\xi}\right|_{\xi=\xi_i} u(\xi_i). \tag{1.41}$$

For properly characterising the differentiation matrix, it is necessary to choose (as for the numerical integration) a set of quadrature points. The differentiation matrices employed in this work use either Gauss-Legendre or Gauss-Lobatto-Legendre quadrature points. The interested reader can refer to Karniadakis and Sherwin (2005) for further details.

Eq. 1.41 can be easily extended to multidimensional cases. Specifically, for a two-dimensional tensor product element we have:

$$\begin{aligned}
\frac{du^{\delta}(\xi_{1_i},\xi_{2_j})}{d\xi_1} &= \sum_{p=0}^{Q_1-1}\sum_{q=0}^{Q_2-1} \left.\frac{d\ell_p(\xi_1)}{d\xi_1}\right|_{\xi_{1_i}} \ell(\xi_{2_j})u_{pq}, \\
\frac{du^{\delta}(\xi_{1_i},\xi_{2_j})}{d\xi_2} &= \sum_{p=0}^{Q_1-1}\sum_{q=0}^{Q_2-1} \left.\frac{d\ell_q(\xi_2)}{d\xi_2}\right|_{\xi_{2_j}} \ell(\xi_{1_i})u_{pq},
\end{aligned} \tag{1.42}$$

where Q_1 and Q_2 are the number of quadrature points in the ξ_1 and ξ_2 directions, respectively. For a three-dimensional tensor product element the extension reads as:

$$\begin{aligned}
\frac{du^{\delta}(\xi_{1_i},\xi_{2_j},\xi_{3_k})}{d\xi_1} &= \sum_{p=0}^{Q_1-1}\sum_{q=0}^{Q_2-1}\sum_{r=0}^{Q_3-1} \left.\frac{d\ell_p(\xi_1)}{d\xi_1}\right|_{\xi_{1_i}} \ell(\xi_{2_j}))l(\xi_{3_k})u_{pqr}, \\
\frac{du^{\delta}(\xi_{1_i},\xi_{2_j},\xi_{3_k})}{d\xi_2} &= \sum_{p=0}^{Q_1-1}\sum_{q=0}^{Q_2-1}\sum_{r=0}^{Q_3-1} \left.\frac{d\ell_q(\xi_2)}{d\xi_2}\right|_{\xi_{2_j}} \ell(\xi_{1_i}))l(\xi_{3_k})u_{pqr}, \\
\frac{du^{\delta}(\xi_{1_i},\xi_{2_j},\xi_{3_k})}{d\xi_3} &= \sum_{p=0}^{Q_1-1}\sum_{q=0}^{Q_2-1}\sum_{r=0}^{Q_3-1} \left.\frac{d\ell_r(\xi_3)}{d\xi_3}\right|_{\xi_{3_k}} \ell(\xi_{1_i}))l(\xi_{2_j})u_{pqr},
\end{aligned} \tag{1.43}$$

where Q_1, Q_2 and Q_3 are the number of quadrature points in the ξ_1, ξ_2 and ξ_3 directions, respectively. As for the numerical integration we can introduce the matrix form of the differentiation matrices for the one-dimensional, two-dimensional and three-dimensional cases. Specifically, in one dimension the differentiation matrix is denoted by:

$$\mathbf{D}_{\xi} = \left.\frac{d\ell_j(\xi)}{d\xi}\right|_{\xi=\xi_i}, \qquad 0 \leq i,j \leq Q-1. \tag{1.44}$$

In the two-dimensional case we have:

$$\mathbf{D}_{\xi_1} = \left.\frac{d\ell_p(\xi)}{d\xi_1}\right|_{\xi_{1_i}}, \qquad \mathbf{D}_{\xi_2} = \left.\frac{d\ell_q(\xi)}{d\xi_2}\right|_{\xi_{2_j}}, \tag{1.45}$$

where $\mathbf{D}_{\xi_1}$ and $\mathbf{D}_{\xi_2}$ are the one-dimensional derivative matrices in the ξ_1 and ξ_2 directions, respectively and the indices associated to the ξ_1 direction are defined as a set of Q_1 points: $0 \leq i, p \leq Q_1 - 1$, while those associated to the ξ_2 directions are defined on a set of Q_2 points: $0 \leq j, q \leq Q_2 - 1$. The extension to the three-dimensional case reads as follows:

$$\mathbf{D}_{\xi_1} = \left.\frac{d\ell_p(\xi)}{d\xi_1}\right|_{\xi_{1_i}}, \qquad \mathbf{D}_{\xi_2} = \left.\frac{d\ell_q(\xi)}{d\xi_2}\right|_{\xi_{2_j}}, \qquad \mathbf{D}_{\xi_3} = \left.\frac{d\ell_r(\xi)}{d\xi_3}\right|_{\xi_{3_k}}, \tag{1.46}$$

where $\mathbf{D}_{\xi_1}$, $\mathbf{D}_{\xi_2}$ and $\mathbf{D}_{\xi_3}$ are the one-dimensional derivative matrices in the ξ_1 and ξ_2 directions, respectively and the indices associated to the ξ_1 direction are defined on Q_1 points: $0 \leq i, p \leq Q_1 - 1$, those associated to the ξ_2 direction are defined on Q_2 points: $0 \leq j, q \leq Q_2 - 1$, while those associated to the ξ_3 direction are defined on a set of Q_3 points: $0 \leq k, r \leq Q_3 - 1$.

Backward and forward transformations

In this paragraph we introduce the transformation to evaluate the physical values from the basis coefficients (backward transformation) and the transformation for evaluating the coefficients of the expansion basis starting from the physical values (forward transformation).

Backward transformation
Reusing the matrix form defined previously for the basis function we can write the generic polynomial expansion u^δ in vector form as follows (such as that of Eq. 1.27):

$$\mathbf{u}^\delta(\xi) = \mathbf{B}(\boldsymbol{\xi})\mathbf{u}, \tag{1.47}$$

where $\mathbf{u}^\delta$ can be a one-, two- or three-dimensional expansion. Equation 1.47 is the discrete backward transformation and involves the transformation from the coefficients space to the physical space. Note that, if we use a nodal expansion basis, such as Lagrange polynomials, through a set of nodal points and the basis is evaluated at the same nodal points, then $\mathbf{B} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In this case we recover the Kronecker delta property, which in matrix form reads

$$\mathbf{u}^\delta = \mathbf{B}\mathbf{u} = \mathbf{I}\mathbf{u} = \mathbf{u}. \tag{1.48}$$

However, if the basis is evaluated at a set of points which does not correspond to the nodal points, then the matrix $\mathbf{B}$ is full. This arises in the case of quadrilateral

tensor product basis when the number of quadrature points is not equal to the polynomial order plus one, that is $Q \neq P + 1$.

Forward transformation
The forward transformation can be formulated by reusing some concepts of the method of weighted residuals introduced in section 1.2 . Specifically, we first define the error between the continuous function and its discrete representation as follows:

$$u^{\delta}(\xi_1, \xi_2, \xi_3) - u(\xi_1, \xi_2, \xi_3) = \sum_{pqr} u_{pqr}\phi_{pqr} - u = R(u), \tag{1.49}$$

where $R(u)$ is the residual introduced in Eq. 1.6.

Following the method of weighted residuals we multiply both the sides of Eq. 1.49 by a test function v and set the right hand-side to zero:

$$\left(v, \sum_{pqr} u_{pqr}\phi_{pqr}\right) - (v, u) = \underbrace{(v, R(u))}_{=0} \rightarrow \left(v, \sum_{pqr} u_{pqr}\phi_{pqr}\right) = (v, u). \tag{1.50}$$

The choice of the test function determines the type of projection (i.e. of elemental forward transformation). In particular, we distinguish between a collocation and a Galerkin projection.
The first is simply obtained by setting $v = \delta(\xi_{1_i}, \xi_{2_j})$ where δ is the Kronecker delta function. In matrix form this becomes:

$$\mathbf{u} = \mathbf{B}_{\mathcal{N}}^{-1}\mathbf{u}^{\delta} \tag{1.51}$$

and it is known as collocation forward transformation. $\mathbf{B}_{\mathcal{N}}$ is defined as $\mathbf{B}_{\mathcal{N}} = \phi_{pq}(\xi_{1_i}, \xi_{2_j})$. The second is instead obtained by setting the test functions equal to the expansion functions, that is $v = \phi$. The Galerkin transformation can be written in matrix form as follows:

$$\mathbf{u} = (\mathbf{B}^T\mathbf{W}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}\mathbf{u}^{\delta} = \mathbf{M}^{-1}\mathbf{B}^T\mathbf{u}^{\delta}, \tag{1.52}$$

where $\mathbf{W}$ is the weight matrix previously introduced and $\mathbf{M} = \mathbf{B}^T\mathbf{W}\mathbf{B}$ is the elemental mass matrix. The manipulations that produce the matrix form in Eq. 1.52 are not reported here for the sake of brevity. However, the interested reader can refer to chapter 4 in Karniadakis and Sherwin (2005). Note that also in this case, if we use a nodal expansion onto a set of nodal points and $Q = P + 1$ where Q is the number of points and P is the order of the expansion, we recover the Kronecker delta property, therefore Eq. (2.43) simplifies as follows:

$$\mathbf{u} = (\mathbf{B}^T\mathbf{W}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}\mathbf{u}^{\delta} = (\mathbf{I}^T\mathbf{W}\mathbf{I})^{-1}\mathbf{I}^T\mathbf{W}\mathbf{u}^{\delta} = \mathbf{W}^{-1}\mathbf{W}\mathbf{u}^{\delta} = \mathbf{u}^{\delta}, \tag{1.53}$$

where we used $\mathbf{B} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

Connectivity

After having decomposed the domain into N non-overlapping elements Ω_n and performed the required operations on each of these subdomains, it is necessary to impose connectivity across the elements in order to construct the global problem. There are different ways of reinforcing the connectivity. For example, when dealing with continuous Galerkin methods, we require the solution to be continuous between two adjacent elements, whereas when dealing with discontinuous-type approaches, such as the discontinuous Galerkin and the flux reconstruction approaches, we require the flux and not the solution to be continuous between two adjacent elements. Therefore, we describe separately how to impose connectivity for a continuous and a discontinuous discretisation.

Continuous discretisation
For explaining the connectivity on a continuous framework we focus on the classical continous Galerkin method. When using this approach, C^0 continuity is imposed across the elements through the so-called *global assembly* strategy (also referred to as *direct stiff summation*). We describe this methodology by defining a generic function $u(x_1, x_2, x_3)$ expanded as follows:

$$u^{\delta}(x_1, x_2, x_3) = \sum_{n=0}^{N-1} \sum_{p=0}^{P_1-1} \sum_{q=0}^{P_2-1} \sum_{r=0}^{P_3-1} u_{pqr_n} \phi_{pqr_n}(x_1, x_2, x_3), \tag{1.54}$$

where N is the number of subdomains (elements), u_{pqr_n} are the local expansion coefficients for a given element Ω_n. In vectorial notation 1.54 can be represented as:

$$\mathbf{u}_l = \underline{\mathbf{u}}_n \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N-1} \end{bmatrix}, \tag{1.55}$$

where $\underline{\mathbf{u}}_n$ is the concatenation of the local expansion coefficients. The relation between the global expansion coefficients $\mathbf{u}_g$ and the local ones, $\mathbf{u}_l$, can be represented through a matrix-vector multiplication:

$$\mathbf{u}_l = \mathbf{A}\mathbf{u}_g, \tag{1.56}$$

where $\mathbf{A}$ is a very sparse matrix with entries typically equal to ± 1. This matrix applies the connectivity rules and is called the assembly matrix.

Discontinuous discretisation
In a discontinuous discretisation, such as the discontinuous Galerkin and the flux reconstruction approaches, the solution is allowed to be discontinuous across two adjacent elements but the fluxes are required to be continuous. In this case the connectivity across the elements is obtained through a boundary term which is composed by a numerical flux $\boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta)$. This flux depends on the solution on the left, u_+^δ, and on the right, u_-^δ, with respect to a given interface as shown in Figure 1.4.

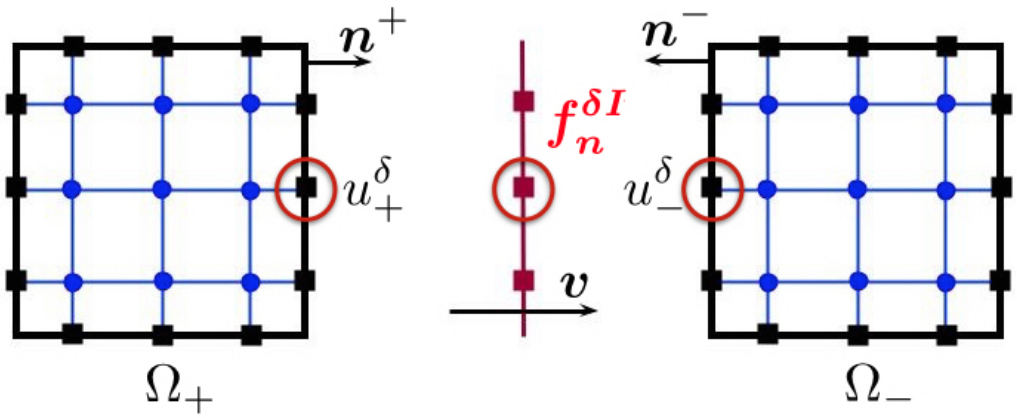


Figure 1.4: Conventions adopted for calculating the numerical fluxes in discontinuous spectral/hp element methods.

The form of the numerical flux depends on the problem being solved. In case of a linear advection equation for instance, we can use the following upwind numerical flux:

$$\boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta) = \begin{cases} \boldsymbol{v}u_+^\delta, & \boldsymbol{v}\cdot\boldsymbol{n}_+ \geq 0\,, \\ \boldsymbol{v}u_-^\delta, & \boldsymbol{v}\cdot\boldsymbol{n}_+ < 0\,, \end{cases} \tag{1.57}$$

where $\boldsymbol{n}_+$ is the outward pointing normal of the left element and $\boldsymbol{v}$ is the advection velocity, as illustrated in Figure 1.4.

When solving more complicated hyperbolic problems, such as compressible flows we need to apply connectivity through more complex forms of numerical fluxes which are based on the physics of the problem. Specifically, for discretising a first order flux (e.g. the flux of the Euler equations or the advective flux of the Navier-Stokes equations) we need to use either approximate or exact Riemann solvers. This implies solving a Riemann problem at each interface between two elements. In chapter 3 we describe the form of the numerical fluxes when approximating the advection operator of the equations governing compressible flows. In

the case of a second order flux (e.g. the diffusion flux of the Navier-Stokes equations) connectivity is again imposed through a numerical flux with a form slightly different from that of the advection term. The numerical flux associated with a second order problem is described in section 1.5 when discussing the discontinuous Galerkin and flux reconstruction diffusion operators.

1.3.2 Time discretisation

After having discretised the problem in space, if the problem is time-dependent, the partial differential problem reduces to a set of ordinary differential equations (ODEs), where the only differentiation variable is the time t. There are two main categories of time-integration methods for integrating an ODE or a system of ODEs:

- multi-step methods: they combine the information from previous time-levels (or time-steps) to calculate the solution at the new time-level;
- multi-stage methods: they combine the information at various stages between two time-steps discarding the information from previous time-steps.

The first strategy is more memory demanding because it is necessary to store the information of the previous time-steps but, on the other hand, it requires a relatively limited number of floating point operations. The second strategy is instead more CPU intensive (more floating point operations per time-step) but needs less memory. Both strategies can be explicit - the current time-step is calculated using information from the previous time-step(s) only - or implicit - the current time-step is computed solving a nonlinear problem which involves the current time-step. Multi-step and multi-stage methods, can be grouped together by means of the unified General Linear (GL) method presented by Butcher (1987). In this work the solver adopted makes use of explicit time-integration schemes, specifically Runge-Kutta schemes. In the following we introduce the GL method and successively we describe the explicit Runge-Kutta schemes which can be directly derived from the GL method.
Consider the discretised linear differential equation:

$$\mathcal{L}(u^\delta) = R(u^\delta) \tag{1.58}$$

and assume that we have chosen one of the spatial discretisation described before. The problem reduces to an ODE dependent only on time t. We have obtained an initial-value problem that is:

$$\frac{du^\delta}{dt} = f(u^\delta), \qquad u^\delta(t_0) = u_0^\delta, \tag{1.59}$$

where, with f we indicate the approximation of the spatial derivative of the differential problem. We now need to find a suitable approximation to solve Eq. 1.59 in order to propagate the solution in time. By defining the discrete time as $t \to t^k$ and introducing a given finite interval of time or time-step $\Delta t = (t^{k+1} - t^k)$, the GL method is stated as finding $u^\delta(t^{k+1})$ such that

$$\begin{cases} U_i^\delta = \Delta t \sum_{j=1}^{s} z_{ij}^{(1)} F_j + \sum_{j=1}^{r} z_{ij}^{(3)} u_j^\delta(t^k) & 1 \leq i \leq s, \\ u_i^\delta(t^{k+1}) = \Delta t \sum_{j=1}^{s} z_{ij}^{(2)} F_j + \sum_{j=1}^{r} z_{ij}^{(4)} u_j^\delta(t^k) & 1 \leq i \leq r, \end{cases} \tag{1.60}$$

where $z_{ij}^{(1)}$, $z_{ij}^{(2)}$, $z_{ij}^{(3)}$ and $z_{ij}^{(4)}$ are the elements of the matrices which define the time-integration scheme, s and r are the stages and the steps, respectively, U_i^δ are the stage values and $F_i = f(U_i^\delta)$ are the stage derivatives. The explicit Runge-Kutta schemes adopted in this work can be defined within the GL method framework by rewriting Eq. 1.60 in matrix form:

$$\begin{bmatrix} \mathbf{U}^\delta \\ \mathbf{u}^\delta(t^{k+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{Z}^1 \otimes \mathbf{I} & \mathbf{Z}^3 \otimes \mathbf{I} \\ \mathbf{Z}^2 \otimes \mathbf{I} & \mathbf{Z}^4 \otimes \mathbf{I} \end{bmatrix} = \begin{bmatrix} \Delta t\, \boldsymbol{F} \\ \mathbf{u}^\delta(t^k) \end{bmatrix}, \tag{1.61}$$

where $\mathbf{I}$ is the identity matrix. For a 4^{th}-order Runge-Kutta scheme the matrices $\mathbf{Z}^1$, $\mathbf{Z}^2$, $\mathbf{Z}^3$ and $\mathbf{Z}^4$ are as follows:

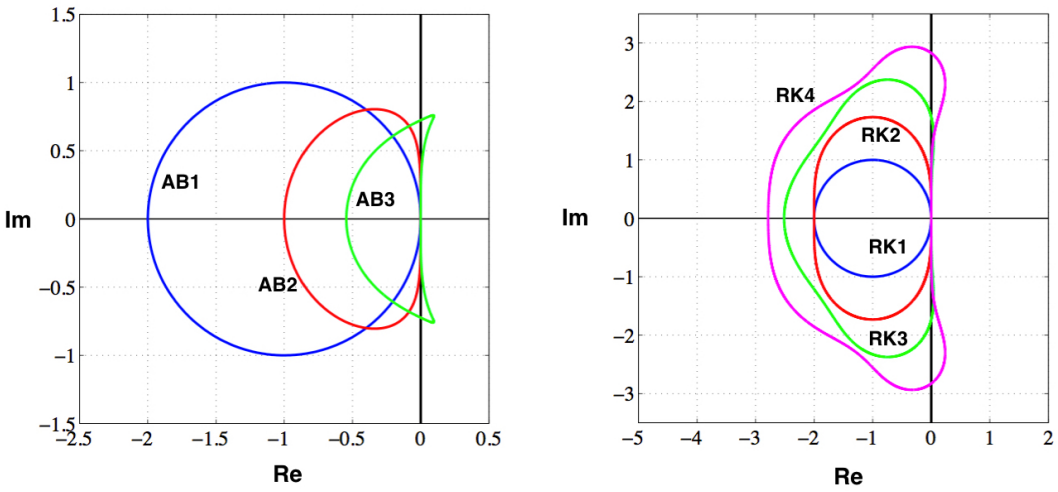
$$\left[\begin{array}{c|c} \mathbf{Z}^{(1)} & \mathbf{Z}^{(3)} \\ \hline \mathbf{Z}^{(2)} & \mathbf{Z}^{(4)} \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ \hline 1/6 & 1/3 & 1/3 & 1/6 & 1 \end{array}\right]. \tag{1.62}$$

In Nektar++ the implementation of the time-integration schemes is performed using the GL method strategy and therefore by defining a matrix whose coefficients change depending on the scheme chosen. In theory, we could have also used other explicit methods for the time-integration, such as forward Euler, Adams-Bashforth and other Runge-Kutta schemes. In practice, although more expensive than the others, we mainly used the 4th-order Runge-Kutta scheme whose stability region was satisfactory for the majority of the studies of this work. Specifically, the semi-discrete system of Eq. 1.59 can be expressed as:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}u \tag{1.63}$$

where $\mathbf{A}$ represents the discretisation of the advection operator and $\mathbf{u}$ is the vector of expansion coefficients. To maintain numerical stability, the eigenvalue spectrum

of **A** multiplied by the time-step Δt must lie within the stability region of the chosen time-integration scheme. In figure 1.5 the stability regions of different Adams-Bashforth and Runge-Kutta schemes[1] are depicted. In the following the time-step restriction which arises when solving an ODE with an explicit time integration scheme is described.



(a) (b)

Figure 1.5: Regions of stability of the ODE solvers. On the left the Adam-Bashforth schemes 1,2,3. On the right the Runge-Kutta schemes 1,2,3,4.

Numerical evaluation of the time-step restriction

The stability of an explicit time-integration scheme is governed by the CFL condition. This condition imposes that the space-time numerical domain of dependence has to include the analytical one. To formalise the definition of the CFL condition we consider the following hyperbolic equation:

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = 0 \tag{1.64}$$

where $\mathbf{v}$ is the local advection velocity. For a general domain with minimum mesh spacing h with local advection velocity $\mathbf{v}$ and given the definition of Courant number:

$$C = \frac{|\mathbf{v}|\Delta t}{h}, \tag{1.65}$$

[1]Note that 1st-order Adams-Bashforth and 1st-order Runge-Kutta are equivalent to forward Euler.

the CFL condition translates into defining the admissible values of C, and therefore, selecting h and Δt such that $C \leq c_{\max}$. With an explicit time-integration method the numerical stability is guaranteed if $c_{\max} = 1$. In the example of Eq. 1.64 the CFL condition can be easily fulfilled but in real CFD application the scenario becomes more complex.
In general, after having discretised the spatial domain, the evaluation of the time-step limit restriction is equivalent to bounding the largest eigenvalue $\lambda_{\max}$ of the advection operator:

$$\Delta t \cdot \lambda_{\max} \leq const\,. \tag{1.66}$$

It is possible to demonstrate that the maximum eigenvalue in the standard region can be bounded by:

$$c_\lambda P^2 > \lambda^{st}_{\max} \tag{1.67}$$

where c_λ is a constant (see Karniadakis and Sherwin (2005)). In a general domain of characteristic length h with local advection velocity $\mathbf{v}$ we expect that the maximum eigenvalue will be of the form:

$$C_\lambda(\mathbf{v}, h)P^2 > \lambda_{\max}\,. \tag{1.68}$$

The evaluation of $c_\lambda(\mathbf{v}, h)$ highly depends on the problem considered and so it is difficult to formalise it in an exact algebraic expression. Nevertheless, we now have a first condition that the time-step restriction is bounded by:

$$\Delta t \leq \frac{\alpha}{C_\lambda(\mathbf{v}, h)P^2} \tag{1.69}$$

where α represents the distance from the origin of the boundary of the stability region of the time-integration scheme along the azimuthal of the dominating eigenvalue $\lambda_{\max}$.
To apply Eq. 1.69 in a numerical algorithm we still require an estimate of $C_\lambda(\mathbf{v}, h)$. We note that an estimate of c_λ in the standard region can be obtained from numerical experiments and analytical results. Due to the local elemental nature of the problem we can apply the mapping from the elemental region to the standard region also to the advection velocity $\mathbf{v}$ in order to evaluate a local velocity in the standard element. Having obtained the local velocity, we can then scale the previous approximation of the maximum eigenvalue in the standard element which was done for a unit local velocity to approximate the maximum eigenvalue for a non-unit velocity in a general-shaped element. Denoting $\mathbf{v}^e = [v^e_1, v^e_2, v^e_3]$ as the velocity in the element e and $\mathbf{v}^{st} = [v^{st}_1, v^{st}_2, v^{st}_3]$ as the velocity in the standard

region we can evaluate $\mathbf{v}^{st}$ in terms of $\mathbf{v}^{e}$ by applying the chain rule:

$$\begin{aligned} v_1^{st} &= v_1^e \frac{\partial \xi_1}{\partial x_1} + v_2^e \frac{\partial \xi_1}{\partial x_2} + v_3^e \frac{\partial \xi_1}{\partial x_3}\,, \\ v_2^{st} &= v_1^e \frac{\partial \xi_2}{\partial x_1} + v_2^e \frac{\partial \xi_2}{\partial x_2} + v_3^e \frac{\partial \xi_2}{\partial x_3}\,, \\ v_3^{st} &= v_1^e \frac{\partial \xi_3}{\partial x_1} + v_2^e \frac{\partial \xi_3}{\partial x_2} + v_3^e \frac{\partial \xi_3}{\partial x_3}\,. \end{aligned} \tag{1.70}$$

By considering the maximum value of the local velocity $|\mathbf{v}^{st}|$ and using the bound on the local maximum eigenvalue of Eq. 1.67 we can write an estimate of the maximum eigenvalue $\lambda_{\max}$ as:

$$\max|\mathbf{v}^{st}| c_\lambda P^2 \approx \lambda_{\max}\,, \tag{1.71}$$

which is equivalent to approximating the constant $C_\lambda(\mathbf{v}, h)$ by:

$$C_\lambda(\mathbf{v}, h) \approx c_\lambda \max|\mathbf{v}^{st}|\,. \tag{1.72}$$

In practice we evaluate $\mathbf{v}$ at each quadrature point to determine the maximum and typically we choose a value of $c_\lambda = 0.2$. In one dimension we note that the geometric factor which map an element of size h to the standard element is $\partial\xi_1/\partial x_1 = 2/h$ and so the constant becomes:

$$C_\lambda(\mathbf{v}, h) \approx \frac{c_\lambda}{h} \max|\mathbf{v}^{e}|\,. \tag{1.73}$$

Substituting Eq. 1.73 into Eq. 1.69 we can write:

$$\max|\mathbf{v}^{e}| \frac{c_\lambda}{h} P^2 \Delta t \leq \alpha\,. \tag{1.74}$$

The term $h/(c_\lambda P^2)$ can be considered as an approximation of the minimum mesh spacing Δx and so we ca see that Eq. 1.74 recovers the definition of the CFL condition of Eq. 1.65 used in finite differences. For the simple case of the one-dimensional linear advection equation, we can compute the maximum admissible $\Delta t_{\max}$ from the following relation:

$$\Delta t_{\max} = \frac{\alpha}{\lambda_{\max}} \tag{1.75}$$

where $\lambda_{\max}$ and α depend on the spatial discretisation (the information of minimum h is included in the right-hand side of the equation) used and on the polynomial order of the solution. In Table 1.1 it is reported a study of time-restriction for

different polynomial order for the simple case of a two-dimensional linear advection equation for different polynomial orders using a 4th-order Runge-Kutta scheme.

The author wants to remark that the limitation on the time-step would be much less restrictive ($C_{\max}$ greater than 1 in Eq. 1.65) when using an implicit time-integration scheme. Furthermore, even if the benefits of the high order methods can be better appreciated when using a large final time, in many of the simulations[2] of this thesis the final output time selected is relatively short. This choice can be explained with the very demanding time-step selected in order for the time-integration error to become negligible ($\Delta t = \Delta t_{\max}/100$). In addition in most of these studies the goal was examining the connections between different schemes or techniques and, for this purpose, even a short final time was sufficient.

1.3.3 Summary

The concepts and the nomenclature introduced in the last two sections will be extensively used in this thesis. In the rest of this chapter we will describe in detail two discontinuous spatial discretisations, the discontinuous Galerkin (DG) and the flux reconstruction (FR) approaches for both first and second order problems by reusing some of the conventions presented in section 1.3.1.

1.4 DG and FR schemes: advection operator

In this section, we describe the advection operator of the discontinuous Galerkin method and of the flux reconstruction approach for solving a general first order problem. For these types of problems the implementation of the FR approach in the spectral/hp element library Nektar++ has been carried out for one- and two-dimensional (quadrilateral) tensor product elements. The advection operator for the DG approach was already existing. For describing the two approaches we consider the following three-dimensional first order conservation law:

$$\frac{\partial u}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(u) = 0, \tag{1.76}$$

within a domain $\Omega \in \mathbb{R}^3$, with $\boldsymbol{f} = [f_1, f_2, f_3]$, where $f_1 = f_1(u)$, $f_2 = f_2(u)$ and $f_3 = f_3(u)$ are the advection fluxes in the x_1, x_2 and x_3 Cartesian directions, respectively and where t is the time, $u = u(x_1, x_2, x_3, t)$ is the conserved variable and $\nabla_{\boldsymbol{x}} = [\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3]$.

In both the discontinuous Galerkin method and the flux reconstruction approach we split the domain Ω into N non-overlapping subdomains Ω_n as shown in

[2]See for example the studies on the advection equation and isentropic vortex in chapters 1, 2 and 3 and unsteady diffusion equation in chapter 1

Eq. 1.11, and then we define the problem on a local element Ω_n. The solution of the global problem is given by the summation of the elemental contribution of the computational domain.

All the steps described in the following two sections are very general and they can therefore be applied to any first order PDE, such as the linear advection equation, the Burgers' equation and the compressible Euler equations. They can also be applied to advection-diffusion problems where we need to discretise an advection term, (e.g. compressible Navier-Stokes equations).

1.4.1 Discontinuous Galerkin approach

To describe the DG method applied to Eq. 1.76 we introduce the space of the test function ℓ:

$$\mathcal{V}^\delta = \left\{ \ell \in \mathrm{L}^2(\Omega) : \ell|_{\Omega_n} \in \mathbb{P}_\mathrm{P}(\Omega_n), \ \forall \Omega_n \right\}, \tag{1.77}$$

where $\mathbb{P}_\mathrm{P}$ is the space of the three-dimensional piecewise continuous polynomials of order $P \geq 1$ on Ω and

$$\mathrm{L}^2(\Omega) = \left\{ \ell \ : \ \Omega_n \rightarrow \mathbb{R} \mid \iiint_{\Omega_n} |\ell(\boldsymbol{x}^2| d\boldsymbol{x} \leq \infty \right\}, \tag{1.78}$$

with $\mathbb{R}$ being the space of the real numbers. We now approximate the solution u by a piecewise three-dimensional polynomial $u^\delta \in \mathcal{V}^\delta$. In general the approximation space can be of modal or nodal type. If we use a nodal basis, such as Lagrange polynomials, we obtain the common class of nodal DG schemes. We then multiply Eq. 1.76 by the test function ℓ and integrate over a local element Ω_n such that:

$$\iiint_{\Omega_n} \ell(\boldsymbol{x}) \frac{\partial u_n^\delta(\boldsymbol{x})}{\partial t} d\boldsymbol{x} + \iiint_{\Omega_n} \ell(\boldsymbol{x}) \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}_n^\delta \right) d\boldsymbol{x} = 0. \tag{1.79}$$

We then integrate by parts the second term of Eq. 1.79 and we obtain the weak form of the DG method:

$$\begin{gathered} \iiint_{\Omega_n} \ell(\boldsymbol{x}) \frac{\partial u_n^\delta(\boldsymbol{x})}{\partial t} d\boldsymbol{x} - \\ \underbrace{\iiint_{\Omega_n} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}) \cdot \boldsymbol{f}_n^{\delta D} d\boldsymbol{x}}_{\text{volumetric term}} + \underbrace{\iint_{\partial\Omega_n} \ell(\boldsymbol{x}) \left(\boldsymbol{f}_n^{\delta I} \cdot \mathbf{n} \right) ds}_{\text{boundary term}} = 0, \end{gathered} \tag{1.80}$$

where the flux $\boldsymbol{f}_n^{\delta D}$ is called discontinuous since it is evaluated from the piecewise discontinuous solution between the elements, $\mathbf{n}$ is the outward pointing normal with respect to a given face of the local element Ω_n and the solution u^δ satisfies Eq. 1.80 for all the $\ell(\boldsymbol{x}) \in \mathcal{V}^\delta$.

The second term of Eq. 1.80 represents the volumetric term and involves only elemental operations. In order to couple the various elements of the computational domain together, thus solve the global problem, it is necessary to propagate the solution across the domain by defining a numerical flux as seen in section 1.3.1. The third term represents this numerical flux $\boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta)$ which couples the adjacent elements and allows the solution to propagate across the domain. As seen in section 1.3.1 the values u_+^δ and u_-^δ represent the solution on the two faces of a given interface. In order to calculate this flux, the values of the approximate solution at the boundaries of the element must be evaluated. This can be easy if the set of solution points comprises points on the faces of the element, otherwise an interpolation has to be performed (see figure 1.6). The boundary conditions are also enforced through the numerical flux $\boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta)$. In this case, u_-^δ represents the value of the boundary contained on a ghost point. In order for this coupling to generate a stable numerical scheme, it is necessary to make some considerations based on the natural propagation of the information.

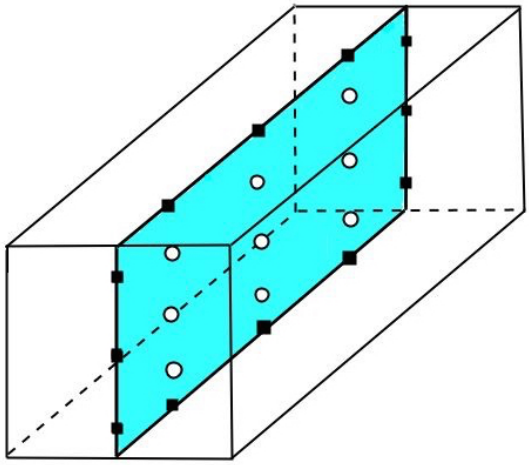


Figure 1.6: Distribution of solution points (circles) and interface flux points (squares) along a plane which cuts the hexahedral element.

Specifically, if we are for instance solving the linear advection equation, we could use a simple Riemann solver based on an upwind flux as that of Eq. 1.57. For more complicated hyperbolic problems, such as the compressible Euler equations, we need to use more complex one-dimensional Riemann solvers, either approximate or exact. In chapter 3 we will detail some of these solvers used in simulations of compressible flows. For further details the interested reader can refer to Toro

(2009).

Eq. 1.80 can be rewritten in a more compact form as follows:

$$\left(\ell(\boldsymbol{x}), \frac{\partial u_n^\delta(\boldsymbol{x})}{\partial t}\right)_{\Omega_n} - \left(\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x}), \boldsymbol{f}_n^{\delta D}\right)_{\Omega_n} + \left\langle \ell(\boldsymbol{x}), \boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta)\right\rangle_{\partial\Omega_n} = 0, \tag{1.81}$$

where the elemental inner products are:

$$\begin{gathered}(g_1, g_2) = \iiint_{\Omega_n} g_1 g_2 d\boldsymbol{x}, \qquad (\mathbf{g}_1, \mathbf{g}_2) = \iiint_{\Omega_n} \mathbf{g}_1 \mathbf{g}_2 d\boldsymbol{x}, \\ \langle g_1, \mathbf{g}_2\rangle = \iint_{\partial\Omega_n} g_1 \mathbf{g}_2 \cdot \mathbf{n} ds\,.\end{gathered} \tag{1.82}$$

For implementation convenience, it is common to transform Eq. 1.80 into the standard space. This means redefining each elemental contribution of Eq. 1.80 into the standard element Ω_s by using the mapping $\boldsymbol{x} = \boldsymbol{\Theta}(\boldsymbol{\xi})$ defined in section 1.3.1, such that:

$$\begin{gathered}\iiint_{\Omega_s} \ell(\boldsymbol{\Theta}(\boldsymbol{\xi}))\frac{\partial \widehat{u}_n^\delta(\boldsymbol{\Theta}(\xi))}{\partial t} J_n d\boldsymbol{\xi} - \iiint_{\Omega_s} \nabla_{\boldsymbol{\xi}}\ell(\boldsymbol{\Theta}(\boldsymbol{\xi}))\cdot \widehat{\boldsymbol{f}}_n^{\delta D} J_n d\boldsymbol{\xi} + \\ \iint_{\partial\Omega_s} \ell(\boldsymbol{\Theta}(\boldsymbol{\xi}))\widehat{(\boldsymbol{f}_n^{\delta I}\cdot \mathbf{n})} J_{2D} d\widehat{s} = 0,\end{gathered} \tag{1.83}$$

where $\widehat{\boldsymbol{f}}_n^{\delta D} = J_n \boldsymbol{G}_{3D}^{-1}\boldsymbol{f}_n^{\delta D}$, J_{2D} is the two-dimensional Jacobian referred to the face of the element and where we used the following formula for the normals:

$$\boldsymbol{n} = J_{2D}\boldsymbol{G}_{3D}^{-T}\hat{\boldsymbol{n}}. \tag{1.84}$$

Eq.(1.83) can be written in matrix form using the nomenclature introduced in section 1.3.1 and removing the subscript "n" for sake of clarity. In particular, the (P + 1) × (P + 1) × (P + 1) equations defined at each solution point within each given element are:

$$\begin{aligned}&(\mathbf{B})^{\mathrm{T}}\mathbf{W}\mathbf{B}\frac{\mathrm{d}\widehat{\mathbf{u}}}{\mathrm{d}t} - (\mathbf{D}_{x_1}\mathbf{B})^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_1^D - (\mathbf{D}_{x_2}\mathbf{B})^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_2^D \\ &-(\mathbf{D}_{x_3}\mathbf{B})^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0,\end{aligned} \tag{1.85}$$

where $\widehat{\mathbf{b}}^{\mathrm{DG}}$ is the boundary surface integral of Eq. 1.83. The matrices $\mathbf{D}_{x_j}$ can be written in terms of partial derivatives in ξ_i by using the chain rule as follows:

$$\mathbf{D}_{x_j} = \sum_{i=1}^{3} \boldsymbol{\Lambda}(G_{i/j})\mathbf{D}_{\xi_i}, \tag{1.86}$$

where $\mathbf{\Lambda}(\cdot)$ is a diagonal matrix which contains the factors:

$$G_{i/j} = \frac{\partial \xi_i}{\partial x_j} \tag{1.87}$$

and $\mathbf{D}_{\xi_i}$ are the differentiation matrices defined in Eq. 1.46. Substituting Eq. 1.86 and 1.87 into Eq. 1.85 we obtain:

$$\begin{aligned}(\mathbf{B})^{\mathrm{T}}\mathbf{W}\mathbf{B}\frac{\mathrm{d}\,\widehat{\mathbf{u}}}{\mathrm{d}t} - \left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/1})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_1^D - \\ \left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/2})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_2^D - \\ \left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/3})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\,\widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0,\end{aligned} \tag{1.88}$$

where the diagonal matrix $\mathbf{\Lambda}(\cdot)$ has on its diagonal the geometric factors defined in Eq. (1.87). The final matrix form of the DG method can be written as follows:

$$\begin{aligned}\mathbf{M}\frac{\mathrm{d}\widehat{\mathbf{u}}}{\mathrm{d}t} - \left\{\left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/1})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_1^D - \\ \left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/2})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_2^D - \\ \left[\left(\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/3})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_3^D\right\} + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0,\end{aligned} \tag{1.89}$$

where $\mathbf{M}$ is the elemental mass matrix:

$$\mathbf{M} = (\mathbf{B})^T\mathbf{W}\mathbf{B}, \tag{1.90}$$

$\mathbf{W}$ is the matrix of weights and we used the backward transformation of Eq. 1.47 to represent the discrete solution polynomial (note that the Jacobian J_n in Eq. 1.83 has been incorporated into the weight matrix $\mathbf{W}$). The matrix form of the DG method can be further reduced by defining the following three matrices:

$$\begin{aligned}\mathbf{S}_{x_1} &= -\left[\left(\textstyle\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/1})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}, \\ \mathbf{S}_{x_2} &= -\left[\left(\textstyle\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/2})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}, \\ \mathbf{S}_{x_3} &= -\left[\left(\textstyle\sum_{i=1}^{3}\mathbf{\Lambda}(G_{i/3})\mathbf{D}_{\xi_i}\right)\mathbf{B}\right]^{\mathrm{T}}\mathbf{W}\mathbf{B}\end{aligned} \tag{1.91}$$

and substituting them into Eq. 1.89 such that:

$$\mathbf{M}\frac{d\widehat{\mathbf{u}}}{dt} + \mathbf{S}_{x_1}\widehat{\mathbf{f}}_1^D + \mathbf{S}_{x_2}\widehat{\mathbf{f}}_2^D + \mathbf{S}_{x_3}\widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0. \tag{1.92}$$

The final form of the DG method can then be advanced in time via any explicit time discretisation scheme, such as the explicit Runge-Kutta methods.

1.4.2 Flux Reconstruction approach

The approximation space of the FR approach is identical to that of the DG method defined in Eq. 1.77. Whilst the DG method is formulated starting from an integral form, the FR approach starts from the differential form of Eq. 1.76. From this perspective, it is convenient to map each element Ω_n into a standard element Ω_s via the generic mapping of equation 1.19 or Eq. 1.21. The procedure for solving the conservation law of Eq. 1.76 can be described inside the standard element Ω_s since the only difference between one element and another is the mapping $\boldsymbol{\Theta}$. In the reference space Eq. 1.76 becomes:

$$\frac{\partial \widehat{u}_n^\delta}{\partial t} + \nabla_{\boldsymbol{\xi}} \cdot \widehat{\mathbf{f}}_n^\delta = 0, \tag{1.93}$$

where the transformed solution is represented as:

$$\widehat{u}_n^\delta(\boldsymbol{\xi}, t) = J_n u_n^\delta(\boldsymbol{\Theta}_n(\boldsymbol{\xi}), t), \tag{1.94}$$

and the transformed fluxes are:

$$\widehat{\boldsymbol{f}}_n^\delta(\boldsymbol{\xi}, t) = J_n \mathbf{G}_n^{-1} \boldsymbol{f}_n^\delta = \begin{bmatrix} \widehat{f}_{1,n}^\delta \\ \widehat{f}_{2,n}^\delta \\ \widehat{f}_{3,n}^\delta \end{bmatrix} = J_n \begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} & \frac{\partial \xi_1}{\partial x_3} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} & \frac{\partial \xi_2}{\partial x_3} \\ \frac{\partial \xi_3}{\partial x_1} & \frac{\partial \xi_3}{\partial x_2} & \frac{\partial \xi_3}{\partial x_3} \end{bmatrix} \cdot \begin{bmatrix} f_{1,n}^\delta \\ f_{2,n}^\delta \\ f_{3,n}^\delta \end{bmatrix} \tag{1.95}$$

and $\nabla_{\boldsymbol{\xi}} = [\partial/\partial\xi_1, \partial/\partial\xi_2, \partial/\partial\xi_3]$. The metric terms $\mathbf{G}_n$ and J_n have been defined in section 1.3.1.

The first step to construct the FR approach is to define the approximate solution which is a multi-dimensional polynomial of degree P on a set of $(\mathrm{P}+1)^3$ solution points within each standard element Ω_s:

$$\widehat{u}_n^\delta(\boldsymbol{\xi}, t) = \sum_{i,j,k}^{P} \ell_{ijk}(\boldsymbol{\xi}) \widehat{u}_{ijk}^\delta = \sum_{i,j,k}^{P} \ell_i(\xi_1)\ell_j(\xi_2)\ell_k(\xi_3)\widehat{u}_{ijk}^\delta, \tag{1.96}$$

where u^{δ}_{ijk} are the values of the known solution and $\ell_{ijk}(\boldsymbol{\xi})$ are three-dimensional continuous polynomials (such as Lagrange polynomials). Note that, the solution expansion in Eq. 1.96 incorporates the Jacobian J_n. This expansion is identical to expanding the Jacobian and the solution separately in the case of a collocation projection (and if the number of quadrature points is equal to $Q = P + 1$, where P is the polynomial order of the expansion) or if the Jacobian is constant. In general, when $Q \geq P + 1$, expanding the Jacobian and the solution separately produces a different result than Eq. 1.96. In Nektar++, the implementation has been carried out using two different expansions for the Jacobian (including the metric terms) and the solution. Figure 1.7 shows the approximate solution in the standard element for a one-dimensional case.

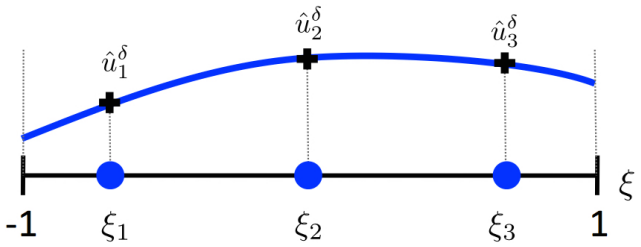


Figure 1.7: Approximation of the solution in a standard 1D element.

In the second step we calculate an order P polynomial flux, called *discontinuous flux* and denoted by $\widehat{\boldsymbol{f}}^{\delta D}_n(\boldsymbol{\xi}) = \left[\widehat{f}^{\delta D}_{1,n}, \widehat{f}^{\delta D}_{2,n}, \widehat{f}^{\delta D}_{3,n}\right]$ using the known solution defined at the previous step:

$$\begin{aligned}
\widehat{f}^{\delta D}_{1,n} &= \sum_{i,j,k=0}^{P} \ell_{ijk}(\boldsymbol{\xi}) \widehat{f}^{\delta D}_{1_{ijk}} = \sum_{i,j,k=0}^{P} \ell_i(\xi_1)\ell_j(\xi_2)\ell_k(\xi_3) \widehat{f}^{\delta D}_{1_{ijk}}, \\
\widehat{f}^{\delta D}_{2,n} &= \sum_{i,j,k=0}^{P} \ell_{ijk}(\boldsymbol{\xi}) \widehat{f}^{\delta D}_{2_{ijk}} = \sum_{i,j,k=0}^{P} \ell_i(\xi_1)\ell_j(\xi_2)\ell_k(\xi_3) \widehat{f}^{\delta D}_{2_{ijk}}, \\
\widehat{f}^{\delta D}_{3,n} &= \sum_{i,j,k=0}^{P} \ell_{ijk}(\boldsymbol{\xi}) \widehat{f}^{\delta D}_{3_{ijk}} = \sum_{i,j,k=0}^{P} \ell_i(\xi_1)\ell_j(\xi_2)\ell_k(\xi_3) \widehat{f}^{\delta D}_{3_{ijk}},
\end{aligned} \tag{1.97}$$

for the ξ_1, ξ_2, ξ_3 directions, respectively. We use the superscript "D" to indicate that the solution can be discontinuous between adjacent elements. In Eq. 1.97 the considerations made for the geometric terms are still valid. In particular, the expansion include the geometric factors and they are identical to representing the fluxes and the metric terms separately if we use a collocation projection or if the element is not deformed. Figure 1.8 shows the approximate flux in the standard element for a one-dimensional case.

The third step is the calculation of the numerical flux denoted by $\boldsymbol{f}_n^{\delta I} = \boldsymbol{f}_n^{\delta I}(u_+^\delta, u_-^\delta)$ with a procedure identical to that adopted in the DG method. Henceforth the numerical interface fluxes associated with the left and right ends of Ω_s will be denoted as $\widehat{f}_L^{\delta RI}$ and $\widehat{f}_R^{\delta RI}$ respectively.

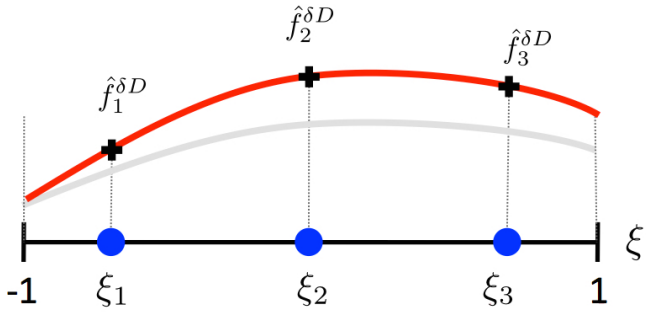


Figure 1.8: Approximation of the discontinuous flux in a standard 1D element.

The fourth step of the FR approach consists in adding a correction flux $\widehat{\boldsymbol{f}}_n^{\delta C}$ to the approximate discontinuous flux $\widehat{\boldsymbol{f}}_n^{\delta D}$. The corrective flux is a polynomial function of degree $P+1$ and thus the total flux is of degree $P+1$ as well. The sum of the discontinuous and corrective fluxes is equal to the transformed Riemann interface flux at the boundaries of each element and follows the approximate discontinuous flux (in some sense) within each standard element. This permits coupling between the elements since the total flux is now continuous. To satisfy all these requirements six correction functions of degree $P+1$ have to be defined, one for each face of the hexahedral standard element, Ω_s and each correction function has to satisfy numerical constraints for each of the flux points of the correspondent face.

We define the correction functions as $\Psi_L(\xi_1)$ and $\Phi_R(\xi_1)$ for the left and right faces of the standard element Ω_s, $\Psi_{BW}(\xi_2)$ and $\Psi_{FW}(\xi_2)$ the backward and forward faces, and as $\Psi_B(\xi_2)$ and $\Phi_T(\xi_2)$ for bottom and top faces. These functions have to approximate zero (in some sense) and satisfy the following constraints:

$$\begin{aligned}
\Psi_L(-1) = 1,\ \ \Psi_L(1) = 0; \qquad & \Psi_R(-1) = 0,\ \ \Psi_R(1) = 1; \\
\Psi_{BW}(-1) = 1,\ \ \Psi_{BW}(1) = 0; \qquad & \Psi_{FW}(-1) = 0,\ \ \Psi_{FW}(1) = 1; \\
\Psi_B(-1) = 1,\ \ \Psi_B(1) = 0; \qquad & \Psi_T(-1) = 0,\ \ \Psi_T(1) = 1; \\
\Psi_L(\xi_1) = \Psi_R(-\xi_1);\ \ \Phi_{BW}(\xi_2) = \Psi_{FW}(-\xi_2);\ \ \Psi_B(\xi_3) = \Psi_T(-\xi_3).
\end{aligned} \tag{1.98}$$

The last constraint follows by symmetry consideration. As already mentioned,

the FR approach allows the recovery of a wide-range of numerical schemes. From this point of view, Vincent et al. (2011a) have identified a class of linearly energy stable FR schemes called VCJH schemes. In particular, these VCJH schemes are recovered using the following expressions for the correction functions:

$$\Psi_L = \Psi_B = \Psi_{BW} = \frac{(-1)^{\mathrm{P}}}{2}\left[L_{\mathrm{P}} - \left(\frac{\eta_{\mathrm{P}} L_{\mathrm{P}-1} + L_{\mathrm{P}+1}}{1+\eta_{\mathrm{P}}}\right)\right], \tag{1.99a}$$

$$\Psi_R = \Psi_T = \Psi_{FW} = \frac{1}{2}\left[L_{\mathrm{P}} + \left(\frac{\eta_{\mathrm{P}} L_{\mathrm{P}-1} + L_{\mathrm{P}+1}}{1+\eta_{\mathrm{P}}}\right)\right], \tag{1.99b}$$

where L_P is a Legendre polynomial of degree P,

$$\eta_P = \frac{c\,(2\mathrm{P}+1)\,(a_{\mathrm{P}}\mathrm{P}!)}{2}, \qquad a_{\mathrm{P}} = \frac{(2\mathrm{P})!}{2^P(\mathrm{P}!)^2} \tag{1.100}$$

and c is a free scalar parameter which must be comprised within the range:

$$\frac{-2}{(2\mathrm{P}+1)(a_{\mathrm{P}}\mathrm{P}!)^2} < c < \infty. \tag{1.101}$$

By varying the scalar parameter c an infinite range of FR schemes can be recovered. In Nektar++ the numerical schemes recovered by three different values of the parameter c have been implemented. These are listed below.

Nodal discontinuous Galerkin scheme ($\mathrm{FR_{DG}}$)

If $c = c_{\mathrm{DG}} = 0$, then $\eta_{\mathrm{P}} = 0$ and the correction functions are the left and right Radau polynomials:

$$\Psi_L = \Psi_B = \Psi_{BW} = \frac{(-1)^{\mathrm{P}}}{2}\left(L_{\mathrm{P}} - L_{\mathrm{P}+1}\right),$$

$$\Psi_R = \Psi_T = \Psi_{FW} = \frac{1}{2}\left(L_{\mathrm{P}} + L_{\mathrm{P}+1}\right).$$

In this case a particular nodal discontinuous Galerkin scheme is recovered.

Spectral difference scheme ($\mathrm{FR_{SD}}$)

In order to recover a spectral difference scheme it is necessary that the correction functions have symmetrical zeros with respect to $\xi = 0$ of a standard element. To satisfy this requirement the parameter c has to be equal to:

$$c_{\mathrm{SD}} = \frac{2\mathrm{P}}{(2\mathrm{P}+1)(\mathrm{P}+1)(a_{\mathrm{P}}\mathrm{P}!)^2}, \;\; \textit{then} \;\; \eta_{\mathrm{P}} = \frac{\mathrm{P}}{\mathrm{P}+1}. \tag{1.102}$$

The correction functions then become:

$$\Psi_L = \Psi_B = \Psi_{BW} = \frac{(-1)^{\mathrm{P}}}{2}\left[L_{\mathrm{P}} - \left(\frac{\mathrm{P}L_{\mathrm{P}-1}L_{\mathrm{P}} + (\mathrm{P}+1)L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right],$$
$$\Psi_R = \Psi_T = \Psi_{FW} = \frac{1}{2}\left[L_{\mathrm{P}} + \left(\frac{\mathrm{P}L_{\mathrm{P}-1}L_{\mathrm{P}} + (\mathrm{P}+1)L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right].$$

Huynh or g_2 scheme ($\mathrm{FR_{g2}}$)

A third important scheme recovered from the FR approach is the Huyhn or g_2 scheme presented for the first time by Huynh (2007). This particular scheme is recovered using c equal to:

$$c_{\mathrm{SD}} = \frac{2\mathrm{P}+1}{(2\mathrm{P}+1)\mathrm{P}(a_{\mathrm{P}}\mathrm{P}!)^2}, \;\; \textit{then} \;\; \eta_{\mathrm{P}} = \frac{\mathrm{P}+1}{\mathrm{P}}. \tag{1.103}$$

The correction functions then become:

$$\Psi_L = \Psi_B = \Psi_{BW} = \frac{(-1)^{\mathrm{P}}}{2}\left[L_{\mathrm{P}} - \left(\frac{(\mathrm{P}+1)L_{\mathrm{P}-1}L_{\mathrm{P}} + \mathrm{P}L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right],$$
$$\Psi_R = \Psi_T = \Psi_{FW} = \frac{1}{2}\left[L_{\mathrm{P}} + \left(\frac{(\mathrm{P}+1)L_{\mathrm{P}-1}L_{\mathrm{P}} + \mathrm{P}L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right].$$

After having defined the correction functions Ψ_L, Ψ_R, Ψ_{BW}, Ψ_{FW}, Ψ_B and Ψ_T it is now possible to calculate the correction flux $\widehat{\boldsymbol{f}}_n^{\delta C}$. In particular, by evaluating the transformed flux jumps defined as the difference between the numerical flux and the discontinuous flux evaluated at the interfaces between two adjacent elements, it is possible to calculate the correction fluxes as:

$$\begin{aligned}
\widehat{f_{1,n}^{\delta C}} &= \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_L - \widehat{f}_{1,L}^{\delta D}\right]}_{\widehat{\Delta f_{1,L}}}\Psi_L, + \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_R - \widehat{f}_{1,R}^{\delta D}\right]}_{\widehat{\Delta f_{1,R}}}\Psi_R \\
\widehat{f_{2,n}^{\delta C}} &= \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_{BW} - \widehat{f}_{2,BW}^{\delta D}\right]}_{\widehat{\Delta f_{2,BW}}}\Psi_{BW}, + \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_{FW} - \widehat{f}_{2,FW}^{\delta D}\right]}_{\widehat{\Delta f_{2,FW}}}\Psi_{FW} \\
\widehat{f_{3,n}^{\delta C}} &= \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_B - \widehat{f}_{3,B}^{\delta D}\right]}_{\widehat{\Delta f_{3,B}}}\Psi_B + \underbrace{\left[\left(\widehat{\boldsymbol{f}_n^{\delta I}\cdot\mathbf{n}}\right)_T - \widehat{f}_{1,T}^{\delta D}\right]}_{\widehat{\Delta f_{3,T}}}\Psi_T
\end{aligned}, \tag{1.104}$$

where $\mathbf{n}$ is the outward normal with respect to the faces of the standard element and $f_{1,L}^{\delta D}$, $f_{1,R}^{\delta D}$, $f_{2,BW}^{\delta D}$, $f_{2,FW}^{\delta D}$, $f_{3,B}^{\delta D}$ and $f_{3,T}^{\delta D}$ are the discontinuous fluxes evaluated

at the faces of the standard element. All the flux jumps have been transformed into the reference space by using the formula introduced in Eq. 1.84 for the DG method.

The fifth step consists in calculating the divergence of the discontinuous flux:

$$\nabla_{\boldsymbol{\xi}} \cdot \widehat{\boldsymbol{f}}_n^{\delta D} = \sum_{i,j,k=0}^{P} \widehat{f}_{1_{ijk}}^{\delta D} \frac{\partial \ell_i(\xi_1)}{\partial \xi_1} \ell_j(\xi_2)\ell_k(\xi_3) + \sum_{i,j,k=0}^{P} \widehat{f}_{1_{ijk}}^{\delta D} \ell_i(\xi_1) \frac{\partial \ell_j(\xi_2)}{\partial \xi_2} \ell_k(\xi_3) + \sum_{i,j,k=0}^{P} \widehat{f}_{1_{ijk}}^{\delta D} \ell_i(\xi_1)\ell_j(\xi_2) \frac{\partial \ell_k(\xi_3)}{\partial \xi_3} \tag{1.105}$$

and of the correction flux:

$$\nabla_{\boldsymbol{\xi}} \cdot \widehat{\boldsymbol{f}}_n^{\delta C} = \widehat{(\Delta f_{1,L})} \frac{\partial \Psi_L}{\partial \xi_1} + \widehat{(\Delta f_{1,R})} \frac{\partial \Psi_R}{\partial \xi_1} + \widehat{(\Delta f_{2,BW})} \frac{\partial \Psi_{BW}}{\partial \xi_2} + \widehat{(\Delta f_{2,FW})} \frac{\partial \Psi_{FW}}{\partial \xi_2} + \widehat{(\Delta f_{3,B})} \frac{\partial \Psi_B}{\partial \xi_3} + \widehat{(\Delta f_{3,T})} \frac{\partial \Psi_T}{\partial \xi_3}. \tag{1.106}$$

Finally, we can write the FR approach in matrix form on the (P + 1) × (P + 1) × (P + 1) solution points as follows:

$$\frac{d\widehat{\mathbf{u}}}{\mathrm{d}t} + \left[\sum_{i=1}^{i=3} \boldsymbol{\Lambda}(G_{i/1}) \mathbf{D}_{\xi_i} \right] \widehat{\mathbf{f}}_1^D + \left[\sum_{i=1}^{i=3} \boldsymbol{\Lambda}(G_{i/2}) \mathbf{D}_{\xi_i} \right] \widehat{\mathbf{f}}_2^D + \left[\sum_{i=1}^{i=3} \boldsymbol{\Lambda}(G_{i/3}) \mathbf{D}_{\xi_i} \right] \widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{FR}} = 0 \tag{1.107}$$

where $\widehat{\mathbf{b}}^{\mathrm{FR}}$ is the boundary term vector also defined in Eq. 1.106. The matrix form of the FR approach can be further reduced substituting Eq. 1.86 into Eq. 1.107 and obtaining:

$$\frac{d\widehat{\mathbf{u}}}{dt} + \mathbf{D}_{x_1} \widehat{\mathbf{f}}_1^D + \mathbf{D}_{x_2} \widehat{\mathbf{f}}_2^D + \mathbf{D}_{x_3} \widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{FR}} = 0. \tag{1.108}$$

Eq. 1.108 can be advanced in time via an explicit time integration scheme (e.g. explicit Runge-Kutta methods).

1.5 DG and FR schemes: diffusion operator

After having introduced the DG and FR schemes for discretising the advection operator, we now present the discretisation of the diffusion operator for second order problems. For this type of problems the implementation of the DG method in

the spectral/hp element library Nektar++ has been carried out for one-, two- and three-dimensional tensor product elements - specifically quadrilaterals and hexahedra, whilst the FR approach for one- and two-dimensional cases. For describing the two approaches we consider the following two-dimensional second order problem:

$$\frac{\partial u}{\partial t} - \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(u, \nabla_{\boldsymbol{x}} u) = 0, \tag{1.109}$$

within a domain $\Omega \in \mathbb{R}^3$, with $\boldsymbol{f} = [f_1, f_2, f_3]$, where $f_1 = f_1(u, \nabla_{\boldsymbol{x}} u)$, $f_2 = f_2(u, \nabla_{\boldsymbol{x}} u)$ and $f_3 = f_3(u, \nabla_{\boldsymbol{x}} u)$ are the diffusive fluxes in the x_1, x_2 and x_3 Cartesian directions, respectively and where t is the time, $u = u(x_1, x_2, x_3, t)$ is the conserved variable and $\nabla_{\boldsymbol{x}} = [\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3]$. We can appreciate how the fluxes in this case depend also on the first order derivatives of the independent variables.
To construct both the DG and FR diffusion operators, it is convenient to rewrite the second order problem in Eq.1.109 as a system of first order equations

$$\frac{\partial u}{\partial t} - \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(u, \boldsymbol{q}) = 0, \tag{1.110a}$$

$$\boldsymbol{q} - \nabla_{\boldsymbol{x}} u = 0, \tag{1.110b}$$

within Ω where $\boldsymbol{q}$ are the auxiliary variables. Henceforth, we will refer to Eq. 1.110a as the principal equation and to Eq. 1.110b as the auxiliary equation.
Similar to the advection operators, we partition the domain into N non-overlapping subdomains Ω_n, on which, we will define the diffusion operators. In the following, we first describe the discontinuous Galerkin method applied to Eq. 1.110 and then we focus on the fux reconstruction approach. All the steps described in the following two sections are very general and they can be applied to any elliptic or second order problem, such as the diffusive flux of the advection-diffusion equation or the viscous part of the compressible Navier-Stokes equation.

1.5.1 Discontinuous Galerkin method

To describe the DG diffusion operator we first define the spaces of the test functions ℓ and $\boldsymbol{\ell}$ associated to the system in Eq1.110:

$$\mathcal{V}^{\delta} = \left\{ \ell \in \mathrm{L}^2(\Omega) : \ell|_{\Omega_n} \in \mathbb{P}_{\mathrm{P}}(\Omega_n), \ \forall \Omega_n \right\}, \tag{1.111}$$

$$\mathcal{W}^{\delta} = \left\{ \boldsymbol{\ell} \in \left[\mathrm{L}^2(\Omega)\right]^2 : \boldsymbol{\ell}|_{\Omega_n} \in \left[\mathbb{P}_{\mathrm{P}}(\Omega_n)\right]^2, \ \forall \Omega_n \right\}, \tag{1.112}$$

where $\mathbb{P}_{\mathrm{P}}$ is the space of the three-dimensional piecewise continuous polynomials of order $P \geq 1$ on Ω and L^2 has been defined in Eq. 1.78. Note that the space in Eq. 1.111 is associated with the principal equation, Eq. 1.110a, while the space in

Eq. 1.112 is associated with the auxiliary equation, Eq. 1.110b.

We now approximate the solution u by a piecewise three-dimensional polynomial $u^\delta \in \mathcal{V}^\delta$. We then multiply Eq. 1.110a by the test function ℓ and the auxiliary equation, Eq. 1.110b, by the test function $\boldsymbol{\ell}$ and integrate over a local element Ω_n such that:

$$\iiint_{\Omega_n} \ell(\boldsymbol{x}) \frac{\partial u_n^{\delta D}(\boldsymbol{x})}{\partial t} d\boldsymbol{x} - \iiint_{\Omega_n} \ell(\boldsymbol{x}) \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}_n^{\delta}\right) d\boldsymbol{x} = 0, \tag{1.113a}$$

$$\iiint_{\Omega_n} \boldsymbol{\ell}(\boldsymbol{x}) \cdot \boldsymbol{q}_n^{\delta D}(\boldsymbol{x}) d\boldsymbol{x} - \iiint_{\Omega_n} \boldsymbol{\ell}(\boldsymbol{x}) \cdot \nabla_{\boldsymbol{x}} u_n^{\delta}(\boldsymbol{x}) d\boldsymbol{x} = 0, \tag{1.113b}$$

where we used the superscript "D" to denote discontinuous variables across adjacent elements. We successively integrate by parts the second term of both the equations:

$$\begin{aligned} &\iiint_{\Omega_n} \ell(\boldsymbol{x}) \frac{\partial u_n^{\delta D}(\boldsymbol{x})}{\partial t} d\boldsymbol{x} + \\ &\underbrace{\iiint_{\Omega_n} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}) \cdot \boldsymbol{f}_n^{\delta D} d\boldsymbol{x}}_{\text{volumetric term}} - \underbrace{\iint_{\partial\Omega_n} \ell(\boldsymbol{x}) \left(\boldsymbol{f}_n^{\delta I} \cdot \mathbf{n}\right) ds}_{\text{boundary term}} = 0, \end{aligned} \tag{1.114a}$$

$$\begin{aligned} &\iiint_{\Omega_n} \boldsymbol{\ell}(\boldsymbol{x}) \cdot \boldsymbol{q}_n^{\delta D}(\boldsymbol{x}) d\boldsymbol{x} + \\ &\underbrace{\iiint_{\Omega_n} \Big(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{\ell}(\boldsymbol{x})\Big) u_n^{\delta D} d\boldsymbol{x}}_{\text{volumetric term}} - \underbrace{\iint_{\partial\Omega_n} \Big(\boldsymbol{\ell}(\boldsymbol{x}) \cdot \mathbf{n}\Big) u_n^{\delta I} ds}_{\text{boundary term}} = 0, \end{aligned} \tag{1.114b}$$

and we obtain the weak form of the DG method for the second order problem of Eq. 1.109. In Eq. 1.114 $u_n^{\delta D}$ denotes the discretised discontinuous solution, $\boldsymbol{f}_n^{\delta D} = \boldsymbol{f}_n^{\delta D}(u^{\delta D}, \boldsymbol{q}_n^{\delta D})$ is the diffusive flux calculated from the discontinuous solution and its gradient, $\boldsymbol{q}_n^{\delta D}$ is the vector of the auxiliary variables and $\boldsymbol{f}_n^{\delta I}$, $u_n^{\delta I}$ are the numerical fluxes associated to the principal and the auxiliary equation, respectively, with $\mathbf{n}$ being the outward pointing normal with respect to a given element face.

In order to solve Eq. 1.114 we start by solving the auxiliary equation. Specifically, we need to find the numerical flux $u_n^{\delta I}$ via an appropriate methodology. There are different techniques (with different solutions) for calculating the numerical flux in the context of a second order problem. These include the *Bassi-Rebay* (BR) flux (Bassi and Rebay (1997)), the *Interior Penalty* (IP) flux (Arnold (1982)), the compact DG (CDG) method (Peraire and Persson (2008)) and the *Local Discontinuous Galerkin* (LDG) flux (Cockburn and Shu (1998a)). In Nektar++ we have implemented the latter, that reads:

$$u_n^{\delta I} = \left\{u_n^{\delta D}\right\} \pm \beta \left[u_n^{\delta D}\right], \tag{1.115}$$

where $\{\cdot\}$ and $[\cdot]$ are the average and the jump operator, respectively and they are defined as follows:

$$\{g\} = \frac{g_+ + g_-}{2}, \qquad [g] = \frac{g_+\mathbf{n}_+ + g_-\mathbf{n}_-}{2}, \tag{1.116}$$

where β was chosen such that it produces a compact stencil[3], that is $\beta = (1/2)\mathbf{n}_+$. The quantities g_- and g_+ denote the variable g on either the sides of the interface between two adjacent elements. $\mathbf{n}_-$ is the outward pointing unit normal from the element on the right side of the interface and $\mathbf{n}_+$ is the outward pointing unit normal from the element on the left side of the interface.

After having solved the auxiliary equation and, therefore, having calculated the derivatives needed for computing the flux $\widehat{\boldsymbol{f}}_n^{\delta D}$, the principal equation, Eq. 1.114a, reduces to a simple first order problem and can be solved using the same procedure used for a first order problem, except for the term containing the numerical flux $\widehat{\boldsymbol{f}}_n^{\delta I}$. To preserve stability and accuracy this numerical flux needs to be computed using the same formulation applied before. In our case we have:

$$\boldsymbol{f}_n^{\delta I} = \left\{\boldsymbol{f}_n^{\delta D}\right\} + \mp\beta \cdot \left[\boldsymbol{f}_n^{\delta D}\right] - \tau\left[u_n^{\delta D}\right], \tag{1.117}$$

where we can add a penalty term, $\tau\left[u_n^{\delta D}\right]$ to control the jump in the solution and we note that the sign prior to the flux jump operator has changed. The factors β and τ ensure that the numerical method converges to the optimal order of accuracy.

1.5.2 Flux Reconstruction approach

The approximation space of FR is identical to the DG expressed in Eq. 1.111 and Eq. 1.112. The procedure to solve Eq. 1.110 by means of the FR approach can be explained within the standard element as done for the FR advection operator. Within the standard domain Ω_s, Eq. 1.110 becomes the following transformed system of equations:

$$\frac{\partial \widehat{u}_n^{\delta D}}{\partial t} - \nabla_{\boldsymbol{\xi}} \cdot \widehat{\boldsymbol{f}}_n^{\delta}(\widehat{u}, \widehat{\boldsymbol{q}}) = 0, \tag{1.118a}$$

$$\widehat{\boldsymbol{q}}_n^{\delta D} - \nabla_{\boldsymbol{\xi}} \widehat{u}_n^{\delta} = 0, \tag{1.118b}$$

where $\widehat{u}_n^{\delta D}$ and $\widehat{\boldsymbol{f}}_n^{\delta}$ have been defined in Eq. 1.94 and Eq. 1.95, respectively and reported here for sake of clarity:

$$\widehat{u}_n^{\delta D}(\boldsymbol{\xi}, t) = J_n u_n^{\delta D}(\boldsymbol{\Theta}_n(\boldsymbol{\xi}, t)) \qquad \widehat{\boldsymbol{f}}_n^{\delta}(\boldsymbol{\xi}, t) = J_n \mathbf{G}_n^{-1} \boldsymbol{f}_n^{\delta},$$

whilst the auxiliary variable $\widehat{\boldsymbol{q}}^{\delta D}$ is defined as:

$$\widehat{\boldsymbol{q}}_n^{\delta D}(\boldsymbol{\xi}, t) = \nabla_{\boldsymbol{\xi}} \widehat{u}^{\delta} = J_n \mathbf{G}_n^T \boldsymbol{q}_n^{\delta D}. \tag{1.119}$$

[3]Note that we could have used different values of β which would have produced wider stencils.

In the FR diffusion operator implementation we need to project the solution onto a set of (P+1)^3 points within Ω_s:

$$\widehat{u}^{\delta D}(\boldsymbol{\xi},t)=\sum_{i,j,k=0}^{P}\ell_{ijk}(\boldsymbol{\xi})\widehat{u}_{ijk}^{\delta D}=\sum_{i,j,k=0}^{P}\ell_i(\xi_1)\ell_j(\xi_2)\ell_k(\xi_3)\widehat{u}_{ijk}^{\delta D}. \tag{1.120}$$

After the projection we first need to solve the auxiliary equation in order to compute the auxiliary variable vector $\widehat{\boldsymbol{q}}_n^{\delta D}$ required for calculating the diffusive flux $\widehat{\boldsymbol{f}}_n^{\delta D}(\widehat{u}_n^{\delta D},\widehat{\boldsymbol{q}}_n^{\delta D})$. The procedure is analogous to that adopted for the advection term. In particular we calculate the numerical flux for the auxiliary equation $u_n^{\delta I}$ in order to propagate the numerical solution from element to element by using Eq. 1.115. After having transformed it into the standard space we calculate the auxiliary correction flux $\widehat{u}_n^{\delta C}$ which is a polynomial of order P+1. Then we add this to the to the discontinuous solution $\widehat{u}_n^{\delta D}$, in order to obtain the following total auxiliary flux:

$$\widehat{u}_n^{\delta}=\widehat{u}_n^{\delta D}+\widehat{u}_n^{\delta C}. \tag{1.121}$$

The auxiliary correction flux is defined as follows:

$$\begin{aligned}\widehat{u}_n^{\delta C}=&\left(\widehat{u}_L^{\delta I}-\widehat{u}_L^{\delta D}\right)\Psi_L+\left(\widehat{u}_R^{\delta I}-\widehat{u}_R^{\delta D}\right)\Psi_R+\left(\widehat{u}_{BW}^{\delta I}-\widehat{u}_{BW}^{\delta D}\right)\Psi_{BW}+\\&\left(\widehat{u}_{FW}^{\delta I}-\widehat{u}_{FW}^{\delta D}\right)\Psi_{FW}+\left(\widehat{u}_B^{\delta I}-\widehat{u}_B^{\delta D}\right)\Psi_B+\left(\widehat{u}_T^{\delta I}-\widehat{u}_T^{\delta D}\right)\Psi_T,\end{aligned} \tag{1.122}$$

where $\widehat{u}_L^{\delta I}$, $\widehat{u}_R^{\delta I}$, $\widehat{u}_{BW}^{\delta I}$, $\widehat{u}_{FW}^{\delta I}$, $\widehat{u}_B^{\delta I}$ and $\widehat{u}_T^{\delta I}$ are the auxiliary numerical fluxes at the left, right, backward, forward, bottom and top faces of the standard element, $\widehat{u}_L^{\delta D}$, $\widehat{u}_R^{\delta D}$, $\widehat{u}_{BW}^{\delta D}$, $\widehat{u}_{FW}^{\delta D}$, $\widehat{u}_B^{\delta D}$ and $\widehat{u}_T^{\delta D}$ are the associated values of the transformed solution $\widehat{u}^{\delta D}$ at the same faces and where we used the correction functions defined in Eq. 1.99. After having calculated the total auxiliary flux, we can calculate the auxiliary variables $\widehat{\boldsymbol{q}}_n^{\delta D}$ by differentiating Eq. 1.121 within Ω_s as follows:

$$\widehat{\boldsymbol{q}}_n^{\delta D}=\begin{bmatrix}\displaystyle\sum_{i,j,k=0}^{P}\widehat{u}_{ijk}^{\delta D}\frac{\partial\ell_i(\xi_1)}{\partial\xi_1}\ell_j(\xi_2)\ell_k(\xi_3)+\left(\widehat{u}_L^{\delta I}-\widehat{u}_L^{\delta D}\right)\frac{\partial\Psi_L}{\partial\xi_1}+\left(\widehat{u}_R^{\delta I}-\widehat{u}_R^{\delta D}\right)\frac{\partial\Psi_R}{\partial\xi_1}\\ \displaystyle\sum_{i,j,k=0}^{P}\widehat{u}_{ijk}^{\delta D}\ell_i(\xi_1)\frac{\partial\ell_j(\xi_2)}{\partial\xi_2}\ell_k(\xi_3)+\left(\widehat{u}_{BW}^{\delta I}-\widehat{u}_{BW}^{\delta D}\right)\frac{\partial\Psi_{BW}}{\partial\xi_2}+\left(\widehat{u}_{FW}^{\delta I}-\widehat{u}_{FW}^{\delta D}\right)\frac{\partial\Psi_{FW}}{\partial\xi_2}\\ \displaystyle\sum_{i,j,k=0}^{P}\widehat{u}_{ijk}^{\delta D}\ell_i(\xi_1)\ell_j(\xi_2)\frac{\partial\ell_k(\xi_3)}{\partial\xi_3}+\left(\widehat{u}_B^{\delta I}-\widehat{u}_B^{\delta D}\right)\frac{\partial\Psi_B}{\partial\xi_3}+\left(\widehat{u}_T^{\delta I}-\widehat{u}_T^{\delta D}\right)\frac{\partial\Psi_T}{\partial\xi_3}\end{bmatrix}$$

This concludes the work on the auxiliary equation and permits the construction of the flux $\widehat{\boldsymbol{f}}_n^{\delta}$. Therefore, we can now solve the principal equation, Eq. 1.118a, which consists in performing the same steps as for the FR advection operator. Specifically, we need to define the discontinuous fluxes $\widehat{f}_{1,n}^{\delta D}$, $\widehat{f}_{2,n}^{\delta D}$ and $\widehat{f}_{3,n}^{\delta D}$ as in Eq. 1.97. We then calculate the correction fluxes $\widehat{f}_{1,n}^{\delta C}$, $\widehat{f}_{2,n}^{\delta C}$ and $\widehat{f}_{3,n}^{\delta C}$ as in Eq. 1.104.

The only difference at this stage is the calculation of the numerical flux $\boldsymbol{f}_n^{\delta I}$ needed for calculating the correction fluxes. In order to have a stable numerical scheme, they must be calculated using the same approach used for the numerical flux of the auxiliary equation. In our case, this means using Eq. 1.117. We finally calculate the divergence of the resulting total flux by using Eq. 1.105 and Eq. 1.106, and we assemble the overall right-hand side.

1.6 Code verification

The implementation of the FR approach and of the DG method in Nektar++ have been thoroughly tested in this thesis work. Additional verification results can be found in Mengaldo (2012) and De Grazia (2013). In this section we present the results obtained on a two-dimensional unsteady linear advection equation and on a two-dimensional unsteady diffusion equation, in order to verify both the advection and diffusion operators.

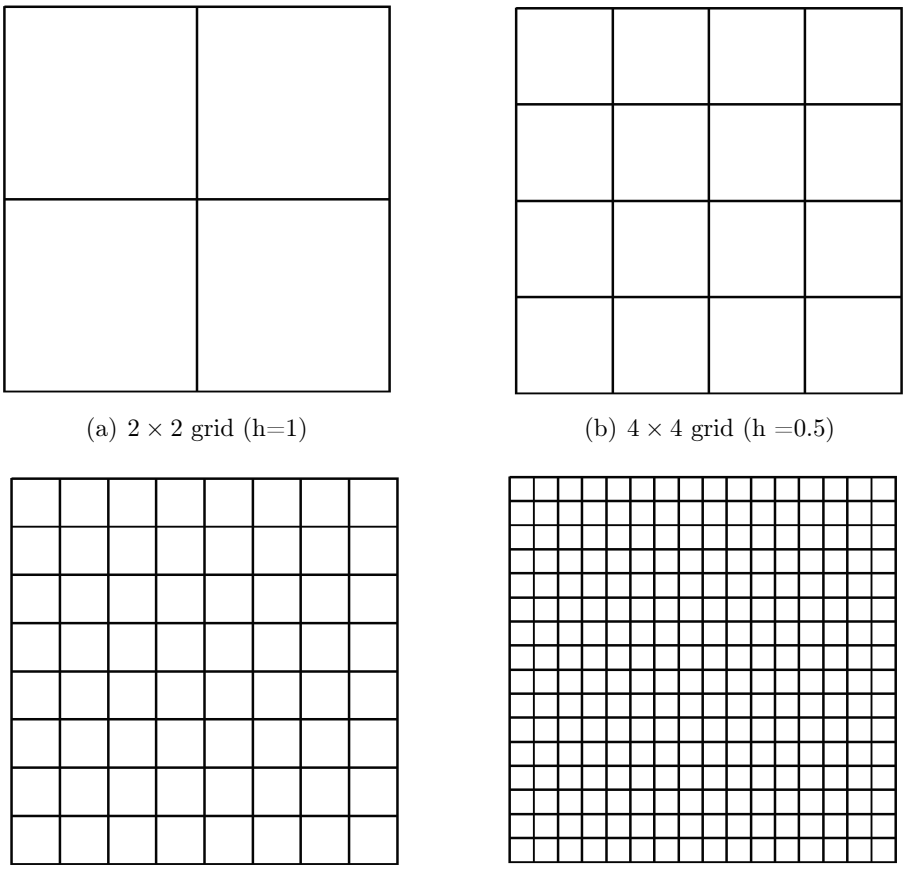
For all the simulations the time-integration scheme used was a fourth order Runge-Kutta scheme and the time-step was chosen such that the time-integration error was negligible. Specifically the time-step chosen was $\Delta t_{\max}/100$, where $\Delta t_{\max}$ is the maximum time-step that allowed the simulation to remain stable.
All the simulations performed aimed to check only the spatial accuracy of the different schemes.

To verify the implementation of the FR approach, we considered the three different schemes presented in section 1.4.2, namely the $\mathrm{FR_{DG}}$, the $\mathrm{FR_{SD}}$, the $\mathrm{FR_{g2}}$ schemes. All the FR scheme considered employed a collocation projection of the solution on a set of Gauss-Lobatto-Legendre points. Concerning the DG method we used the same solution points applying the same type of projection. We consider a nodal DG version testing both the cases with an exact (EMM) and a lumped (LMM) mass matrix. The second derives from the choice of the solution points and the Kronecker delta property of the Lagrange polynomials approximating the solution.

Throughout the following, we will make extensive use of the $\mathrm{L_2}$ error defined as follows:

$$\mathrm{E_{L_2}} = \sqrt{\sum_{i,j=0}^{Q-1} (u_{ij} - u_{ij,exact})^2 w_{ij}}, \tag{1.123}$$

where u_{ij} is the numerical solution calculated at the ij-th quadrature point, $u_{ij,exact}$ is the associated exact solution, Q is the number of quadrature points in each



(a) 2×2 grid (h=1)

(b) 4×4 grid (h =0.5)

(c) 8×8 grid (h =0.25)

(d) 16×16 grid (h =0.125)

Figure 1.9: Unsteady linear advection and unsteady diffusion cases: regular meshes.

direction and w_{ij} are the associated weights:

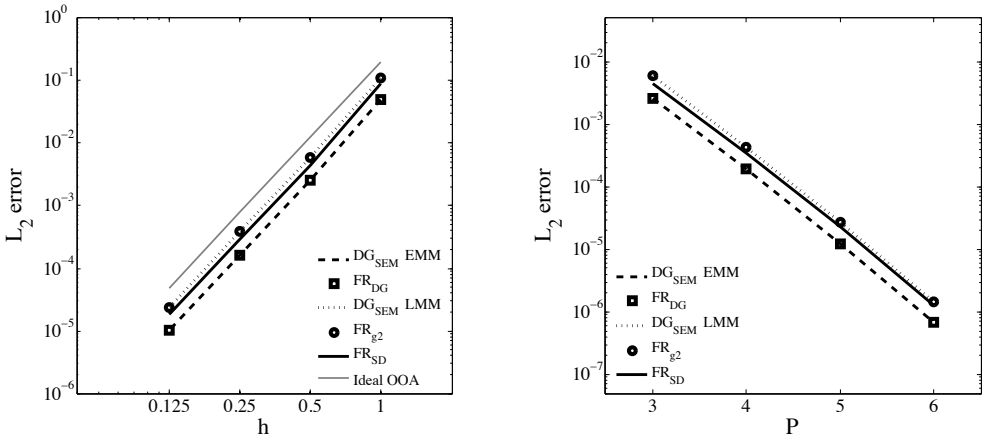
$$w_{ij} = \int_{-1}^{1} \ell_i(\xi_1) d\xi_1 \int_{-1}^{1} \ell_j(\xi_2) d\xi_2 \,. \tag{1.124}$$

We first approximate the solution refining an initial regular grid of $N_x \times N_x = 4$ elements by increasing N_x such that $N_x = 2, 4, 8, 16$ representing the solution with Lagrange polynomial of order P=3. The meshes employed are depicted in Figure 1.9. We then used the grid 1.9(b) refining the solution by increasing the polynomial order P such that P=3,4,5,6.

The test case of the linear advection equation is defined as follows:

$$\begin{cases} \dfrac{\partial u}{\partial t} + a_x \dfrac{\partial u}{\partial x} + a_y \dfrac{\partial u}{\partial y} = 0 \\ u(x, y, t = 0) = \sin(\pi x)\cos(\pi y) \\ u(x_b, y_b, t) = \sin(\pi(x - a_x t))\cos(\pi(y - a_y t)), \end{cases} \tag{1.125}$$

where u is the independent variable, a_x and a_y are the advection velocities along x and y and (x_b, y_b) denote the boundaries of the numerical domain where we used the exact solution of the problem $u_{ex}(x, y, t) = \sin(\pi(x - a_x t))\cos(\pi(y - a_y t))$ as the boundary condition. We used $a_x = 1, a_y = 0$ and a final time of $T = 1s$ in order for the initial condition to travel one advective length.



(a) h-type convergence

(b) p-type convergence

Figure 1.10: Order of accuracy (OOA) and P convergence for the linear advection equation.

In Figure 1.10(a), we show the L_2 error vs. the element size "h" for the four regular meshes depicted in Figure 1.9. The curves are obtained in a logarithmic scale and it is possible to observe an $\mathcal{O}(h^{P+1})$ convergence for both the FR schemes and the DG method.

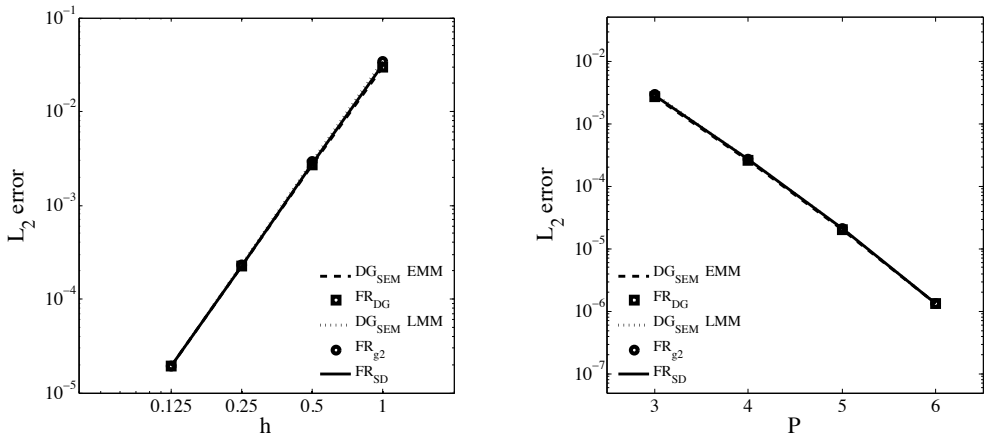
In Figure 1.10(b), we show the L_2 error vs. the polynomial order. The curves are obtained in a semi-logarithmic scale and it is possible to see an exponential convergence of the error for both the FR schemes and the DG method. We then did the same test on the diffusion operator.

The test case of the diffusion equation is defined as follows

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} - \dfrac{\partial^2 u}{\partial y^2} = 0 \\ u(x, y, t = 0) = \sin(\pi x)\cos(\pi y) \\ u(x_b, y_b, t) = e^{-2\pi^2 t}\sin(\pi x)\cos(\pi y), \end{cases} \tag{1.126}$$

where u is the independent variable and (x_b, y_b) denotes the boundaries of the numerical domain where we used the exact solution of the problem $u_{ex}(x, y, t) = e^{-2\pi^2 t}\sin(\pi x)\cos(\pi y)$ as the boundary condition. We used a final time of $T = 0.1s$.

In Figure 1.11(a), we show the L_2 error vs. the element size "h" for the four regular meshes depicted in Figure 1.9 with curves in logarithmic scale. In Figure 1.11(b), we show the L_2 error vs. the polynomial order with the curves in semi-logarithmic scale. In both the pictures the differences between the different approaches are very small compared to those observed in the advection case. However it is possible to appreciate good h-type (Figure 1.11(a)) and p-type convergence (Figure 1.11(b)) for all the schemes employed.



(a) h-type convergence

(b) p-type convergence

Figure 1.11: H and P convergence for the diffusion equation.

In Table 1.1 we report the values of the L_2 error, E_{L_2}, and the maximum time-step for which it is possible to achieve a stable simulation for the linear advection case tested on the mesh in Figure 1.9(b). From this table, we can see how the

P			$\mathbf{DG_{SEM}}$ -EMM	$\mathbf{DG_{SEM}}$ -LMM	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$	$\mathbf{FR_{SD}}$
P =3	$\mathbf{E_{L_2}}$	[-]	2.56×10^{-3}	6.04×10^{-3}	2.56×10^{-3}	6.04×10^{-3}	4.50×10^{-3}
	$\Delta \mathbf{t}_{max}$	[s]	1.046×10^{-2}	2.42×10^{-2}	1.046×10^{-2}	2.42×10^{-2}	1.83×10^{-2}
P =5	$\mathbf{E_{L_2}}$	[-]	1.93×10^{-4}	4.34×10^{-4}	1.93×10^{-4}	4.34×10^{-4}	3.47×10^{-4}
	$\Delta \mathbf{t}_{max}$	[s]	7.50×10^{-4}	1.72×10^{-3}	7.50×10^{-4}	1.72×10^{-3}	1.37×10^{-3}
P =4	$\mathbf{E_{L_2}}$	[-]	1.23×10^{-5}	2.71×10^{-5}	1.23×10^{-5}	2.71×10^{-5}	2.26×10^{-5}
	$\Delta \mathbf{t}_{max}$	[s]	4.98×10^{-5}	1.09×10^{-4}	4.98×10^{-5}	1.09×10^{-4}	9.10×10^{-5}
P =6	$\mathbf{E_{L_2}}$	[-]	6.80×10^{-7}	1.47×10^{-6}	6.80×10^{-7}	1.47×10^{-6}	1.26×10^{-6}
	$\Delta \mathbf{t}_{max}$	[s]	2.62×10^{-6}	5.80×10^{-6}	2.62×10^{-6}	5.80×10^{-6}	4.94×10^{-6}

Table 1.1: L_2 error and maximum Δt vs polynomial order for the different types of DG and FR schemes tested, on the mesh with element size h =0.5.

error associated to the FR_{DG} and the DG_{SEM}-EMM schemes is the lowest amongst the schemes considered for a given polynomial order, while the largest time-step limit is obtained for FR_{g2} and DG_{SEM}-LMM. The results are in agreement with

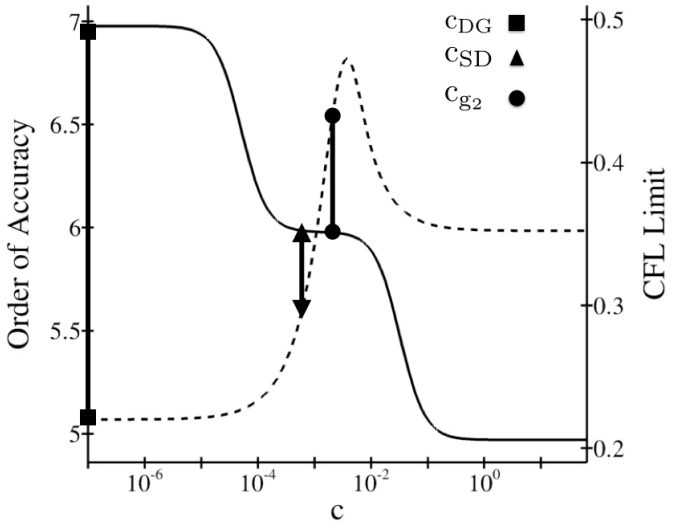


Figure 1.12: Order of accuracy (OOA) and CFL limit for the FR schemes obtained varying c, which is the free scalar parameter described in Eq. 1.101. Figure readapted from Vincent et al. (2011b) for one-dimensional case with P = 3.

the findings of Vincent et al. (2011b), where the authors showed how the FR_{DG} scheme is the most accurate but also the scheme with the most severe restriction in terms of time-step, while the FR_{SD} and FR_{g2} are slightly less accurate but have more favourable time-step restrictions. Interestingly, we can note that the DG_{SEM}-LMM scheme provides the same results of the FR_{g2} scheme in terms of both L_2 error and maximum time-step as shown in De Grazia et al. (2014). In this paper the author of this thesis and co-workers have demonstrated how, for regular quadrilateral meshes, the FR_{DG} and the FR_{g2} schemes are identical, for both linear and nonlinear problems, to the DG_{SEM} method with an exact and lumped mass matrix, respectively. These two identities are particularly important because they indicate that not only the FR approach but also the DG method allows recovering two schemes with different accuracy and time-restriction properties. To better explain this point, Figure 1.12 shows the CFL limit and the super-accuracy[4] of the three FR schemes considered for a one-dimensional case linear advection equation with P=3. We can see how the FR_{g2} scheme allows the largest time-step limit. The FR_{SD} has a similar time-step limit while the FR_{DG} scheme is the most restrictive. On the other hand the FR_{DG} scheme is the most accurate while the FR_{g2} scheme is the least accurate among the FR schemes considered. In the next chapter we will thoroughly investigate these connections between the DG method and FR schemes on quadrilateral tensor product meshes.

[4]The super-accuracy for the current implementation of the FR approach and of the DG method has also been tested for the one-dimensional linear advection equation in De Grazia (2013) providing the expected results.

Chapter 2

Connections between the DG method and the FR schemes

As already mentioned, finding the connections between the DG and FR approaches is crucial for understanding the advantages we can attain by using one or the other method and for exploiting the existing DG machinery for tackling, for instance, the aliasing issues arising in collocated schemes. Furthermore, it allows a better contextualisation of the FR approach within the framework of discontinuous spectral/hp element methods.

In this chapter we closely examine the connections between three nodal versions of tensor product discontinuous Galerkin spectral element approximations and two types of collocated flux reconstruction schemes for solving systems of conservation laws on quadrilateral meshes. The different types of discontinuous Galerkin approximations arise from the choice of the solution nodes of the Lagrange basis representing the solution and from the quadrature approximation used to integrate the mass matrix and the other terms of the discretisation. By considering both a linear and nonlinear advection equation on a regular grid, we examine the mathematical properties which connect these discretisations. These arguments are further confirmed by the results of an empirical numerical study. The work of this chapter has been published by the author of this thesis and the co-workers in De Grazia et al. (2014).

2.1 Introduction

In recent years there has been much interest in high-order discretisations as they offer the potential to obtain more accurate approximations with less computational cost compared to lower order methods. The most widely adopted high-order methods are based on the discontinuous Galerkin method, introduced by Reed and Hill (1973). The DG method is widely used in both industry and academia due to

	$\mathbf{DG_{SEM}}$-GLL Lumped mass matrix		$\mathbf{DG_{SEM}}$ Exact mass matrix		$\mathbf{DG_{Q>P}}$ Exact mass matrix	
Flux type	Linear	Nonlinear	Linear	Nonlinear	Linear	Nonlinear
$\mathbf{FR_{DG}}$	✗	✗	✓	✓	✓	✗
$\mathbf{FR_{g2}}$	✓	✓	✗	✗	✗	✗

Table 2.1: Connections between different types of DG and FR schemes derived in this work for constant Jacobian elements. ✓ indicates that the schemes are equivalent, whereas ✗ indicates differences between the schemes.

the computational efficiency of the scheme and attractive numerical properties, and has well-known formulations to cope with both hyperbolic and elliptic problems, as in Cockburn and Shu (2001) and Arnold et al. (2001). Classically the DG method starts with an expression of the underlying problem in a weak form, whereby the governing system is multiplied by a test function and integrated over the domain, which is itself partitioned into non-overlapping elements. A high-order discretisation therefore entails selecting both a basis of polynomials which represent the solution locally on each element and a set of quadrature points on which the inner products arising in the weak formulation can be calculated. Some of the most efficient and ubiquitous forms of the DG method are so-called nodal DG schemes, whereby Lagrange interpolants are combined with a set of nodal solution points on a given element as in Hesthaven and Warburton (2008).

More recently, another set of methods, which are based upon the system being expressed in a differential form, have offered an alternative route which avoids the need for quadrature rules. This makes these schemes easier and potentially more efficient to implement. The first of these was introduced by Kopriva and Kolias (1996) and was extended in 2006 to quadrilateral and triangular elements under the name of spectral-difference (SD) schemes by Liu et al. (2006). Most recently, a new scheme called the flux reconstruction (FR) method was presented in Huynh (2007) and has undergone significant development in recent years, with applications to the compressible Euler and Navier-Stokes equations (Castonguay et al. (2011a); Williams et al. (2011)), extension from a tensor product formulation to simplex elements (Castonguay et al. (2011b)), allowing for the use of the method in a wide variety of geometries.

One of the most appealing properties of the FR method is the ability to encapsulate other types of numerical discretisations. Recently a new range of energy stable FR schemes have been identified in Vincent et al. (2011a), referred to as Vincent-Castonguay-Jameson-Huynh (VCJH) schemes. Here, a single real-valued

parameter c dictates the scheme which is recovered, and enables one to recover not only differential-type schemes such as a particular SD method, but also integral-type schemes such as nodal DG schemes. However, whilst a general connection between FR and nodal DG schemes has been examined in other work (Huynh (2007); Vincent et al. (2011a)), there are many aspects of this equivalence that are yet to be investigated. A particularly important point that has so far remained widely unaddressed is that nodal DG schemes take on different numerical properties based on the choice of solution and quadrature points. We note that in a general nodal DG scheme, given an expansion on each element in terms of P Lagrange interpolants, we may perform quadrature on a separate set of Q quadrature points. If $P = Q$ and the distribution of solution and quadrature points is identical, then we recover the so-called discontinuous Galerkin spectral element method ($\mathrm{DG_{SEM}}$) which diagonalises the mass matrix, allowing for further computational optimisation as in Kopriva et al. (2000); Gassner and Kopriva (2010). However, exactly which of these DG schemes can be recovered by the FR approach has not yet been examined.

In this chapter we examine connections between various DG and FR schemes for the case of an advection equation. We consider a nodal DG scheme with $Q > P$ and two $\mathrm{DG_{SEM}}$ schemes, where the subscript SEM[1] indicates that a collocation quadrature rule is used for the inner product of the advection term of the conservation law equation.

In Huynh (2007) the author proposed that one can recover a nodal DG scheme with the FR approach by using Radau polynomials for the correction functions. We refer to this FR scheme as $\mathrm{FR_{DG}}$. In Huynh's work, there is an implicit assumption that sufficiently accurate quadrature is used to resolve any potential nonlinearity of the flux function used in the FR framework. Since a collocation projection of the solution and hence of the flux is applied, this is always true only in case of linear flux. Thus, only the case of linear advection was investigated and aliasing issues arising from a nonlinear flux, which can lead to differences between the $\mathrm{FR_{DG}}$ and DG schemes, were not considered.

The essential relationships that are derived in this chapter are summarized in table 2.1. Firstly we demonstrate that the $\mathrm{FR_{DG}}$ and DG schemes, on a regular grid, are also equivalent for a nonlinear advection equation provided that a collocation quadrature rule is used for the the inner product of the advection term of the DG discretisation. Additionally we show that the use of this rule leads to identical aliasing errors in both schemes. We perform a rigorous analysis for the case of 2D advection equation on a quadrilateral regular grid, which may be easily extended

[1]In Kopriva et al. (2000); Gassner and Kopriva (2010) the $\mathrm{DG_{SEM}}$ method implied a collocation quadrature rule for the inner product of all the terms of the discretisation and not only the advection term. This led to a diagonal mass matrix which, in the case of Gauss-Lobatto-Legendre points, is not an exact mass matrix but a lumped one.

to 3D regular grids. We state that this derivation is always valid on regular grids, whilst some differences might exist on deformed grids. In the rest of the chapter we will refer to this DG scheme as $\mathrm{DG}_{\mathrm{SEM}}$ with exact mass matrix (second column of Table 2.1).

We observe that if an over-integration is used for the advection term of the DG method the equivalence is valid only in the case of a linear flux function on a regular grid, since by over-integrating we are reducing the aliasing errors arising from nonlinearities. Hereafter we will refer to this DG scheme as $\mathrm{DG}_{\mathrm{Q>P}}$ with exact mass matrix (third column of Table 2.1).

Finally, for the case of a 1D conservation law on a regular grid (which again can be extended to 2D/3D regular tensor product grids), we show that a DG method with solution nodes collocated at Gauss-Lobatto-Legendre points recovers the FR g2 scheme introduced by Huynh (2007) when the mass matrix of the DG discretisation is lumped. The equivalence is valid for both the cases of a linear or nonlinear flux function provided that a collocation quadrature rule is used for the inner product of the advection term of the DG discretisation. In the following we refer to this DG sheme as $\mathrm{DG}_{\mathrm{SEM}}$-GLL with lumped mass matrix (first column of Table 2.1), and to the type of FR scheme recovered as $\mathrm{FR}_{\mathrm{g2}}$ scheme. For the same reasons cited above, when an over-integration of the advection term of the DG method is used the proof is valid only in the case of a linear flux function on a regular grid.

We validate these theoretical results by performing a series of experiments using both linear and nonlinear fluxes on regular and deformed grids using the spectral element framework *Nektar++*. Finally, we conclude with a brief study of the relative computational efficiencies of each scheme. Comparisons of the efficiency of differential-form based high-order methods were already performed in Liang et al. (2013a) and Liang et al. (2013b), where the CPU processing time of the CPR[2] (correction procedure via reconstruction) method was measured and compared with that of the SD method on quadrilateral meshes. In Yu and Wang (2013) numerical accuracy and efficiency of several high-order methods, both differential-form based (SD, CPR) and integral-form based (DG) were analysed and compared on linear and curved quadrilateral elements.

2.2 Theory

In this section, we first describe the DG and FR methods in the setting of a 1D scalar conservation law and based on the arguments of Huynh (2007), we prove the equality of $\mathrm{DG}_{\mathrm{SEM}}$ and $\mathrm{FR}_{\mathrm{DG}}$ schemes on a regular grid of quadrilateral elements.

[2]The CPR method combines the FR idea with the lifting collocation penalty (LCP) framework.

We further show that in one dimension the $\mathrm{DG_{SEM}}$-GLL scheme is mathematically identical to the $\mathrm{FR_{g2}}$ scheme.

2.2.1 Discontinuous Galerkin method for 1D scalar conservation law

Consider the 1D scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{2.1}$$

within a domain $\Omega \subset \mathbb{R}$, where $f = f(u)$ is the flux function. The aim of the DG method is to find an approximate weak solution to Eq. 2.1 when the domain Ω is partitioned into N non-overlapping elements Ω_n such that

$$\Omega = \bigcup_{n=1}^{N} \Omega_n.$$

From an implementation perspective, it is convenient to map each element Ω_n into a reference (standard) element $\Omega_s = \{\xi \mid -1 \leq \xi \leq 1\}$ through the map defined in Eq. 1.16 and here reported for sake of clarity:

$$x = \Theta_n(\xi) = \left(\frac{1-\xi}{2}\right) x_n + \left(\frac{1+\xi}{2}\right) x_{n+1}.$$

In the reference domain the transformed equation into Ω_s becomes

$$\frac{\partial \widehat{u}_n^\delta}{\partial t} + \frac{\partial \widehat{f}_n^\delta}{\partial \xi} = 0, \tag{2.2}$$

where

$$\widehat{u}_n^\delta = \widehat{u}_n^\delta(\xi, t) = u_n^\delta(\Theta_n^{-1}(\xi), t),$$
$$\widehat{f}_n^\delta = \widehat{f}_n^\delta(\xi, t) = \frac{f_n^\delta(\Theta_n^{-1}(\xi), t)}{J_n},$$

with $J_n = (x_{n+1} - x_n)/2$. The transformed solution $\widehat{u}_n^\delta$ in the reference space is approximated by a polynomial of degree P and can be written as

$$\widehat{u}_n^\delta = \sum_{i=0}^{P} \widehat{u}_i^\delta \ell_i(\xi), \tag{2.3}$$

where ℓ_i denote the Lagrange polynomials. Analogously the approximate transformed flux $\widehat{f}_n^\delta$ in the reference space can be written as

$$\widehat{f}_n^{\delta D} = \sum_{i=0}^{P} \widehat{f}_i^{\delta D} \ell_i(\xi), \tag{2.4}$$

where, as mentioned in section 1.4, the superscript D stands for discontinuous since the flux is calculated directly from the approximate solution, which is in general piecewise discontinuous between elements.

Now, we write the residual of Eq. 2.2

$$R^{\delta} = \frac{\partial \widehat{u}_n^{\delta}}{\partial t} + \frac{\partial \widehat{f}_n^{\delta}}{\partial \xi}$$

which we require to be orthogonal to a set of smooth test functions. Following the standard Galerkin approach we choose test functions defined by the basis $\ell_i(\xi)$ leading to the condition that

$$\int_{\Omega_s} R^{\delta} \cdot \ell_i \, d\xi \quad \forall \, i.$$

After performing integration by parts, substituting the boundary terms by the numerical interface fluxes and integrating by parts once more we derive the strong form of the DG method,

$$\begin{aligned} &\frac{\partial}{\partial t} \int_{\Omega_s} \widehat{u}_n^{\delta}(\xi) \cdot \ell_i(\xi) \, d\xi \; + \int_{\Omega_s} \frac{\partial}{\partial \xi} \widehat{f}_n^{\delta D}(\xi) \cdot \ell_i(\xi) \, d\xi \; + \\ &\left[\left(\widehat{f}_n^{\delta I} - \widehat{f}_n^{\delta D} \right) \cdot \ell_i(\xi) \right]_{-1}^{+1} = 0 \quad \forall \, i. \end{aligned} \tag{2.5}$$

The $P+1$ equations can be written in matrix form as

$$\mathbf{M} \frac{d\widehat{\mathbf{u}}_n^{\delta}}{dt} + \mathbf{S}\widehat{\mathbf{f}}_n^{\delta D} = \left[\left(\widehat{f}_n^{\delta D} - \widehat{f}_n^{\delta I} \right) \mathbf{l}(\xi) \right]_{-1}^{+1}, \tag{2.6}$$

where $\mathbf{M}$ and $\mathbf{S}$ are the local mass matrix and stiffness matrices

$$M_{i,j} = \int_{\Omega_s} \ell_i(\xi) \ell_j(\xi) \, d\xi, \qquad S_{i,j} = \int_{\Omega_s} \ell_i(\xi) \frac{d\ell_j(\xi)}{d\xi} \, d\xi \tag{2.7}$$

and $\mathbf{l}(\xi) = [\ell_0(\xi), ..., \ell_P(\xi)]^T$.

2.2.2 Flux reconstruction method for 1D scalar conservation law

After having obtained the formulation of DG scheme for the 1D scalar equation on a regular grid, we follow the approach of Huynh (2007) to write the formulation of the FR scheme.

Consider again the 1D scalar conservation law of Eq. 2.1. As before, we discretise the domain into N non-overlapping standard elements and we map each

of them into a standard element Ω_s. The solution $\widehat{u}_n^\delta$ in the reference space can again be expanded in a polynomial basis as in Eq. 2.3. The approximate flux in the standard element can be written as

$$\widehat{f}_n^\delta = \widehat{f}_n^\delta(\xi, t) = \widehat{f}_n^{\delta D} + \widehat{f}_n^{\delta C}$$

where $\widehat{f}_n^{\delta D}$ remains defined as in Eq. 2.4 and

$$\widehat{f}_n^{\delta C} = \Psi_L(\xi)\left[\widehat{f}_n^{\delta I} - \widehat{f}_n^{\delta D}\right]\Big|_{\xi=-1} + \quad \Psi_R(\xi)\left[\widehat{f}_n^{\delta I} - \widehat{f}_n^{\delta D}\right]\Big|_{\xi=1} = \widehat{\Delta f}_L\Psi_L + \widehat{\Delta f}_R\Psi_R.$$

In the above, as already described in section 1.4, D stands for discontinuous, C stands for corrective, $\widehat{\Delta f}_L$ and $\widehat{\Delta f}_R$ are the left and right jump fluxes at the boundary, $\Psi_L(\xi)$ and $\Psi_R(\xi)$ are the left and right correction functions which have to approximate zero in some sense and satisfy

$$\Psi_L(-1) = 1, \quad \Psi_L(1) = 0, \quad \Psi_R(-1) = 0, \quad \Psi_R(1) = 1.$$

The derivative of the flux with respect to ξ is

$$\frac{\partial \widehat{f}_n^\delta}{\partial \xi} = \frac{\partial \widehat{f}_n^{\delta D}}{\partial \xi} + \frac{\partial \widehat{f}_n^{\delta C}}{\partial \xi} = \frac{\partial \widehat{f}_n^{\delta D}}{\partial \xi} + \widehat{\Delta f}_L\frac{d\Psi_L}{\partial \xi} + \widehat{\Delta f}_R\frac{d\Psi_R}{\partial \xi}$$

As noted by Allaneau and Jameson (2011) the functions $\Psi_L(\xi)$ and $\Psi_R(\xi)$ can be defined in the space of polynomials of degree at most $P+1$. As a consequence the derivatives $\frac{d\Psi_L}{\partial \xi}$ and $\frac{d\Psi_R}{\partial \xi}$ can be represented in the same basis as the solution $\widehat{u}_n^\delta$. This gives the formulation of the FR scheme:

$$\frac{\partial \widehat{u}_n^\delta}{\partial t} + \frac{\partial \widehat{f}_n^{\delta D}}{\partial \xi} + \widehat{\Delta f_L}\frac{d\Psi_L}{\partial \xi} + \widehat{\Delta f_R}\frac{d\Psi_R}{\partial \xi} = 0. \tag{2.8}$$

Now, we require the residual of Eq. 2.8 to be orthogonal to a set of smooth function as before and we therefore obtain

$$\begin{aligned}
&\frac{\partial}{\partial t}\int_{\Omega_s} \widehat{u}_n^\delta(\xi)\cdot\ell_i(\xi)\,d\xi \; + \int_{\Omega_s}\frac{\partial}{\partial \xi}\widehat{f}_n^{\delta D}(\xi)\cdot\ell_i(\xi)\,d\xi \; + \\
&\int_{\Omega_s}\left(\widehat{\Delta f_L}\Psi'_L(\xi) + \widehat{\Delta f_R}\Psi'_R(\xi)\right)\cdot\ell_i(\xi)\,d\xi \quad = 0 \quad \forall\, i.
\end{aligned}$$

Performing an integration by parts on the last term we see that

$$\begin{aligned}
&\frac{\partial}{\partial t}\int_{\Omega_s} \widehat{u}_n^\delta(\xi)\cdot\ell_i(\xi)\,d\xi \; + \int_{\Omega_s}\frac{\partial}{\partial \xi}\widehat{f}_n^{\delta D}(\xi)\cdot\ell_i(\xi)\,d\xi \; + \\
&\widehat{\Delta f_L}\cdot\ell_i(-1) - \widehat{\Delta f_R}\cdot\ell_i(1) - \\
&\int_{\Omega_s}\left(\widehat{\Delta f_L}\Psi_L(\xi) + \widehat{\Delta f_R}\Psi_R(\xi)\right)\cdot\ell'_i(\xi)\,d\xi \quad = 0 \quad \forall\, i,
\end{aligned} \tag{2.9}$$

and so eq. 2.9 becomes

$$\begin{aligned}
&\frac{\partial}{\partial t}\int_{\Omega_s} \widehat{u}_n^{\delta}(\xi)\cdot\ell_i(\xi)\,d\xi \;+\int_{\Omega_s}\frac{\partial}{\partial\xi}\widehat{f}_n^{\delta D}(\xi)\cdot\ell_i(\xi)\,d\xi \;+\\
&\Big[\Big(\widehat{f}_n^{\delta I}-\widehat{f}_n^{\delta D}\Big)\cdot\ell_i(\xi)\Big]_{-1}^{+1} \;-\\
&\int_{\Omega_s}\Big(\widehat{\Delta f_L}\Psi_L(\xi)+\widehat{\Delta f_R}\Psi_R(\xi)\Big)\cdot\ell_i'(\xi) \;=0\quad\forall\, i.
\end{aligned}\tag{2.10}$$

Eq. 2.10 is equal to Eq. 2.5 except for the last term. As pointed out by Huynh (2007), that term vanishes if we define Ψ_L and Ψ_R using Radau polynomials of order $P+1$ because a Radau polynomial of order $P+1$ is orthogonal to all the polynomials of order less than or equal to $P-1$. Therefore, we recover exactly the DG method, at least for the case of a linear flux function. We refer to this type of FR scheme with correction functions equal to Radau polynomials as $\mathrm{FR_{DG}}$.

The $P+1$ equations can be written in matrix form as:

$$\frac{d\widehat{\mathbf{u}}_n^{\delta}}{dt}+\mathbf{D}\widehat{\mathbf{f}}_n^{\delta D}+\widehat{\Delta f}_L\boldsymbol{\Psi}'_L(\xi)+\widehat{\Delta f}_R\boldsymbol{\Psi}'_R(\xi)=0$$

where $\mathbf{D}=\mathbf{M}^{-1}\mathbf{S}$ is the local differentiation matrix with $D_{i,j}=\ell_j'(\xi_i)$, $\boldsymbol{\Psi}'_L(\xi)=[\Psi'_L(\xi_0),...,\Psi'_L(\xi_P)]^T$ and $\boldsymbol{\Psi}'_R(\xi)=[\Psi'_R(\xi_0),...,\Psi'_R(\xi_P)]^T$.

2.2.3 2D scalar conservation law on quadrilateral grids

In this subsection we demonstrate the equality of DG and $\mathrm{FR_{DG}}$ on meshes of regular quadrilateral elements. Consider the 2D scalar conservation law

$$\frac{\partial u}{\partial t}+\nabla_{\boldsymbol{x}}\cdot\boldsymbol{f}=0\tag{2.11}$$

within a domain $\Omega\subset\mathbb{R}^2$, with $\mathbf{f}=(f_1,f_2)$, where $f_1=f_1(u)$ and $f_2=f_2(u)$ are the fluxes in the x and y directions respectively. The domain Ω is partitioned into N non-overlapping, conforming quadrilateral elements Ω_n such that

$$\Omega=\bigcup_{n=1}^{N}\Omega_n.$$

As in the one dimensional case, we map each quadrilateral element Ω_n into a reference element $\Omega_s=[-1,1]^2$ in the transformed space $\boldsymbol{\xi}=(\xi_1,\xi_2)$ with the relation of Eq. 1.20 which can be redefined as

$$\boldsymbol{x}=\Theta_n(\boldsymbol{\xi})=\sum_{i=1}^{K}\boldsymbol{M}_i\boldsymbol{x}_{i,n}.\tag{2.12}$$

Here $\boldsymbol{M}_i$ are the transformation functions, $\boldsymbol{x}_{i,n}$ are the physical coordinates of the points which describe each physical element Ω_n and K is the number of points. In the reference domain the transformed equation into any individual Ω_s becomes

$$\frac{\partial \widehat{u}_n^\delta}{\partial t} + \nabla_{\boldsymbol{\xi}} \cdot \widehat{\boldsymbol{f}}_n^\delta = 0, \tag{2.13}$$

where

$$\widehat{u}_n^\delta = \widehat{u}_n^\delta(\xi_1, \xi_2, t) = J_n u_n^\delta(\boldsymbol{\Theta}^{-1}(\xi_1, \xi_2), t),$$

$$\widehat{\boldsymbol{f}}_n^\delta = \widehat{\boldsymbol{f}}_n^\delta(\xi_1, \xi_2, t) = (\widehat{f}_{1,n}^\delta, \widehat{f}_{2,n}^\delta) = \left(\frac{\partial y}{\partial \xi_2} f_{1,n}^\delta - \frac{\partial x}{\partial \xi_2} f_{2,n}^\delta, -\frac{\partial y}{\partial \xi_1} f_{1,n}^\delta + \frac{\partial x}{\partial \xi_1} f_{2,n}^\delta \right),$$

and the metric terms $J_n, \frac{\partial x}{\partial \xi_1}, \frac{\partial x}{\partial \xi_2}, \frac{\partial y}{\partial \xi_1}, \frac{\partial y}{\partial \xi_2}$ can be evaluated from Eq. 2.12. The transformed solution $\widehat{u}_n^\delta$ in the reference space within each element is approximated by a polynomial of degree P and can be written as

$$\widehat{u}_n^\delta = \sum_{i,j=0}^{i,j=P} \widehat{u}_{ij}^\delta \ell_{ij}(\xi_1, \xi_2) = \sum_{i,j=0}^{i,j=P} \widehat{u}_{ij}^\delta \ell_i(\xi_1) \ell_j(\xi_2), \tag{2.14}$$

where $\ell_i(\xi_1)$ and $\ell_j(\xi_2)$ are one-dimensional Lagrange polynomials of order P of the nodal basis associated with the 1D solution points along ξ_1 and ξ_2 directions. The polynomial $\ell_{ij}(\xi_1, \xi_2)$ is obtained by the tensor product of the 1D nodal bases $\ell_i(\xi_1)$ and $\ell_j(\xi_2)$ such that

$$\ell_{ij}(\xi_{1_\ell}, \xi_{2_m}) = \begin{cases} 1, & \text{if } (\xi_{1_\ell}, \xi_{2_m}) = (\xi_{1_i}, \xi_{2_j}), \\ 0, & \text{if } (\xi_{1_\ell}, \xi_{2_m}) \neq (\xi_{1_i, \xi_{2_j}}). \end{cases} \tag{2.15}$$

Analogously the approximate transformed fluxes $\widehat{f}_{1,n}^{\delta D}$ and $\widehat{f}_{2,n}^{\delta D}$ in the reference space within each element can be written as

$$\widehat{f}_{1,n}^{\delta D} = \sum_{i,j=0}^{i,j=P} \widehat{f}_{1_{ij}}^{\delta D} \ell_{ij}(\xi_1, \xi_2) = \sum_{i,j=0}^{i,j=P} \widehat{f}_{1_{ij}}^{\delta D} \ell_i(\xi_1) \ell_j(\xi_2), \tag{2.16}$$

$$\widehat{f}_{2,n}^{\delta D} = \sum_{i,j=0}^{i,j=P} \widehat{f}_{2_{ij}}^{\delta D} \ell_{ij}(\xi_1, \xi_2) = \sum_{i,j=0}^{i,j=P} \widehat{f}_{2_{ij}}^{\delta D} \ell_i(\xi_1) \ell_j(\xi_2), \tag{2.17}$$

where the superscript D again stands for discontinuous since the fluxes are calculated directly from the approximate solution, which is in general piecewise discontinuous between elements.

To prove that the DG method is identical to $\mathrm{FR_{DG}}$ for the case of a 2D scalar equation we first write the residual of Eq. 2.13 as

$$R^\delta = \frac{\partial \widehat{u}_n^\delta}{\partial t} + \frac{\partial \widehat{f}_{1,n}^\delta}{\partial \xi_1} + \frac{\partial \widehat{f}_{2,n}^\delta}{\partial \xi_2},$$

and we require the residual R^δ to be orthogonal to a set of smooth test functions. The use of test functions defined by the basis $\ell_{ij}(\xi_1, \xi_2)$ leads to the following equation:

$$\iint_{\Omega_s} R^\delta \cdot \ell_{ij}\, d\xi_1 d\xi_2 = 0 \quad \forall\, i, j.$$

Performing the integration by parts, substituting the boundary terms by the numerical interface fluxes and integrating by parts once more we derive the strong form of the DG method,

$$\begin{aligned}
&\frac{\partial}{\partial t} \iint_{\Omega_s} \widehat{u}_n^\delta(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \\
&\iint_{\Omega_s} \frac{\partial}{\partial \xi_1} \widehat{f}_{1,n}^{\delta D}(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \iint_{\Omega_s} \frac{\partial}{\partial \xi_2} \widehat{f}_{2,n}^{\delta D}(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \\
&\int_{-1}^{1} \left[\widehat{f}_{1,n}^{\delta I}(1, \xi_2) - \widehat{f}_{1,n}^{\delta D}(1, \xi_2) \right] \cdot \ell_{ij}(1, \xi_2)\, d\xi_2 - \\
&\int_{-1}^{1} \left[\widehat{f}_{1,n}^{\delta I}(-1, \xi_2) - \widehat{f}_{1,n}^{\delta D}(-1, \xi_2) \right] \cdot \ell_{ij}(-1, \xi_2)\, d\xi_2 + \\
&\int_{-1}^{1} \left[\widehat{f}_{2,n}^{\delta I}(\xi_1, 1) - \widehat{f}_{2,n}^{\delta D}(\xi_1, 1) \right] \cdot \ell_{ij}(\xi_1, 1)\, d\xi - \\
&\int_{-1}^{1} \left[\widehat{f}_{2,n}^{\delta I}(\xi_1, -1) - \widehat{f}_{2,n}^{\delta D}(\xi_1, -1) \right] \cdot \ell_{ij}(\xi_1, -1)\, d\xi = 0, \quad \forall\, i, j.
\end{aligned}$$

Via the tensor product the above equation can be rewritten as:

$$\begin{aligned}
&\frac{\partial}{\partial t} \iint_{\Omega_s} \widehat{u}_n^\delta(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \\
&\iint_{\Omega_s} \frac{\partial}{\partial \xi_1} \widehat{f}_{1,n}^{\delta D}(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \iint_{\Omega_s} \frac{\partial}{\partial \xi_2} \widehat{f}_{2,n}^{\delta D}(\xi_1, \xi_2) \cdot \ell_{ij}(\xi_1, \xi_2)\, d\xi_1\, d\xi_2 + \\
&\int_{-1}^{1} \left[\widehat{f}_{1,n}^{\delta I}(1, \xi_2) - \widehat{f}_{1,n}^{\delta D}(1, \xi_2) \right] \cdot \ell_i(1) \ell_j(\xi_2)\, d\xi_2 - \\
&\int_{-1}^{1} \left[\widehat{f}_{1,n}^{\delta I}(-1, \xi_2) - \widehat{f}_{1,n}^{\delta D}(-1, \xi_2) \right] \cdot \ell_i(-1) \ell_j(\xi_2)\, d\xi_2 + \\
&\int_{-1}^{1} \left[\widehat{f}_{2,n}^{\delta I}(\xi_1, 1) - \widehat{f}_{2,n}^{\delta D}(\xi_1, 1) \right] \cdot \ell_i(\xi_1) \ell_j(1)\, d\xi_1 - \\
&\int_{-1}^{1} \left[\widehat{f}_{2,n}^{\delta I}(\xi_1, -1) - \widehat{f}_{2,n}^{\delta D}(\xi_1, -1) \right] \cdot \ell_i(\xi_1) \ell_j(-1)\, d\xi_1 = 0, \quad \forall\, i, j.
\end{aligned}$$

As shown in Huynh (2007), we can write $\ell(-1)$ through the expression

$$\int_{-1}^{1} \Psi'_L(\xi_1)\ell(\xi_1)\, d\xi_1 = \int_{-1}^{1} \Psi'_B(\xi_2)\ell(\xi_2)\, d\xi_2 = -\ell(-1), \tag{2.18}$$

where L and B denote left and bottom edges respectively, $\Psi_L(\xi_1)$ and $\Psi_B(\xi_2)$ are the right Radau polynomials of order $P+1$ on $[-1,1]$ which vanish at the right boundary $\xi_1 = \xi_2 = 1$. This is due to the orthogonality of the right Radau polynomial of order $P+1$ to all polynomials of order up to $P-1$. Analogously, we can write $\ell(1)$ using the expression

$$\int_{-1}^{1} \Psi'_R(\xi_1)\ell(\xi_1)\, d\xi_1 = \int_{-1}^{1} \Psi'_T(\xi_2)\ell(\xi_2)\, d\xi_2 = \ell(1), \tag{2.19}$$

where R and T denote right and top edges and $\Psi_R(\xi_1)$ and $\Psi_T(\xi_2)$ are the left Radau polynomials of order $P+1$ on $[-1,1]$ which vanish at the left boundary $\xi_1 = \xi_2 = -1$.

Substituting Eqs. 2.18 and 2.19 into the strong form of the DG method leads to

$$\begin{aligned}
&\frac{\partial}{\partial t} \iint_{\Omega_s} \widehat{u}_n^{\delta}(\xi_1,\xi_2) \cdot \ell_{ij}(\xi_1,\xi_2)\, d\xi_1\, d\xi_2 + \\
&\iint_{\Omega_s} \Big\{ \widehat{f}_{1,n}^{\delta D}(\xi_1,\xi_2) + \Big[\widehat{f}_{1,n}^{\delta I}(1,\xi_2) - \widehat{f}_{1,n}^{\delta D}(1,\xi_2) \Big] \Psi_R(\xi_1) + \\
&\Big[\widehat{f}_{1,n}^{\delta I}(-1,\xi_2) - \widehat{f}_{1,n}^{\delta D}(-1,\xi_2) \Big] \Psi_L(\xi_1) \Big\}_{/\xi_1} \cdot \ell_i(\xi_1)\ell_j(\xi_2)\, d\xi_2 + \\
&\iint_{\Omega_s} \Big\{ \widehat{f}_{2,n}^{\delta D}(\xi_1,\xi_2) + \Big[\widehat{f}_{2,n}^{\delta I}(\xi,1) - \widehat{f}_{2,n}^{\delta D}(\xi_1,1) \Big] \Psi_T(\xi_2) + \\
&\Big[\widehat{f}_{2,n}^{\delta I}(\xi_1,-1) - \widehat{f}_{2,n}^{\delta D}(\xi_1,-1) \Big] \Psi_B(\xi_2) \Big\}_{/\xi_2} \cdot \ell_i(\xi_1)\ell_j(\xi_2)\, d\xi_2 \; = 0, \;\; \forall\, i,j.
\end{aligned}$$

Finally, by defining the fluxes $F_{1,n}^{\delta}(\xi_1,\xi_2)$ and $F_{2,n}^{\delta}(\xi_1,\xi_2)$ to be

$$\begin{cases}
F_{1,n}^{\delta}(\xi_1,\xi_2) = \widehat{f}_{1,n}^{\delta D}(\xi_1,\xi_2) + \Big[\widehat{f}_{1,n}^{\delta I}(1,\xi_2) - \widehat{f}_{1,n}^{\delta D}(1,\xi_2) \Big] \Psi_R(\xi_1) + \\
\qquad\qquad \Big[\widehat{f}_{1,n}^{\delta I}(-1,\xi_2) - \widehat{f}_{1,n}^{\delta D}(-1,\xi_2) \Big] \Psi_L(\xi_1) \\
F_{2,n}^{\delta}(\xi_1,\xi_2) = \widehat{f}_{2,n}^{\delta D}(\xi_1,\xi_2) + \Big[\widehat{f}_{2,n}^{\delta I}(\xi_1,1) - \widehat{f}_{2,n}^{\delta D}(\xi_1,1) \Big] \Psi_T(\xi_2) + \\
\qquad\qquad \Big[\widehat{f}_{2,n}^{\delta I}(\xi_1,-1) - \widehat{f}_{2,n}^{\delta D}(\xi_1,-1) \Big] \Psi_B(\xi_2)
\end{cases}$$

we can write

$$\iint_{\Omega_s} \left\{ \left(\widehat{u}_n^{\delta}(\xi_1,\xi_2)\right)_{/t} + \left(F_{1,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_1} + \left(F_{2,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_2} \right\} \cdot \ell_{ij}(\xi_1,\xi_2)\, d\xi_1\, d\xi_2 = 0.$$

Since $\widehat{u}_n^{\delta}(\xi_1,\xi_2)$, $\left(F_{1,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_1}$ and $\left(F_{2,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_2}$ are polynomials of degree P and Eq. 2.2.3 is valid for any polynomial $\ell_{ij}(\xi_1,\xi_2)$ of degree P we can eliminate the integrals and write:

$$\left(\widehat{u}_n^{\delta}(\xi_1,\xi_2)\right)_{/t} + \left(F_{1,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_1} + \left(F_{2,n}^{\delta}(\xi_1,\xi_2)\right)_{/\xi_2} = 0. \tag{2.20}$$

This proves that the FR approach recovers the DG method, at least for the case of a linear flux function on a tensor product grid, when Radau polynomials are used as correction functions leading to the $\mathrm{FR_{DG}}$ scheme.

By using Eqs. 2.14, 2.15, 2.16 and 2.17 the equation for the generic point (ξ_{1_i},ξ_{2_j}) becomes:

$$\begin{aligned}
\frac{d\widehat{u}_{ij}}{dt} = &-\left.\frac{\partial \widehat{f}_{1,n}^{\delta D}}{\partial \xi_1}\right|_{\xi_{1_i},\xi_{2_j}} - \left.\frac{\partial \widehat{f}_{2,n}^{\delta D}}{\partial \xi_2}\right|_{\xi_{1_i},\xi_{2_j}} \\
&- \left[\widehat{f}_{1,n}^{\delta I}(1,\xi_{2_j}) - \widehat{f}_{1,n}^{\delta D}(1,\xi_{2_j})\right] \left.\frac{\partial \Psi_R}{\partial \xi}\right|_{\xi_{1_i}} - \left[\widehat{f}_{1,n}^{\delta I}(-1,\xi_{2_j}) - \widehat{f}_{1,n}^{\delta D}(-1,\xi_{2_j})\right] \left.\frac{\partial \Psi_L}{\partial \xi_1}\right|_{\xi_{1_i}} \\
&- \left[\widehat{f}_{2,n}^{\delta I}(\xi_{1_i},1) - \widehat{f}_{2,n}^{\delta D}(\xi_{1_i},1)\right] \left.\frac{\partial \Psi_T}{\partial \xi_2}\right|_{\xi_{2_j}} - \left[\widehat{f}_{2,n}^{\delta I}(\xi_i,-1) - \widehat{f}_{2,n}^{\delta D}(\xi_{1_i},-1)\right] \left.\frac{\partial \Psi_B}{\partial \xi_2}\right|_{\xi_{2_j}}.
\end{aligned}$$

2.2.4 $\mathrm{FR_{g2}}$ as $\mathrm{DG_{SEM}}$-GLL with lumped mass matrix

In Allaneau and Jameson (2011) it was shown that some linearly filtered DG methods can be expressed in the flux reconstruction framework. Equivalently, some FR schemes can be described as a DG method for which a linear filtering operator is applied on the residual. In particular the entire class of energy-stable FR schemes introduced in Vincent et al. (2011a) can be recast as a filtered DG method. In this subsection we demonstrate the equivalence between the Huynh's g2 FR scheme and a particular type of filtered DG scheme.

For Eq. 2.2 the classical DG method can be written as

$$\mathbf{M}\frac{d\widehat{\mathbf{u}}_n^{\delta}}{dt} + \mathbf{S}\widehat{\mathbf{f}}_n^{\delta D} = \left[\left(\widehat{f}_n^{\delta D} - \widehat{f}_n^{\delta I}\right)\mathbf{l}(\xi)\right]_{-1}^{+1}, \tag{2.21}$$

where $\mathbf{M}$ and $\mathbf{S}$ again denote the local mass and stiffness matrices. The filtered DG method is

$$\widetilde{\mathbf{M}}\frac{d\widehat{\mathbf{u}}_n^{\delta}}{dt} - \mathbf{S}\widehat{\mathbf{f}}_n^{\delta D} = \left[\widehat{f}_n^{\delta I}\mathbf{l}(\xi)\right]_{-1}^{+1},$$

where

$$\widetilde{\mathbf{M}} = \mathbf{M} \cdot \mathbf{F}^{-1} \tag{2.22}$$

and $\mathbf{F}$ denotes a linear filter. For Huynh's g2 FR scheme the correction functions Ψ_L and Ψ_R defined in section 1.4 and here reported for sake of clarity are:

$$\Psi_L = \frac{(-1)^P}{2}\left[L_{\mathrm{P}} - \left(\frac{(\mathrm{P}+1)L_{\mathrm{P}-1}L_{\mathrm{P}} + \mathrm{P}L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right]$$

and

$$\Psi_R = \frac{1}{2}\left[L_{\mathrm{P}} + \left(\frac{(\mathrm{P}+1)L_{\mathrm{P}-1}L_{\mathrm{P}} + \mathrm{P}L_{\mathrm{P}+1}}{2\mathrm{P}+1}\right)\right],$$

where L_{P} is a Legendre polynomial of order P. The explicit form of the filter in the non-normalized Legendre basis is a diagonal matrix of the form

$$\mathbf{F_P} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \frac{\mathrm{P}}{1+\frac{2\mathrm{P}+1}{2}(\mathrm{P}!a_{\mathrm{P}})^2} \end{bmatrix},$$

where

$$a_{\mathrm{P}} = \frac{(2\mathrm{P})!}{2^{\mathrm{P}}(\mathrm{P}!)^2}.$$

The filter can be transformed to the computational basis B so that

$$\mathbf{F_B} = {\mathbf{V_{B,P}}}^{-1} \cdot \mathbf{F_P} \cdot \mathbf{V_{B,P}}, \tag{2.23}$$

where $\mathbf{V_{B,P}}$ is the transformation matrix from a general and unspecified basis B to basis P. If we consider a basis B which represents the solution $\widehat{u}_n^\delta$ in terms of Lagrange polynomials with solution values at $P+1$ Gauss-Lobatto-Legendre (GLL) points, substituting Eq. 2.23 in Eq. 2.22 we obtain

$$\widetilde{\mathbf{M}} = \begin{bmatrix} \ddots & & \\ & w_i & \\ & & \ddots \end{bmatrix}, \tag{2.24}$$

where w_i denote the weights of the GLL quadrature rule at the $P+1$ points.

Consider now Eq. 2.21 with solution coefficients $\widehat{u}_n^\delta$ represented by Lagrange polynomials with solution values at $P+1$ GLL points. If we additionally choose a GLL quadrature to utilise a $\mathrm{DG_{SEM}}$ scheme, then, as it is well known, this rule is exact only for polynomials up to order $2P-1$, so the coefficients of the mass matrices as defined in Eq. 2.7 will contain a numerical quadrature error since

each integrand is a polynomial of order $2P$. However, since, as already explained in section 1.3.1, the Lagrangian basis has the property that $\ell_i(\xi_j) = \delta_{ij}$ where δ_{ij} represents the Kronecker delta function, with this quadrature rule the mass matrix $\mathbf{M}$ obtained is diagonal:

$$\mathbf{M} = \begin{bmatrix} \ddots & & \\ & w_i & \\ & & \ddots \end{bmatrix}, \tag{2.25}$$

where w_i are the weights of the GLL quadrature rule. This makes the scheme extremely efficient as the mass matrix is now trivially invertible. The efficiency of the scheme can also be seen in the FR framework. In fact, with solution values located at GLL points, the derivatives of the correction functions of Ψ_L and Ψ_R vanish at all solution points except at the left boundary if evaluating Ψ'_L or at the right boundary if evaluating Ψ'_R (the zeros of Ψ' are at GLL points). Thus, in Huynh's g2 FR scheme, the corrective flux modifies only the boundary points.

The mass matrix of Eq. 2.25 is equal to that of Eq. 2.24, so Huynh's g2 FR scheme can be seen as a DG scheme where mass matrix $\mathbf{M}$ is lumped. We refer to this type of DG scheme as $\mathrm{DG_{SEM}}$-GLL with lumped mass matrix and the type of FR scheme recovered as $\mathrm{FR_{g2}}$. Therefore, in this case we recover exactly the $\mathrm{DG_{SEM}}$-GLL with lumped mass matrix method at least for the case of a linear flux function.

2.2.5 Summary

In the previous sections we have derived the connections between the DG method and FR schemes in the case of an advection problem. A summary of the results of this section can be seen in Table 2.1, where the connections between the three DG and two FR schemes is presented.

In the first two columns of the table we consider two nodal types of $\mathrm{DG_{SEM}}$ scheme.
The $\mathrm{DG_{SEM}}$ method with exact mass matrix is equivalent to the $\mathrm{FR_{DG}}$ scheme.
The $\mathrm{DG_{SEM}}$-GLL (solution values at Gauss-Lobatto-Legendre points) method with lumped mass matrix is equivalent to the $\mathrm{FR_{g2}}$ scheme.
The equivalences are valid both for linear and nonlinear flux functions.

In the third column of this table we consider an arbitrary nodal DG scheme and state that Q (the quadrature order) must be larger than P (the polynomial order). We note that this statement does not necessarily mean that the number of quadrature points is higher than the number of solution points but only that the precision of the quadrature rule chosen is higher than that of a collocation quadrature rule. For example, if the solution values are at P Gauss-Lobatto-Legdendre (GLL) points and the quadrature points are at Q Gauss-Legendre (GL)

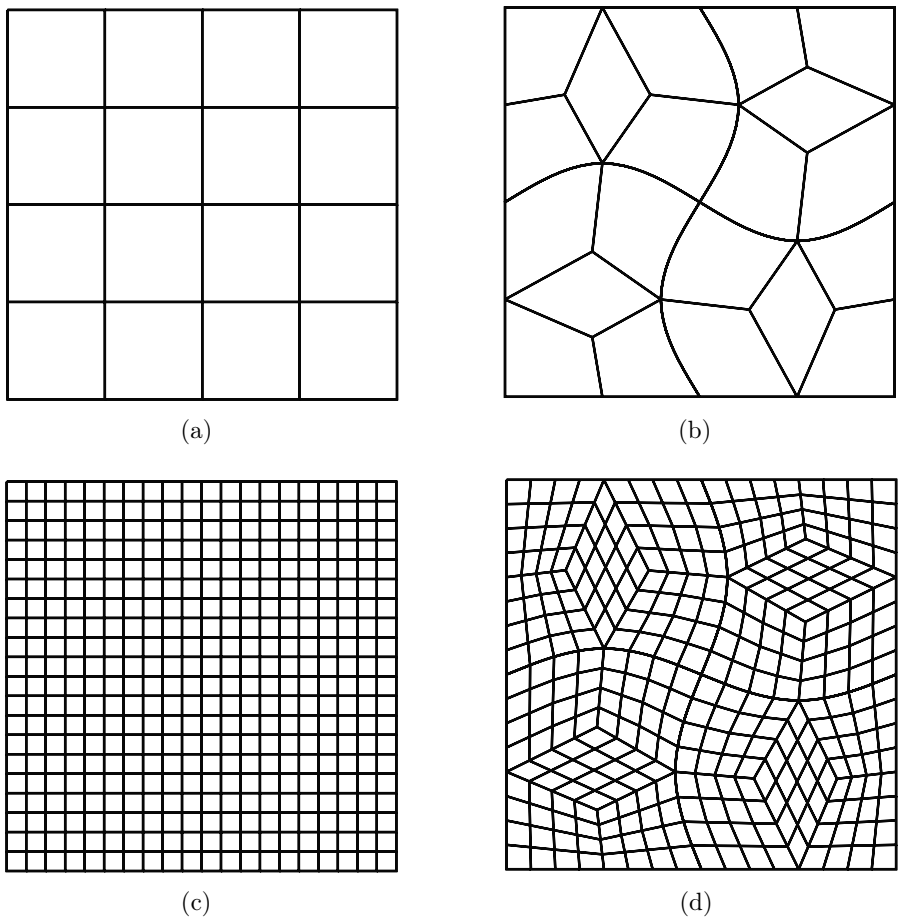


Figure 2.1: Meshes utilised in the numerical experiments. The grids on the left are regular and on the right are deformed.

points with $Q = P$, the quadrature precision is higher than that obtained with the same number of quadrature points collocated at GLL points. We additionally note that the $\mathrm{DG}_{\mathrm{Q>P}}$ scheme with exact mass matrix is equivalent to the $\mathrm{FR}_{\mathrm{DG}}$ scheme when the flux function is linear. If the flux function is nonlinear the schemes are different because the aliasing errors arising from the nonlinearities are minimised in the DG scheme by using a quadrature with a higher precision.

All of the derivations above assume that the elemental Jacobian is constant in space, so that each element is not deformed. In the case of a deformed grid it is not ensured that the results are always valid. The deformation of the grid introduces a nonlinearity in the equations and this implies that the equivalences depend on

the approximation of the geometric factors. For details about the connections between the DG method and FR approach on deformed grids the reader can refer to Mengaldo et al. (2015c) where the author and co-workers extended the work in De Grazia et al. (2014) by considering irregular/curvilinear meshes.

2.3 Numerical experiments

A range of numerical experiments were undertaken in order to confirm the theoretical equality of the schemes as proven in the previous section. These simulations take into consideration both linear and nonlinear flux functions. Four grids were generated, the first two of domain $\Omega = [-1, 1] \times [-1, 1]$, the other two of domain $\Omega = [0, 10] \times [-5, 5]$, as shown in figure 2.1. The first of these (figure 2.1(a)) is a structured subdivision of the domain into a 4×4 array of regular quadrilaterals (i.e. the Jacobian is constant across each element). To demonstrate the schemes' equivalence for curvilinear (i.e. spatially varying Jacobian), an additional unstructured grid has been generated as shown in figure 2.1(b). These meshes were used for simulations of the linear flux function which will be presented below. Additionally the grids depicted in Figures 2.1(c) and 2.1(d), which possess 400 and 320 elements respectively, were obtained through subdivision of these meshes and were used for simulations of the nonlinear flux functions.

2.3.1 Linear flux

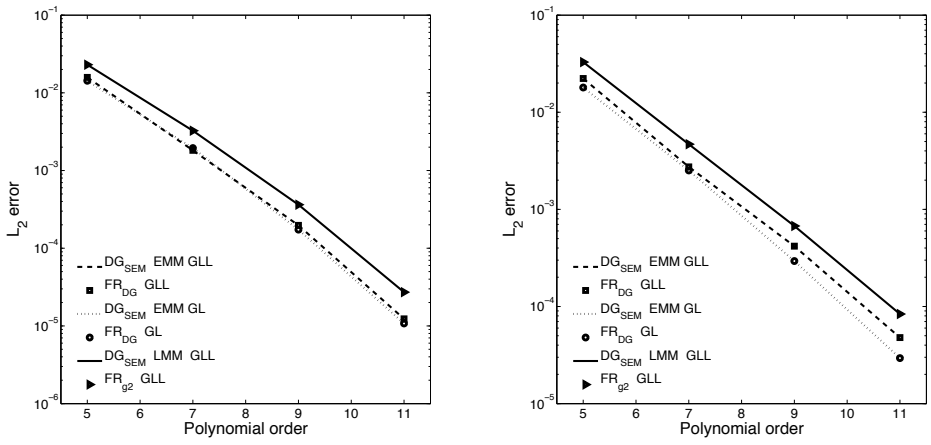
This series of experiments were undertaken on the 2D linear advection equation:

$$\frac{\partial u}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \mathbf{f} = 0, \quad \mathbf{f} = \mathbf{a}\, u, \quad \mathbf{a} = [1, 0]. \tag{2.26}$$

The solution was represented by a fifth, seventh, ninth and eleventh order polynomial. First the polynomials were defined by a collocation projection of the solution values at six, eight, ten and twelve GLL points for each element. Then the calculations were repeated using GL points. In all the cases a collocation quadrature rule was used for the inner product of the advection term of the DG scheme. With GLL points both the cases of exact mass matrix (EMM) and lumped mass matrix (LMM) were considered. Periodic conditions were applied at the boundary of the grids, and the Gaussian profile

$$u(x, y, 0) = e^{-40(x^2+y^2)} \tag{2.27}$$

was used as an initial condition at $t = 0$. A fourth-order explicit Runge-Kutta time integration scheme was used to discretise the equations in time. The final time of the simulation was fixed at $T = 2s$ so that the exact solution is equal to the initial



(a) (b)

Figure 2.2: L_2 error of the final solution vs. polynomial order P. On the left the results on the regular grid. On the right the results on the deformed grid. Results from the DG simulations are indicated by lines, whereas results from FR simulations are indicated by markers.

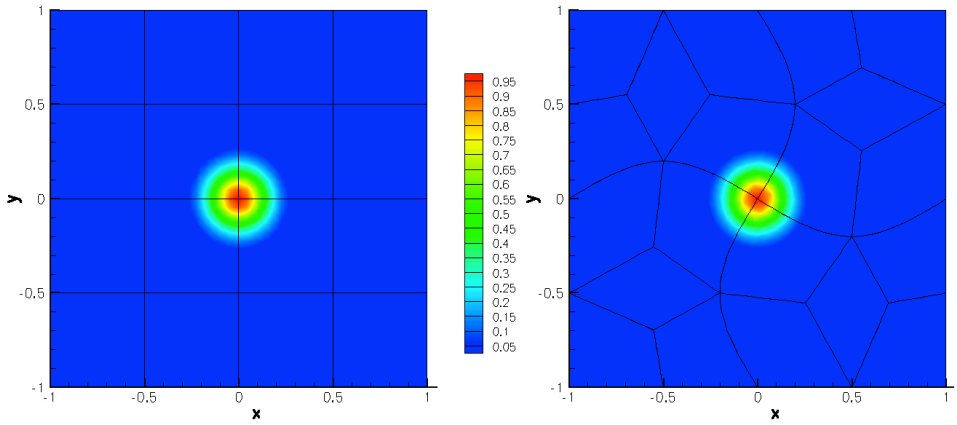


Figure 2.3: Visualization of the solution field at the final time $T = 2s$. On the left we depict the regular grid with solution values at twelve Gauss-Legendre points. On the right the deformed grid with solution values at twelve Gauss-Lobatto-Legendre points is shown.

condition of Eq. 2.27. The time-step was chosen as $\Delta t_{max}/100$, where Δt_{max} is the maximum time-step that allowed the simulation to remain stable. This choice for the time-step was made to ensure that the error from the time-stepping scheme was negligible compared to the spatial discretization.

As the theoretical results suggest, figure 2.2(a) confirms that for regular elements the L_2 error of the final solution field for each DG scheme in table 2.1 precisely matches the equivalent FR scheme. Figure 2.2(b) also shows that the results match on the deformed grid because the approximation of the geometric factors is performed in the same way for both approaches. For both regular and deformed grids the results are identical up to machine precision. In figure 2.3 we see that the solution at the final time on the regular grid and on the deformed grid also correlate. We recall that in figure 2.2, EMM and LMM denote results obtained with exact and lumped mass matrices respectively, and GLL and GL solution values denote Gauss-Lobatto-Legendre and Gauss-Legendre points. This terminology will be used in the following comparison figures.

2.3.2 Nonlinear flux

This series of experiments were undertaken using the two-dimensional Euler equations which govern the compressible flow of an inviscid fluid.
These can be written as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}_1^i}{\partial x} + \frac{\partial \mathbf{F}_2^i}{\partial y} = 0,$$

where $\mathbf{q} = (\rho, \rho u, \rho v, E)^\top$ is the vector of the conserved variables and $\mathbf{F}^i = \mathbf{F}^i(\mathbf{q})$ and $\mathbf{G}^i = \mathbf{G}^i(\mathbf{q})$ are the vectors of the inviscid fluxes,

$$\mathbf{F}_1^i = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(E+p) \end{Bmatrix}, \quad \mathbf{F}_2^i = \begin{Bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(E+p) \end{Bmatrix}. \tag{2.28}$$

In the above, ρ is the density, u and v are the velocity components in x and y directions, p is the pressure and E is the total energy.

For a perfect gas the pressure is related to the total energy by the expression

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

where γ denotes the constant ratio of specific heats of the gas and is equal to 1.4 for air. The solution was represented by a third, fourth, fifth and sixth order polynomial. First the polynomials were defined by a collocation projection of the solution values at four, five, six and ten GLL points for each element. Then the

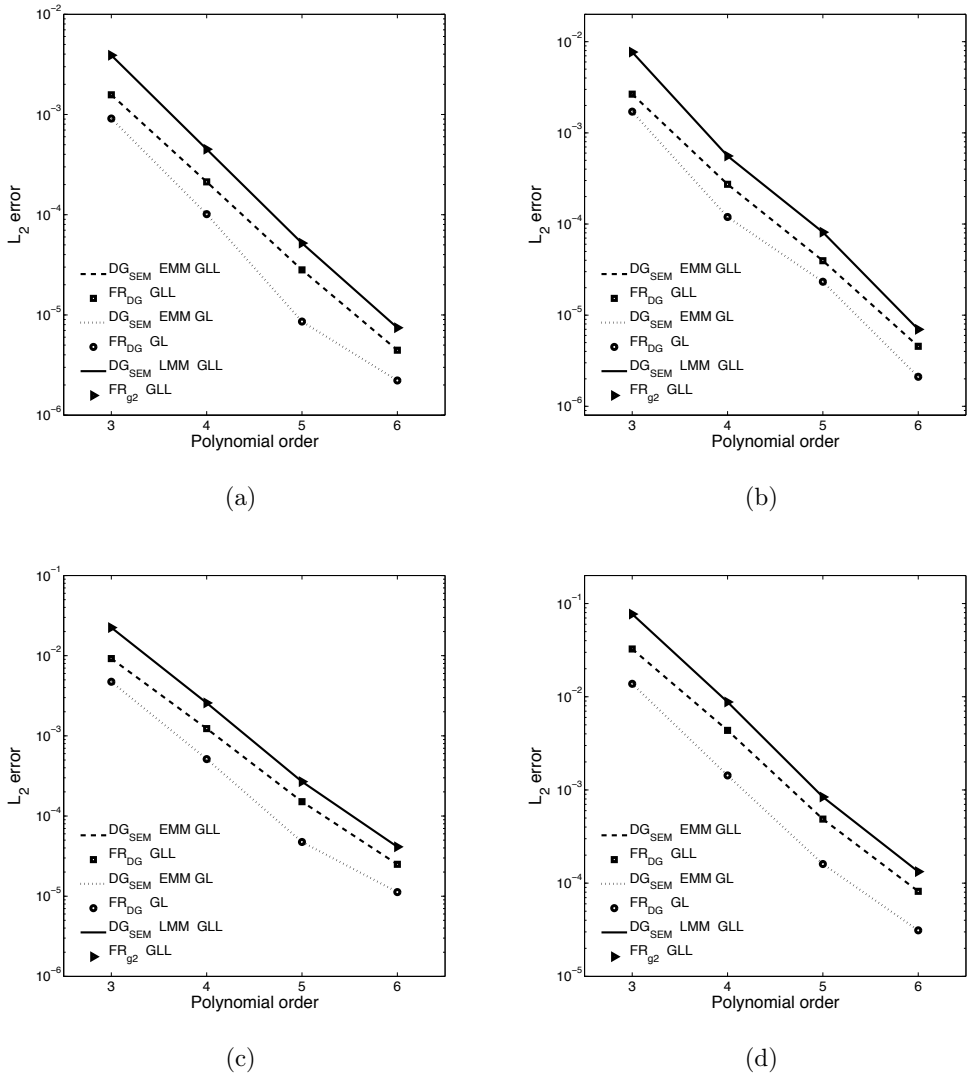


Figure 2.4: L_2 error vs polynomial order P on the regular grid at the final time $T = 2s$. Figures (a)-(d) depict ρ, ρu, ρv and E respectively.

calculations were repeated using GL points. In all the cases a collocation quadrature rule was used for the inner product of the advection term of the DG scheme. With GLL both the cases of exact mass matrix (EMM) and lumped mass matrix (LMM) were considered.

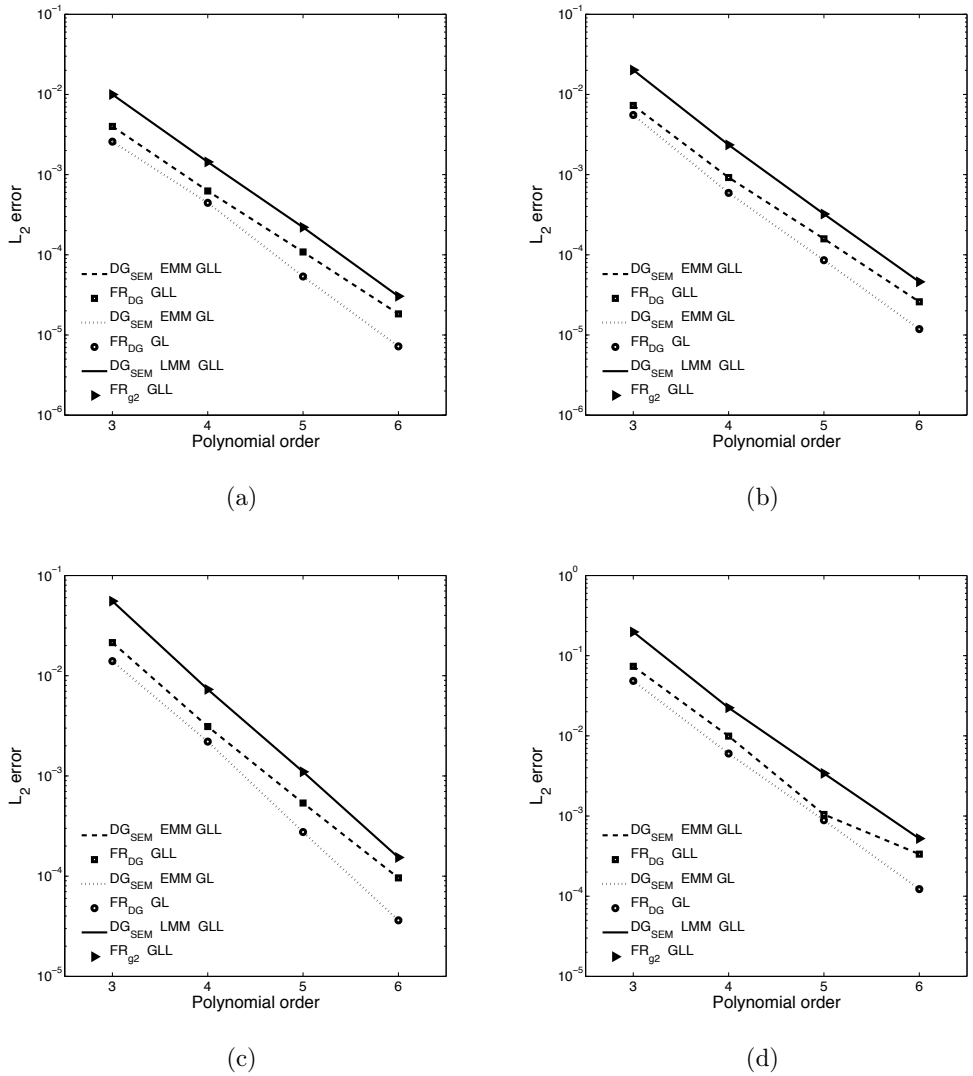


Figure 2.5: L_2 error vs polynomial order P on the deformed grid at the final time $T = 2s$. Figures (a)-(d) depict ρ, ρu, ρv and E respectively.

Periodic conditions were applied at the boundary of the grids and an isentropic vortex in a free-stream flow (in the positive y direction) was prescribed within $\mathbf{\Omega}$

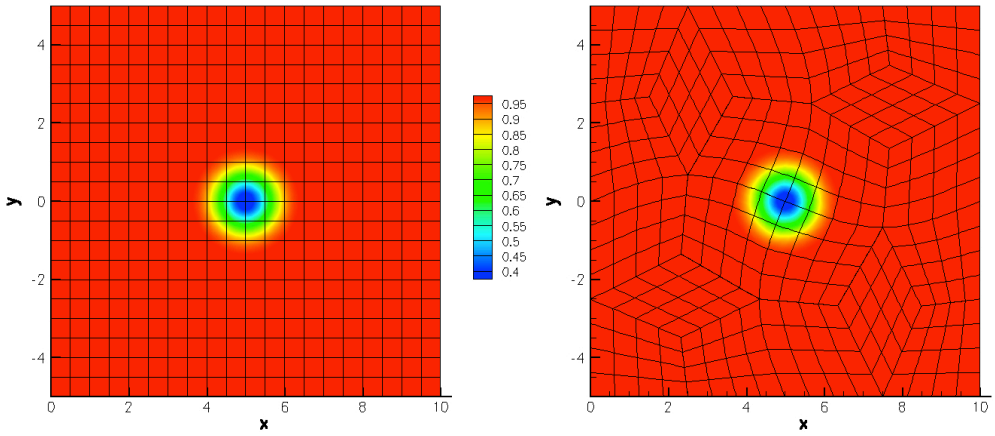


Figure 2.6: Solution at the final time $T = 2s$ for the simulation of the compressible Euler equations. The density contour is visualized on both the regular (left) and deformed (right) grids using seven GLL points.

at $t = 0$:

$$\begin{aligned} \rho &= \left(1 - \frac{\beta^2(\gamma-1)e^{2f}}{16\gamma\pi^2}\right)^{\frac{1}{\gamma-1}}, \\ u &= \left(u_0 - \frac{\beta e^f (y-y_0)}{2\pi R}\right), \\ v &= \left(v_0 + \frac{\beta e^f (x-x_0)}{2\pi R}\right), \\ E &= \frac{\rho^\gamma}{\gamma-1} + \frac{1}{2}\rho\left(u^2+v^2\right), \end{aligned} \tag{2.29}$$

where

$$f = 1 - r^2,$$
$$r = \sqrt{(x-x_0)^2 + (y-y_0)^2},$$

with $(u_0, v_0) = (0, 5)$, $(x_0, y_0) = (5, 0)$, $R = 1$, $\beta = 5$ and $\gamma = 1.4$. An explicit Runge-Kutta time integration scheme of 4 stages is used to discretise the equations in time. The final time step was $T = 2s$, therefore the exact solution at the final time is equal to the initial of Eq. 2.29. The time-step was chosen as $\Delta t_{max}/100$, where Δt_{max} is the maximum time-step that allowed the simulation to remain stable. This choice for the time-step was made to ensure that the error from the time-stepping scheme was negligible compared to the spatial discretization.
As predicted by the mathematical derivation, figures 2.4 demonstrates that the

appropriate DG and FR schemes exhibit identical numerical errors at the final time on a regular grid. As in the linear case Figure 2.5 also shows that the results match on the deformed grid because the approximation of the geometric factors is performed in the same way for both approaches. In particular we additionally note that the error produced with a collocation projection at GL points is minor when compared to GLL points. This is related to the property of GL points for minimising the aliasing errors due to nonlinear flux functions, as shown in Jameson et al. (2012) and Castonguay et al. (2011a). For both regular and deformed grids the results are identical up to machine precision. Figure 2.6 shows the solution at the final time on the regular grid and on the deformed grid.

2.3.3 Comparison of computational time

Figures 2.7(a), 2.7(b) show a comparison between CPU processing time of different schemes for two different variants of our codebase, where figure 2.7(a) is an earlier version. In these figures we examine the average CPU time required for a time-step of a linear advection problem on the mesh of figure 2.1(a) using the $\mathrm{FR_{g2}}$ scheme, the $\mathrm{DG_{SEM}}$-GLL with lumped mass matrix and the nodal $\mathrm{DG_{Q>P}}$ scheme with exact mass matrix, where a quadrature rule with $Q > P$ was used for the inner product of the advection term. All of these schemes have solution values which are collocated at Gauss-Lobatto-Legendre points.

The solid and dashed lines represent two mathematically identical schemes and so necessarily one might expect the same trend of CPU processing time as in figure 2.7(b). In figure 2.7(a) the lines with the same trend are the dashed and the dotted since the implementation of $\mathrm{DG_{SEM}}$-GLL is not exploiting the possibility of saving time by using the diagonal nature of the mass matrix, but performs calculation as if the mass matrix were full, similar to the nodal $\mathrm{DG_{Q>P}}$ scheme. From the point of view of operations the schemes should all scale as P^3 and whilst a non-linear trend in polynomial order is obvious in the DG timing is theoretically not clear for the FR scheme. In figure 2.7(a) we note the implementation based on the FR approach has a much better scaling with polynomial order suggesting that the implementation in the standard element space has led to a design which is more suited to the memory bandwidth limited nature of current CPUs. Clearly the same implementation is possible with the DG-SEM scheme but potentially greater insight is required by the developer due to the local element space construction of this scheme. However, interestingly for low-order expansions the implementation of the $\mathrm{DG_{SEM}}$-GLL scheme performs better suggesting how the collocation nature of this scheme helps recover some of the inefficiencies that are likely to arise in the DG approach where integration over volume and trace spaces are necessary requiring addition computational cost and more complicated memory access. In figure 2.7(b) the $\mathrm{DG_{SEM}}$-GLL and $\mathrm{FR_{DG}}$ have the same trend as expected since the

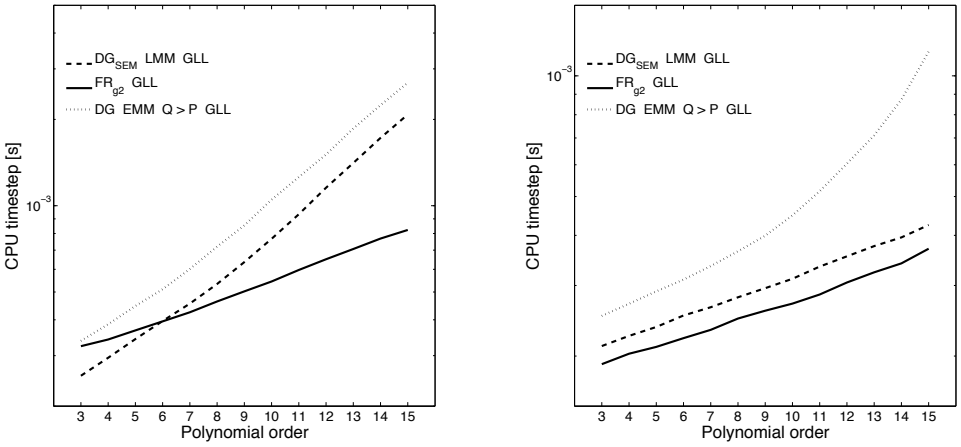


Figure 2.7: The CPU time required to calculate an average time-step of a scalar advection problem is shown as a function of polynomial order for three different schemes: DG_{SEM}-GLL, $DG_{Q>P}$ and FR_{g2}. The solution values are at Gauss-Lobatto-Legendre points in all cases. On the left the CPU time with the implementation not exploiting the diagonal mass matrix of DG_{SEM}-GLL. On the right the CPU time with the implementation exploiting the diagonal mass matrix but where additional changes have been made to the discrete operators on the trace space.

implementation of DG_{SEM}-GLL is exploiting the possibility of saving time deriving from the diagonal mass matrix. The only difference between the two schemes is a shift in the CPU time. Since no attempt has been made to explicitly optimise the code, this difference is likely to be due to the different data structures used in the DG approach and in the FR. In particular, as already mentioned, in the DG implementation contained in Nektar++, all the calculations are performed in the local element space (x, y), whilst, for the FR scheme, all the calculations are in the standard elements space (ξ_1, ξ_2). Further, in neither approach have we attempted to optimise the algorithm by calculating core operations over multiple elements and this type of operations may also perform differently in each method. However, we note that, in general, the efficiency of a solver is obviously highly dependent not only on the choice of scheme but also on its implementation, so that whilst some methods may be claimed as extremely efficient it is only through the consideration

of detailed implementation issues that this claim could be corroborated.

2.4 Conclusions

The connections between three nodal versions of tensor product discontinuous Galerkin spectral element approximations and two types of flux reconstruction schemes were shown both through a mathematical derivation and a numerical study. It was demonstrated that, for an advection equation, the $\mathrm{DG_{SEM}}$ scheme with an exact mass matrix and the $\mathrm{FR_{DG}}$ scheme are equivalent on a quadrilateral grid. It was demonstrated that the $\mathrm{DG_{SEM}}$-GLL scheme with a lumped mass matrix and the $\mathrm{FR_{g2}}$ scheme are equivalent for the case of a 1D conservation law. These equivalences are valid for both the cases of linear and nonlinear fluxes. The $\mathrm{DG_{Q>P}}$ scheme with an exact mass matrix recovers the $\mathrm{FR_{DG}}$ scheme only when the flux function is linear. In fact, when the flux is nonlinear, the aliasing issues due to the nonlinearities are minimised by the higher precision of the quadrature rule of the $\mathrm{DG_{Q>P}}$ scheme and this implies that there are differences with the $\mathrm{FR_{DG}}$ scheme. These results can be extended to higher dimensional grids. Additionally, the connections established here are always valid on regular grids regardless of whether the flux function is linear or nonlinear. However in the case of a deformed mesh, the equivalence of the schemes depends on the approximation of the geometric factors.

Note that in this chapter we always considered a collocated version of the FR schemes and we showed in section 2.3.2 that this leads to aliasing errors which are identical (up to machine precision) to its DG counterpart. This suggests that if we treat the nonlinear flux of the FR and DG formulation with an identical technique, the equivalences obtained can be extended to non collocated schemes as well. For instance, using an over-integration, we can have a $\mathrm{FR_{DG}}$ scheme with $Q > P$ which is identical to the $\mathrm{DG_{Q>P}}$ scheme.

In chapter 4, we will better characterise the sources of aliasing in discontinuous spectral/hp element methods and we will detail the dealiasing strategies based on the concept of consistent integration of the nonlinearities which can be used for addressing the aliasing errors arising in the DG and FR approaches. In the next chapter we will introduce the two nonlinear systems of compressible aerodynamic equations that have been widely used in this thesis: the Euler and Navier-Stokes equations.

Chapter 3

Governing equations

After having presented, in the previous chapters, the numerical schemes applied in the study and their connections, in this chapter, we present the equations governing inviscid and viscous compressible flows, namely the Euler and Navier-Stokes equations. We also introduce briefly their numerical discretisation by means of the DG and the FR approaches as well as the implementation strategy adopted for the boundary conditions. In the final part of this chapter we show some results to verify the compressible flow solver, whose implementation has been an important part of this thesis work.

3.1 Euler equations

The Euler equations are a first-order hyperbolic system of equations and they describe an inviscid and adiabatic flow. They are a subset of the compressible Navier-Stokes equations and can be used to describe flows where the viscosity effects can be neglected - for example the free-stream region of high-Reynolds number flows. On a physical domain Ω and in a Cartesian frame of reference (x_1, x_2, x_3) they can be expressed as a hyperbolic conservation law in the form:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}_{i,1}}{\partial x_1} + \frac{\partial \boldsymbol{f}_{i,2}}{\partial x_2} + \frac{\partial \boldsymbol{f}_{i,3}}{\partial x_3} = 0, \tag{3.1}$$

where $\boldsymbol{u}$ is the vector of the conserved variables:

$$\boldsymbol{u} = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{array} \right\} \tag{3.2}$$

and $\boldsymbol{f}_{i,1} = \boldsymbol{f}_{i,1}(\boldsymbol{u})$, $\boldsymbol{f}_{i,2} = \boldsymbol{f}_{i,2}(\boldsymbol{u})$ and $\boldsymbol{f}_{i,3} = \boldsymbol{f}_{i,3}(\boldsymbol{u})$ are the vectors of the inviscid fluxes

$$\boldsymbol{f}_{i,1} = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ \rho uw \\ u(E+p) \end{Bmatrix}, \quad \boldsymbol{f}_{i,2} = \begin{Bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ \rho vw \\ v(E+p) \end{Bmatrix}, \quad \boldsymbol{f}_{i,3} = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ p + \rho w^2 \\ w(E+p) \end{Bmatrix}, \tag{3.3}$$

where ρ is the density, u, v and w are the velocity components in x_1, x_2 and x_3 directions, p is the pressure and E is the total energy. In this work we considered a perfect gas law for which the pressure is related to the total energy by the following expression

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2), \tag{3.4}$$

where γ is the ratio of specific heats.

3.1.1 Numerical discretisation

Following the steps presented in chapter 1 for the discretisation of a first order problem (section 1.4), the numerical discretisation of the Euler equations (Eq. 3.1) by means of the DG method and the FR approach involves dividing the computational domain Ω into N non-overlapping subdomains $\Omega \longrightarrow \Omega_n$, and, on each subdomain representing the solution by a polynomial of degree P, $\boldsymbol{u}_n \in \mathbb{P}^{\mathbb{P}}$. As previously described, the solution over the entire domain is allowed to be discontinuous between the elements and the coupling between the elements is obtained through the boundary terms.

In the following the formulations of the DG method and of the FR approach are reported by using a compact form, where we use the index ℓ to indicate the $\ell - th$ equation of the system in Eq. 3.1.

DG scheme :

The matrix form of the DG scheme applied to the three dimensional Euler equations is:

$$\mathbf{M}\frac{d\widehat{\mathbf{u}}}{dt} + \mathbf{S}_{x_1}\widehat{\mathbf{f}}^D_{i,1} + \mathbf{S}_{x_2}\widehat{\mathbf{f}}^D_{i,2} + \mathbf{S}_{x_3}\widehat{\mathbf{f}}^D_{i,3} + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0 \quad \text{for } \ell = 1, ..., 5, \tag{3.5}$$

where $\mathbf{M}$ is the elemental mass matrix, $\mathbf{S}_{x_1}$, $\mathbf{S}_{x_2}$ and $\mathbf{S}_{x_3}$ are the elemental stiffness matrices defined in Eq. 1.91, $\widehat{\mathbf{u}}$ is the transformed vector of the ℓ-th variable defined on the solution points of a given element (note that we changed notation here; $\mathbf{u}$ is a vector containing the values of a conserved variable on the solution points of a given element, while $\boldsymbol{u}$ is the vector

of the conserved variables), $\widehat{\mathbf{f}}^D_{i,1}$, $\widehat{\mathbf{f}}^D_{i,2}$ and $\widehat{\mathbf{f}}^D_{i,3}$ are the elemental transformed nonlinear fluxes defined on the solution points of a given element (note again the change of notation) and finally $\widehat{\mathbf{b}}^{\mathrm{DG}}$ is the boundary term (also defined on the solution points of a given element) which, in the physical space, is represented by the following surface integral:

$$\widehat{\mathbf{b}}^{\mathrm{DG}}\Big|_{\ell} = \iint_{\partial\Omega_s} \ell(\boldsymbol{\Theta}(\boldsymbol{\xi}))\Big(\widehat{\mathcal{H}^{\delta I}_n \cdot \mathbf{n}}\Big)_{\ell} J_{2D} d\widehat{s} \quad \text{for } \ell = 1, ..., 5, \tag{3.6}$$

with $\ell(\boldsymbol{\Theta}(\boldsymbol{\xi}))$ being a three-dimensional polynomial expansion basis, either modal or nodal, $\mathbf{n}$ being the outward pointing normal with respect to a given element Ω_n, and $\mathcal{H}^{\delta I}_i$ being the tensor of the inviscid fluxes $\mathcal{H}^{\delta I}_i = [\mathbf{f}^{\delta I}_{i,1}, \mathbf{f}^{\delta I}_{i,2}, \mathbf{f}^{\delta I}_{i,3}]$ at the interface between two adjacent elements.

FR approach :

The matrix form of the FR approach applied to the three dimensional Euler equations is:

$$\frac{d\widehat{\mathbf{u}}}{dt} + \mathbf{D}_{x_1}\widehat{\mathbf{f}}^D_{i,1} + \mathbf{D}_{x_2}\widehat{\mathbf{f}}^D_{i,2} + \mathbf{D}_{x_3}\widehat{\mathbf{f}}^D_{i,3} + \widehat{\mathbf{b}}^{\mathrm{FR}} = 0 \quad \text{for } \ell = 1, ..., 5, \tag{3.7}$$

where $\mathbf{D}_{x_1}$, $\mathbf{D}_{x_2}$ and $\mathbf{D}_{x_3}$ are the differentiation matrices with respect to x_1, x_2 and x_3, respectively and defined in Eq. 1.86, $\widehat{\mathbf{f}}^D_{i,1}$, $\widehat{\mathbf{f}}^D_{i,2}$ and $\widehat{\mathbf{f}}^D_{i,3}$ are the elemental transformed nonlinear fluxes of the Euler equations and $\widehat{\mathbf{b}}^{\mathrm{FR}}$ is the boundary term:

$$\widehat{\mathbf{b}}^{\mathrm{FR}}\Big|_{\ell} = \Big(\widehat{\mathcal{H}^{\delta C}_i \cdot \mathbf{n}}\Big)_{\ell} \boldsymbol{\Psi}' = \left[\Big(\widehat{\mathcal{H}^{\delta I}_i \cdot \mathbf{n}}\Big)_{\ell} - \Big(\widehat{\mathcal{H}^{\delta DI}_i \cdot \mathbf{n}}\Big)_{\ell}\right] \boldsymbol{\Psi}' \quad \text{for } \ell = 1, ..., 5, \tag{3.8}$$

with $\boldsymbol{\Psi}'$ being the derivative of a suitable correction function, $\mathcal{H}^{\delta I}_i$ being the tensor of the inviscid fluxes $\mathcal{H}^{\delta I}_i = [\mathbf{f}^{\delta I}_{i,1}, \mathbf{f}^{\delta I}_{i,2}, \mathbf{f}^{\delta I}_{i,3}]$ at the interface between two adjacent elements and $\mathcal{H}^{\delta DI}_i$ being again the tensor of the inviscid fluxes evaluated at the boundary of a given element Ω_n.

As anticipated in chapter 1, one of the crucial features of both the methods and more in general of this class of schemes is the way the boundary term $\widetilde{\mathcal{H}}^{\delta I}_i$ is calculated. For the Euler equations and for the advection term of the compressible Navier-Stokes equations it is common to use either exact or approximated Riemann solvers. They both calculate an intermediate numerical flux at the interface between two adjacent elements by using some characteristic information coming from an eigenvalue analysis of the equations and may contain values coming from the left (+) or right (-) state of a given interface (see Fig. 1.4):

$$\mathcal{H}^{\delta I}_i\Big|_{\ell} = \mathcal{H}^{\delta I}_i(\mathbf{u}^{\delta}_+, \mathbf{u}^{\delta}_-)\Big|_{\ell} \quad \text{for } \ell = 1, ..., 5. \tag{3.9}$$

When adopting a weak strategy for implementing the boundary conditions, the boundary term $\mathcal{H}^{\delta I}$ is also responsible for the boundary conditions to be correctly transferred into the domain via a ghost state. In this thesis we used this approach and the implementations of the boundary conditions are shown in section 3.4.

In addition, during the implementation of the compressible flow solver in Nektar++ different Riemann solvers have been considered, including an exact and various approximated Riemann solvers. In section 3.3.1 we describe three of the Riemann solvers implemented in Nektar++: exact, HLL and HLLC.

3.2 Compressible Navier-Stokes equations

Differently from the Euler equations, the compressible Navier-Stokes equations include the effects of fluid viscosity and heat conduction and are consequently composed of an inviscid and a viscous tensor. They are a second-order system of equations where the viscous tensor depend not only on the conserved variables but also, indirectly, on their first order derivatives. The second order partial differential equations for the three-dimensional case can be written as:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}_1}{\partial x} + \frac{\partial \boldsymbol{f}_2}{\partial y} + \frac{\partial \boldsymbol{f}_3}{\partial z} = 0, \tag{3.10}$$

where $\boldsymbol{u}$ is the vector of the conserved variables, $\boldsymbol{f}_1 = \boldsymbol{f}_1(\boldsymbol{u}, \nabla_{\boldsymbol{x}}\boldsymbol{u})$, $\boldsymbol{f}_2 = \boldsymbol{f}_2(\boldsymbol{u}, \nabla_{\boldsymbol{x}}\boldsymbol{u})$ and $\boldsymbol{f}_3 = \boldsymbol{f}_3(\boldsymbol{u}, \nabla_{\boldsymbol{x}}\boldsymbol{u})$ are the vectors of the fluxes which can also be written as:

$$\begin{aligned} \boldsymbol{f}_1 &= \boldsymbol{f}_{i,3} - \boldsymbol{f}_{v,1}, \\ \boldsymbol{f}_2 &= \boldsymbol{f}_{i,2} - \boldsymbol{f}_{v,1}, \\ \boldsymbol{f}_3 &= \boldsymbol{f}_{i,3} - \boldsymbol{f}_{v,3} \end{aligned} \tag{3.11}$$

where $\boldsymbol{f}_{i,1}$, $\boldsymbol{f}_{i,2}$ and $\boldsymbol{f}_{i,3}$ are the inviscid fluxes of Eq. (3.3) and $\boldsymbol{f}_{v,1}$, $\boldsymbol{f}_{v,2}$ and $\boldsymbol{f}_{v,3}$ are the viscous fluxes which take the following form:

$$\mathbf{f}_{v,1} = \begin{Bmatrix} 0 \\ \tau_{x_1x_1} \\ \tau_{x_2x_1} \\ \tau_{x_3x_1} \\ u\tau_{x_1x_1} + v\tau_{x_2x_1} + w\tau_{x_3x_1} + kT_{x_1} \end{Bmatrix},$$

$$\mathbf{f}_{v,2} = \begin{Bmatrix} 0 \\ \tau_{x_1x_2} \\ \tau_{x_2x_2} \\ \tau_{x_3x_2} \\ u\tau_{x_1x_2} + v\tau_{x_2x_2} + w\tau_{x_3x_2} + kT_{x_2} \end{Bmatrix}, \tag{3.12}$$

$$\mathbf{f}_{v,3} = \left\{ \begin{array}{c} 0 \\ \tau_{x_1x_3} \\ \tau_{x_2x_3} \\ \tau_{x_3x_3} \\ u\tau_{x_1x_3} + v\tau_{x_2x_3} + w\tau_{x_3x_3} + kT_{x_3} \end{array} \right\},$$

where $\tau_{x_1x_1}$, $\tau_{x_1x_2}$, $\tau_{x_1x_3}$, $\tau_{x_2x_1}$, $\tau_{x_2x_2}$, $\tau_{x_2x_3}$, $\tau_{x_3x_1}$, $\tau_{x_3x_2}$, and $\tau_{x_3x_3}$ are the components of the stress tensor:

$$\tau_{x_1x_1} = 2\mu\frac{\partial u}{\partial x_1} + \lambda\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right) = 2\mu\left[\frac{\partial u}{\partial x_1} - \frac{1}{3}\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right)\right],$$

$$\tau_{x_2x_2} = 2\mu\frac{\partial v}{\partial x_2} + \lambda\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right) = 2\mu\left[\frac{\partial v}{\partial x_2} - \frac{1}{3}\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right)\right],$$

$$\tau_{x_3x_3} = 2\mu\frac{\partial w}{\partial x_3} + \lambda\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right) = 2\mu\left[\frac{\partial w}{\partial x_3} - \frac{1}{3}\left(\frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2} + \frac{\partial w}{\partial x_3}\right)\right],$$

$$\tau_{x_1x_2} = \tau_{x_2x_1} = \mu\left(\frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2}\right),$$

$$\tau_{x_2x_3} = \tau_{x_3x_2} = \mu\left(\frac{\partial w}{\partial x_2} + \frac{\partial v}{\partial x_3}\right),$$

$$\tau_{x_3x_1} = \tau_{x_1x_3} = \mu\left(\frac{\partial u}{\partial x_3} + \frac{\partial w}{\partial x_1}\right),$$

where μ is the dynamic viscosity calculated using the Sutherland's law:

$$\mu = \mu_\infty \left(\frac{T}{T_\infty}\right)^{3/2} \frac{T_\infty + 110.4}{T + 110.4}, \tag{3.13}$$

λ is the second coefficient of viscosity[1] and k is the thermal conductivity.

Numerical discretisation

The numerical discretisation of the compressible Navier-Stokes equations for the inviscid fluxes is identical to that of the Euler equations (the inviscid fluxes of the Euler equations and of the compressible Navier-Stokes are identical) while the additional viscous fluxes are treated in a different manner. Specifically, we use the approach presented for second-order problems in section 1.5. We split Eq. 3.10 into two first order equations: an auxiliary equation, through which we calculate the derivatives of the conserved variables $\boldsymbol{u}$, and the principal equation, which is constituted by the original problem (see Eq. 1.110). The procedure is then

[1] In the Navier-Stokes equations for all the simulations carried out it was used the Stokes hypothesis $\lambda = -2/3$.

divided in two steps. The first involves the calculation of the spatial derivatives of the auxiliary variables $\boldsymbol{u}_{aux} = [u,\, v,\, w,\, T]^T$, necessary to construct the viscous flux. The second step consists in the computation of the viscous fluxes and in the solution of the principal system following either the DG method or the FR approach presented in section 1.5.

For both, DG and FR, the numerical fluxes at the interface between two adjacent elements are calculated using the local discontinuous Galerkin (LDG) method also presented in section 1.5 and here briefly reported for the reader's convenience:

$$\begin{aligned} \boldsymbol{u}_{aux}^{\delta I} &= \{\boldsymbol{u}_{aux}^{\delta D}\} \pm \boldsymbol{\beta} \cdot [\boldsymbol{u}_{aux}^{\delta D}] \\ \mathcal{H}_v^{\delta I} &= \{\mathcal{H}_v^{\delta D}\} \mp \boldsymbol{\beta} \cdot [\mathcal{H}_v^{\delta D}] - \tau [\boldsymbol{u}_{aux}^{\delta D}] \end{aligned} \tag{3.14}$$

where:

$$\{\mathbf{g}\} = \frac{\mathbf{g}_+ + \mathbf{g}_-}{2}, \qquad [\mathbf{g}] = \frac{\mathbf{g}_+ \boldsymbol{n}_+ + \mathbf{g}_- \boldsymbol{n}_-}{2}, \qquad \boldsymbol{\beta} = \frac{1}{2}\boldsymbol{n}_+ \tag{3.15}$$

with the quantities $\mathbf{g}_-$ and $\mathbf{g}_+$ being the variable $\mathbf{g}$ on the right and on the left side of the interface between two elements and with $\boldsymbol{n}_+$ and $\boldsymbol{n}_-$ being the respective normals (for additional details the interested reader can refer to Cockburn and Shu (1998a)).

As in the case of the inviscid flux, where the boundary term (i.e. the interface flux) transfers the boundary conditions into the domain, for the viscous flux the interface fluxes of the auxiliary variables $\boldsymbol{u}_{aux}^{\delta I}$ and of the primitive system $\mathcal{H}^{\delta I}$ are responsible for the boundary conditions to be correctly transferred into the domain, when a weak approach is adopted.

3.3 The Riemann problem

A conservation law together with piecewise constant data having a single or multiple discontinuity is known as the Riemann problem. In order to propagate the information across the element at each interface a Riemann problem must be solved. The Riemann problem for the one-dimensional time-dependent Euler equations is the Initial Value Problem (IVP) for the conservation laws:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} = 0\,, \tag{3.16}$$

where:

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \qquad \boldsymbol{f} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \tag{3.17}$$

with initial condition:

$$\boldsymbol{u}(x,0)\begin{cases}\boldsymbol{u}_+ & \text{if } x<0\\ \boldsymbol{u}_- & \text{if } x\geq 0\,.\end{cases} \tag{3.18}$$

The domain of interest in the $x-t$ plane are points (x, t) with $-\infty < x < \infty$ and $t > 0$. In practice one lets x vary in a finite interval $[x_L, x_R]$ around the point $x = 0$. The Riemann problem of Eq. 3.17 and 3.18 is the simplest, non-trivial, IVP for Eq. 3.16. The Riemann problem consists of just two constant states, which in terms of primitive variables are $(\rho_+, u_+, p_+)^T$ to the left of $x = 0$ and $(\rho_-, u_-, p_-)^T$ to the right of $x = 0$, separated by a discontinuity at $x = 0$. Physically, in the context of the Euler equations, the Riemann problem is a slight generalisation of the so called shock-tube problem: two stationary gases ($u_+ = u_- = 0$) in a tube are separated by a diaphragm. In case of rupture of the diaphragm a nearly centred wave system is generated which typically consists of a rarefaction wave, a contact discontinuity and a shock wave. This physical problem is reasonably well approximated by solving the shock-tube problem for the Euler equations. In the Riemann problem the particle speeds u_+ and u_- are allowed to be non-zero, but the structure of the solution is the same as that of the shock-tube problem.

The solution of the Riemann problem of Eq. 3.16 can be obtained through a Riemann solver either exact or approximate. The Riemann solver calculates the solution pattern at a given interface and, in particular, the solution for $x/t = 0$. This can then be used to calculate the associated numerical flux between the elements in a Godunov-type approach (Godunov (1959)). In the next section the exact and two approximate Riemann solvers are described in the general case of the three-dimensional Euler equations.

3.3.1 One-dimensional Riemann solvers

As already mentioned before, the one-dimensional Riemann solvers allow the various elements of the DG/FR approach (applied to the Euler equations and to the advection term of the compressible Navier-Stokes equations) to communicate and propagate the information. They can also be responsible for the boundary conditions to be correctly transferred into the domain via a ghost state if these are applied with a weak approach.

In the following we describe some relevant Riemann solvers implemented in Nektar++ and used in this thesis work. A good reference on this topic is the the book by Toro (2009). In the next section we will adopt a similar nomenclature to that used in the book. In particular, we consider the three-dimensional Euler equations and their associated Riemann problem (Fig. 3.1) which is composed of two nonlinear waves (dotted lines) associated with the maximum and minimum

eigenvalue of the system (shock waves or rarefaction waves), and by three contact waves (dashed line) associated with the three coincident eigenvalues of the problem.

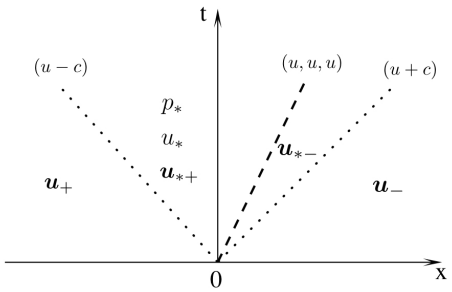


Figure 3.1: Riemann problem for the Euler equations with ideal gas law.

The region between the two nonlinear waves is denoted as the star region in the following and the left and right states are denoted with the subscripts "+" and "−", respectively. The nature of the solution strictly depends on the equation of state. In all the cases considered in this work, we used an ideal gas law.

When using a Riemann solver at a given interface it is necessary to rotate the two-/three dimensional problem into a one-dimensional problem in the normal direction with respect to the given interface. By exploiting the rotational invariance of the Euler equations it is possible to write:

$$[\boldsymbol{f}_{i,1} \quad \boldsymbol{f}_{i,2} \quad \boldsymbol{f}_{i,2}] \begin{bmatrix} \cos(\theta_{x_2})\cos(\theta_{x_3}) \\ \cos(\theta_{x_2})\sin(\theta_{x_3}) \\ \sin(\theta_{x_2}) \end{bmatrix} = \mathbf{R}^{-1}\boldsymbol{f}(\mathbf{R}\boldsymbol{u}), \tag{3.19}$$

where θ_{x_2}, θ_{x_3} are the angles which link the normal frame of reference (with respect to a given interface) to the Cartesian frame of reference, $\mathbf{R}$ is the following rotation matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos(\theta_{x_2})\cos(\theta_{x_3}) & \cos(\theta_{x_2})\sin(\theta_{x_3}) & \sin(\theta_{x_2}) & 0 \\ 0 & -\sin(\theta_{x_3}) & \cos(\theta_{x_3}) & 0 & 0 \\ 0 & -\sin(\theta_{x_2})\cos(\theta_{x_3}) & -\sin(\theta_{x_2})\sin(\theta_{x_3}) & \cos(\theta_{x_2}) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

and $\mathbf{R}^{-1}$ is its inverse. From a practical implementative point of view it is common to first apply the rotation matrix $\mathbf{R}$ to the variables $\boldsymbol{u}$ in order to calculate the augmented one-dimensional flux:

$$\boldsymbol{f}(\boldsymbol{u}_n) = \boldsymbol{f}(\mathbf{R}\boldsymbol{u}). \tag{3.21}$$

Successively one rotates the augmented one-dimensional flux back to the cartesian frame of reference by multiplying by the inverse of the rotation matrix using Eq. (3.19). It is important to note here that the final flux obtained is projected into the normal direction with respect to the interface.

The intermediate step between the two rotations is to calculate the augmented one-dimensional flux. At this stage different Riemann solvers can be used depending on the performance (dissipation, dispersion properties, stability, computational costs) one desires to obtain.

In the the following we briefly describe three Riemann solvers which have been implemented into Nektar++: an exact Riemann solver and two approximated Riemann solvers, HLL and HLLC.

Exact :

The Exact Riemann solver uses the exact solution of the Riemann problem for the Euler equations. The first exact Riemann solver for the Euler equations can be found in Godunov (1959) and was further developed in Godunov (1976). Examples of exact Riemann solvers for ideal gases can be found in Gottlieb and Groth (1988) and Toro (1989). It is well-known that there are ten possible wave patterns allowed at an interface and if we use a local frame of reference with respect to the interface, this reduces in solving the Riemann problem in the space-time plane for x/t = 0. The first step to perform is therefore to identify which pattern has arisen at a given interface and successively calculate the boundary state $\boldsymbol{u}_n$ (and consequently the augmented one-dimensional flux $\boldsymbol{f}(\boldsymbol{u}_n)$) for $x/t = 0$.

In particular we have two macro cases each of those in turn divided into five sub-cases:

- Positive particle speed in the region between the two nonlinear waves (star region), $u_* \geq 0$
  1. Left rarefaction - $\boldsymbol{u}_+$: if $p_* \leq p_+$ and $u_+ \geq c_+$ we have a left rarefaction wave with a left state which is supersonic and the state for x/t = 0 is the left state. Consequently the boundary fluxes in the normal frame of reference with respect to a given interface are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ u_+ \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ u_+^2 + p_+ \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ u_+ v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ u_+ w_+ \\ \mathcal{H}_{E}^{\delta I} = u_+ \left[\frac{p_+}{\gamma - 1} + \frac{1}{2} \rho_+ (u_+^2 + v_+^2 + w_+^2) + p_+ \right] \end{cases} \tag{3.22}$$

  2. Left rarefaction - $\boldsymbol{u}_{*+}$: if $p_* \leq p_+$, $u_+ < c_+$, and $u_* < c_+ \left(\frac{p_*}{p_+}\right)^{\frac{\gamma-1}{2\gamma}}$ we have a left rarefaction wave with a left state which is subsonic and

the state for x/t = 0 is the star left state. The boundary fluxes in this case are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ \left(\frac{p_*}{p_+}\right)^{\frac{1}{\gamma}} u_* \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ \left(\frac{p_*}{p_+}\right)^{\frac{1}{\gamma}} u_*^2 + p_* \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ \left(\frac{p_*}{p_+}\right)^{\frac{1}{\gamma}} u_* v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ \left(\frac{p_*}{p_+}\right)^{\frac{1}{\gamma}} u_* w_+ \\ \mathcal{H}_{E}^{\delta I} = u_* \left[\frac{p_*}{\gamma-1} + \frac{1}{2}\rho_+ \left(\frac{p_*}{p_+}\right)^{\frac{1}{\gamma}} (u_*^2 + v_+^2 + w_+^2) + p_*\right] \end{cases} \tag{3.23}$$

3. Left rarefaction - $\boldsymbol{u}_{+Fan}$: if $p_* \leq p_+$, $u_+ < c_+$ and $u_* \geq c_+ \left(\frac{p_*}{p_+}\right)^{\frac{\gamma-1}{2\gamma}}$ we have a left rarefaction wave with a left state which is subsonic and the state for x/t = 0 is the rarefaction fan. The boundary fluxes in this case are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2}{\gamma-1}} \left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right) \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2}{\gamma-1}} \left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right)^2 + \\ \qquad + p_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2\gamma}{\gamma-1}} \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2}{\gamma-1}} \left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right) v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2}{\gamma-1}} \left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right) w_+ \\ \mathcal{H}_{E}^{\delta I} = \left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right) \left\{ \frac{p_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2\gamma}{\gamma-1}}}{\gamma-1} + \right. \\ \qquad + \frac{1}{2}\rho_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2}{\gamma-1}} \left[\left(\frac{2c_+}{\gamma+1} + \frac{\gamma-1}{\gamma+1} u_+\right)^2 + \\ \qquad \left. + v_+^2 + w_+^2\right] + p_+ \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}\left(\frac{u_+}{c_+}\right)\right]^{\frac{2\gamma}{\gamma-1}} \right\} \end{cases} \tag{3.24}$$

4. Left shock - $\mathbf{q}_+$: if $p_* > p_+$ and $u_+ \geq c_+ \left(\frac{\gamma+1}{2\gamma}\frac{p_*}{p_+} + \frac{\gamma-1}{2\gamma}\right)^{\frac{1}{2}}$, we have a left shock with the state for x/t = 0 being the left state. The associated boundary fluxes are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ u_+ \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ u_+^2 + p_+ \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ u_+ v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ u_+ w_+ \\ \mathcal{H}_{E}^{\delta I} = u_+ \left[\frac{p_+}{\gamma-1} + \frac{1}{2}\rho_L (u_+^2 + v_+^2 + w_+^2) + p_+\right] \end{cases} \tag{3.25}$$

5. Left shock - $\boldsymbol{u}_{*+}$: if $p_* > p_+$ and $u_+ < c_+ \left(\frac{\gamma+1}{2\gamma}\frac{p_*}{p_+} + \frac{\gamma-1}{2\gamma}\right)^{\frac{1}{2}}$, we have a left shock with the state for x/t = 0 being the left star state. The

associated boundary fluxes are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ \left(\frac{\frac{p_*}{p_+} + \frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_+}\frac{\gamma-1}{\gamma+1}+1} \right) u_* \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ \left(\frac{\frac{p_*}{p_+} + \frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_+}\frac{\gamma-1}{\gamma+1}+1} \right) u_*^2 + p_* \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ \left(\frac{\frac{p_*}{p_+} + \frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_+}\frac{\gamma-1}{\gamma+1}+1} \right) u_* v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ \left(\frac{\frac{p_*}{p_+} + \frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_+}\frac{\gamma-1}{\gamma+1}+1} \right) u_* w_+ \\ \mathcal{H}_{E}^{\delta I} = u_* \left[\frac{p_*}{\gamma-1} + \frac{1}{2}\rho_+ \left(\frac{\frac{p_*}{p_+} + \frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_+}\frac{\gamma-1}{\gamma+1}+1} \right) (u_*^2 + v_+^2 + w_+^2) + p_* \right] \end{cases} \tag{3.26}$$

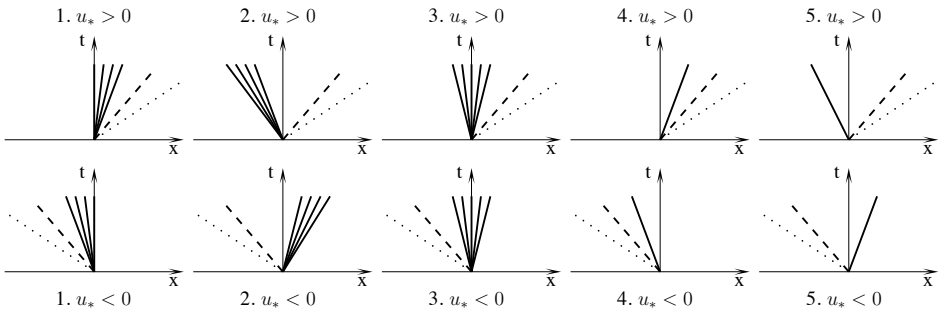


Figure 3.2: Possible wave patterns of the exact Riemann solver for the Euler equation. The dashed line is a contact discontinuity. The group of solid lines (first three top and bottom figures from the left) is a rarefaction wave. The single solid line is a shock wave. The dotted line can be either a rarefaction wave or a shock.

- Negative particle speed in the region between the two nonlinear waves, $u_* < 0$

  1. Right rarefaction - $\boldsymbol{u}_-$: if $p_* \leq p_-$ and $u_- + c_- \leq 0$ then we have a right rarefaction wave and the state in x/t = 0 is the right state. The boundary fluxes are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- u_- \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- u_-^2 + p_- \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- u_- v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- u_- w_- \\ \mathcal{H}_{E}^{\delta I} = u_- \left[\frac{p_-}{\gamma-1} + \frac{1}{2}\rho_L (u_-^2 + v_-^2 + w_-^2) + p_- \right] \end{cases} \tag{3.27}$$

2. Right rarefaction - $\boldsymbol{u}_{*-}$: if $p_* \leq p_-$, $u_- + c_- > 0$ and $u_* + c_-\left(\frac{p_*}{p_-}\right)^{\frac{\gamma-1}{2\gamma}} \geq 0$, then we have a right rarefaction wave and the state for x/t = 0 is the star right state The boundary fluxes in this case are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_-\left(\frac{p_*}{p_-}\right)^{\frac{1}{\gamma}} u_* \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_-\left(\frac{p_*}{p_-}\right)^{\frac{1}{\gamma}} u_*^2 + p_* \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_-\left(\frac{p_*}{p_-}\right)^{\frac{1}{\gamma}} u_* v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_-\left(\frac{p_*}{p_-}\right)^{\frac{1}{\gamma}} u_* w_- \\ \mathcal{H}_{E}^{\delta I} = u_*\left[\frac{p_*}{\gamma-1} + \frac{1}{2}\rho_-\left(\frac{p_*}{p_-}\right)^{\frac{1}{\gamma}}(u_*^2 + v_-^2 + w_-^2) + p_*\right] \end{cases} \tag{3.28}$$

3. Right rarefaction - $\boldsymbol{u}_{-Fan}$: if $p_* \leq p_-$, $u_- + c_- > 0$ and $u_* + c_-\left(\frac{p_*}{p_-}\right)^{\frac{\gamma-1}{2\gamma}} < 0$ we have a right rarefaction wave with a right state which is supersonic and the state for x/t = 0 is the rarefaction fan. The boundary fluxes in this case are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2}{\gamma-1}}\left(-\frac{2c_-}{\gamma+1} + \frac{\gamma-1}{\gamma+1}u_-\right) \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2}{\gamma-1}}\left(-\frac{2c_-}{\gamma+1} + \frac{\gamma-1}{\gamma+1}u_-\right)^2 + \\ \qquad + p_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2\gamma}{\gamma-1}} \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2}{\gamma-1}}\left(\frac{2c_-}{\gamma+1} - \frac{\gamma-1}{\gamma+1}u_-\right)v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2}{\gamma-1}}\left(\frac{2c_-}{\gamma+1} - \frac{\gamma-1}{\gamma+1}u_-\right)w_- \\ \mathcal{H}_{E}^{\delta I} = \left(-\frac{2c_-}{\gamma+1} + \frac{\gamma-1}{\gamma+1}u_-\right)\left\{\frac{p_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2\gamma}{\gamma-1}}}{\gamma-1} + \right. \\ \qquad + \frac{1}{2}\rho_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2}{\gamma-1}}\left[\left(-\frac{2c_-}{\gamma+1} + \frac{\gamma-1}{\gamma+1}u_-\right)^2 + \\ \qquad \left. + v_-^2 + w_-^2\right] + p_-\left[\frac{2}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\left(\frac{u_-}{c_-}\right)\right]^{\frac{2\gamma}{\gamma-1}}\right\} \end{cases} \tag{3.29}$$

4. Right shock - $\boldsymbol{u}_-$: if $p_* > p_-$ and $u_- + c_-\left[\frac{\gamma+1}{2\gamma}\left(\frac{p_*}{p_-}\right) + \frac{\gamma-1}{2\gamma}\right] \leq 0$ we have a right shock with the state for x/t = 0 being the right state. The associated boundary fluxes are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- u_- \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- u_-^2 + p_- \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- u_- v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- u_- w_- \\ \mathcal{H}_{E}^{\delta I} = u_-\left[\frac{p_-}{\gamma-1} + \frac{1}{2}\rho_R(u_-^2 + v_-^2 + w_-^2) + p_-\right] \end{cases} \tag{3.30}$$

5. Right shock - $\boldsymbol{u}_{*-}$: f $p_* > p_-$ and $u_- + c_-\left[\frac{\gamma+1}{2\gamma}\left(\frac{p_*}{p_-}\right) + \frac{\gamma-1}{2\gamma}\right] > 0$ we have a right shock with the state for x/t = 0 being the star right state. The associated boundary fluxes are:

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- \left(\frac{\frac{p_*}{p_-}+\frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_-}\frac{\gamma-1}{\gamma+1}+1} \right) u_* \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- \left(\frac{\frac{p_*}{p_-}+\frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_-}\frac{\gamma-1}{\gamma+1}+1} \right) u_*^2 + p_* \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- \left(\frac{\frac{p_*}{p_-}+\frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_-}\frac{\gamma-1}{\gamma+1}+1} \right) u_* v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- \left(\frac{\frac{p_*}{p_-}+\frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_-}\frac{\gamma-1}{\gamma+1}+1} \right) u_* w_- \\ \mathcal{H}_{E}^{\delta I} = u_* \left[\frac{p_*}{\gamma-1} + \frac{1}{2}\rho_- \left(\frac{\frac{p_*}{p_-}+\frac{\gamma-1}{\gamma+1}}{\frac{p_*}{p_-}\frac{\gamma-1}{\gamma+1}+1} \right) (u_*^2 + v_-^2 + w_-^2) + p_* \right] \end{cases} \tag{3.31}$$

Note that the pressure in the star region, p_*, is pre-calculated from the following nonlinear equation:

$$f(p_*, \boldsymbol{u}_+, \boldsymbol{u}_-) = f_+(p_*, \boldsymbol{u}_+) + f_-(p_*, \boldsymbol{u}_-) + u_- - u_+ = 0, \tag{3.32}$$

where:

$$\begin{cases} f_+(p_*, \boldsymbol{u}_+) = (p_* - p_+)\left[\frac{\frac{2}{(\gamma+1)\rho_+}}{p_* + \frac{(\gamma-1)}{(\gamma+1)}p_+} \right] & \text{if } p_* > p_+ \text{ (shock)} \\ f_+(p_*, \boldsymbol{u}_+) = \frac{2c_+}{(\gamma-1)}\left[\left(\frac{p_*}{p_+}\right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] & \text{if } p_* \leq p_+ \text{ (rarefaction)} \\ f_-(p_*, \boldsymbol{u}_-) = (p_* - p_-)\left[\frac{\frac{2}{(\gamma+1)\rho_-}}{p_* + \frac{(\gamma-1)}{(\gamma+1)}p_-} \right] & \text{if } p_* > p_- \text{ (shock)} \\ f_-(p_*, \boldsymbol{u}_-) = \frac{2c_-}{(\gamma-1)}\left[\left(\frac{p_*}{p_-}\right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] & \text{if } p_* \leq p_- \text{ (rarefaction)} \end{cases}$$

After having pre-calculated the pressure p_*, the velocity in the star region, u_*, is calculated as follows:

$$u_* = \frac{1}{2}(u_+ + u_-) + \frac{1}{2}\big[f_-(p_*) + f_+(p_*)\big]. \tag{3.33}$$

HLL :

The HLL (Harten, Lax, van Leer) Riemann solver, first proposed by Harten et al. (1983), is an approximated solver which assumes a two-wave configuration by neglecting the middle contact wave (i.e. neglecting the shear waves). The possible patterns in this case are three and they are selected through the

evaluation of the fastest signal velocities which perturb the initial data. The approach produced practical schemes after the contributions of Davis (1988) and Einfeldt (1988), who independently proposed various ways of computing the wave speeds required to completely determine the intercell flux. In order to determine completely the numerical fluxes in the HLL Riemann solvers we need to provide an algorithm for computing the fastest wave speeds, namely S_+ and S_-. Different approaches have been proposed in literature to compute the bounding for the minimum and maximum signal velocities. Davis (1988) suggested the simple estimates:

$$\begin{cases} S_+ = \min(u_+ - c_+, u_- - c_-) \\ S_- = \max(u_- + c_-, u_- + c_-) \end{cases} . \tag{3.34}$$

These estimates make use of data values only, are exceedingly simple but are not recommended for practical computations. Both Davis (1988) and Einfeldt (1988) proposed to use the Roe (1981) average eigenvalues for the left and right nonlinear waves, that is:

$$\begin{cases} S_+ = u_{Roe} - c_{Roe} \\ S_- = u_{Roe} + c_{Roe} \end{cases} , \tag{3.35}$$

where u_{Roe} and c_{Roe} are the Roe averages of the normal speed and of the speed of sound respectively. These are defined as:

$$u_{Roe} = \frac{\sqrt{\rho_+}u_+ + \sqrt{\rho_-}u_-}{\sqrt{\rho_+} + \sqrt{\rho_-}}, \quad c_{Roe} = \left[(\gamma - 1) \left(H_{Roe} - \frac{1}{2} u_{Roe}^2 \right)^{1/2} \right], \tag{3.36}$$

with the enthalpy $H = \dfrac{E + p}{\rho}$ approximated as:

$$H_{Roe} = \frac{\sqrt{\rho_+}H_+ + \sqrt{\rho_-}H_-}{\sqrt{\rho_+} + \sqrt{\rho_-}}. \tag{3.37}$$

In Nektar++ we use the following estimates:

$$\begin{cases} S_+ = \min(u_+ - c_+, u_{Roe} - c_{Roe}) \\ S_- = \max(u_- + c_-, u_{Roe} + c_{Roe}) \end{cases} , \tag{3.38}$$

and the three different states are:

1. $S_+ \geq 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ u_+ \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ u_+^2 + p_+ \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ u_+ v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ u_+ w_+ \\ \mathcal{H}_{E}^{\delta I} = u_+ \left[\frac{p_+}{\gamma - 1} + \frac{1}{2} \rho_L (u_+^2 + v_+^2 + w_+^2) + p_+ \right] \end{cases} \tag{3.39}$$

2. $S_- \leq 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- u_- \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- u_-^2 + p_- \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- u_- v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- u_- w_- \\ \mathcal{H}_{E}^{\delta I} = u_- \left[\frac{p_-}{\gamma - 1} + \frac{1}{2} \rho_R (u_-^2 + v_-^2 + w_-^2) + p_- \right] \end{cases} \tag{3.40}$$

3. $S_+ < 0$ or $S_- > 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \frac{S_- \rho_+ u_+ - S_+ \rho_- u_- + S_- S_+ (\rho_- - \rho_+)}{S_- - S_+} \\ \mathcal{H}_{\rho u}^{\delta I} = \frac{S_- (\rho_+ u_+^2 + p_+) - S_+ (\rho_- u_-^2 + p_-) + S_- S_+ (\rho_- u_- - \rho_+ u_+)}{S_- - S_+} \\ \mathcal{H}_{\rho v}^{\delta I} = \frac{S_- (\rho_+ u_+ v_+) - S_+ (\rho_- u_- v_-) + S_- S_+ (\rho_- v_- - \rho_+ v_+)}{S_- - S_+} \\ \mathcal{H}_{\rho w}^{\delta I} = \frac{S_- (\rho_+ u_+ w_+) - S_+ (\rho_- u_- w_-) + S_- S_+ (\rho_- w_- - \rho_+ w_+)}{S_- - S_+} \\ \mathcal{H}_{E}^{\delta I} = \frac{S_- u_+ (E_+ + p_+) - S_+ u_- (E_- + p_-) + S_- S_+ (E_- - E_+)}{S_- - S_+} \end{cases} \tag{3.41}$$

It should be noted that the HLL Riemann solver does not allow for jumps in the tangential velocities as well as in the density. This feature can penalise some problems but can be beneficial for others.

HLLC :

The HLLC (Harten, Lax, van Leer + Contact) Riemann solver is an approximated solver which restores a three-wave configuration by contrast to the HLL solver. This approach was proposed by Toro et al. (1992, 1994) to remedy the problem of intermediate waves in the HLL approach. A precursor to HLLC was also anticipated in Toro (1992). The possible patterns in this case are four and they are selected through the evaluation of the fastest signal velocities which perturb the initial data, namely, S_+ and S_- (computed as in Eq. (3.38)) as well as through the calculation of the middle (contact) wave speed S_M. This can be computed once S_+ and S_- are known as:

$$S_M = \frac{p_- - p_+ + \rho_+ u_+ (S_+ - u_+) - \rho_- u_- (S_- - u_-)}{\rho_+ (S_+ - u_+) - \rho_- (S_- - u_-)} \tag{3.42}$$

1. $S_+ \geq 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ u_+ \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ u_+^2 + p_+ \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ u_+ v_+ \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ u_+ w_+ \\ \mathcal{H}_{E}^{\delta I} = u_+ \left[\frac{p_+}{\gamma - 1} + \frac{1}{2} \rho_L (u_+^2 + v_+^2 + w_+^2) + p_+ \right] \end{cases} \tag{3.43}$$

2. $S_- \leq 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- u_- \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- u_-^2 + p_- \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- u_- v_- \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- u_- w_- \\ \mathcal{H}_{E}^{\delta I} = u_- \left[\frac{p_-}{\gamma-1} + \frac{1}{2}\rho_R(u_-^2 + v_-^2 + w_-^2) + p_-\right] \end{cases} \tag{3.44}$$

3. $S_+ < 0$ and $S_M \geq 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_+ u_+ + S_+\left[\rho_+ \frac{(S_+ - u_+)}{(S_+ - S_M)} - \rho_+\right] \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_+ u_+^2 + p_+ + S_+\left[\rho_+ \frac{(S_+ - u_+)}{(S_+ - S_M)} - \rho_+ u_+\right] \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_+ u_+ v_+ + S_+\left[\rho_+ \frac{(S_+ - v_+)}{(S_+ - S_M)} - \rho_+ v_+\right] \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_+ u_+ w_+ + S_+\left[\rho_+ \frac{(S_+ - w_+)}{(S_+ - S_M)} - \rho_+ w_+\right] \\ \mathcal{H}_{E}^{\delta I} = u_+(E_+ + p_+) + S_+\Big\{\rho_+ \frac{(S_+ - u_+)}{(S_+ - S_M)}\big[\frac{E_+}{\rho_+} + \\ \qquad (S_M - u_+)\big(S_M + \frac{p_+}{\rho_+(S_+ - u_+)}\big)\big] - E_+\Big\} \end{cases} \tag{3.45}$$

4. $S_M < 0$ and $S_- > 0$

$$\begin{cases} \mathcal{H}_{\rho}^{\delta I} = \rho_- u_- + S_-\left[\rho_- \frac{(S_- - u_-)}{(S_- - S_M)} - \rho_-\right] \\ \mathcal{H}_{\rho u}^{\delta I} = \rho_- u_-^2 + p_- + S_-\left[\rho_- \frac{(S_- - u_-)}{(S_- - S_M)} - \rho_- u_-\right] \\ \mathcal{H}_{\rho v}^{\delta I} = \rho_- u_- v_- + S_-\left[\rho_- \frac{(S_- - v_-)}{(S_- - S_M)} - \rho_- v_-\right] \\ \mathcal{H}_{\rho w}^{\delta I} = \rho_- u_- w_- + S_-\left[\rho_- \frac{(S_- - w_-)}{(S_- - S_M)} - \rho_- w_-\right] \\ \mathcal{H}_{E}^{\delta I} = u_-(E_- + p_-) + S_-\Big\{\rho_- \frac{(S_- - u_-)}{(S_- - S_M)}\big[\frac{E_-}{\rho_-} + \\ \qquad (S_M - u_-)\big(S_M + \frac{p_-}{\rho_-(S_- - u_-)}\big)\big] - E_-\Big\} \end{cases} \tag{3.46}$$

For a more detailed discussion and some additional implementative insights on this topic the interested reader can refer to Toro (2009).

3.4 Boundary conditions

The implementation of boundary conditions (BCs) can have a significant impact on the overall accuracy of a numerical simulation especially when dealing with compressible flows, either inviscid or viscous. It is therefore important to implement them correctly in CFD solvers.

Usually BCs are imposed in two different ways: by directly modifying the boundary solution at each iteration/time-step (*strong* BCs) or by modifying the state from which the numerical flux is calculated[2] (*weak* BCs).

In this thesis work, we considered (as already mentioned in chapter 1) a weak strategy - i.e. we applied the boundary conditions to the fluxes rather than to the conserved variables.

Specifically, for the inviscid fluxes we explored two weak approaches. The first approach consists in applying the BCs in an indirect manner through a Riemann solver (Approach A) whereas the second applies the BCs by calculating the fluxes directly from the known value of the variables at the boundary (Approach B):

$$Approach\ A: \qquad \mathcal{H}_i^{\delta I} = \mathcal{H}_i^{\delta I}(\mathbf{u}_+^{\delta}, \mathbf{u}_-^{\delta}). \tag{3.47}$$

$$Approach\ B: \qquad \mathcal{H}_i^{\delta I} = \mathcal{H}_i^{\delta I}(\mathbf{u}_{BC}^{\delta}). \tag{3.48}$$

These two approaches are common in the family of methods considered in this paper and one implementation or the other depends on a series of factors which range from solver architecture, computational costs as well as robustness and effectiveness of the BC implementation.

Approach A is less intrusive with respect to the underlying numerics used at the inner interfaces of the numerical discretisation. On the other hand, the use of a Riemann solver for applying BCs implies the usage of a ghost point where it is necessary to apply a consistent ghost state (indicated with the subscript "$-$" in the rest of this section) which is not always trivial. Additionally, different Riemann solvers have different performance in transferring the BCs into the domain and one Riemann solver can perform better than another depending on the BCs being applied.

While for the advection term (i.e. the inviscid fluxes) we can make a distinction between approach A and B, the only way for applying the boundary conditions to the viscous fluxes in a weak manner is by means of approach B. A detailed comparison between approach A and B for the advection term and some considerations on the treatment of the diffusion term have been presented by the author of this thesis and his co-workers in (Mengaldo et al. (2014)), they are not reported here. Nevertheless, we describe the most robust BC implementation strategies adopted for both the Euler and compressible Navier-Stokes equations.

We additionally remark that another common way of applying the BCs is by setting directly the boundary solution, *strong* reinforcement, which is not taken into account in the present thesis.

[2]Note that this is independent of the mathematical type of the BCs (Dirichlet, Neumann or Robin).

3.4.1 BCs for the Euler equations

Since the applications which are of interest in this thesis are mainly concerned with external aerodynamics, the BCs considered for the compressible Euler equations are essentially two: farfield BCs and slip wall BCs. The first are concerned with boundaries located in the free-stream, and their main objective is to damp spurious reflections which may arise when an outgoing wave is passing through the truncated computational domain. The second, slip wall BCs, are applied when the flow hits a solid non-transpiring wall.

Farfield BCs

Farfield BCs for the Euler equations have been widely explored in the past decades and the interested reader can refer to Thompson (1987, 1990) and Giles (1990) for a detailed analysis.

According to the hyperbolic nature of the Euler equations the flux evaluated at the boundary is a combination of the information coming from inside and outside the domain.

In setting the BCs all the variables at the boundary are specified from free-stream conditions. The complete ghost state is:

$$\boldsymbol{u}_{-} = \left\{ \begin{array}{c} \rho_{-} \\ (\rho u)_{-} \\ (\rho v)_{-} \\ (\rho w)_{-} \\ E_{-} \end{array} \right\} = \left\{ \begin{array}{c} \rho_{\infty} \\ \rho_{\infty} u_{\infty} \\ \rho_{\infty} v_{\infty} \\ \rho_{\infty} w_{\infty} \\ E_{\infty} \end{array} \right\} \tag{3.49}$$

and the flux is then calculated by a Riemann solver trough Eq. 3.47. In evaluating the boundary flux the Riemann solver takes automatically into account the eigenvalues (characteristic lines) of the Euler equations and evaluates the flux by combining the inner and outer states, where the outer state is described by Eq. 3.49. This way of applying the boundary conditions is identical to adopting a characteristic approach based on the Riemann invariants (see appendix C). Specifically, by imposing the boundary conditions in this way, the Riemann solver takes automatically into account the ingoing and outgoing characteristic waves of the Euler equation system. Therefore the boundary conditions are perfectly non-reflective (absorbent) and can be implemented straightforward.

Slip wall BCs

In the case of an inviscid flow hitting a solid surface or wall (with no transpiration), the BCs must prevent the fluid from penetrating the wall. For setting the slip condition we extrapolate the interior velocity at the boundary $\boldsymbol{v}_{in} = [u_{in}, v_{in}, w_{in}]^T$

and then we define the ghost state $\boldsymbol{v}_- = [u_-, v_-, w_-]^T$ by negating the internal velocity:

$$\boldsymbol{v}_- = -\boldsymbol{v}_{in} \tag{3.50}$$

The density and internal energy are extrapolated from the interior such that the complete ghost state is

$$\boldsymbol{u}_- = \left\{ \begin{array}{c} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{array} \right\} = \left\{ \begin{array}{c} \rho_{in} \\ -\rho_{in} u_{in} \\ -\rho_{in} v_{in} \\ -\rho_{in} w_{in} \\ E_{in} \end{array} \right\} \tag{3.51}$$

and the boundary flux is calculated through a Riemann solver using Eq. 3.47. The normal component of the velocity evaluated by the Riemann solver is zero since the only non-zero contributions to the flux function are those coming from the pressure in the momentum equations. If the impermeable wall moves with a prescribed velocity $\boldsymbol{v}_w$ Eq. 3.50 becomes:

$$\boldsymbol{v}_- = -\boldsymbol{v}_{in} + 2\boldsymbol{v}_w \tag{3.52}$$

and Eq. 3.51 will be based on this velocity.

3.4.2 BCs for the compressible Navier-Stokes equations

For the compressible Navier-Stokes equations we consider three typologies of BCs: farfield BCs with and without presence of vortical structures passing through the boundaries, inflow/outflow BCs with laminar viscous effects (e.g. the inflow and outflow regions of a flat plate) and no-slip wall BCs both isothermal and adiabatic.

Farfield BCs

Farfield BCs for the compressible Navier-Stokes equations are still a very active topic of research. In regions where the viscosity effects can be neglected it has been shown that a characteristic approach like that used for the Euler equation will work properly for the Navier-Stokes equations. However, in regions where the viscosity effects become important, such as in the case when vortical structures (e.g. shedding behind an object, turbo machinery wakes, etc.) pass through the farfield boundaries, the characteristic treatment of the BCs fails, generating spurious reflections which pollute the overall solution and may lead to numerical instabilities. Various techniques have been presented in the literature to minimise and possibly eliminate these spurious reflections. For further details, the interested

reader can refer to Colonius (2004). In Nektar++, we set the BCs as in Eq. 3.49 for boundaries where the viscosity effects can be neglected, while we have implemented a method based on the so-called sponge terms, for boundaries where viscosity effects may generate the spurious reflections just mentioned. The latter implementation is based on Israeli and Orszag (1981) and Bodony (2006) and is formulated by modifying the right-hand side of the compressible Navier-Stokes equations as follows:

$$\frac{\partial \boldsymbol{u}}{\partial} + \frac{\partial \boldsymbol{f}_2}{\partial x_1} + \frac{\partial \boldsymbol{f}_2}{\partial x_2} + \frac{\partial \boldsymbol{f}_3}{\partial x_3} = \sigma(\overline{\boldsymbol{x}})(\boldsymbol{u}_{ref} - \boldsymbol{u}) \tag{3.53}$$

where $\sigma(\overline{\boldsymbol{x}})$ is a damping coefficient defined in a region $\overline{\boldsymbol{x}}$ in proximity to the boundaries and $\boldsymbol{u}_{ref}$ is a known reference solution. The unphysical terms on the right-hand side are active only near external boundaries where they damp the flow variables to a known reference solution. The length and the shape of the damping coefficient depends on the problem being solved and has been throughly investigated in Mani (2012).

Inflow/Outflow BCs in presence of laminar viscous effects

In the case of inflow/outflow BCs where laminar viscous effects are important, such as the inflow and outflow regions of a flat plate with a developing boundary layer, to appropriately recover the physical solution we need to apply a different set of BCs than the farfield BCs we have shown in the previous section. In the following we show the treatment of the advection and of the diffusion terms separately.

Advection term

For the advection term we distinguish the following four different cases:

Subsonic inflow ($M_n = |u_n|/c < 1$)

We impose the density and the velocity on the ghost state and extrapolate the pressure from the interior of the domain:

$$\boldsymbol{u}_- = \left\{ \begin{array}{c} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{array} \right\} = \left\{ \begin{array}{c} \rho_e \\ \rho_e u_e \\ \rho_e v_e \\ \rho_e w_e \\ \frac{p_{in}}{\gamma - 1} + \frac{1}{2}\rho_e(u_e^2 + v_e^2 + w_e^2) \end{array} \right\}. \tag{3.54}$$

We can use this strategy when dealing with a flat plate simulation for instance, where the inflow state is known from the computation of the similarity solution.

Supersonic inflow ($M_n = |u_n|/c > 1$)

We impose all the variables on the ghost state:

$$\boldsymbol{u}_- = \begin{Bmatrix} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{Bmatrix} = \begin{Bmatrix} \rho_e \\ \rho_e u_e \\ \rho_e v_e \\ \rho_e w_e \\ \frac{p_e}{\gamma-1} + \frac{1}{2}\rho_e(u_e^2 + v_e^2 + w_e^2) \end{Bmatrix}. \tag{3.55}$$

Subsonic outflow ($M_n = |u_n|/c < 1$)

We impose an estimated or exact pressure p_e on the ghost state and extrapolate density and velocity from the interior of the domain:

$$\boldsymbol{u}_- = \begin{Bmatrix} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{Bmatrix} = \begin{Bmatrix} \rho_{in} \\ \rho_{in} u_{in} \\ \rho_{in} v_{in} \\ \rho_{in} w_{in} \\ \frac{p_e}{\gamma-1} + \frac{1}{2}\rho_{in}(u_{in}^2 + v_{in}^2 + w_{in}^2) \end{Bmatrix}. \tag{3.56}$$

This approach is particularly accurate when a good pressure estimate is known a priori.

Supersonic outflow ($M_n = |u_n|/c > 1$)

We extrapolate the entire ghost state from the interior values:

$$\boldsymbol{u}_- = \begin{Bmatrix} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{Bmatrix} = \begin{Bmatrix} \rho_{in} \\ \rho_{in} u_{in} \\ \rho_{in} v_{in} \\ \rho_{in} w_{in} \\ \frac{p_{in}}{\gamma-1} + \frac{1}{2}\rho_{in}(u_{in}^2 + v_{in}^2 + w_{in}^2) \end{Bmatrix}. \tag{3.57}$$

This approach is also known as do-nothing approach.

The boundary flux is then calculated by Eq. 3.47 using one of the ghost states above, depending on the boundary. In Nektar++, the identification of a subsonic

or a supersonic region is automatic, therefore the user, in this case, needs just to specify if the region is an inflow or an outflow region. Note also that, all the above implementations are reflective - if an acoustic or a vortical wave hits the boundary this set of inflow/outflow BCs generate spurious reflections. In this case we can use these BCs in conjunction with the previously defined sponge region, so that the inflow/outflow regions become fully non-reflective.

Diffusion term

Concerning the diffusion term, we need to impose the auxiliary variables, $\boldsymbol{u}_{aux}$, which are used to calculate the first order derivatives, at the boundary. Since we are using an LDG approach, with $\boldsymbol{\beta} = 1/2$, we directly impose these values at the boundary without making any distinction between the four cases shown for the advection term:

$$\boldsymbol{u}_{aux_{BC}} = \left\{ \begin{array}{c} u_{BC} \\ v_{BC} \\ w_{BC} \\ T_{BC} \end{array} \right\} = \left\{ \begin{array}{c} u_{in} \\ v_{in} \\ w_{in} \\ T_{in} \end{array} \right\}. \tag{3.58}$$

The velocity at the boundary is left to be free. The boundary viscous flux is then calculated as $\mathcal{H}_v^{\delta I} = \mathcal{H}_v^{\delta I}(\boldsymbol{u}_{BC}^{\delta}, (\nabla_{\boldsymbol{x}} \boldsymbol{u}^{\delta})_{in})$.

Eq. 3.58 is used as boundary condition for the auxiliary variables in the LDG method, where the derivatives of u, v, w and T are calculated, and successively used to evaluate the boundary viscous flux. It is important to note that the boundary viscous flux is computed using the gradients of the inner state since they are unknown at the boundary.

No-slip isothermal and adiabatic wall BCs

For the compressible Navier-Stokes equations a no-slip condition must be applied to the velocity field at a solid wall (with no transpiration). We first consider the treatment of the BCs on the advection term and successively we describe how to apply the BCs to the diffusion term.

Advection term

The most robust and reliable implementation strategy for the no-slip wall BCs on the advection term is achieved by setting the ghost state as follows (as for the Euler equations):

$$\boldsymbol{u}_- = \left\{ \begin{array}{c} \rho_- \\ (\rho u)_- \\ (\rho v)_- \\ (\rho w)_- \\ E_- \end{array} \right\} = \left\{ \begin{array}{c} \rho_{in} \\ -\rho_{in} u_{in} \\ -\rho_{in} v_{in} \\ -\rho_{in} w_{in} \\ E_{in} \end{array} \right\}, \tag{3.59}$$

where the density and the internal energy are extrapolated from the interior while the velocity components are negated. The boundary flux is then calculated through a Riemann solver via Eq. 3.47. As for the Euler equations, the normal component of the velocity evaluated by the Riemann solver is zero since the only non-zero contributions to the flux function are those coming from the pressure in the momentum equations. Note that there is no distinction at this level for what concerns the isothermal and adiabatic wall BCs. This arises in the implementation of the diffusion term. If the impermeable wall moves with a prescribed velocity $\boldsymbol{v}_w$ the velocity of the ghost state will be set according to Eq. 3.52 and Eq. 3.59 will be based on this velocity.

Diffusion term

Concerning the diffusion term we impose the boundary conditions on the auxiliary variables $\boldsymbol{u}_{aux}$, which are used to calculate the first order derivatives. Here, we distinguish between an isothermal and an adiabatic wall.

Isothermal wall

In the case of LDG method in conjunction with an isothermal wall with an imposed temperature T_w, the auxiliary variables at the boundary assume the following values:

$$\boldsymbol{u}_{aux_{BC}} = \left\{ \begin{array}{c} u_{BC} \\ v_{BC} \\ w_{BC} \\ T_{BC} \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ T_w \end{array} \right\}. \tag{3.60}$$

It is possible to appreciate that the velocity at the boundary is set to zero. The final boundary viscous flux is then calculated as $\mathcal{H}_v^{\delta I} = \mathcal{H}_v^{\delta I}(\boldsymbol{u}_{BC}^{\delta}, (\nabla_{\boldsymbol{x}} \boldsymbol{u}^{\delta})_{in})$.

Eq. 3.60 is used as boundary conditions for the auxiliary variables in the LDG method, where the derivatives of u, v, w and T are calculated, and successively used to evaluate the boundary viscous flux. As in this case the boundary viscous flux is computed using the gradients of the inner state since they are unknown at the boundary. An exception is the shear stress part of the boundary viscous flux associated to the energy equation which is set to zero. This is due to premultiplication with the velocity which is known to be zero at a no-slip wall.

Adiabatic wall

The diffusion term for an adiabatic wall differs from the isothermal since we apply the thermal flux instead of the wall temperature. Specifically the temperature at the boundary is extrapolated from the interior and the dependent variables at the wall are:

$$\boldsymbol{u}_{aux_{BC}} = \left\{ \begin{array}{c} u_{BC} \\ v_{BC} \\ w_{BC} \\ T_{BC} \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ T_{in} \end{array} \right\}. \tag{3.61}$$

Eq. 3.61 is used as boundary condition for the auxiliary system in the LDG method.

In order to evaluate the normal flux at the boundary, we need to split the flux in Eq. (3.12) into two parts: one which arises form the shear stresses (superscript "s" in Eq. (3.62)), and one which depends on the thermal diffusivity (superscript "t" in Eq. (3.62)):

$$\begin{aligned}
\boldsymbol{f}^s_{v,1} &= \left\{ \begin{array}{c} 0 \\ \tau_{x_1x_1} \\ \tau_{x_2x_1} \\ \tau_{x_3x_1} \\ u\tau_{x_1x_1} + v\tau_{x_2x_1} + w\tau_{x_3x_1} \end{array} \right\}, \quad \boldsymbol{f}^t_{v,1} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ kT_{x_1} \end{array} \right\}, \\
\boldsymbol{f}^s_{v,1} &= \left\{ \begin{array}{c} 0 \\ \tau_{x_1x_2} \\ \tau_{x_2x_2} \\ \tau_{x_3x_2} \\ u\tau_{x_1x_2} + v\tau_{x_2x_2} + w\tau_{x_3x_2} \end{array} \right\}, \quad \boldsymbol{f}^t_{v,2} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ kT_{x_2} \end{array} \right\}, \\
\boldsymbol{f}^s_{v,3} &= \left\{ \begin{array}{c} 0 \\ \tau_{x_1x_3} \\ \tau_{x_2x_3} \\ \tau_{x_3x_3} \\ u\tau_{x_1x_3} + v\tau_{x_2x_3} + w\tau_{x_3x_3} \end{array} \right\}, \quad \boldsymbol{f}^t_{v,3} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ kT_{x_3} \end{array} \right\},
\end{aligned} \tag{3.62}$$

At the boundary, the normal flux arising from the shear stresses is extrapolated from the values obtained in the interior of the domain while the normal flux arising from the thermal diffusivity is set to zero, i.e.

$$k\nabla_{\boldsymbol{x}} T \cdot \mathbf{n} = 0. \tag{3.63}$$

As for the isothermal BCs, the final boundary viscous flux is calculated by using the following relation: $\mathcal{H}_v^{\delta I} = \mathcal{H}_v^{\delta I}(\boldsymbol{u}^{\delta}_{BC}, (\nabla_{\boldsymbol{x}} \boldsymbol{u}^{\delta})_{in})$.

3.5 Code verification

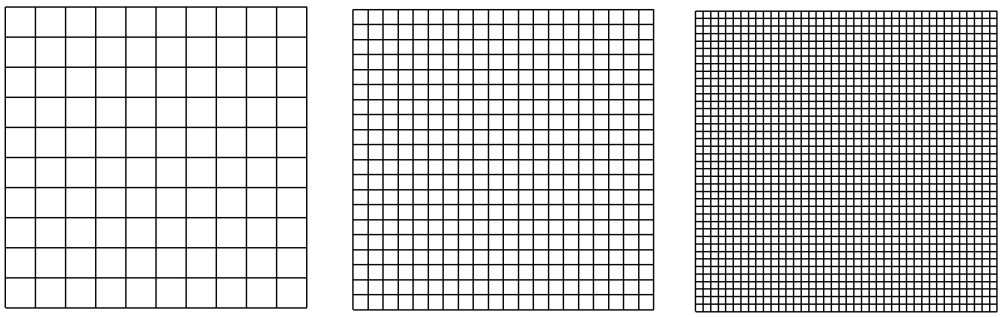
The compressible flow solver implemented in Nektar++ and the underlying numerical discretisations by means of the DG and the FR approaches have been thoroughly tested for both the Euler and the compressible Navier-Stokes equations. In the next two sections we show a few relevant test cases, both in a two- and three dimensional context, and we highlight the main features of discontinuous spectral/hp element methods applied to nonlinear systems of conservation laws.

3.5.1 Euler equations

We start showing the verification results for the Euler equations. Specifically, we consider two test cases, the isentropic vortex test case (Shu (1998)) and a subsonic flow past a sphere. The aim of the first test case is to show the convergence order achieved for the various FR and DG schemes shown in chapter 1 when solving a nonlinear system of equations. In addition, the first test case shows the performance of the three different Riemann solvers introduced in section 3.3.1. The second test case illustrates the generation of numerical entropy when a low-order approximation of the surface of the sphere is employed and highlights the issues arising from collocated schemes when dealing with nonlinear problems.

Isentropic vortex

The aim of the simulations carried out is to recover the exponential order of convergence expected for the FR approach and the DG method in order to show the correctness of the implementation.



(a) 10×10 (b) 20×20 (c) 40×40

Figure 3.3: Meshes utilised in the numerical experiments on the isentropic vortex.

As for the linear advection and the linear diffusion equation in chapter 1, we used three different FR schemes, namely the $\mathrm{FR_{DG}}$, $\mathrm{FR_{g2}}$ and $\mathrm{FR_{SD}}$ schemes (in their collocated version), which have been introduced in section 1.4.1 and both the $\mathrm{DG_{SEM}}$-LMM and the $\mathrm{DG_{SEM}}$-EMM methods introduced in section 1.4.2. The solution and quadrature points chosen were Gauss-Lobatto-Legendre. The simulations have been carried out on the three regular squared meshes depicted in Figure 3.3. The meshes are increasingly finer where the number of elements is equal to $N_x \times N_x$ with $N_x = 10, 20, 40$, respectively. The length of each side is equal to 20. Periodic conditions were applied at the boundary of the grids, and an isentropic vortex in a free-stream flow (in the positive y direction) was prescribed within $\mathbf{\Omega}$ at $t = 0$. The vortex is described by Eq. 2.29 here reported for the reader's convenience:

$$\begin{aligned} \rho &= \left(1 - \tfrac{\beta^2(\gamma-1)e^{2f}}{16\gamma\pi^2}\right)^{\frac{1}{\gamma-1}}, \\ u &= \left(u_0 - \tfrac{\beta e^f (y-y_0)}{2\pi R}\right), \\ v &= \left(v_0 + \tfrac{\beta e^f (x-x_0)}{2\pi R}\right), \\ E &= \tfrac{\rho^\gamma}{\gamma-1} + \tfrac{1}{2}\rho\left(u^2+v^2\right), \end{aligned} \tag{3.64}$$

where:

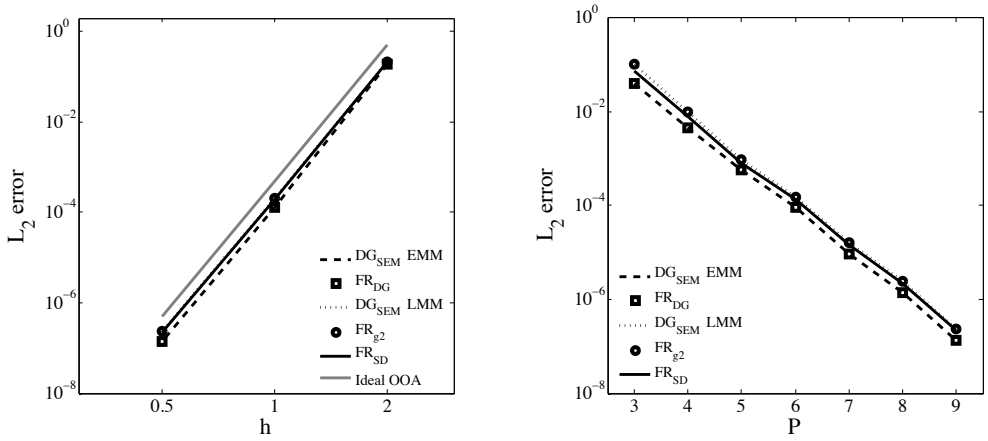
$$\begin{aligned} f &= 1 - r^2, \\ r &= \sqrt{(x-x_0)^2 + (y-y_0)^2}\ . \end{aligned}$$

with $(u_0, v_0) = (0, 5)$, $(x_0, y_0) = (5, 0)$, $R = 1$, $\beta = 5$ and $\gamma = 1.4$. An explicit Runge-Kutta time integration scheme of 4 stages was used to discretise the equations in time. The final time step was $T = 4s$, therefore the exact solution at the final time is equal to the initial. The time-step chosen was small enough in order to consider the temporal error negligible.

In Figure 3.4(a), we show the $\mathrm{L_2}$ error associated to the energy E as a function of the mesh size 'h' in a logarithmic scale along with the ideal order of convergence (also referred to as order of accuracy) denoted with 'OOA'. We represented the solution with a polynomial order $P = 9$ and it is possible to appreciate how the optimal order of convergence is achieved for all the DG and FR schemes taken into account.

In Figure 3.4(b), we show the $\mathrm{L_2}$ error associated to the energy E as a function of the polynomial order in a semi-logarithmic scale. We discretised the problem on the finest mesh (Figure 3.3(c)) and we can see a good P-convergence, as expected.

Finally, Figure 3.5 shows the behaviour of the three different Riemann solvers presented in section 3.3.1. While there are no differences between the exact and HLLC Riemann solvers, it is possible to observe a noticeably different behaviour of the HLL Riemann solver, which provides a higher $\mathrm{L_2}$ error than the other three.This



(a) h-type convergence

(b) p-type convergence

Figure 3.4: Order of accuracy (OOA) and P convergence for the isentropic vortex equation.

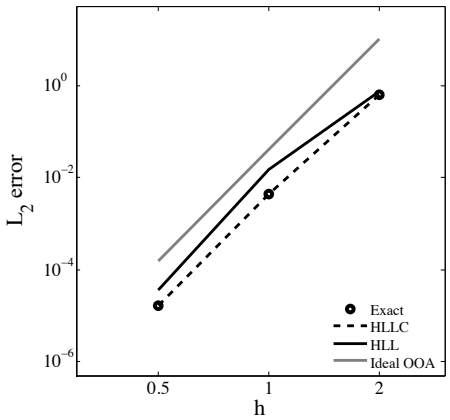


Figure 3.5: Order of accuracy (OOA) for different Riemann solvers detailed in section 3.3.1. Isentropic vortex test case.

result is expected because the HLL Riemann solver does not take into account the full Riemann problem and therefore is the least accurate (for this problem) among those considered. In fact, the central idea of the HLL Riemann solver is to assume for the solution, a wave configuration that consists of two waves separating three constant states. This assumption of a two-wave configuration is correct only for hyperbolic systems of two equations, such as the one-dimensional

shallow water equations. For larger systems, such as the Euler equations or the split two-dimensional shallow water equations for example, the two-wave assumption is incorrect an the results obtained can be very inaccurate.

Subsonic inviscid flow past a sphere

To show how the numerical methods presented deal with curved geometries we consider a subsonic flow past a sphere. In fact, when straight edges are used to describe the sphere and in general when a low-order approximation of the equations is employed (i.e. in case of spectral element methods, when low polynomial orders are employed), the solution can be affected by numerically-generated entropy. In addition, the posterior stagnation point is very sensitive to numerical errors (such as integration errors) due to the strong gradients of the solution in that region. In this point, it is possible to see a numerically-generated entropy that can lead to the failure of the simulation. This behaviour can be avoided by using a very fine mesh in proximity to the posterior stagnation point or by using a higher quadrature which leads to a reduction of the errors arising from the non-exact integration performed when using collocated schemes, such as the case of the nodal DG method or the FR approaches. Figure 3.6 shows the mesh used for the simulations. This is composed by elements $N_s \times N_r = 1014 \times 13$, where N_s is the number of elements on the sphere and N_r is the number of elements in the radial direction (normal to the sphere surface). The sphere surface is approximated by a fourth-order spline function.

For all the simulations we used a Mach number equal to $M_\infty = 0.2$, with the pressure set to $p = 22600\ Pa$ and the density equal to $\rho = 0.364\ kg/m^3$. For the time-integration we used a fourth-order Runge-Kutta scheme and the time-step was chosen such that the time-integration errors were negligible. The numerical fluxes were obtained through an exact Riemann solver (see section 3.3.1) and we employed the DG method both with a collocation projection of fluxes and with a *Global* dealiasing technique described in chapter 4.

	L_∞ entropy error [J/Kg K]			
	P=1 Q =2	P=2 Q =3	P=3 Q =4	P=4 Q =5
Collocated	✗	✗	✗	✗
	P=1 Q = 3	P=2 Q =4	P=3 Q =6	P=4 Q =8
Global dealiasing	23.36	7.37	1.55	0.66

Table 3.1: L_∞ errors associated to the entropy for all the subsonic flow past a sphere test cases performed at steady state.

The quadrature points were Gauss-Lobatto-Legendre type. In Table 3.1, we

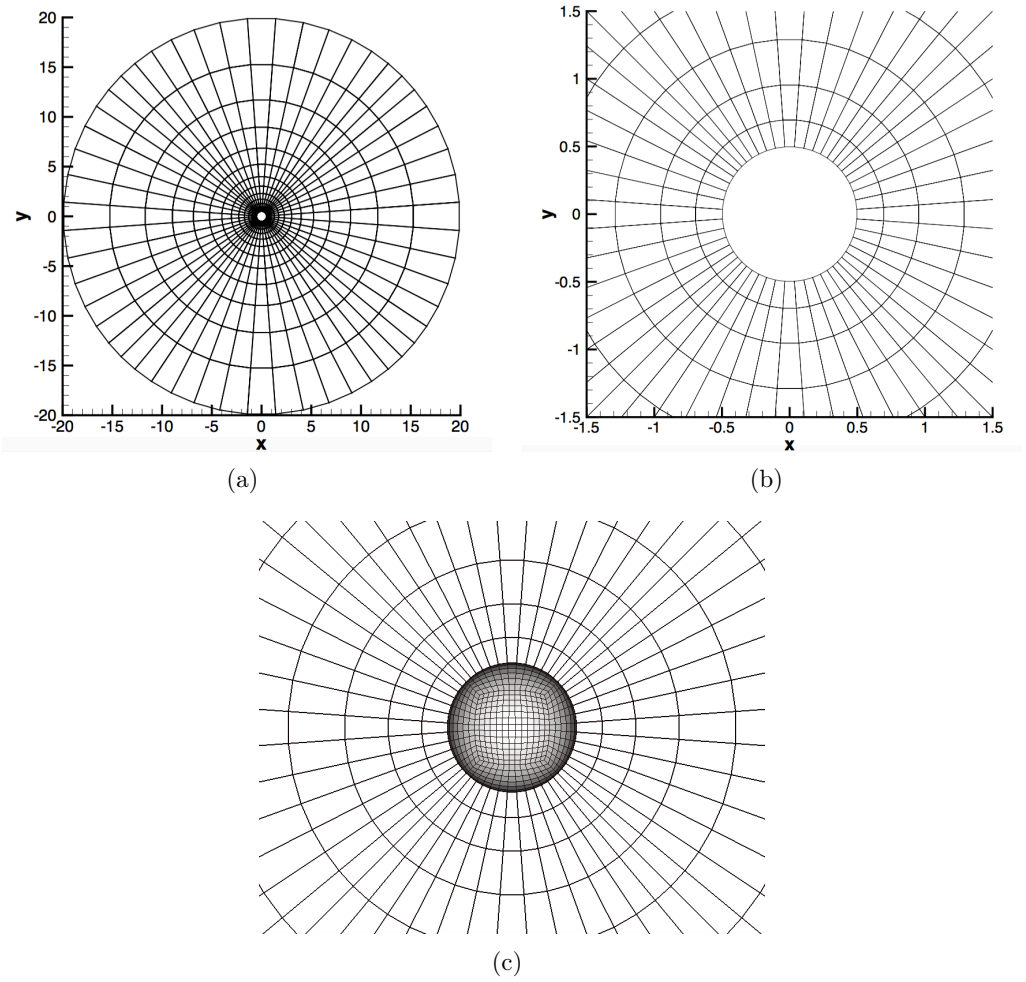
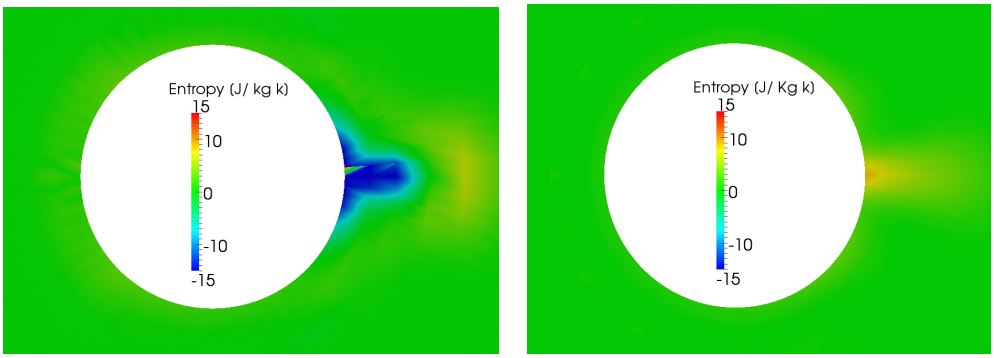


Figure 3.6: Mesh employed for the subsonic inviscid flow past a sphere test case. Figure (a) shows a slice which cuts the domain along the z axis. Figures (b), (c) are zooms which show the grid refinement around the sphere for both surface and volume mesh.

show the L_∞ errors associated to the entropy at steady state. The symbol ✗ denotes that the simulation failed to converge. It is possible to see how all the tests performed with collocated schemes (i.e. Q=P+1, where P is the polynomial order and Q is the number of quadrature points) failed, while using a higher quadrature produced stable simulations. Note that for the higher quadrature tests we used the

double of the quadrature points with respect to the polynomial order. This allowed keeping the same order of accuracy for the integration of the nonlinear terms in all the test cases. From the latter perspective, the use of P=4 and Q=6 for instance, produced an unstable simulation, because the integration errors introduced are higher than the case P=3, Q=6.



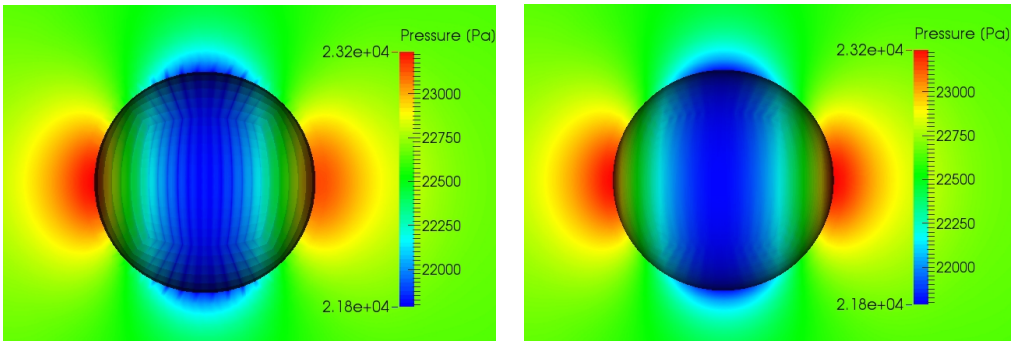
(a) (b)

Figure 3.7: Numerical entropy generation at the posterior stagnation point for P =2. On the left the collocated schemes. On the right the dealiased scheme.

To better illustrate this aspect, in Figure 3.7 we show the numerical entropy generated at the posterior stagnation point of the sphere for P=2 at the same physical time on a plane that cuts the domain along the central line. The image on the left (Figure 3.7(a)) shows the collocated case, while the image on right (Figure 3.7(b)) shows the non-collocated one.

For the collocated case it is possible to see a buildup of numerical entropy which will lead to the failure of the simulation after a few time-steps; the non-collocated case instead shows a more contained distribution of the numerical entropy which will be convected downstream and will lead to a stable numerical simulation. Note that, in the two figures, we kept the same colour gradient to the highlight differences in terms of entropy distribution. However, the maximum value of entropy for the collocated case is approximately two orders of magnitude bigger than the non-collocated case.

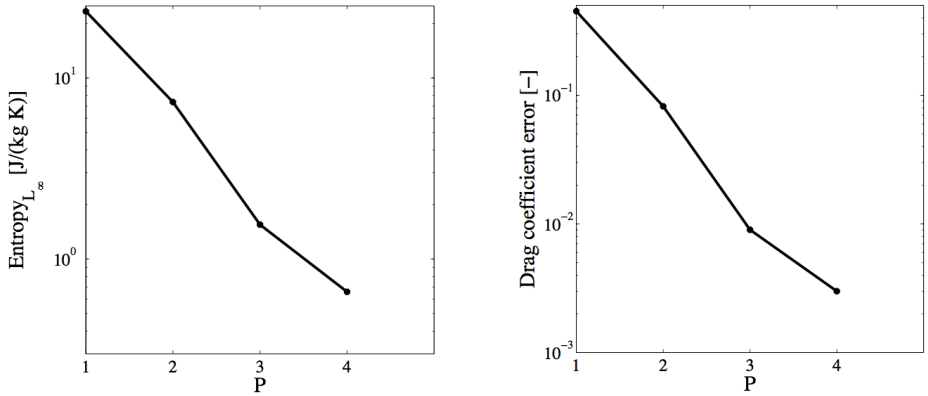
Figure 3.8 shows a comparison between the pressure distribution obtained using a low-order approximation of the geometry (Figure 3.8(a)) and using a high-order description (Figure 3.8(b)). Both the simulations are run with P = 4 and Q = 8. We can appreciate how, if we do not describe the geometry with enough accuracy, we lose all the benefits which derive from a high-order approximation of



(a) (b)

Figure 3.8: Pressure distribution around the sphere. On the left the mesh with low-order description of the geometry. On the right the mesh with high-order description. Both the cases are run with P = 4 and Q = 8.

the solution.



(a) (b)

Figure 3.9: Entropy L_∞ error vs. polynomial order (a). Drag Coefficient error vs polynomial order (b).

Figure 3.9(a) shows the L_∞ norm of the entropy error calculated on the entire domain as a function of the polynomial order. It is possible to note a decay of the error as the polynomial order increases, as expected. Finally, in Figure 3.9(b) the

error in the calculation of the drag coefficient is represented. This, according to the d'Alembert's paradox, should be zero and it is possible to appreciate a decrease of the error as we increase the polynomial order. These results were obtained using the dealiased DG scheme when the simulation reached the steady state (i.e. the relative variation of the solution between two successive time-steps was less than $10^{-}15$). The results show that the numerical method is accurate and reliable.

As we will see in chapter 4, it is clear from the result of this section that, when treating nonlinear problems, the under-integration of the nonlinear terms, which arises in the collocated methods, plays a crucial role in the numerical stability of a simulation, This under-integration leads to aliasing errors, which in turn can lead to the failure of the simulation. We have seen that, using a higher quadrature, i.e. using more quadrature points than a collocated method, improves the numerical stability of the simulations carried out. The dealiasing techniques based on consistent integration of the nonlinearities detailed in chapter 4 have been applied with success to the inviscid three-dimensional flow around a sphere.

In addition we remark the importance of an appropriate description of the geometry when dealing with curved surfaces. The high level of accuracy typical of high-order spectral element methods is strongly affected when we have a low-order representation of the geometry. The generation of high-order meshes on complex geometries is not trivial and, together with the robustness of high-order schemes, represents the main reason that has hitherto limited the adoption of these methods in both academia and industry.

3.5.2 Compressible Navier-Stokes equations

For verifying the Navier-Stokes equations we consider two test cases, a two-dimensional subsonic viscous flow over a flat plate and a three-dimensional subsonic viscous flow past a cylinder. With the first we aim to recover the Blasius solution for a low Mach number flow, whilst with the second we aim to predict the correct Strouhal number of the cylinder wake.

Subsonic viscous flow past a flat plate

The two-dimensional incompressible laminar flow which forms over a semi-infinite flat plate can be described by the following third-order ordinary differential equation and its associated boundary conditions which were derived by Blasius in 1908:

$$2f'''(\eta) + f''(\eta) = 0, \quad f(\eta = 0) = 0, \quad f'(\eta = 0) = 0, \quad f'(\eta = \infty) = 0\,, \qquad (3.65)$$

where $\eta = y\sqrt{U_\infty/(\nu x)}$ is the similarity variable, with x and y being the streamwise and wall-normal coordinates, respectively, ν being the kinematic viscosity and U_∞ being the free-stream velocity. Equation 3.65 can be solved numerically and from

its solution we can derive the flow variable values inside the boundary layer and on the flat plate surface. In particular the skin friction coefficient derived from the Blasius solution is defined as:

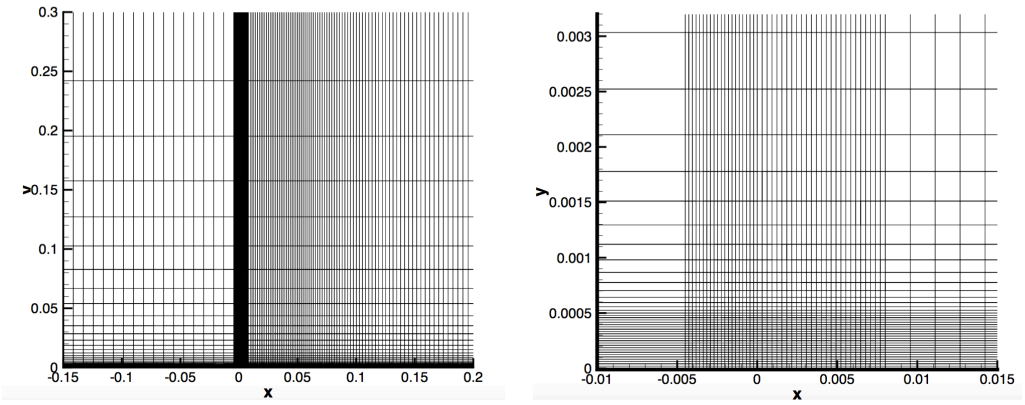
$$C_f = \frac{0.664}{\sqrt{Re_x}}, \tag{3.66}$$

where $Re_x = U_\infty x/\nu$ is the local Reynolds number based on the streamwise location. Note that a compressible solution can be recovered by using the transformation described by Schlichting (1979). In the following we show the results obtained on a nearly incompressible flow (Ma = 0.1) over a flat plate, for which we assume the Blasius solution to be valid. Specifically we compare the skin friction obtained by the Navier-Stokes solver with that calculated via Eq. 3.66.

The computational domain used is depicted in Figure 3.10. At the inlet region, located at $x = -0.15m$, the inflow characteristic boundary conditions were imposed, while at the outlet, located at $x = 0.2m$, the outflow boundary conditions were applied. In the region $x \in [-0.15; 0.0]m,\ \ y = 0m$, symmetry boundary conditions were set, while in the region $x \in [0.0; 0.2]m,\ \ y = 0m$, isothermal no-slip boundary conditions were applied. Finally, at the top boundary, located at $y \approx 100\delta_{99}$, where δ_{99} is the thickness[3] of the boundary layer at the outlet, the farfield boundary conditions were used. The mesh employed for the simulations is composed by $N_x \times N_y = 183 \times 48$ elements and is depicted in Figure 3.10. The scheme adopted for the spatial discretisations of all the simulations was the $\mathrm{FR_{SD}}$ method for both the convective and viscous terms. For the numerical fluxes associated to the convective term we used the exact Riemann solver described in section 3.3.1 and we applied the LDG numerical flux for the diffusion term. We applied both a collocation projection of the fluxes and a *Local* dealiasing technique described in chapter 4. The time-integration was performed through a fourth-order Runge-Kutta scheme and the time-step was approximately ten times lower than the time-step limit imposed by the Courant-Friedrichs-Lewy (CFL) condition (Courant (1967)). The solution and quadrature points were Gauss-Lobatto-Legendre type.

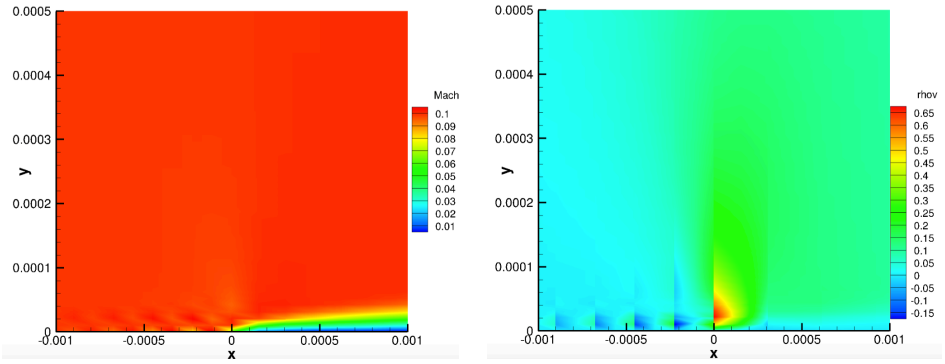
The Reynolds number chosen was equal to $Re = 1.6 \times 10^6$, based on the length of the flat plate which was equal to $L = 0.2m$, while the Prandtl number Pr was 0.72. The thermodynamic parameters, pressure and density, were $p_\infty = 22600Pa$, $\rho = 0.364kg/m^3$. Figure 3.11 shows the contour of the Mach number and of the y-component of the momentum at the leading edge of the flat plate. From Figure 3.11(b) it is possible to see that the resolution at the leading edge is not sufficient to describe the solution and therefore we expect to encounter under-resolution issues, especially if we use the discontinuous spectral/hp element methods in a collocated fashion. In Table 3.2, we report the differences between the skin friction

[3]In this study the boundary-layer thickness is defined as the distance from the wall to the point where the velocity is 99% of the free-stream velocity



(a) (b)

Figure 3.10: Mesh employed for the flat plate simulations. On the left the full domain. On the right a zoom which shows the grid refinement around the leading edge.



(a) Mach (b) ρv

Figure 3.11: Mach and y-component of the momentum at the leading edge of the flag plate.

coefficient derived from the Blasius solution of Eq 3.65 and the numerical solution with the compressible Navier-Stokes solver implemented in Nektar++. In the table all the different combinations of polynomial order and number of quadrature points are written. This error, also referred to as global L_2 error, represents the

	L_2 error for C_f [-]		
	P=1 $Q_V = 2$ $Q_I = 2$	P=2 $Q_V = 3$ $Q_I = 3$	P=3 $Q_V = 4$ $Q_I = 4$
Collocated	0.27	✗	✗
	P=1 $Q_V = 3$ $Q_I = 3$	P=2 $Q_V = 4$ $Q_I = 4$	P=3 $Q_V = 6$ $Q_I = 6$
Local dealiasing	0.18	0.039	0.068

Table 3.2: L_2 errors associated skin friction coefficient C_f for the flat plate simulations.

difference between the two curves along the entire flat plate (excluding the point x=0 which is singular). It is possible to see that the simulations carried out using the collocated schemes failed to converge and, when converging (i.e. for P=1 and Q=2), provided the highest L_2 error. This confirms what expected - the under-resolution at the leading edge couples with the integration errors associated to the collocated schemes leading to unstable simulations. On the other hand, the non-collocated simulations provided lower L_2 errors and they all converge. Note that

	L_2 error for C_f [-]		
	P=1 $Q_V = 2$ $Q_I = 2$	P=2 $Q_V = 3$ $Q_I = 3$	P=3 $Q_V = 4$ $Q_I = 4$
Collocated	0.013	✗	✗
	P=1 $Q_V = 3$ $Q_I = 3$	P=2 $Q_V = 4$ $Q_I = 4$	P=3 $Q_V = 6$ $Q_I = 6$
Local dealiasing	0.011	0.0017	0.00023

Table 3.3: L_2 errors associated skin friction coefficient C_f for the flat plate simulations at $x = 0.02m$ (x/L = 0.1).

the errors reported in the table are driven to higher values than those obtained if excluding a few points in proximity to the leading edge. Interestingly, the global L_2 error increases from P= 2 to P= 3. This is because the under-resolved leading edge introduces more marked spikes in the solution for higher polynomial orders. However, we recover a decreasing trend for the local error when increasing the polynomial order, if we measure a local L_2 error downstream of the leading edge. From this perspective, in Table 3.3, we report the L_2 errors at $x = 0.02m$. We can appreciate that the errors decrease approximately by one order of magnitude than the global L_2 errors presented in Table 3.2 and that the errors decrease as the polynomial order increases.

To better illustrate the leading edge under-resolution errors, in Figure 3.12, we show the skin friction coefficient calculated for two non-collocated simulations with P=1 and P=3 against the skin friction obtained from the Blasius solution. It is possible to see that the leading edge is particularly badly resolved when we use a low-order polynomial like P = 1, whilst it becomes well resolved for a higher

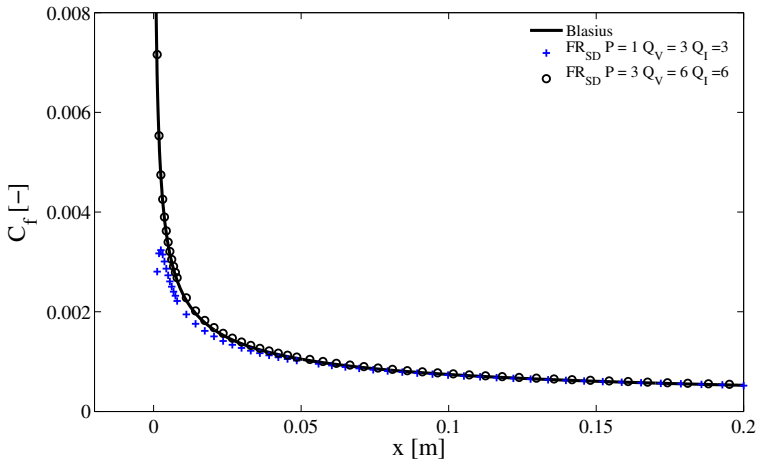
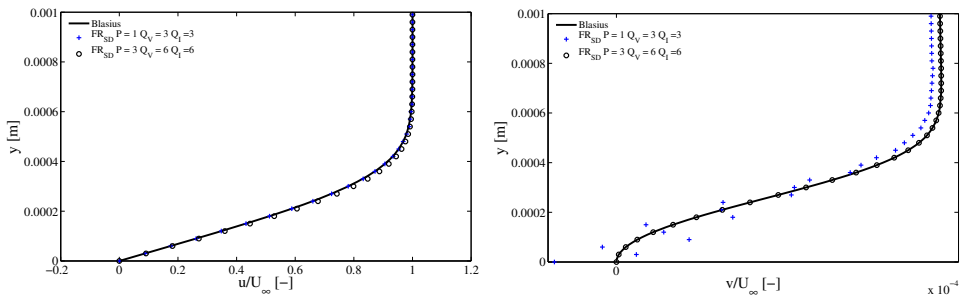


Figure 3.12: Skin coefficient on the flat plate.

polynomial order like P=3. Finally, in Figure 3.13, we show the x- and the y-component of the velocity, u and v, (normalised with respect to the free-stream velocity) as a function of the wall-normal coordinate y. We compare the Blasius solution with that obtained with Nektar++ for P = 1 and P = 3 at $x = 0.1m$. We can see that, while the case P=1 provides a poor solution for the wall-normal component, the case obtained with P=3 recovers the Blasius solution well, for both the x- and the y-component of the velocity.



(a) u (b) v

Figure 3.13: Velocity components at $x = 0.1m$.

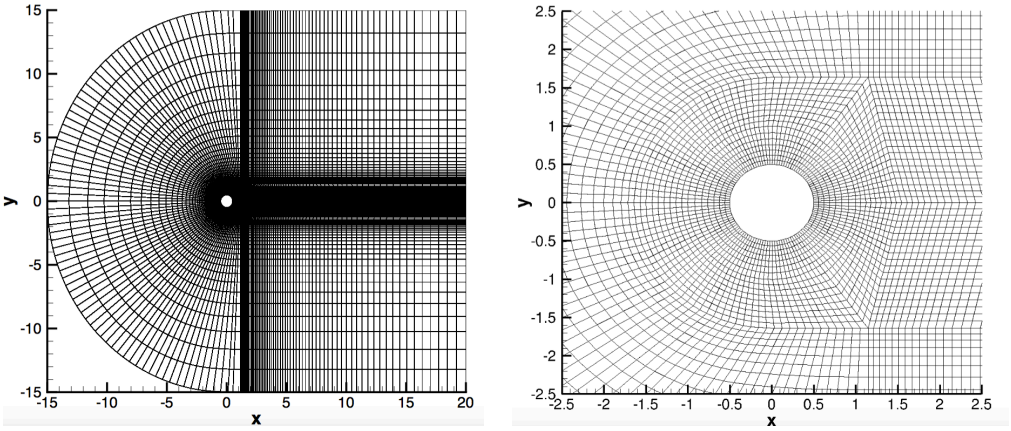
The results of this section confirm what we have already seen for the inviscid subsonic flow past the sphere. When we perform an under-resolved simulation, the

errors related to the under-integration of the nonlinear terms may lead to unstable simulations and the dealiasing techniques like those described in chapter 4 become crucial for the robustness of the simulation.

Subsonic viscous flow past a cylinder

We tested the subsonic viscous flow over a circular cylinder at $Re = \frac{\rho U_\infty D}{\mu} = 3900$. The mesh used is depicted in Figure 3.14 and is composed of approximately 80000 hexahedral elements with 16 elements in the spanwise direction. The cylinder surface is approximated by fourth order polynomials.

For all the simulations we used a Mach number equal to $M_\infty = 0.2$, with the pressure set to $p = 22600\ Pa$ and the density equal to $\rho = 0.364\ kg/m^3$. The additional parameters related to the viscous term are the Reynolds number, which was set to $Re = 3900$ (based on the diameter of the cylinder $D = 1m$) and the Prandtl number, which was set to $Pr = 0.72$. For the time-integration we used a fourth-order Runge-Kutta scheme, the numerical fluxes were computed through an exact Riemann solver (see section 3.3.1) and we employed the DG method both with collocation projection of fluxes and with *Global* dealiasing technique representing the solution with a third order polynomial. The solution and quadrature points were Gauss-Lobatto-Legendre type.



(a) (b)

Figure 3.14: Mesh employed for the subsonic viscous flow past a cylinder test case. Figure (a) shows a slice which cuts the domain along the z axis. Figure (b), is a zoom which shows the grid refinement around the cylinder.

Figure 3.15 shows the Q-criterion contour around the cylinder at Re = 3900 with Q defined as:

$$Q = -\frac{1}{2}\left(S_{ij}S_{ij} - \Omega_{ij}\Omega_{ij}\right) = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i}, \tag{3.67}$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = Sym\left(\frac{\partial u_i}{\partial x_j}\right), \tag{3.68}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) = Asym\left(\frac{\partial u_i}{\partial x_j}\right) \tag{3.69}$$

and u is the generic component of the velocity. The Q-criterion measures the excess of rotation rate over the strain rate magnitude in all directions. In regions where rotation predominates over the strain $Q > 0$. In regions where strain is larger than rotation $Q < 0$. It is possible to appreciate the fully turbulent wake generated behind the cylinder by the complex three-dimensional vortex shedding. The frequency f_s of the shedding can be calculated and compared against available experimental and numerical data to show the reliability of the algorithms adopted. Specifically, we measured the so-called Strouhal number, defined as follows:

$$St = \frac{f_s D}{U_\infty} \tag{3.70}$$

where D is the cylinder diameter and U_∞ is the free stream velocity.

In Table 3.4 we report the results relative to the Strouhal number of the simulations performed compared with a LES of Kravchenko and Moin (2000) and a DNS of Ma et al. (2000) at the same Reynolds number.

Strouhal number [-]						
P=3 Q = 4	P=3 Q =5	P=3 Q =6	P=3 Q = 7	P = 3 Q = 8	Kravchenko	Ma
✗	✗	✗	✗	0.215	0.210	0.219

Table 3.4: Strouhal number evaluation with a fourth order accurate solution increasing the number of quadrature points.

As before, the symbol ✗ denotes that the simulation diverged. It is possible to see how it was necessary to use a number of quadrature point $Q = 2(P + 1)$ in order to stabilise the simulation, which is the number of quadrature points necessary to exactly integrate a cubic nonlinearity like those present in the Navier-Stokes equations. We can see there is a good agreement with the other results found in the literature.

We ran the simulation for ≈ 800 convective times starting from an initial solution equal to the farfield condition. After that we averaged the flow quantities

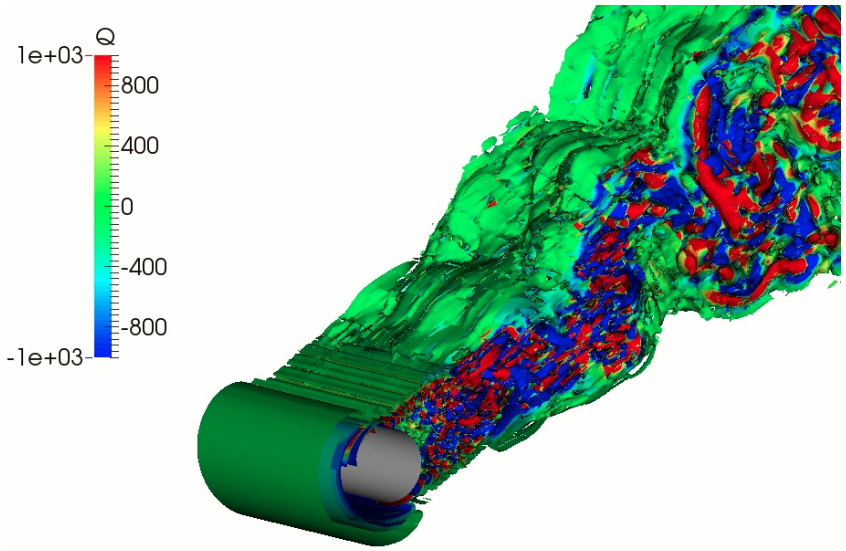
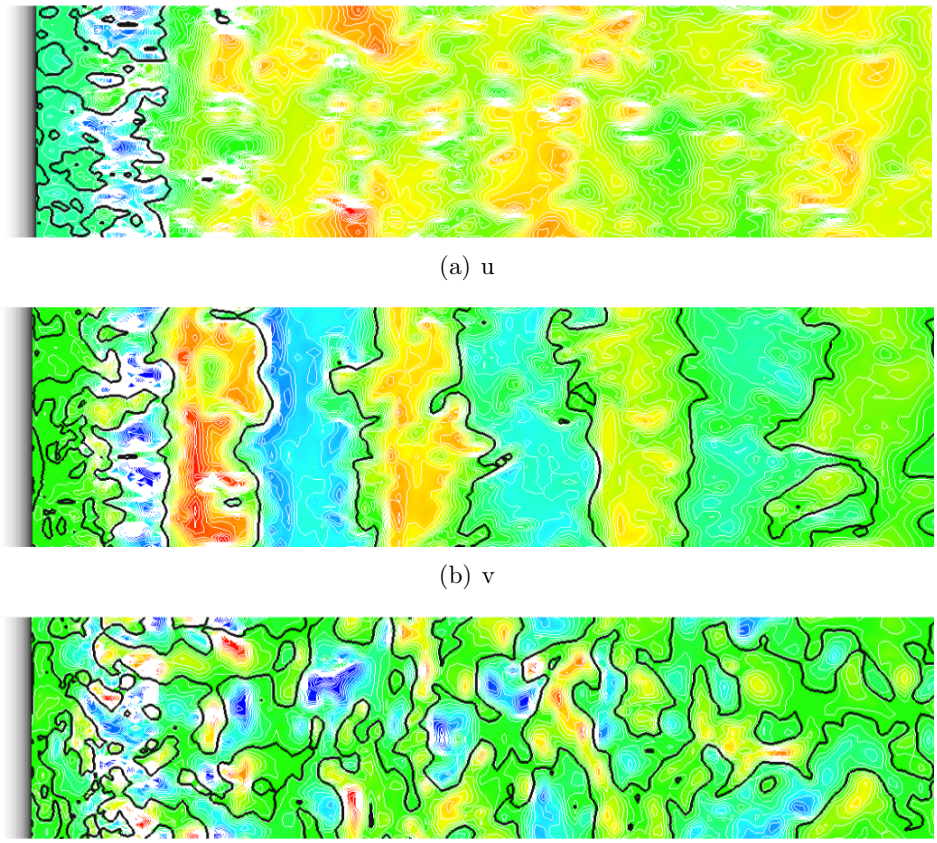


Figure 3.15: Q-criterion contour of the Navier Stokes simulation of the cylinder at Re = 3900.

using an extended average time of 400 convective times which corresponds to approximately 80 shedding periods. Figure 3.16 shows the instantaneous streamwise, crosswise and spanwise velocities in the $(x - z, y = 0)$ plane in the wake of the cylinder.

Figure 3.17(a) shows the averaged profile of the streamwise velocity in the centerline of the cylinder wake, whilst Figures 3.17(b), 3.17(c), 3.17(d) show the time-span-average streamwise velocity profiles at different locations in the wake. In particular, recently, Lehmkuhl et al. (2013) demonstrated that the wake for this test case fluctuates between a high energy mode and a low energy mode. From Figure 3.17 we can see a good agreement of the results with the low energy mode of other numerical (Witherden et al. (2015); Lehmkuhl et al. (2013)) and experimental (Parnaudeau et al. (2008)) data. The agreement of the results only with the low energy mode can be explained with the time interval used for the the averaging and the refinement of the grid different from the other numerical experiments.

As for the flat plate experiment, in this case it is even more evident the need for a consistent integration of the nonlinear terms. In fact, due to the high Reynolds number of the turbulent flow we remark that, for increasing the stability of the simulation we needed a number quadrature points $Q = 2(P + 1)$. This is in



(a) u

(b) v

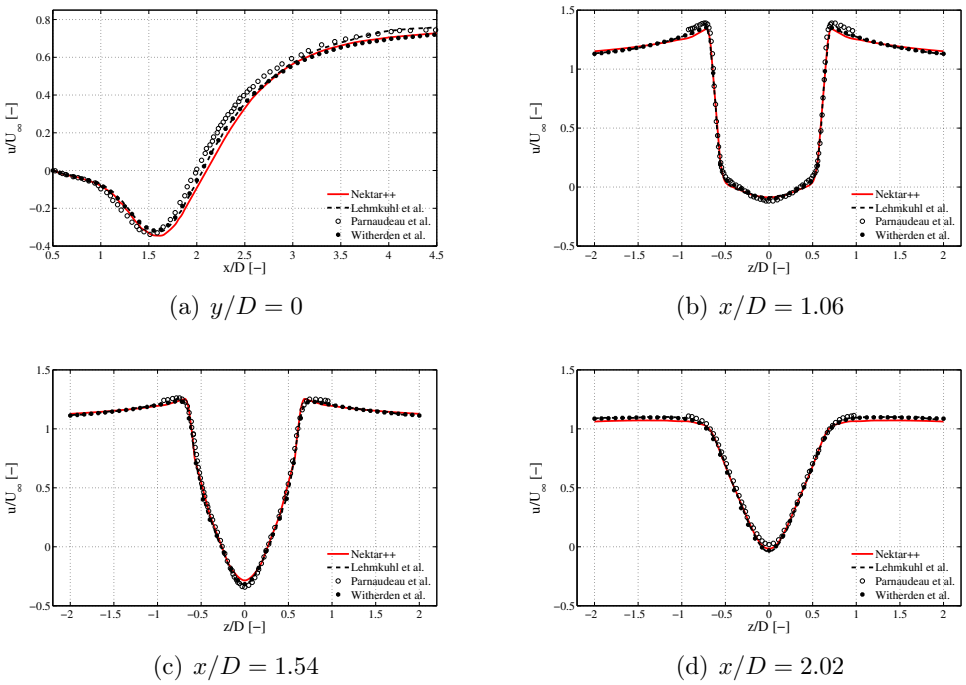
(c) w

Figure 3.16: Instantaneous velocity components in the $(x - z, y = 0)$ plane in the wake of a circular cylinder at $Re = 3900$. The thick black solid lines shows the contour where the velocity components are zero.

agreement with that reported in Table 4.1 for the cubic nonlinearity present in the Navier-Stokes equations and is higher than that needed for the laminar flat plate case for the same polynomial order of the solution.

3.6 Summary

In the previous sections we have detailed some experiments performed for testing the Compressible Flow solver. The cases studied consider both two- and three-dimensional flows either inviscid or viscous. We have seen that the solver is accurate for predicting high Reynolds number flows like those that we want to



(a) $y/D = 0$ (b) $x/D = 1.06$

(c) $x/D = 1.54$ (d) $x/D = 2.02$

Figure 3.17: Averaged profile of the streamwise velocity in the centerline of a circular cylinder wake at $Re = 3900$ (a).Time-span-average streamwise velocity profiles at different locations in the wake of the circular cylinder at $Re = 3900$ (b, c, d).

investigate. We want to remark the importance of a correct high-order representation of the geometry investigated which can otherwise decrease the accuracy of the method. In addition it is clear that, especially for under-resolved simulations (which is often the case when considering high-Reynolds number flows), the numerical errors introduced by the under integration of the nonlinear terms, also referred to as aliasing errors, can easily lead to the failure of the simulation. In this case the dealiasing techniques which will be described in chapter 4 can always be applied to increase the robustness of the simulation either if we use a DG or a FR discretisation of the equations. In the next chapter we will describe in detail the dealiasing techniques applied in this thesis work.

Chapter 4

Dealiasing strategies for spectral/hp element methods

In this chapter, we investigate the aliasing issues arising in discontinuous spectral/hp element methods - specifically in the DG method and in the FR approach. We identify the aliasing sources appearing in both discretisations presenting two strategies to tackle these issues. Specifically we detail two techniques based on the concept of consistent integration of the nonlinearities. The first is a localised approach which targets the nonlinear flux functions in the problem, the second is a global approach which targets both the nonlinear flux functions and the deformed/curved geometry. Part of the contents of this chapter has been published by the author of this thesis and his co-workers in Mengaldo et al. (2015b).

4.1 Introduction

Numerical stability is a fundamental requirement for a numerical tool (e.g. a computational fluid dynamics solver) especially when used in an engineering design process where reliability and robustness are key factors for its efficacy in the industrial pipeline. While low-order finite element methods demonstrate a usually satisfying level of numerical stability due to their dissipation properties, high-order spectral element methods typically exhibit low dissipation errors, and are therefore affected by a lack of stability which in turn affects their robustness and reliability.

Over the last decade, various attempts have been made to address this problem. Different formulations of the nonlinear terms have different aliasing properties. In Blaisdell et al. (1996) it is shown that the skew-symmetric form of the convective term results in a reduced amplitude of the aliasing errors in comparison to the conservative and nonconservative forms. Examples of the skew-symmetric methodology for discontinuous Galerkin spectral element methods can be found in Gassner (2013a, 2014). Approaches such as spectral vanishing viscosity (Karamanos and

Karniadakis (2000)) modify the underlying discretised operators in order to add artificial dissipation at high polynomial orders, thereby reducing high-frequency oscillations in the approximate solution, stabilising the method and making it suitable for high Reynolds number large-eddy simulations (Kirby and Karniadakis (2002); Pasquetti (2006); Kirby and Sherwin (2006a); Xu (2006)). Polynomial filtering (Gottlieb and Hesthaven (2011); Fischer and Mullen (2001); Fischer et al. (2002); Hesthaven and Kirby (2008)) adopts a similar strategy, in which an interpolation-based filter is applied locally to each element at each time-step to prevent the buildup of unwanted oscillations in the solution field. In Warburton (2013) a DG method with guaranteed stability is obtained using a non-polynomial modified basis. Variational multiscale schemes (Hughes et al. (1998)) offer an alternative form of stabilisation based on the model used to approximate coupling between unresolved and small resolved scales (Wasberg et al. (2009); Farhat et al. (2006); Marras et al. (2012)). Finally, in Gassner (2013b) the impact of an exponential based modal filtering and of over-integration on underresolved high-order computations of turbulent flows is investigated.

In this chapter we focus on the concept of over-integration (Kirby and Karniadakis (2003), Deville et al. (2002)), where over-integration refers to the use of more quadrature points than are necessary for a linear operator, but which might equally well be considered as consistent integration of the non-linear operators. For further details of the impact of over-integration on problems with non-constant coefficients and deformed geometries see Maday and Rønquist (1990) and Kopriva (2006). In the formulation of spectral element methods, on any given element one may choose the number of expansion modes used to represent the approximation of a function, and the number of quadrature points used to calculate numerical estimates of integrals which may appear in the formulation.

A popular choice, as we have already seen, is to couple a set of quadrature points with an equal number of polynomials (such as Lagrange polynomials) defined at the same points, leading to a collocation method. There are many examples of this throughout the literature, both in terms of the more traditionally utilised continuous Galerkin (CG) and discontinuous Galerkin (DG) formulations, as well as newer extensions such as the flux reconstruction (FR) technique as presented by Huynh (2007). In collocation methods, while most linear operators can be exactly integrated in this setting depending on the choice of quadrature, integrals of nonlinear terms typically incur numerical error. However, the computational efficiencies that can be attained through the use of a collocation formulation, in particular the presence of a diagonal mass matrix, often outweigh the numerical error that is incurred.

To illustrate this point, it is well-known that polynomials of order P can be exactly integrated up to machine precision given some minimum number of quadrature points Q_{min} (Karniadakis and Sherwin (2005)); for example Gauss-Lobatto-

Legendre quadrature, which is widely adopted in the context of spectral element methods, allows one to obtain exact values of integrals for order $2P-3$ polynomials given P quadrature points (Orszag (1972)). However, when nonlinear quantities are calculated at the quadrature points, a collocation projection leads to an error known as polynomial aliasing being introduced into the obtained integrals due to an insufficient number of quadrature points. This numerical error may be negligible for well-resolved simulations, for example flow simulations at low Reynolds numbers, but can become a critical issue for under- and marginally-resolved simulations such as typically arise in large eddy simulations (LES). In addition, when we have non-polynomial functions (such as the compressible Euler and Navier-Stokes equations fluxes written in conservative form), consistent integration cannot be applied with a specific rule as in the case of polynomial functions. In this case, it is still possible to reduce consistently the aliasing errors but the number of quadrature points required might become computationally too costly.

The aim of this chapter is to illustrate effective and computationally efficient approaches to resolve the aliasing issues arising in discontinuous spectral/*hp* element discretisations such as the DG method and the FR approach. Specifically, for a scalar PDE, we can write these two formulations by separating the linear terms from the nonlinear ones, as follows:

$$\begin{aligned} \text{DG}: \quad & \frac{du}{dt} = \mathbf{M}^{-1} \left(\mathcal{L}_V + \mathcal{L}_I + \mathcal{N}_V + \mathcal{N}_I \right) , \\ \text{FR}: \quad & \frac{du}{dt} = \mathcal{L}_V + \mathcal{L}_I + \mathcal{N}_V + \mathcal{N}_I , \end{aligned} \tag{4.1}$$

where u is the approximate solution field, $\mathbf{M}$ is the mass matrix, $\mathcal{L}$ is the linear term and $\mathcal{N}$ is the nonlinear term. Both linear and nonlinear terms are composed by a volumetric term (V) over each spectral element and an interface term (I) on the trace space connecting elements. Henceforth in the term $\mathcal{N}$ we are including the aliasing issues due to both nonlinear and "variable coefficient[1]" fluxes as well as to a possible curved geometry. Both, nonlinear/variable coefficient fluxes and spatially varying geometric terms can result in aliasing errors if not exactly integrated and may generate aliasing-driven numerical instabilities.

To address these aliasing issues we consider two different strategies both based on the concept of consistent integration. With the first approach we *locally* consistently integrate only the nonlinear terms produced by the the nonlinear or "variable coefficient" fluxes present in the partial differential equation (PDE) itself. Eq. (4.1)

[1]Note that the advection equation with spatially varying coefficients used in section 4.4 is still a linear equation (the advection velocity does not depend on the solution). In this case the aliasing depends on the non constant velocity coefficients.

can therefore be re-written as:

$$\begin{aligned} \text{DG}: \quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_{Q_V} + \mathcal{N}_I\Big|_{Q_I} \right), \\ \text{FR}: \quad & \frac{du}{dt} = \mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_{Q_V} + \mathcal{N}_I\Big|_{Q_I}, \end{aligned} \tag{4.2}$$

where Q is the number of quadrature points used for the linear term and the mass matrix, and Q_V and Q_I indicate the number of quadrature points used for the consistent integration of the volumetric and the interface part of the nonlinear term respectively. We note the local behaviour of the strategy, which allows us to selectively adjust the quadrature used for the nonlinear term.

In the second approach, we consistently integrate every term of the numerical discretisation which typically implies we have over-integrated the linear terms. In this strategy we can address both PDE-aliasing driven instabilities as well as aliasing sources arising from deformed/curved elements, which we refer to as geometrical-aliasing sources. Eq. (4.1) becomes:

$$\begin{aligned} \text{DG}: \quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_Q + \mathcal{N}_I\Big|_Q \right), \\ \text{FR}: \quad & \frac{du}{dt} = \mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_Q + \mathcal{N}_I\Big|_Q, \end{aligned} \tag{4.3}$$

where the same number of quadrature points is used *globally* for all the terms. To summarise, the main contributions of this chapter are:

- Generalised analysis which encompasses DG, FR discretisations, highlighting the contribution of interface and volumetric terms.
- Local dealiasing strategy which exploits sum-factorisation type approach to improve computational efficiency.
- Comparison of geometrical- and PDE-aliasing.

The chapter is structured as follows. In section 4.2 we briefly summarise the different DG and FR discretisations, and highlight the sources of aliasing errors which arise in each formulation. Section 4.3 outlines the *Local* and *Global* dealiasing strategies we adopt to reduce the aliasing errors and improve the stability of the different discretisations considered. In this section we also describe the implementation details which are required to apply such methodologies within a spectral/hp framework. We present applications to some numerical examples in section 4.4 for each discretisation by highlighting key examples in compressible and incompressible flows. Finally, in section 4.5, we draw a brief summary and the conclusions.

4.2 Aliasing sources in spectral element methods

Aliasing errors in polynomial spectral element methods typically arise through an under-integration of terms appearing in the weak form of the equations being discretised. When using Gauss-Lobatto-Legendre (GLL) quadrature points, for instance, the minimum number of quadrature points Q_{min} necessary to exactly integrate any given order P polynomial $u(\xi) \in \mathbb{P}^P$ up to machine precision is

$$Q_{\text{min}} = \frac{P+3}{2}. \tag{4.4}$$

In Galerkin methods, we are usually interested in integrating the L^2 inner product of two polynomials (ϕ_p, ϕ_q) in order to compute the mass matrix of our discretised problem, where $\phi_p(\xi), \phi_q(\xi) \in \mathbb{P}^P$ are the so-called test or virtual functions, which in our studies are chosen to be the same of the solution expansion. By using Eq. (4.4) it is possible to calculate the minimum number of GLL quadrature points necessary for the quadrature to be exact as a function of the polynomial order P, as shown in Table 4.1.

Polynomial order P	Q_{min}
$[\phi(\xi)]^2 \in \mathbb{P}_{2\text{P}}$	$Q \geq P + 3/2$
$[\phi(\xi)]^3 \in \mathbb{P}_{3\text{P}}$	$Q \geq 3P/2 + 3/2$
$[\phi(\xi)]^4 \in \mathbb{P}_{4\text{P}}$	$Q \geq 2P + 3/2$

Table 4.1: Number of GLL quadrature points for the GLL quadrature to be exact up to machine precision as a function of the polynomial order of the integrand.

We note that to exactly integrate a linear problem it is necessary to use $Q_{\text{min}} = P + 2$ GLL quadrature points. For nonlinear problems the number of quadrature points needed increases and for quadratic nonlinearities such as the convective term of the incompressible Navier-Stokes equations it becomes $Q_{\text{min}} = \frac{3}{2}(P+1)$ whereas for cubic nonlinearities such as those present in the compressible Navier-Stokes equations (if written in primitive variables) $Q_{\text{min}} = 2(P+1)$ quadrature points are needed for the numerical integration to be exact. In case of compressible flows (Euler/full Navier-Stokes equations) written in conservative variables the flux functions are rational. Due to their non-polynomial nature it is not possible to exactly integrate the resulting flux functions by a Gauss-type quadrature. However it is possible to reduce the integration error to machine precision by choosing an integration quadrature rule with a sufficiently high degree.

In addition to the nonlinearities present in the equations (PDE-aliasing sources), additional aliasing may arise from the use of deformed or curved elements. The Jacobian of the isoparametric mapping, which takes a standard element and projects it

into the Cartesian space, is by definition a non-constant, non-linear polynomial for curved elements. This introduces a geometrical-aliasing source which may corrupt the solution in a similar manner as the PDE-aliasing and may eventually increase the overall degree of the nonlinearity.

In this section we consider two different high-order methods, namely the discontinuous Galerkin (DG) method and the flux reconstruction (FR) approach. However, the dealiasing techniques presented can be applied to any spectral element method, including the continuous Galerkin approach. In the following, for both, the DG and the FR approaches, we highlight the sources of PDE- and geometrical-aliasing by considering the multidimensional conservation law

$$\frac{\partial u}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(u) = 0, \tag{4.5}$$

where $u = u(\boldsymbol{x}, t)$ is the conserved variable and $\boldsymbol{f}(u) = [f_1(u), f_2(u), f_3(u)]^{\mathrm{T}}$ is the flux vector which governs the transport of u.

Note that the form of the advection term in Eq. (4.5) can affect the numerical stability. There is a rich literature, especially for the incompressible Navier-Stokes equations, where the convective term is represented in a skew-symmetric form in order to enhance stability despite losing the computational efficiency[2] of the conservative form of the flux. In this regard, Malm et al. (2013) detailed the relationship between consistent integration and the skew-symmetric form of the convection operator, pointing out that for any Galerkin-based method the use of consistent integration recovers imaginary eigenvalues (i.e. skew-symmetry of the convective operator) and therefore stability. In this work we do not consider other possible choices for improving stability than consistent integration. Nevertheless, the connections between consistent integration and the skew-symmetric form of the convective term remains an interesting aspect which needs to be further explored (see also Gassner (2013a, 2014)).

4.2.1 DG formulation

The matrix form of the weak DG method applied to Eq. 4.5 was derived in chapter 1 and it is reported here for the sake of clarity

$$\mathbf{M}\frac{\mathrm{d}\widehat{\mathbf{u}}}{\mathrm{d}t} - \Bigg\{ \Bigg[\Bigg(\sum_{i=1}^{3} \boldsymbol{\Lambda}(G_{i/1}) \mathbf{D}_{\xi_i} \Bigg) \mathbf{B} \Bigg]^{\mathrm{T}} \mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_1^D - \Bigg[\Bigg(\sum_{i=1}^{3} \boldsymbol{\Lambda}(G_{i/2}) \mathbf{D}_{\xi_i} \Bigg) \mathbf{B} \Bigg]^{\mathrm{T}} \mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_2^D -$$

$$\Bigg[\Bigg(\sum_{i=1}^{3} \boldsymbol{\Lambda}(G_{i/3}) \mathbf{D}_{\xi_i} \Bigg) \mathbf{B} \Bigg]^{\mathrm{T}} \mathbf{W}\mathbf{B}\widehat{\mathbf{f}}_3^D \Bigg\} + \widehat{\mathbf{b}}^{\mathrm{DG}} = 0,$$

[2] The skew-symmetric form requires the nonlinear term to be evaluated twice and, therefore, it is more computationally expensive than the conservative form.

From this equation it is possible to identify the aliasing sources arising in the context of the DG method. Specifically, the flux functions $\widehat{\mathbf{f}}_1^D$, $\widehat{\mathbf{f}}_2^D$ and $\widehat{\mathbf{f}}_3^D$ are responsible for PDE-aliasing sources. If they are nonlinear or quasi-linear (i.e. spatially varying coefficient), the overall degree of the integrand will increase leading to PDE-aliasing errors if the number of quadrature points is not incremented to tackle the degree of the nonlinearity.

On the other hand, the mass matrix $\mathbf{M}$ through the Jacobians and the geometric factors $\partial\xi_i/\partial x_j$ appearing in the volumetric and boundary terms can generate geometrical aliasing when the mesh is deformed or curved. In these cases in fact, the Jacobians and the geometric factors are spatially varying inside the elements and they are eventually represented through a polynomial of a certain degree depending on the curvature of the edges/faces. These polynomials couple with the other terms in the discretised equations leading to integrand functions of a higher degree. Therefore, the number of quadrature points necessary to integrate exactly these higher-degree functions will increase and if not taken into account will produce geometrical-aliasing errors.

As already mentioned, finding the exact number of quadrature points necessary for the integration to be exact might not be trivial. In particular, if the nonlinearities are rational functions, a fixed number of quadrature points for the exact integration does not exist and this may affect the performance of the solver.

Note also that one of the peculiarities of a discontinuous discretisation like the DG method, is the boundary term. This term introduces aliasing issues which, depending on the simulation carried out, may be bigger or smaller than those due to the volumetric term. In this work, we have investigated this issue, by evaluating separately the contribution of the boundary term and of the volumetric term to the overall aliasing errors.

4.2.2 FR formulation

As for the DG method, we show again here the matrix form of the FR approach applied to Eq. 4.5 obtained in chapter 1:

$$\frac{d\widehat{\mathbf{u}}}{\mathrm{d}t} + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}(G_{i/1})\mathbf{D}_{\xi_i}\right]\widehat{\mathbf{f}}_1^D + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}(G_{i/2})\mathbf{D}_{\xi_i}\right]\widehat{\mathbf{f}}_2^D + \\ + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}(G_{i/3})\mathbf{D}_{\xi_i}\right]\widehat{\mathbf{f}}_3^D + \widehat{\mathbf{b}}^{\mathrm{FR}} = 0$$

in order to highlight the nonlinearity sources. As in the case of the DG method, PDE-aliasing sources are related to the flux functions $\widehat{\mathbf{f}}_1^D$, $\widehat{\mathbf{f}}_2^D$ and $\widehat{\mathbf{f}}_3^D$, if they are either nonlinear or they have spatially varying coefficients (such as the case of

the quasi-linear advection equation, where the advection velocities are spatially varying across the domain). On the other hand, the geometrical nonlinearities are related to the metric terms $\partial \xi_i / \partial x_j$ and the Jacobians.

As we have seen in the previous chapter, the DG and the FR approaches are identical for particular choices of the correction functions. This indicates that the aliasing sources are the same and, when the nonlinearities are polynomials, the aliasing errors can be completely eliminated using the appropriate number of quadrature points. On the other hand, it also indicates that the dealiasing techniques presented in this chapter can be applied to the overall class of FR schemes recovered by the different forms of the correction functions presented in chapter 1.

Finally, the consideration made for the boundary term of the DG method holds for the FR approach too. Specifically, we have evaluated separately the contribution of the boundary and volumetric terms to the overall aliasing errors for one of the cases considered.

4.3 Dealiasing strategies

In this section we describe the two dealiasing approaches considered in this work, which are both based on the concept of consistent integration. These are:

- *Local dealiasing*: PDE-dealiasing through consistent integration of the nonlinear terms[3];

- *Global dealiasing*: PDE- and geometrical-dealiasing through consistent integration of all the terms of the discretisations.

The first takes into account only PDE-aliasing errors and it locally uses a consistent integration of the nonlinear terms of the PDE. The second takes into account both PDE- and geometrical-aliasing effects.

We remark here that the word consistent integration refers to polynomial nonlinearities, where it is effectively possible to know a priori the number of quadrature points for the integration to be exact, thus consistently integrate the nonlinearities. When non-polynomial functions are present, the concept of consistent integration is out of focus because it is not possible to fully control the quadrature error, although it might still be possible for a higher number of quadrature points.

[3]Note that, as aforementioned, in these nonlinear terms we are including both nonlinear fluxes and variable coefficient fluxes. The second are linear but they are introducing aliasing errors because of the spatial varying nature of the coefficients.

4.3.1 Local dealiasing

The first dealiasing approach presented consists of consistent integration of the nonlinear terms in Eq. (4.5). This approach can be applied to both the discontinuous formulations previously considered, and addresses the aliasing effects arising from the PDE discretisation. In the following, we first consider a one-dimensional case before outlining the extension to two- and three-dimensions through a tensor product expansion. The implementation presented here has been carried out in the spectral/*hp* element library *Nektar++*. This implementation, and in particular its tensor product extension in both structured and unstructured sub-domains, is computationally efficient for high polynomial orders due to its use of the sum factorisation technique.

Local dealiasing in one dimension

Consider a one-dimensional approximate polynomial solution $u(\xi)$ of order P represented by an expansion in terms of nodal polynomial functions $\ell_i(\xi)$, such as Lagrange polynomials, on a set of GLL points.

$$u^{\delta Q_P}(\xi) = \sum_{i=0}^{P} u_i \ell_i(\xi), \qquad Q_P = P + 1. \tag{4.6}$$

The dealiasing strategy consists of 3 steps.

A.) *One-dimensional interpolation*

The solution $u(\xi)$ is interpolated onto a larger set of points $Q = \widetilde{P}+1$ where $\widetilde{P} > P$ (note that this operation does not change the order of the polynomial which remains unaltered and equal to P):

$$u^{\delta Q}(\xi_j) = \sum_{i=0}^{P} u_i \ell_i(\xi_j), \qquad 0 \leq j \leq \widetilde{P}, \qquad Q = \widetilde{P} + 1 \tag{4.7}$$

where the number of points Q required depends on the degree of the nonlinearity and for polynomial functions can be determined a priori (as shown in Table 4.1 for GLL points). This step can be achieved through a matrix-vector multiplication

$$\boldsymbol{u}^{\delta Q} = \mathbf{Q}_{Q_P \to Q} \boldsymbol{u}^{Q_P}, \tag{4.8}$$

where $\mathbf{Q}_{Q_P \to Q}$ is the interpolation matrix

$$\mathbf{Q}_{Q_P \to Q}[j, i] = \ell_i^{Q_P}(\xi_j) \tag{4.9}$$

whose dimensions are $Q \times Q_P$. After having performed this operation, the solution is represented in a richer set of points. The larger set of points

does not necessarily need to have the same distribution of the original set of points. Effectively, Gauss-Legendre points do produce better results in terms of evaluation of the nonlinear terms than Gauss-Lobatto-Legendre. Therefore, if the original points are of Gauss-Lobatto-Legendre type for example, it is always possible to interpolate the solution from the original Gauss-Lobatto-Legendre points onto the different (and possibly larger) set of Gauss-Legendre points.

B.) *Collocation product*

The nonlinear term(s) is evaluated on the expanded set of points using the interpolated values of the solution $u^Q(\xi_j)$ obtained at the previous step. If, for simplicity we consider a quadratic nonlinearity then the product is calculated as follows:

$$f^{\delta Q}(\xi_j) = u^{\delta Q}(\xi_j) \cdot u^{\delta Q}(\xi_j) \qquad 0 \leq j \leq \widetilde{P}. \tag{4.10}$$

It is important to remark here that the nonlinearity $f^Q(\xi_j)$ can well be a non-polynomial function (such as in the case of the compressible Euler and Navier-Stokes equations). In this case, the aforementioned rules for the exact integration (see Table 4.1) no longer hold.

C.) *Galerkin projection*

Finally we perform a projection onto the original set of points. In the following we use a Galerkin projection, which is equivalent to a least squares projection of the nonlinear term evaluated on the larger set of points onto the original set of points, so that

$$\left[f^{\delta Q}(\xi_j) = u^{\delta Q}(\xi_j) \cdot u^{\delta Q}(\xi_j)\right] \xrightarrow{P_{Q\to Q_P}} f^{\delta Q_P}(\xi_i). \tag{4.11}$$

In a similar manner to step A, this can be interpreted as a matrix-vector multiplication

$$\boldsymbol{f}^{\delta Q_P} = \mathbf{P}_{Q\to Q_P}\boldsymbol{f}^{\delta Q}, \tag{4.12}$$

where $\mathbf{P}_{Q\to Q_P}$ is the Galerkin projection matrix

$$\mathbf{P} = \mathbf{M}^{-1}\mathbf{Q}^T_{Q_P\to Q}\mathbf{W}, \tag{4.13}$$

with $\mathbf{M}[i,j] = \int \ell_i^{Q_P}\ell_j^{Q_P}d\xi$ being the mass matrix and $\mathbf{W}[i,i] = \int \ell_i^Q d\xi$ is the diagonal Gauss quadrature weight matrix using Q points. Note that the dimensions of $\mathbf{P}_{Q\to Q_P}$ are $Q_P \times Q$.

To summarise the steps, Fig. (4.1) depicts a schematic representation of the dealiasing procedure described above for one-dimensional problems.

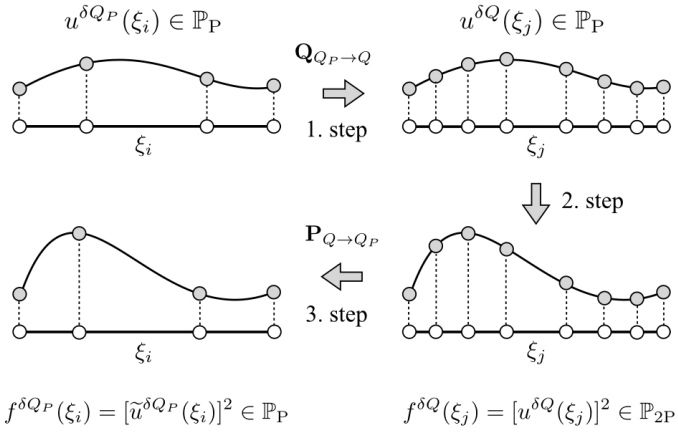


Figure 4.1: Conceptual flow chart of the *Local* dealiasing approach through consistent integration of the nonlinear terms for one-dimensional problems.

This procedure is exact when the isoparametric mapping $\boldsymbol{x} = \Theta(\boldsymbol{\xi})$ which describes the coordinates of a physical element Ω_n is affine and therefore the Jacobian is constant. In addition we note that if $\mathbf{M}$ is diagonal then $\mathbf{M}$ has diagonal components of the weights at Q_P set of points. In this case we observe the projection is an interpolation matrix with the rows scaled by the inverse of the Q_P quadrature point weights whereas the columns are rescaled by the quadrature point integration weights.

Local dealiasing for two-/three dimensional tensor product grids

For the sake of simplicity, in this section we consider a two-dimensional tensor product element and we remark where necessary the differences in terms of computational costs between the two- and the three-dimensional case.

Consider a two-dimensional approximate polynomial solution $u(\xi_1, \xi_2)$ of order P represented through a nodal expansion basis, such as Lagrange polynomials, onto a set of $Q_P \times Q_P = (P+1) \times (P+1)$ points:

$$u^{\delta Q_P}(\xi_1, \xi_2) = \sum_{i,j=0}^{P} u_{ij} \ell_i(\xi_1) \ell_j(\xi_2). \tag{4.14}$$

In the two-dimensional (equivalently three-dimensional) case, the dealiasing technique consists of the same conceptual steps as the one-dimensional case but the

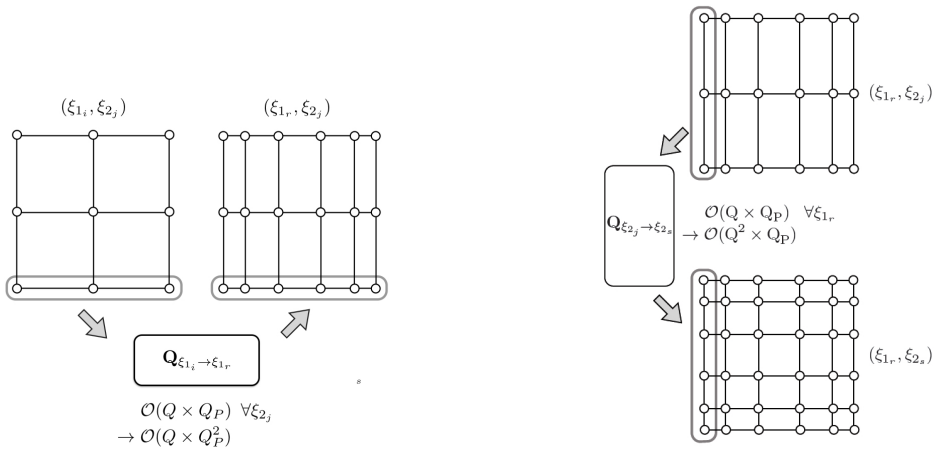
operations to perform are more numerous and consequently more computationally costly. However, the exploitation of the tensor product properties allows an efficient implementation through the use of sum-factorisation technique.

A.) *Two-/three dimensional interpolation*

The first step involves interpolating the solution $u(\xi_1, \xi_2)$ onto a larger set of $Q \times Q = (\widetilde{P}+1) \times (\widetilde{P}+1)$ points, where $\widetilde{P} > P$:

$$u^{\delta Q}(\xi_{1_r}, \xi_{2_s}) = \sum_{i,j=0}^{Q_P} u_{ij} \ell_i(\xi_{1_r}) \ell_j(\xi_{2_s}), \qquad 0 \leq r, s \leq \widetilde{P}. \tag{4.15}$$

As noted in the one-dimensional case, the number of points Q required depends on the degree of the nonlinearity and on the point distribution chosen. The interpolation performed is divided into two (three in the three-dimensional case) sub-steps as shown in Fig. (4.2). The first sub-step, in Fig. (4.2(a)), is the interpolation of the solution along ξ_1 whereas the second, in Fig. (4.2(b)) is the interpolation along ξ_2. This implementation allows a reduction of the floating point operations required.



(a) Interpolation along ξ_1 (b) Interpolation along ξ_2

Figure 4.2: Interpolation for two-dimensional (equivalently three-dimensional) tensor product elements.

Specifically the overall number of floating point operations required is proportional to $\mathcal{O}(Q \times Q_P^2 + Q^2 \times Q_P)$ while by doing a two-dimensional interpolation for a non-tensor product basis the floating point operations requested

become proportional to $\mathcal{O}(Q^2 \times Q_P^2)$. Following a similar procedure for the three-dimensional case gives a number of floating point operations proportional to $\mathcal{O}(Q \times Q_P^3 + Q^2 \times Q_P^2 + Q^3 \times Q_P)$ versus $\mathcal{O}(Q_P^3 \times Q^3)$ operations otherwise required for a full three-dimensional interpolation. Further details of this type of sum factorisation can be found in Karniadakis and Sherwin (2005).

B.) *Collocation product*

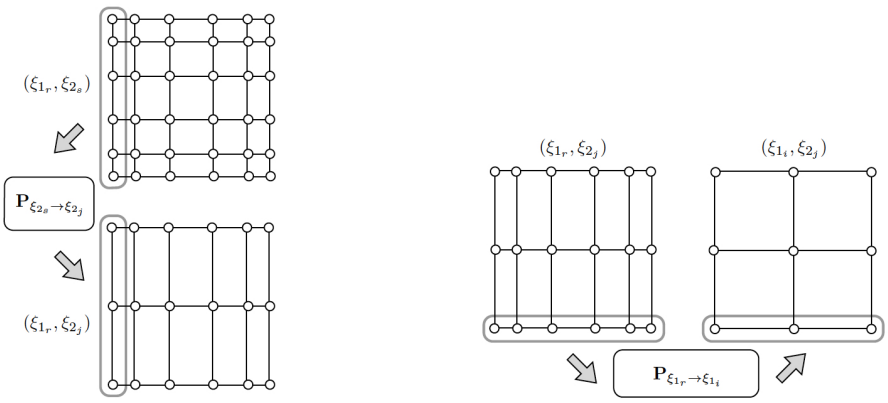
The second step involves evaluating the nonlinear term on the expanded set of points using the interpolated values of the solution $u^{\delta Q}(\xi_{1_r}, \xi_{2_s})$ obtained at the previous step. Similar to the one-dimensional case, the evaluation of the product of the interpolated values $u^Q(\xi_{1_r}, \xi_{2_s})$ is achieved using a collocation projection as follows:

$$f^Q(\xi_{1_r}, \xi_{2_s}) = u(\xi_{1_r}, \xi_{2_s}) \cdot u(\xi_{1_r}, \xi_{2_s}), \quad 0 \leq r, s \leq Q, \tag{4.16}$$

where we again considered a quadratic nonlinearity for the sake of simplicity.

C.) *Galerkin projection*

The third step involves performing a projection of the nonlinear term onto the original set of points $Q_P \times Q_P$. To do so we use a Galerkin projection as for the one-dimensional case. Similarly to the first step we split this operation into two sequential sub-operations as shown in Figure 4.3. First in Figure 4.3(a) we perform a Galerkin projection in direction ξ_2 and successively in Figure 4.3(b) we perform a Galerkin projection in direction ξ_1.



(a) Galerkin projection along ξ_2 (b) Galerkin projection along ξ_1

Figure 4.3: Galerkin projection for two-dimensional tensor product elements.

In Figure 4.4 we show an overview of the steps above for the two-/three-dimensional tensor product case.

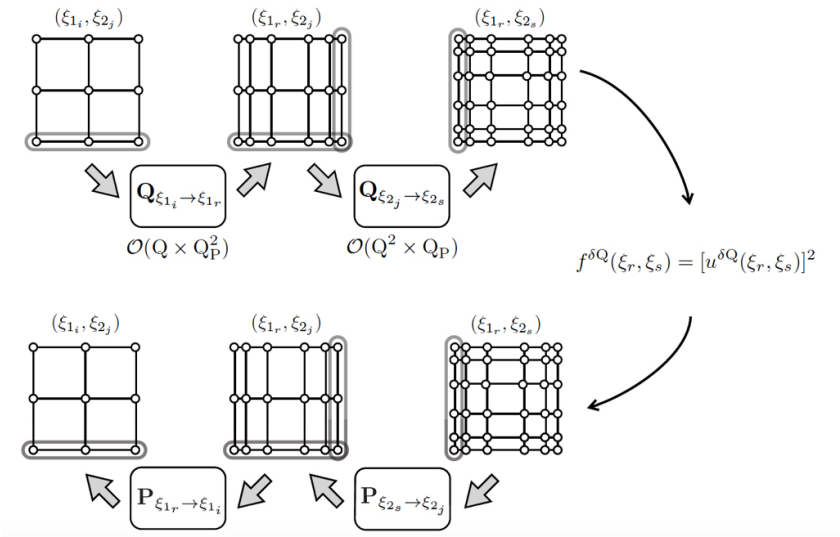


Figure 4.4: Conceptual flow chart of the *Local* dealiasing approach for the two-/three-dimensional tensor product case.

A similar tensor product based approach can also be used for triangles in 2D as well as prismatic and tetrahedral elements in 3D as reported in appendix A.

4.3.2 Global dealiasing

In order to address both PDE- and geometrical-aliasing issues, one can apply the consistent integration to both PDE and geometrical nonlinearities. From this perspective, it is necessary to perform the following steps:

A.) interpolate the solution onto a larger set of points, $u^{Q_P} \to u^Q$ with $Q > Q_P$;

B.) evaluate the nonlinear term(s) onto the larger set of points $\boldsymbol{f}(u^{Q_P}) \to \boldsymbol{f}(u^Q)$;

C.) interpolate the geometric factors $\frac{\partial x_i}{\partial \xi_j}$ onto the larger set of points Q;

D.) construct the derivation matrices $\mathbf{D}_{\xi_i}$ using the larger set of points Q;

E.) interpolate the boundary terms $\widehat{\mathbf{b}}^{\mathrm{DG}}/\widehat{\mathbf{b}}^{\mathrm{FR}}$ onto the larger set of points Q;

F.) assemble the overall right-hand side onto the larger set of points Q;

G.) project the right-hand side to the original set of points Q_P through a Galerkin projection.

Note that we do not make any distinctions between the one- or two-/three- dimensional cases, nor between different element shapes, since here we do not explicitly exploit the tensor product advantages (although some operators may still have a tensor product form) used in the previous section for the *Local* dialysing technique through consistent integration of the nonlinear terms. This approach is similar in nature to the one presented by Kirby and Karniadakis (2003). However, here we additionally perform an over-sampling of the geometric terms, the Jacobian and the geometric factors defined in Eq. 1.22 and 1.23, onto a larger set of points (and therefore the overall integrals are evaluated onto the larger set of points). Based on the greater number of floating operations to be performed the *Global* dealiasing technique is more computationally expensive than the *Local* dealiasing but can it address both PDE- and geometrical-aliasing issues.

4.4 Numerical results

In this section, we apply the dealiasing techniques presented above to a simple transport advection equation and to compressible and incompressible flow simulations.

With the experiments on the transport advection equation (for which a periodic solution is known) we evaluate quantitatively the effects of the dealiasing techniques in terms of both PDE- and geometrical-aliasing errors.

Specifically, to study the effect of PDE-dealiasing, we perform a first set of experiments on regular meshes where the advection equation considered had an advection velocity that was a nonlinear polynomial in space. The results obtained on this first set of simulations permitted a better understanding of the role of the boundary terms in discontinuous numerical discretisations. Through these results it was possible to evaluate the inequalities which lead to exact integration of discretised nonlinear functionals (note that the word 'nonlinear' refers to the polynomial representing the advection velocity). The experiments also allowed a numerical quantification of the volumetric and boundary contributions to the aliasing errors observed.

To study the effect of geometrical-dealiasing, a second set of experiments was instead carried out on various deformed and curvilinear meshes. In this case the advection velocity used was constant in order to have only geometrical nonlinearities and the experiments highlighted the link between geometrical- and PDE-aliasing errors.

After these quantitative analyses, we consider some qualitative effects that dealiasing plays in compressible and incompressible flow simulations. Specifically, we carried out a set of simulations of a compressible inviscid flow past a cylinder for different quadrature orders and a simulation of an incompressible flow past a wing tip with and without dealiasing. The results obtained in both cases allows us to show how consistent integration has an effective role in enhancing the stability of these simulations. However, even if in the examples presented the dealiasing techniques show a stabilising effect, dealiasing does not guarantee stability and in other flow studies some numerical instabilities were experienced even applying a consistent integration of the flux functions and of the geometric terms. In these cases for stabilising the simulation we needed to add an artificial viscosity of the type described in Maday and Tadmor (1989) and Kirby and Sherwin (2006a).

In all the experiments performed we chose Gauss-Lobatto-Legendre points for the consistent integration. The use of a Gauss-Legendre quadrature would have been more efficient for reducing the aliasing issues but, for the scope of this chapter, the results on Gauss-Lobatto-Legendre points are sufficient.

4.4.1 PDE-aliasing

To understand the effect of the PDE-dealiasing techniques presented in the previous section, we considered the following transport equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + a_x \dfrac{\partial u}{\partial x} + a_y \dfrac{\partial u}{\partial y} = 0 \\ u(x, y, t = 0) = \exp\{-41[(x + 0.3)^2 + y^2]\} \\ u(x_b, y_b, t) = 0, \\ a_x = \pi y^{P_{adv}} g(t), \;\; a_y = -\pi x^{P_{adv}} g(t), \end{cases} \tag{4.17}$$

where u is the conserved variable, a_x and a_y are the advection velocities along the x and y directions, (x_b, y_b) denote the boundaries of the numerical domain, P_{adv} is the polynomial order of the flux and $g(t)$ is a time dependent periodic function

$$g(t) = \cos(\pi t/T) \tag{4.18}$$

on a time interval $[0, T]$. The solution u evolves in time in such a way that the initial data is recovered at every period T, so that

$$u(x, y, 0) = u(x, y, T). \tag{4.19}$$

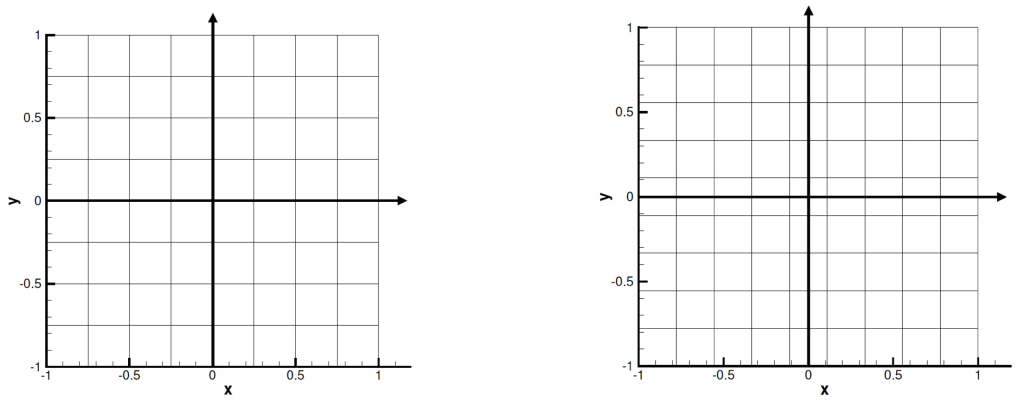
In all the experiments, we selected a period $T = 1$ and a final time $4T$ so that the final solution is equal to the initial in order to calculate the error. This strategy

is the same as that adopted in LeVeque (1996) and it is widely used to evaluate properties of numerical methods when the exact solution is not available.

In the following we first present a comparative study between the *Local* and *Global* dealiasing approaches on the case where the advection velocity is linear, so that $P_{adv} = 1$. This will highlight the connections between the two dealiasing approaches and ensures that some general properties hold true for all the results presented in the following sections. We then show how increasing P_{adv}, and therefore the order of the advection velocities, can lead to a more important role of the interfaces in the case of discontinuous discretisations. In all the simulations we used a tensor product lagrangian polynomial basis of order $P = 4$.

Comparative study of the different dealiasing techniques

In this section we show the results obtained using *Local* and *Global* dealiasing techniques for the test case in Eq. (4.17) when $P_{adv} = 1$ on a regular grid of quadrilaterals, so that no geometrical aliasing effects are present.



(a) Without interfaces across $\pm$ (x, y) (b) With interfaces across $\pm$ (x, y)

Figure 4.5: Meshes for the 2D linear advection test cases: Case A in Figure 4.5(a) and Case B in Figure 4.5(b).

For each simulation we used a forward Euler time integration scheme and the time step was chosen to be sufficiently small so as to consider the temporal error negligible. We remark that any explicit time integration scheme such as 2^{nd} or 4^{th} order Runge-Kutta schemes can be used. In space we consider both the DG and the FR discretisation in turn, where the polynomial order is set to $P = 4$ and the initial condition is given by a collocation projection (i.e. using $Q = 5$ quadrature points).

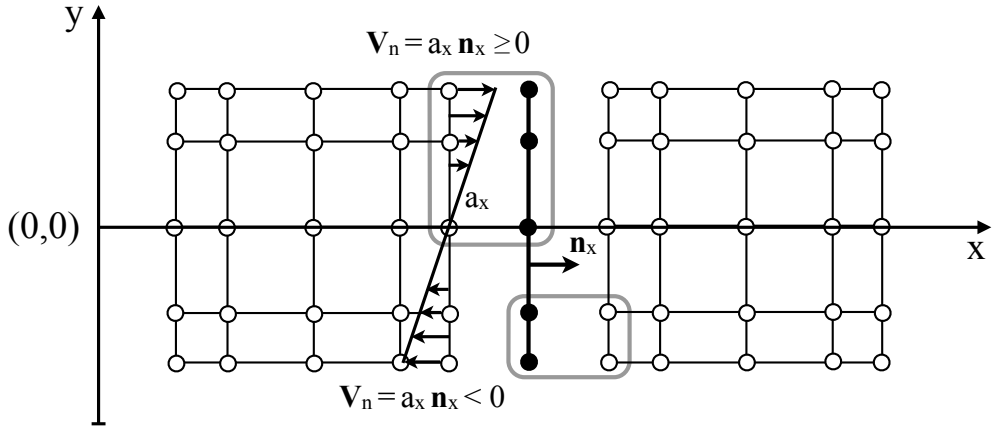


Figure 4.6: Upwind flux at the interface between two elements: Aliasing arising from mixed information coming from the left and the right element.

Throughout the following we will make extensive use of relative L_2 errors defined in Eq. 1.123 and here reported for the reader's convenience:

$$E_{L_2} = \sqrt{\sum_{i,j=0}^{N} (u_{ij} - u_{ij,exact})^2 w_{ij}} \ , \tag{4.20}$$

where:

$$w_{ij} = \int_{-1}^{1} \ell_i(\xi) d\xi_1 \int_{-1}^{1} \ell_j(\xi_2) d\xi_2 \tag{4.21}$$

and $\ell_i(\xi_1), \ell_j(\xi_2)$ are Lagrange polynomials.

In Table 4.2 we show the L_2 errors for the different dealiasing strategies applied to the DG and FR formulations for the mesh shown in Figure 4.5(a). Here, 'LMM' stands for lumped mass matrix, which is the diagonal matrix obtained when $Q = 5$ quadrature points are used to obtain a collocation scheme, while 'EMM' refers to the exact mass matrix obtained at higher quadrature orders. This table illustrates the saturation of the L_2 error up to machine precision for $Q = 6$ as well as the equivalence between *Local* and *Global* dealiasing when the number of quadrature points used for the nonlinear terms are the same in the two approaches, since the geometric terms for a regular grid are constant. We also note that the equivalence of the DG and FR schemes presented in chapter 2 also holds when using a larger number of quadrature points than a collocation method.

Regarding the dealiasing of the interface fluxes, we note that we obtain a saturation of the error up to machine precision. This is because the interface flux for the mesh in Figure 4.5(a) is a polynomial function since its values come from just

one side with respect to a given boundary[4].

	DG-EMM	**DG-LMM**	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00258809184895	0.00517893325001	0.00258809184895	0.00517893325001
$Q = 5, Q_V = 6, Q_I = 5$	0.00245321968401	0.00484792022346	0.00245321968401	0.00484792022346
$Q = 5, Q_V = 7, Q_I = 5$	0.00245321968401	0.00484792022346	0.00245321968401	0.00484792022346
$Q = 5, Q_V = 8, Q_I = 5$	0.00245321968401	0.00484792022346	0.00245321968401	0.00484792022346
$Q = 5, Q_V = 6, Q_I = 6$	0.00238095562953	0.00483526988468	0.00238095562953	0.00483526988468
$Q = 5, Q_V = 7, Q_I = 7$	0.00238095562953	0.00483526988468	0.00238095562953	0.00483526988468
$Q = 5, Q_V = 8, Q_I = 8$	0.00238095562953	0.00483526988468	0.00238095562953	0.00483526988468
$Q = 6$	0.00238095562953	-	0.00238095562953	0.00483526988468
$Q = 7$	0.00238095562953	-	0.00238095562953	0.00483526988468
$Q = 8$	0.00238095562953	-	0.00238095562953	0.00483526988468

Table 4.2: L_2 errors for the different dealiasing strategies applied to the DG and FR formulations: Case A mesh as in Figure 4.5(a).

	DG-EMM	**DG-LMM**	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00164993651569	0.00279358970635	0.00164993651569	0.00279358970635
$Q = 5, Q_V = 6, Q_I = 5$	0.00157503364988	0.00265346210971	0.00157503364988	0.00265346210971
$Q = 5, Q_V = 7, Q_I = 5$	0.00157503364988	0.00265346210971	0.00157503364988	0.00265346210971
$Q = 5, Q_V = 8, Q_I = 5$	0.00157503364988	0.00265346210971	0.00157503364988	0.00265346210971
$Q = 5, Q_V = 6, Q_I = 6$	0.00155395894760	0.00266014915428	0.00155395894760	0.00266014915428
$Q = 5, Q_V = 7, Q_I = 7$	0.00154977027283	0.00265747641942	0.00154977027283	0.00265747641942
$Q = 5, Q_V = 8, Q_I = 8$	0.00155235541132	0.00265532284115	0.00155235541132	0.00265532284115
$Q = 6$	0.00155395894760	-	0.00155395894760	0.00266014915428
$Q = 7$	0.00154977027283	-	0.00154977027283	0.00265747641942
$Q = 8$	0.00155235541132	-	0.00155235541132	0.00265532284115

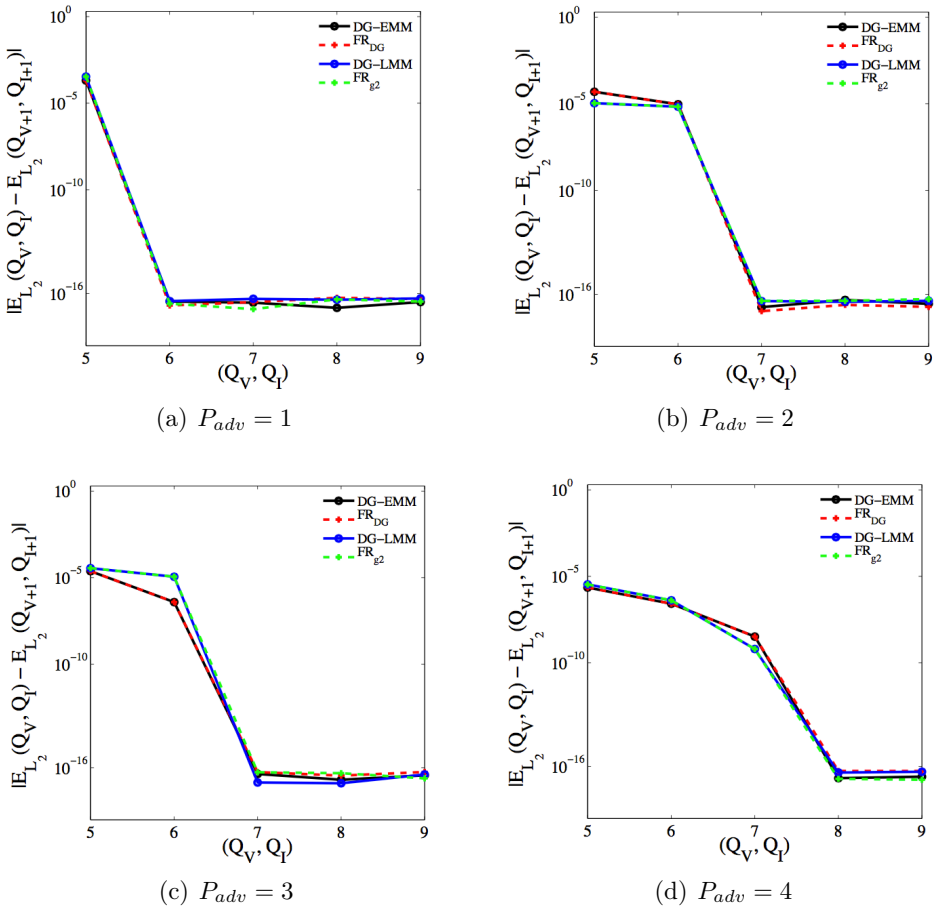
Table 4.3: L_2 errors for the different dealiasing strategies applied to the DG and FR formulations: Case B mesh as in Figure 4.5(b).

If we instead consider the mesh in Figure 4.5(b) we are in a case where the interface fluxes come from both sides of a given interface as shown in Figure 4.6; therefore they do not lie in the polynomial space and error saturation up to machine precision cannot be expected as presented in Table 4.3. This means that in general, it is difficult to fully control the PDE-aliasing arising from the interface because in this case the function representing the interface flux lies outside a polynomial space. However, the error does decrease as the number of quadrature points is increased.

Role of dealiasing for higher order advection velocities

In this section, we show the results with advection velocities which are nonlinear, so that the order P_{adv} is chosen to be greater than one in Eq. (4.17).

[4]Note that all the simulations in this and the following section were run using an upwind interface flux.



(a) $P_{adv} = 1$ (b) $P_{adv} = 2$

(c) $P_{adv} = 3$ (d) $P_{adv} = 4$

Figure 4.7: Differences between the L_2 errors obtained with $(Q_I,\, Q_V)$ and $(Q_I + 1,\, Q_V + 1)$ vs $(Q_I,\, Q_V)$ using the *Local* dealiasing technique.

The numerical parameters are chosen to be identical to that of the previous section, and we use the grid depicted in Figure 4.5(a) to avoid non-saturation of the error when dealiasing the interfaces. In Figure 4.7, each point represents the difference between the L_2 error calculated using (Q_V, Q_I) quadrature points and the error calculated using $(Q_V + 1, Q_I + 1)$ quadrature points in the LMM case[5]. We note that the machine-precision saturation of the error satisfies the inequality:

$$\begin{aligned} Q_{min} &\geq P + \frac{P_{adv}}{2} + \frac{3}{2}, \\ P &\geq P_{adv}, \end{aligned} \tag{4.22}$$

[5]Note that the L_2 error was calculated on an enriched quadrature space.

where Q_{min} is the minimum number of quadrature points to exactly integrate the PDE-aliasing for a given polynomial advection velocity possessing a nonlinearity of order P_{adv} and for a given expansion order P.

In appendix B, Figure B.1 shows the actual L_2 errors vs (Q_V, Q_I) while Tables B.1, B.2, B.3 and B.4, show the tabulated values of the L_2 errors for $P_{adv} = 1, 2, 3, 4$. From these results it is possible to see how, for all the formulations, the use of *Local* dealiasing on both the volumetric and interface fluxes was the best choice in terms of L_2 error among the dealiasing techniques applied. Also note that increasing the number of quadrature points does not always guarantee a better L_2 error (see for example the last two lines of Table 4.3). This can be explained with fact the flux function are not polynomials and, therefore, a complete control of the aliasing errors cannot be achieved.

To further quantify the effects of interface dealiasing, in Table 4.4 we show the difference between the L_2 errors obtained for *Local* dealiasing of the volumetric flux only, and of both volumetric and interface fluxes:

$$\Delta \mathrm{E}_{\mathrm{L2_V}} = \frac{\left(\mathrm{E}_{\mathrm{L}_2}\left(Q_V\right)_{sat} - \mathrm{E}_{\mathrm{L}_2}\left(Q_V, Q_I\right)_{sat}\right)}{\mathrm{E}_{\mathrm{L}_2}\left(Q_V, Q_I\right)_{sat}} \cdot 100 \tag{4.23}$$

for the different DG and FR schemes. In the above equation, Eq. 4.23, the subscript "sat" indicates saturation (i.e. the associated L_2 errors were taken at the point where they saturate, according to Figure 4.7).

Table 4.5 shows the difference between the L_2 errors obtained for *Local* dealiasing of the interface flux only, and of both volumetric and interface fluxes:

$$\Delta \mathrm{E}_{\mathrm{L2_I}} = \frac{\left(\mathrm{E}_{\mathrm{L}_2}\left(Q_I\right)_{sat} - \mathrm{E}_{\mathrm{L}_2}\left(Q_V, Q_I\right)_{sat}\right)}{\mathrm{E}_{\mathrm{L}_2}\left(Q_V, Q_I\right)_{sat}} \cdot 100 \tag{4.24}$$

again for the different DG and FR schemes.

	$\Delta \mathbf{E}_{\mathbf{L2_V}}$			
	DG-EMM	**DG-LMM**	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$P_{adv} = 1$	3.0351	0.2616	3.0351	0.2616
$P_{adv} = 2$	17.8755	5.5913	17.8755	5.5913
$P_{adv} = 3$	19.2840	10.410	19.2840	10.410
$P_{adv} = 4$	7.6039	5.9054	7.6039	5.9054

Table 4.4: Percentage difference between the L_2 errors for *Local* (Q_V) and *Local* (Q_V, Q_I) at error saturation.

	$\Delta\mathbf{E}_{\mathrm{L2_I}}$			
	DG-EMM	**DG-LMM**	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$P_{adv} = 1$	15.4071	7.7022	15.4071	7.7022
$P_{adv} = 2$	15.7231	3.2610	15.7231	3.2610
$P_{adv} = 3$	19.1376	12.4640	19.1376	12.4640
$P_{adv} = 4$	6.4057	3.5643	6.4057	3.5643

Table 4.5: Percentage difference between the L_2 errors for *Local* (Q_I) and *Local* (Q_V, Q_I) at error saturation.

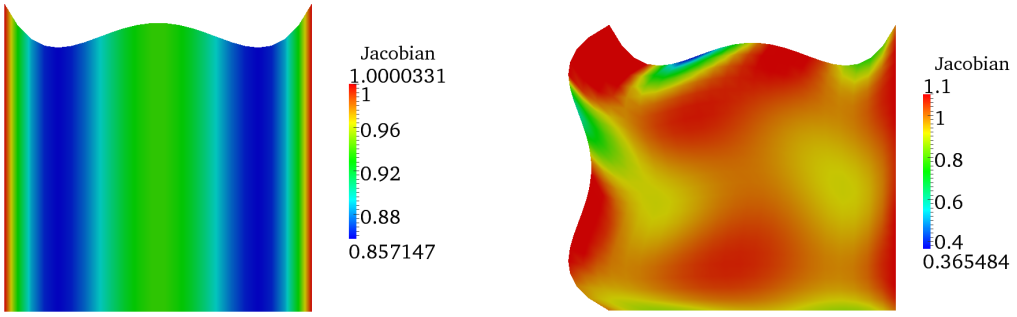
From these tables we can see how the difference between performing only volumetric dealiasing and volumetric plus interface dealiasing increases as P_{adv} increases except for the last case ($P_{adv} = 4$). This is related to the fact that the test case being considered is likely to underestimate the role of the interfaces, because as P_{adv} increases, the magnitude of the advection velocity decreases with the power of the order of the advection velocity P_{adv} (since the mesh is contained within the square [-1,1]×[-1,1]). Nevertheless, it is still possible to see that the contribution of the interfaces becomes greater as P_{adv} increases and it is effectively comparable to the contribution of the volumetric dealiasing for $P_{adv} = 2$, 3 and 4.

This result is crucial, because it indicates that, in some cases, the interface dealiasing is as important as the volumetric dealiasing and, therefore the boundary terms arising in discontinuous discretisations must be taken into account when implementing a dealiasing strategy.

On the other hand, it is also important to remember that aliasing errors arising in the boundary terms cannot be always fully controlled because of their generally non-polynomial nature.

4.4.2 Link between geometrical- and PDE-aliasing

Now that the aliasing effects of the discretisation of the PDE have been examined, in this section we show how geometrical-aliasing can be linked to PDE-aliasing with some differences arising from the fact that geometrical nonlinearities play a slightly different role in the underlying formulation compared to PDE nonlinearities.

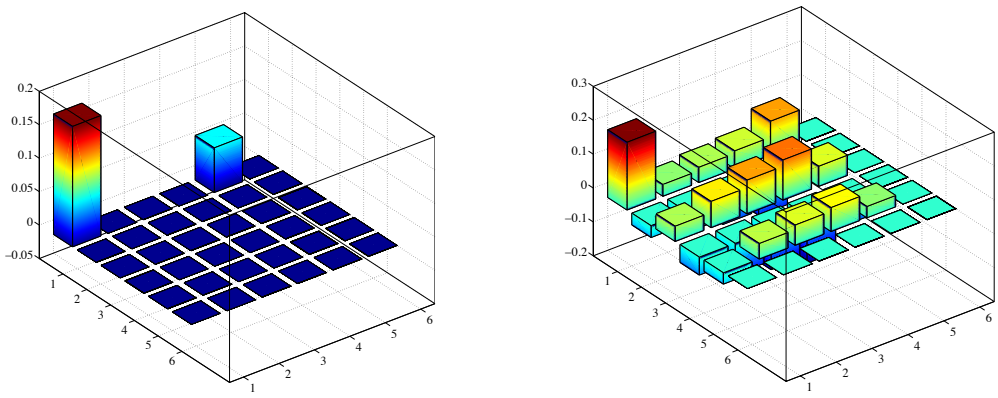


(a) Top edge curved (b) Top and left edges curved

Figure 4.8: Examples of 4^{th} order meshes employed to investigate geometrical aliasing.

We consider a mesh composed of a single standard quadrilateral element $\Omega_{\text{st}} = [-1,1] \times [-1,1]$), to which we curve either only the top edge, or both top and left edges. For each configuration, we applied different curvatures to the edges by means of Legendre polynomials of order $P = 1, 2, 3$ and 4. In Figure 4.8(a) and (4.8(b)) we show two examples for the mesh of order four.

In each case, the isoparametric mapping, and therefore the Jacobian determinant, is represented by an expansion of basis functions over the standard element, so that for example a $P = 4$ expansion contains $5 \times 5 = 25$ different unique modes.

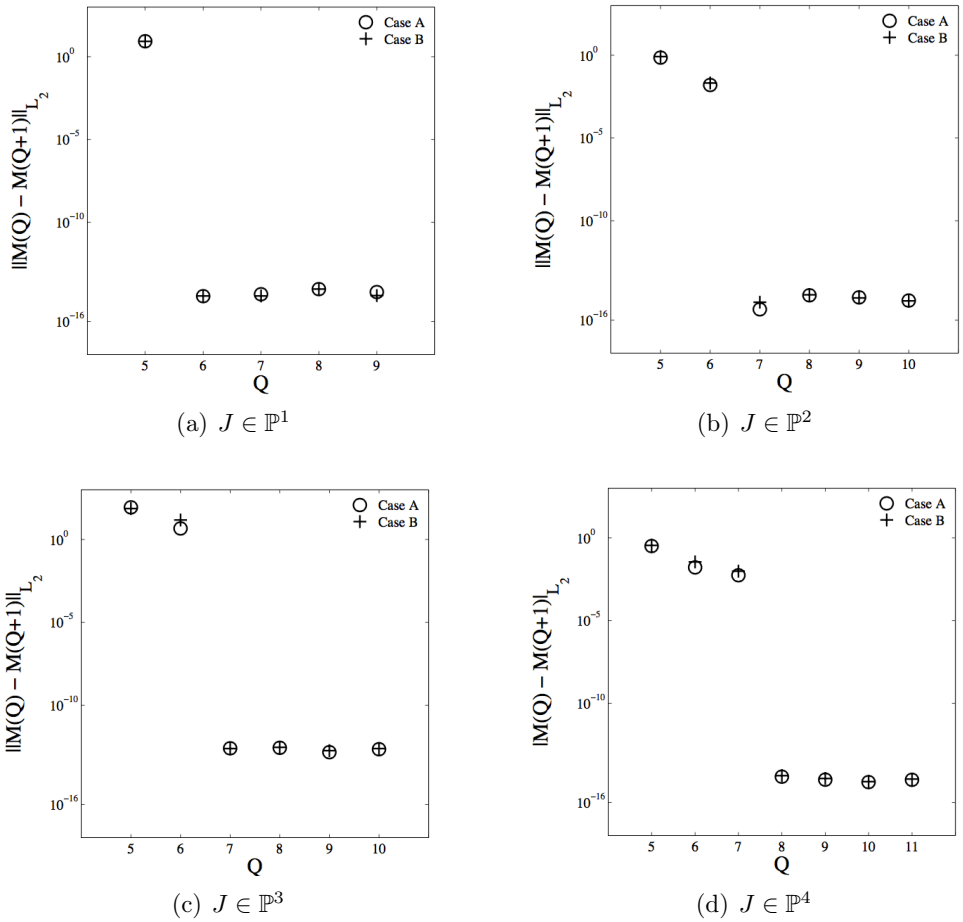


(a) Top edge curved (b) Top and left edges curved

Figure 4.9: Coefficients in an orthonormal expansion basis for the 4^{th} order meshes showed in Fig.(4.8).

Figure 4.9 shows the magnitude of each modal coefficients of the Jacobian

determinant for the 4^{th} order case when projected onto an orthonormal basis of order 5 (leading to 36 coefficients), in order to emphasise that the polynomial space is sufficiently large to capture the Jacobian mapping. In particular we note that the Jacobian for both cases is effectively of fourth order, but in the case of Figure 4.9(a) we have nonzero modes only in the x-direction up to the fifth coefficient whereas in the case of Figure 4.9(b) the nonzero modes span both x and y directions again up to the fifth coefficient.

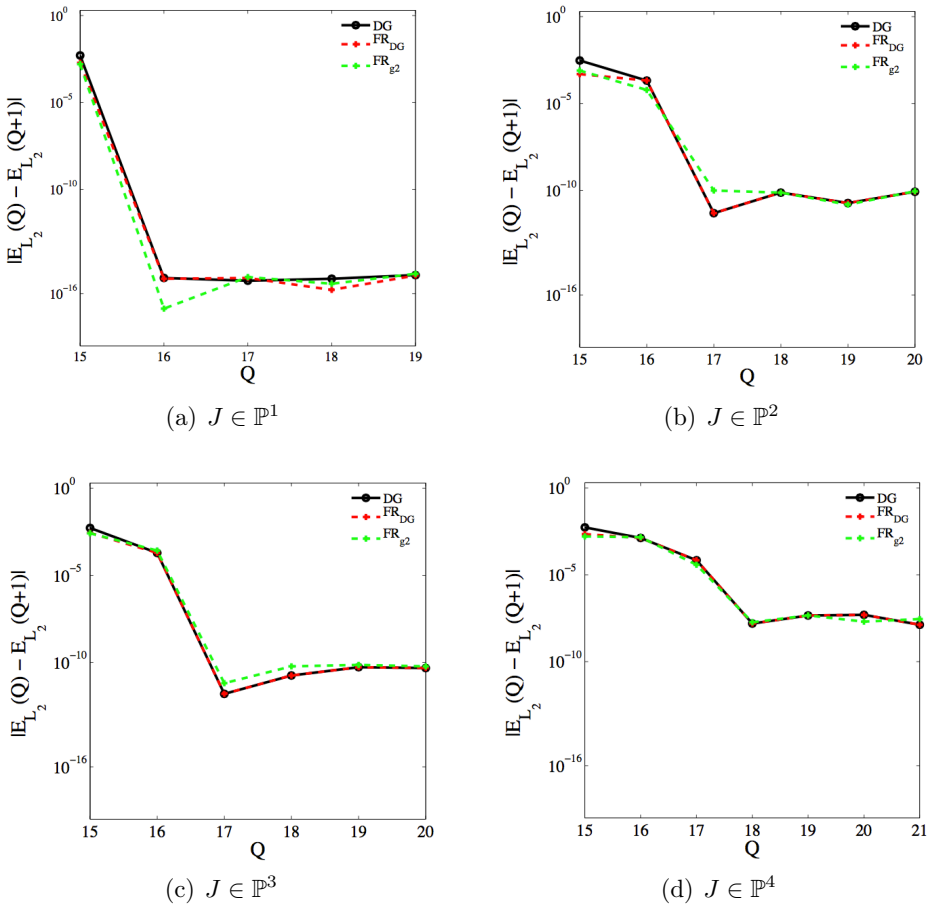


(a) $J \in \mathbb{P}^1$ (b) $J \in \mathbb{P}^2$

(c) $J \in \mathbb{P}^3$ (d) $J \in \mathbb{P}^4$

Figure 4.10: L_2 norm of the difference between mass matrices calculated with Q and $Q+1$ GLL points for the four orders considered and using both the mesh configurations shown in Figure 4.8. Case A corresponds to Figure 4.9(a) and Case B corresponds to Figure 4.9(b).

The sixth coefficient in both directions is zero, showing adequate support for the Jacobian determinant expansion.



(a) $J \in \mathbb{P}^1$ (b) $J \in \mathbb{P}^2$

(c) $J \in \mathbb{P}^3$ (d) $J \in \mathbb{P}^4$

Figure 4.11: Differences between the L_2 errors obtained with Q and $Q+1$ GLL points for the four orders considered using the mesh in Figure 4.8(a).

The test case we consider is identical to that of Eq. (4.17), aside from the use of spatially-constant advection velocities:

$$a_x = \frac{\pi}{2}\, g(t), \quad a_y = -\frac{\pi}{2}\, g(t), \tag{4.25}$$

where $g(t)$ is defined in Eq. (4.18). We chose a period $T = 0.5$ and a final time $40T$. For the time-integration we use a 2^{nd}-order Runge-Kutta scheme. The polynomial

order was $P = 14$ in order to have a sufficiently resolved problem. We used a collocation projection of the initial condition.

Throughout the results in this subsection we used a *Global* dealiasing technique in order to target the geometrical aliasing. We first present estimates of the quadrature necessary for correctly integrating a deformed or curved mesh. For doing so we take into account the leading order within the problem under investigation, which is the mass matrix in the case of the DG discretisation. In Figure 4.10 each point denotes the difference between two mass matrices, the first obtained for Q and the second for $Q+1$ quadrature points and it is calculated as follows:

$$||\Delta \mathbf{M}||_{\mathrm{L}_2} = \sqrt{\sum_{i,j} \left[\mathbf{M}_{i,j}(Q) - \mathbf{M}_{i,j}(Q+1)\right]^2}. \tag{4.26}$$

Table B.5 in appendix B quantifies the results presented in Figure 4.10. We note that the mass matrix does not change (approximately up to machine precision) after we have reached the minimum number of quadrature points to exactly integrate the polynomial order used for describing the geometry, given by the inequality:

$$\begin{aligned} Q_{min} &\geq P + \frac{P_{geom}}{2} + \frac{3}{2}, \\ P &\geq P_{geom}, \end{aligned} \tag{4.27}$$

where Q_{min} is the minimum number of quadrature points to exactly integrate the mass matrix for a given polynomial of order P_{geom} describing the geometry and for a given expansion order P. Eq. (4.27) is identical to Eq. (4.22) with the only exception being that P_{geom} now refers to the polynomial order of the geometry deformation. The above result is consistent with the Gauss-Lobatto-Legendre quadrature rules.

Figure 4.11 shows the differences in the L_2 errors as defined in Eq. (4.26) instead of differences between mass matrices. We can see how both, the DG and the FR discretisations show machine-precision saturation of the error only for the linear mesh (i.e. $J \in \mathbb{P}^1$). For the other three cases, the saturation level is not at machine precision and it increases as P_{geom} increases. This behaviour is due to the numerical flux (i.e. the boundary term) which might come from both the left (internal domain) and right (boundary condition) sides of a given edge, since the normals are spatially varying across the curved edge(s). Therefore, for complex geometries, it is not possible to fully control the geometrical aliasing arising from the interfaces.

As a final comment we also note that the connections between DG and FR established in De Grazia et al. (2014) and presented in chapter 2 also hold true for a deformed/curved mesh when using a larger number of quadrature points than a collocation method.

4.4.3 Flow applications

Having considered the quantitative effects of the *Local* and *Global* dealiasing techniques, in this section we consider two examples that highlight how stability can be enhanced by using appropriate dealiasing strategies. We begin by considering a compressible Euler test case, and show that the lack of dealiasing can lead to the buildup of numerical entropy in the solution field. Secondly, we consider the LES of a NACA 0012 wingtip, and show how dealiasing can increase the robustness of the simulation. As previously stated, however, the consistent integration does not imply stability and in some simulations the high frequency modes of the nonlinear flux functions are not eliminated through dealiasing. In this case, other strategies need to be applied (eventually in conjunction with consistent integration) such as the use of artificial viscosity or modal filtering.

Compressible inviscid subsonic flow past a cylinder

In this section, we present an inviscid compressible flow past a cylinder governed by the compressible Euler equations.

We set the Mach number to 0.2, used a FR_{DG} scheme for the spatial discretisation and a 4^{th}-order Runge-Kutta scheme for the time-integration. The mesh used is a circular mesh composed of $N_t \times N_r = 178 \times 54$ quadrilateral elements in the tangential- and radial-direction, respectively.

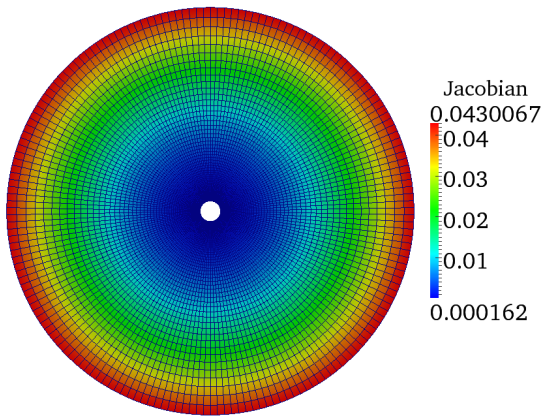
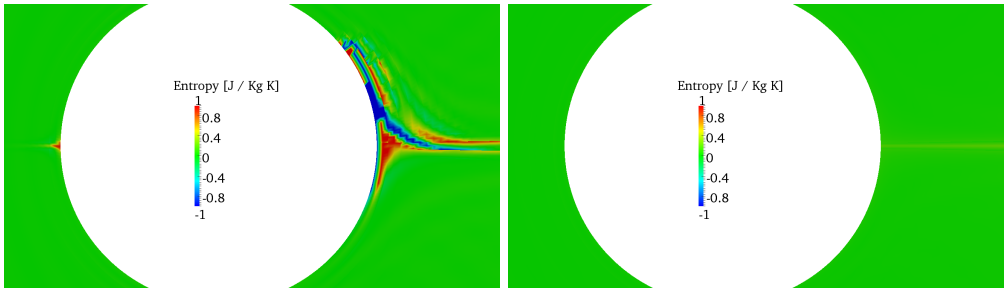


Figure 4.12: Jacobian distribution on the mesh used for the cylinder compressible inviscid simulation.

The cylinder is described by third order splines in order to achieve a high-order description of the geometry. In Figure 4.12 we show the mesh used for simulations with the Jacobian determinant distribution. We use two different polynomial orders: $P = 2$ and $P = 3$ and the *Global* dealiasing technique. Figure 4.12 indicates

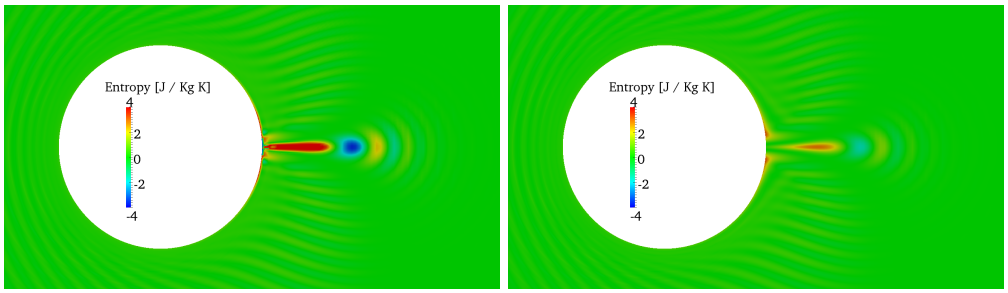
that the elements of the mesh are of good quality and not highly distorted, although a third order curvature is applied to the cylinder wall. We therefore expect PDE-aliasing to play a bigger role than geometrical aliasing.

In Figure 4.13(a) we show the numerically-generated entropy for the case $P = 2$, $Q = 3$ without dealiasing techniques applied. We can clearly see a buildup of entropy from the posterior stagnation point, which is sensitive to aliasing-driven instabilities due to the strong gradients of the solution in this region. Note that ideally for an inviscid simulation, the entropy should be numerically zero. In this case the simulation diverged after a few time-steps; however by applying the *Global* dealiasing techniques using $Q = 4$ we were able to reduce the aliasing errors. In Figure 4.13(b) we show a snapshot of the solution field taken at the same value of time as Figure 4.13(a). We note that the numerically-generated entropy at the posterior stagnation point is no longer present. The simulation for $P = 3$ and $Q = 4$ depicted in Figure 4.14 was also unstable. As in the previous case (i.e. $P = 2$, $Q = 3$), the use of the *Global* dealiasing technique stabilised the simulation.



(a) $P = 2$, $Q = 3$, no dealiasing (b) $P = 2$, $Q = 4$, *Global* dealiasing

Figure 4.13: Numerically-generated entropy for $P = 2$.



(a) $P = 3$, $Q = 4$, no dealiasing (b) $P = 3$, $Q = 5$, *Global* dealiasing

Figure 4.14: Numerically-generated entropy for $P = 3$.

In Table 4.6 we report the maximum and minimum values of entropy across the computational domain for all the cases considered. As we can see, the numerically-generated entropy, which in this case can be seen as a measure of numerical error, roughly saturates at $Q = 6$ for $P = 2$ and at $Q = 8$ for $P = 3$. Clearly the saturation is not up to machine precision because of the high-order description of the geometry and the deformed elements throughout the domain which introduce additional 'nonlinearities' into the problem. Also, the right-hand side of the compressible Euler equations is composed by rational functions which in turn means that it cannot be represented in a polynomial space, and therefore a machine-precision saturation of the solution cannot be expected. However, a minimisation of the relative error can be seen as Q is increased, and it is associated with the cubic term of the compressible Euler equations which requires $Q \geq 2P + 2$ GLL points to guarantee its exact integration. This also may indicate that the PDE-aliasing is potentially dominant for this problem and does not have a strong coupling with geometrical nonlinearities.

Note that, for both the polynomial orders considered, we also applied the *Local* dealiasing approach not shown here since the results were similar to those obtained using the *Global* dealiasing technique. This further reinforces the role of the PDE-aliasing for this problem.

		P=2	
		Entropy-Min [J/Kg K]	**Entropy-Max** [J/Kg K]
NO		-96.36	94.35
Global	Q=4	-0.86666943708297	1.59384922648788
Global	Q=5	-0.85200075803792	1.54735068797723
Global	Q=6	-0.85130979874330	1.54637692206548
Global	Q=7	-0.85130979868331	1.54637692206548
		P=3	
		Entropy-Min	**Entropy-Max**
NO		-1118.61	4347.60
Global	Q=5	-1.67319315760229	3.49726118185079
Global	Q=6	-1.16291305386484	3.39484597010811
Global	Q=7	-1.13721148529237	3.39967822044820
Global	Q=8	-1.13547005462042	3.39977179691589
Global	Q=9	-1.13517999136988	3.39977189419113

Table 4.6: Maximum and minimum values across the mesh of numerically-generated entropy for both $P = 2$ and $P = 3$.

Incompressible viscous flow past a wing tip

In this section we show the results of an incompressible viscous flow past a NACA 0012 wingtip, originally studied experimentally by Chow et al. (1997).

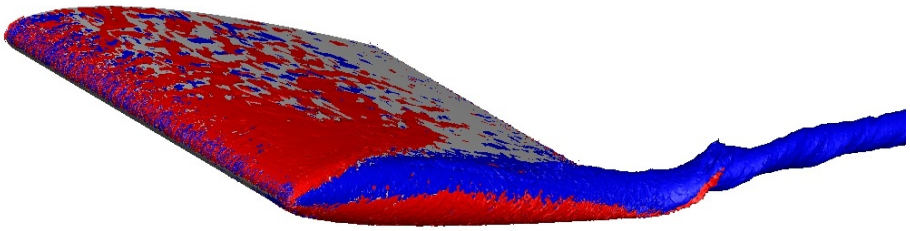
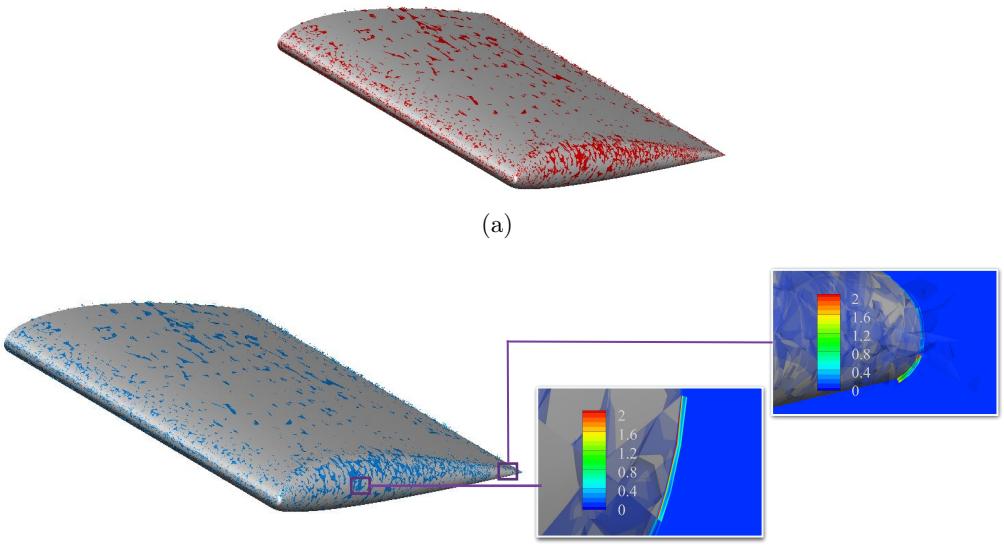


Figure 4.15: Helicity for an incompressible viscous flow over a NACA 0012 wing. [Courtesy of J-E. Lombard]



(a)

(b)

Figure 4.16: Aliasing errors at 30% of the magnitude of the nonlinear terms (Figure 4.16(a)) and close up regions near the wing surface of aliasing errors (Figure 4.16(b)). [Courtesy of J-E. Lombard]

The Large Eddy Simulation performed was an Implicit Large Edgy Simulations (ILES). ILES are different from typical Large Edgy Simulations (LES), where a subgrid-scale model (SGS) is necessary to model the effects of unresolved scales. The aim of this ILES approach is to use a numerical scheme whose truncation errors can play the role of the subgrid model, thus avoiding the need for a separate SGS. This approach is plausible because many SGS take the form of a non-linear, scale-selective dissipation term and some modern numerical schemes have truncation errors of this form. The absence of an explicit SGS equation also makes the

ILES approach easy to implement. A good reference in this field can be found in Grinstein et al. (2011).

The simulation was obtained using the continuous Galerkin (CG) approach, with a velocity correction scheme being used to discretise the incompressible Navier-Stokes equations (Karniadakis et al. (1991)). Although the CG approach is not the main target of this thesis, the results presented in this section illustrate how the dealiasing techniques presented can be effectively applied to high-Reynolds flows and they help stabilising the simulations.

While the experimental Reynolds number was set at $\mathrm{Re} = 4.6 \times 10^6$, initial simulations were performed at a lower Reynolds number of $\mathrm{Re} = 1.2 \times 10^6$. *Local* dealiasing, when combined with a spectral vanishing viscosity to dampen high-frequency energy buildup, yielded a stable simulation.
However, in order to increase the Reynolds number and bring the simulation in line with experimental conditions, we found that without an increase in the global quadrature order (i.e. *Global* dealiasing technique), the simulation quickly became unstable. This highlights the role of geometric as well as PDE-aliasing under these conditions for the mesh and polynomial resolution considered.

In Figure 4.15 we illustrate the dynamics of the flow by showing the isocontours of helicity for a $P = 3$ approximation, where the domain is represented using prismatic elements around the boundary and tetrahedral elements in the rest of the domain. In Figures 4.16(a) and 4.16(b), for the same simulation, we represent the aliasing errors. The top image shows up to 30% aliasing error with respect to the magnitude of the non-linear terms while the bottom image shows close up regions of aliasing near the wing surface where we observe up to 600% error, highlighting the crucial nature of dealiasing in this problem.

4.5 Summary and conclusions

We have presented two dealiasing approaches based on the concept of consistent integration of the nonlinear terms or over-integration if the integration order is based on exact integration of linear operators.

The first technique presented, namely the *Local* approach, targets the PDE-aliasing sources which arise from the nonlinearities contained in the PDE itself. This approach exploits the tensor product structure of the elements to reduce the number of floating point operations, making it computationally efficient at higher polynomial orders. The application of *Local* dealiasing is particularly useful when we have localised nonlinearities such as in the incompressible Navier-Stokes equations if geometric aliasing is not significant.

The second technique, namely the *Global* approach, involves the use of richer quadrature order than is typically necessary for linear PDEs. Therefore this

strategy, targets both PDE- and geometrical-aliasing sources which arise when deformed or curved elements are used, as often occurs in industrially-relevant problems where complex geometries are present. In contrast to the first approach, this technique is more computationally intensive, but can address all the aliasing sources arising in any spectral element method when coupled with a sufficiently large number of quadrature points to tackle the nonlinearities within the problem.

The implementation details of both techniques have been thoroughly explained and we applied them to two high-order discontinuous discretisations, the DG method and the FR approach. We first perform a set of experiments considering a linear and a quasi-linear advection equation on both regular and deformed/curvilinear meshes with the main aim of quantifying the effects that both dealiasing techniques have on the discontinuous discretisations considered. We then showed how the dealiasing strategies can be applied to compressible and incompressible flow problems improving the numerical stability of the simulations.

Based on the above analysis and supported by numerical examples shown, the main findings of this chapter can be summarised as follows:

1. The consistent integration rules presented in Kirby and Karniadakis (2003) are generally true and, for GLL points, can be complemented by the following expression

$$Q_{min} \geq P_{exp} + \frac{P_{order}}{2} + \frac{3}{2},$$
$$P_{exp} \geq P_{order},$$

   where Q_{min} is the minimum number of quadrature points to exactly integrate the highest-degree of nonlinearity P_{order} within the problem considered and for a given expansion order P_{exp}. Note that in the above expression P_{order} can be a combination of geometrical and PDE nonlinearities and therefore the overall degree of the nonlinearity can be higher than that dictated by the PDE itself.

2. In a discontinuous discretisation, it is generally not possible to fully control (i.e. up to machine precision) all aliasing effects, since the boundary terms which dictate interface contributions introduce non-polynomial functions into the problem being solved.

3. The interface dealiasing may play a bigger role as the order of the interface flux function increases, becoming comparable to the volumetric dealiasing contribution. Therefore, when implementing these class of dealiasing techniques, the dealiasing of the boundary terms must be taken into account.

4. Geometrical dealiasing, in contrast to PDE-aliasing, is responsible for modifying the mass matrix and it may therefore change the numerical dissipation

and dispersion of the discretisation employed as shown in Kopriva (2009) and Gassner and Kopriva (2011).

5. The connections between DG and FR presented in De Grazia et al. (2014) and in chapter 2 hold true also using the dealiasing strategies described.

6. The two dealiasing strategies can be used in a complementary manner and when applied to challenging applications have proven effective and have increased the numerical stability of the simulations. This result is in agreement with the findings in the recent work of Malm et al. (2013) where a direct connection between numerical stability and consistent integration has been shown.

As a future step, one can think to include an efficient dealiasing technique addressing geometrical aliasing sources separately from PDE aliasing, by performing a pre-conditioning of the mesh. Specifically, it is possible to define an appropriate geometrical deformation estimator which carries out an analysis of the mesh a priori and identifies the elements needing additional quadrature points to avoid geometrical aliasing issues. In the next chapter we aim to apply the numerical techniques developed throughout this thesis work to three-dimensional compressible boundary-layer flows at high Reynolds numbers.

Chapter 5

Compressible boundary layers: flow past an isolated 3D hump

In this chapter we apply the numerical approaches described in this thesis work to three-dimensional compressible boundary-layer flows. Specifically, we have made extensive use of the DG_{SEM} method with diagonal mass matrix in order to exploit the reduced computational costs when the stability of the simulation permitted it. In case of aliasing-driven instabilities we use the DG method with *Global* dealiasing technique, thus preventing the simulation to suffer from this instability.
In this study we aim to analyse the flow past an isolated three-dimensional Gaussian roughness element (also referred to as hump) in a high speed subsonic regime. In the following we will present a Direct Numerical Simulation (DNS) of a high-speed subsonic boundary-layer flow encountering a three-dimensional hump. This work was motivated by the lack of DNS data of boundary-layer flows past roughness elements in a similar regime which is typical of civil aircraft.
We considered different heights of the hump: the smaller heights resulted in a weakly nonlinear regime, whilst the bigger in a fully nonlinear regime with an increasing laminar separation bubble arising downstream of the roughness element. The Reynolds number was sufficiently high to be considered in the regime of aeronautical applications (especially when considering small imperfections at the leading edge of wings). This chapter is structured as follows, in section 5.1 we overview the literature concerning boundary-layer flows. Section 5.2 details the model problem considered and in section 5.3 we describe the DNS process used. In section 5.4 we present the numerical results and, finally, in section 5.5, we briefly summarise the main conclusions of the chapter.

5.1 Introduction

Modern theory of separation of a fluid flow over a surface relies on an asymptotic analysis of the Navier-Stokes equations at large Reynolds numbers. From this point of view the most fruitful method is that of matched asymptotic expansions. The basic idea of this method belongs to Prandtl who first used it in its classical study on boundary layers in 1904.

According to Prandtl's theory, a high Reynolds number flow past a rigid body has to be subdivided into two regions: the main part of the flow that should be treated as inviscid and a thin[1] region near the wall that is predominantly viscous. Prandtl termed the latter region boundary layer and, in his opinion, the flow separation takes place according to the specific behaviour of the flow in this region. Flow development in boundary layers depends on the pressure distribution along the wall. If the pressure gradient is negative, so that the pressure decreases downstream, then the boundary layer remains well-attached. However if the pressure gradient is adverse, so that the pressure starts to rise in the direction of the flow, the boundary layer tends to separate from the rigid surface.

The classical boundary-layer theory, intended by Prandtl for predicting flow separation, was based on the so-called hierarchical approach where the outer inviscid flow is calculated first ignoring the existence of the viscous region, and only after that one can study the boundary layer. Although it was found that the classical form of the boundary-layer theory was insufficient for describing the separation phenomenon, the mathematical approach suggested by Prandtl for analysing high Reynolds number flows formed the cornerstone of all the subsequent studies of the boundary layer. Prandtl's idea of subdividing the entire flow field in different regions is the basis of one of the most powerful tools in modern asymptotic analysis, the method of matched asymptotic expansions.

A key element on the separation process, which was not fully appreciated in Prandtl's theory, is a mutual interaction between the boundary layer and the external inviscid flow. The asymptotic theory of viscous-inviscid interaction is known as triple-deck theory and was specifically designed with the purpose of describing the phenomenon of the boundary-layer separation at large values of the Reynolds number. The theory was formulated by Neiland (1969) and Stewartson (1969b) for the self-induced separation in supersonic flow and by Stewartson (1969a) and Messiter (1970) for incompressible fluid flows near the trailing edge of a flat plate. Later it became clear that the viscous-inviscid interaction plays a key role in many fluid flows. For instance, it governs upstream influence in supersonic boundary layers as well as development of different modes of instabilities. When a separation phenomenon is involved, the theory has been extended to describe boundary-layer

[1]The thickness of this regions depends on value of the Reynolds number. The higher the Reynolds number the thinner the region.

separation from a smooth body surface in an incompressible regime, supersonic flow separation due to a shock wave impinging upon the boundary-layer interaction, separation at the trailing edge of a thin airfoil as result of an increase in the angle of attack, leading-edge separation, separation due to a wall roughness, etc.; for a detailed description of the theory see Sychev et al. (1998).

According to the triple-deck theory, the flow near the separation point is described by the Prandtl boundary-layer equations. The main alteration to Prandtl's classical theory is that the pressure acting on the boundary layer is not known in advance, but has to be found as a part of the solution, as it is affected by the displacement effect of the boundary layer. The principal advantage of the triple-deck theory is that there are no restrictions on how large the Reynolds number is. In fact, the larger the Reynolds number the more accurate the theory is. However, numerical solutions of triple-deck equations are rather difficult, especially when the separation region is not small. It took a decade before reliable numerical techniques were developed.

Triple-deck theory assumed a crucial role in predicting flow separation in boundary layers especially when computational resources were not available. Yet, also today, triple-deck theory can be seen as an effective reduced model which allows the fast prediction of the main features of a flow although it can eventually fail when treating complex geometries as commonly demanded in industrial applications. Nevertheless, it can be used as an a priori analysis tool within an industrial process for better understanding the underlying physics of a given flow configuration.

In recent years a number of experimental and computational studies have been focused on the interaction between boundary-layer flows (both internal and external) and a small wall roughness, a hump or and indentation. In Smith et al. (1977) the authors analysed the flow structure generated by a shallow three-dimensional hump in a two-dimensional boundary layer. They showed that a corridor emerged in the wake behind the obstacle where the disturbance was much greater than anywhere else in the flow and decayed much more slowly. They applied the three-dimensional counterpart of the classical triple-deck theory. This is applicable provided that the spanwise and longitudinal dimensions of the hump are the same, and both are $O(Re^{-3/8})$, where Re is the Reynolds number based on the distance of the hump from the leading edge.

The theory of Smith et al. was then generalised by Sykes (1978) to make it applicable to the atmospheric boundary layer over hills. In this layer the flow stratification is important and, if this is increased or the surface roughness is made smaller than that assumed in Smith et al. (1977), then the so called compensation flow is realised. The three-dimensional boundary-layer equations can still describe the flow, but the displacement thickness of the boundary layer fully annuls the roughness topology. This condition on the displacement must be used for finding the

pressure distribution. The properties of these flows were studied in Smith (1976, 1980) and Sykes (1980). They found that there is a significant difference between three-dimensional boundary layers with compensation and two-dimensional (see Bogolepov and Neiland (1971)). In two-dimensional flows the disturbances do not propagate upstream, whilst in three-dimensional flows this is possible. Comparison of the linear solution of two- and three-dimensional flows with compensation is shown by Bogolepov and Neiland (1985). Recently the boundary-layer interaction with small wall roughness was studied by Korolev (2007) - who studied the compensation flow regime. In Goldstein et al. (2010, 2011) the authors analysed the perturbations produced in the two-dimensional boundary layer by a row of identical and equally spaced roughnesses placed in a straight line perpendicular to the oncoming flow. The horizontal dimensions of each roughness are assumed to be of the order of the local boundary-layer thickness. It was found that the linear perturbations decay downstream of the line of the roughnesses. However, the analysis of the $O(h^2)$ approximation showed that there might be a slow growth of the perturbations in the main part of the boundary layer, and also in the lower viscous part downstream of the flow regime with compensation. In Ruban and Kravtsova (2013) the two-dimensional boundary layer encountering a three dimensional roughness was studied in hypersonic flow regime. It was found that, if the roughness height is small, then the perturbations decay rather fast everywhere except in the wake behind the roughness where they decay slowly. However if the height is greater than some critical value, two symmetric vortices form in the wake and they appear to support themselves and grow with the distance from the roughness.

All of the work cited above is based on triple-deck theory and analytical solution of simplified equations. Because of their high computational cost, direct numerical simulations have not been widely used in studies concerning roughness-induced transition in high speed flows. However, a significant improvement in numerical techniques for solving the Navier-Stokes equations occurred and the fast expansion of computational resources have made the DNS a real possibility. A comprehensive summary of the role of the DNS in fluid mechanics until 1998 can be found in Moin (1998) and references therein. More recently DNS has been used to simulate the effect of roughness elements on high speed boundary-layer transition and we outline these contributions in the following paragraphs. Marxen and Iaccarino (2008) investigated the effect of two-dimensional localised roughness on boundary-layer instability in a supersonic flow at Mach = 4.8. They observed a strong amplification of the disturbances in the separation zones upstream and downstream of the roughness, while along the roughness element the perturbation was strongly damped. A similar study was performed by Groskopf et al. (2008), who found that instabilities can occur in the recirculation zone around the roughness element and that the trailing edge vortices generate streaks that sustain strongly growing convective instabilities similar to those studied by Chen et al. (20); Choudari et

al. (2013).

Redford et al. (2010) studied transition due to an isolated smooth bump at Mach = 3.0 and 6.0 and found a strong compressibility effect on the detached shear layer behind the roughness element. They also discovered a correlation based on roughness height Reynolds number, Mach number and wall temperature. From this correlation they could see a separation between the cases that went through transition and those that remained laminar. Bernardini et al (2012) did comparable simulations extending the range of flow conditions. They found a similar correlation and- so- validated this transition criterion. Bernardini et al (2014) incorporated the effect of the projected frontal shape and aspect ratio of the roughness element by proposing a new transition criterion based on the momentum deficit induced by a roughness element.

De Tullio and Sandham (2012, 2015) performed a receptivity study at Mach = 6.0 boundary-layer flow over a sharp-edged localised roughness element. They looked at the the effect of roughness height, type of imposed disturbance and wall temperature. They found that the roughness height has a significant effect on instability growth. Specifically, for a roughness height similar to the local displacement thickness the detached shear layer becomes receptive to a broad range of frequencies, increasing the growth rate of instabilities. A roughness element half as high had only a small effect on mode growth. They also found that wall cooling tends to reduce the growth rate of instabilities and different types of disturbances lead to differences in amplitude functions and growth rates.

De Tullio et al. (2013) analysed the instabilities in the wake behind a sharp-edged roughness element at Mach = 2.5 using direct numerical simulations, bi-global stability analysis and analysis of the three-dimensional parabolised stability equations (PSE). They found that bi-global stability analysis can accurately predict the mode shapes but not the growth rate, while the three-dimensional PSE analysis could be used to accurately predict the growth rate of these modes. A similar study was performed by Van den Eynde (2015) who extended the work of De Tullio et al. (2013) to roughness elements with three-dimensional geometries of various type considering a hypersonic flow at Mach = 6.0.

5.2 Problem formulation

In all the numerical studies reported in the previous section the authors considered a supersonic or hypersonic regime. In this work we want to study a high speed subsonic flow of a perfect gas over a rigid flat surface, which has a small roughness located at a distance L from the leading edge AB (see Figure 5.1). Our task is to analyse the disturbances produced in the flow by the interaction between the

boundary layer and the roughness. The roughness shape is defined by the equation:

$$f(x,y) = h_r e^{-\frac{[(x-L)^2+y^2)]}{\beta^2}} \tag{5.1}$$

where L is the distance from the leading edge of the flat plate to the centre of the roughness element, β is a parameter which controls the size of the roughness element, h_r is the height of the roughness. A positive value of h_r defines a hump, a negative an indentation. The Reynolds number was set equal to 4×10^5 for all the simulations performed and is defined as follows:

$$Re = \frac{\rho_\infty U_\infty L}{\mu_\infty}, \tag{5.2}$$

where U_∞ is the free-stream velocity, ρ_∞, μ_∞ are the density and the dynamic viscosity of the fluid at free-stream condition. L is the typical length scale of the oncoming boundary layer which defines the distance the boundary layer develops along the body surface before encountering the obstacle.

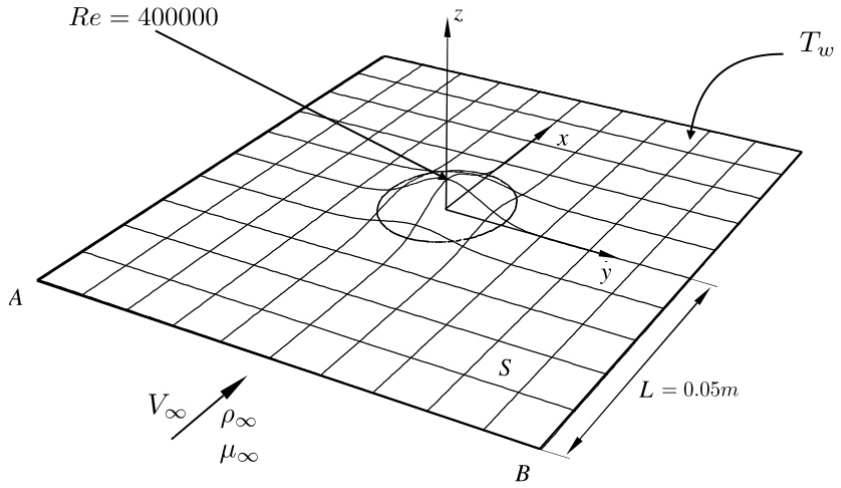


Figure 5.1: Problem formulation. Courtesy of Ruban and Kravtsova (2013).

In this DNS study we considered a transonic condition with a Mach number $Ma_\infty = 0.87$ [2]. This Mach number can be found over aircraft wings in the region

[2]The Mach number is defined as the ratio between the free-stream speed of sound c_∞ and the free-stream velocity U_∞: $Ma_\infty = U_\infty / c_\infty$.

close to the leading edge. In this region the shock effects which can be encountered downstream are not present. Therefore, in the simulations performed there is no need for any shock-capturing technique. The physical parameters adopted for the free-stream conditions are reported in Table 5.1. The free-stream pressure, p_∞, density, ρ_∞, dynamic viscosity, μ_∞, and thermal conducivity, k_∞, correspond to an altitude of eleven thousand meters above sea level and they are based on the International Standard Atmosphere (ISA). The free-stream velocity $\mathbf{V}_\infty = [U_\infty, 0, 0]$ was chosen in such a way that it satisfied the Mach number and we applied isothermal boundary conditions using a temperature T_w at the wall. The temperature T_w was obtained through the Crocco integral defined as follows:

$$c_P T_w = V_\infty^2 \left(\frac{1}{2} + \frac{1}{(\gamma - 1) Ma_\infty^2} \right) . \tag{5.3}$$

The Prandtl number[3] Pr was fixed and corresponds to the value commonly adopted for air.

Ma_∞ [−]	ρ_∞ $[kg/m^3]$	p_∞ $[Pa]$	μ_∞ $[Pa \cdot m]$	k_∞ $[W/(m \cdot K)]$	U_∞ $[m/s]$	T_w $[K]$	Pr [−]
0.87	0.364	22600	$1.167 \cdot 10^{-5}$	0.01629	256.5	258.0	0.72

Table 5.1: Free-stream physical parameters.

We used different roughness heights, in order to investigate both weakly nonlinear and nonlinear regimes. The parameters adopted for the shape (i.e. β) and the height (i.e h_r) of the humps are reported in table 5.2, along with the hump length, ℓ_{999}, that is defined as the length for which the ratio $h(x)/h_r$ becomes smaller than 0.1%.

[3]The Prandtl number is defined as the ratio of kinematic viscosity ν to thermal diffusivity $\alpha = k/(\rho C_p)$ that is: $Pr = C_p \mu / k$ where C_p is the specific heat at constant pressure.

Hump	h_r $[m]$	β $[m]$	ℓ_{999} $[m]$	Ma_r $[-]$	Re_r $[-]$
5 %	$1 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.054	7.71
10 %	$2 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.107	30.70
15 %	$3 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.162	69.45
20 %	$4 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.213	122.00
25 %	$5 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.265	191.4
30 %	$6 \times 1.94 \cdot 10^{-5}$	$1.7961 \cdot 10^{-4}$	$9.60 \cdot 10^{-4}$	0.323	276.9

Table 5.2: Dimensionless quantities relative to the different humps.

In addition we also report the values of the Mach number Ma_r calculated at $[x, y, z] = [L - \ell_{999}/2, 0, h_r]$ and the correspondent Reynolds number based on the roughness height Re_r calculated as follows:

$$Re_r = \frac{\rho_r u_r h_r}{\mu_r}, \tag{5.4}$$

where ρ_r, u_r and μ_r are the density, streamwise velocity and dynamic viscosity at the location defined above.

The values of the first column in Table 5.2 correspond to the percentages with respect to the thickness of the incompressible Blasius boundary layer at that location. This thickness is defined as the distance from the wall to the point where the velocity is 99% of the free-stream velocity. The correspondent values to the compressible boundary-layer thickness we studied are reported in Table 5.3. The values in the third column of Table 5.3 represent the percentages of the hump height with respect to its width.

Hump	h_r/δ [%]	h_r/ℓ_{999} [%]
5 %	3.90	2.02
10 %	7.80	4.04
15 %	11.70	6.06
20 %	15.60	8.08
25 %	19.50	10.10
30 %	23.40	12.12

Table 5.3: Hump percentages.

5.3 Numerical approach

In this section we first present the numerical setup for the DNS of the boundary-layer flow past a roughness element.

5.3.1 DNS domain and boundary conditions

The governing equations considered for the direct numerical simulations are the unsteady three-dimensional compressible Navier-Stokes equations, as presented in chapter 3, Eqs. 3.10, 3.11. The spatial numerical discretisation of the advection operator used is the DG_{SEM} method with a lumped mass matrix, for which, we showed the connections with the FR_{g2} scheme in chapter 2. The use of this scheme allowed a larger time-step, approximately the double than the DG scheme with an exact mass matrix (or equivalently the FR_{DG} scheme) and permitted a consistent speed-up of the simulations because of the diagonal nature of the mass matrix. Note that we used the 'exact' Riemann solver to calculate the numerical fluxes at the element interfaces and we exploited the weak strategy presented in chapter 3 for applying the boundary conditions. The diffusion term was discretised with the local discontinuous Galerkin approach (LDG) presented in chapter 1. The polynomial order used was P = 3 for all the simulations carried out and we applied a *Global* dealiasing technique when needed in order to avoid numerical instabilities in proximity of the hump location. The temporal discretisation used was an explicit 4th-order Runge-Kutta scheme. The time-step used for the simulations was $dt = 2 \times 10^{-10} s$ which is the maximum time-step that allowed the simulation to remain stable.

Figure 5.2 shows a cross-section of the domain of the simulations.. We applied the following boundary conditions described in chapter 3:

- *Inflow*: inflow boundary condition;
- *Outflow*: outflow boundary condition;
- *Top*: farfield boundary condition;
- *Wall*: no-slip, isothermal wall;

For the inflow and top boundary conditions we used the compressible similarity solution described in the next section. The same similarity solution was also used as the initial condition for starting the simulations. In the spanwise direction we applied periodic boundary conditions.

The computational domain is a cartesian box extending for $L_x \approx 66\delta$, $L_y \approx 33\delta$ and $L_z \approx 10\delta$ in the streamwise (x), spanwise (y) and normal (z) directions, where

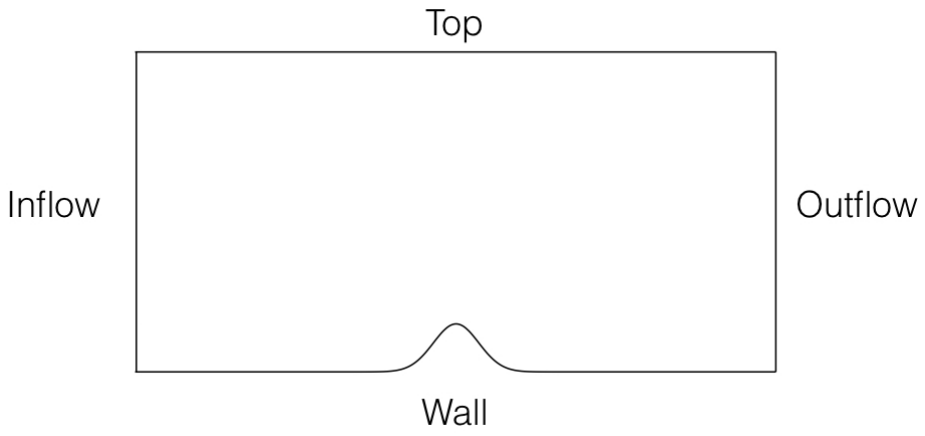


Figure 5.2: DNS domain and boundary conditions.

δ is the compressible boundary-layer thickness at the roughness location chosen as the reference length of the domain.

The size of the domain in the streamwise direction was based on a grid sensitivity study performed by Mengaldo et al. (2015b) in a similar two-dimensional problem. The size of the domain in the spanwise direction was based on considerations of the dimensions of flow structures behind the hump. In particular Ruban and Kravtsova (2013) observed that, the vortices which form behind a roughness element decay rather fast with the distance from the element in the spanwise direction while they decay much more slowly in the streamwise direction. We therefore selected a grid spacing in the spanwise direction that is half of the grid spacing in the streamwise direction. The normal size used was based on consideration on the maximum hump height to simulate and such that any reflection from the top boundary would hit the outflow boundary without affecting the boundary layer.

The final mesh adopted was constituted by $147 \times 36 \times 31$ elements in the streamwise, spanwise and wall-normal directions respectively and is depicted in Figure 5.3 and 5.4. As already, mentioned, we used a polynomial solution of order three within each element and, therefore, the number of solutions points is equal to 588 in the streamwise direction, 124 in the wall-normal and 144 in the spanwise. A stretching technique was used in normal direction to better capture the boundary layer. The minimum Δz in proximity to the wall was equal to $2 \times 10^{-6} m$ which corresponds approximately to $0.75 y^+$. A nonuniform distribution in the streamwise and spanwise direction was also used to allow better resolution in the region close to the obstacle. In the streamwise direction the maximum resolution near the

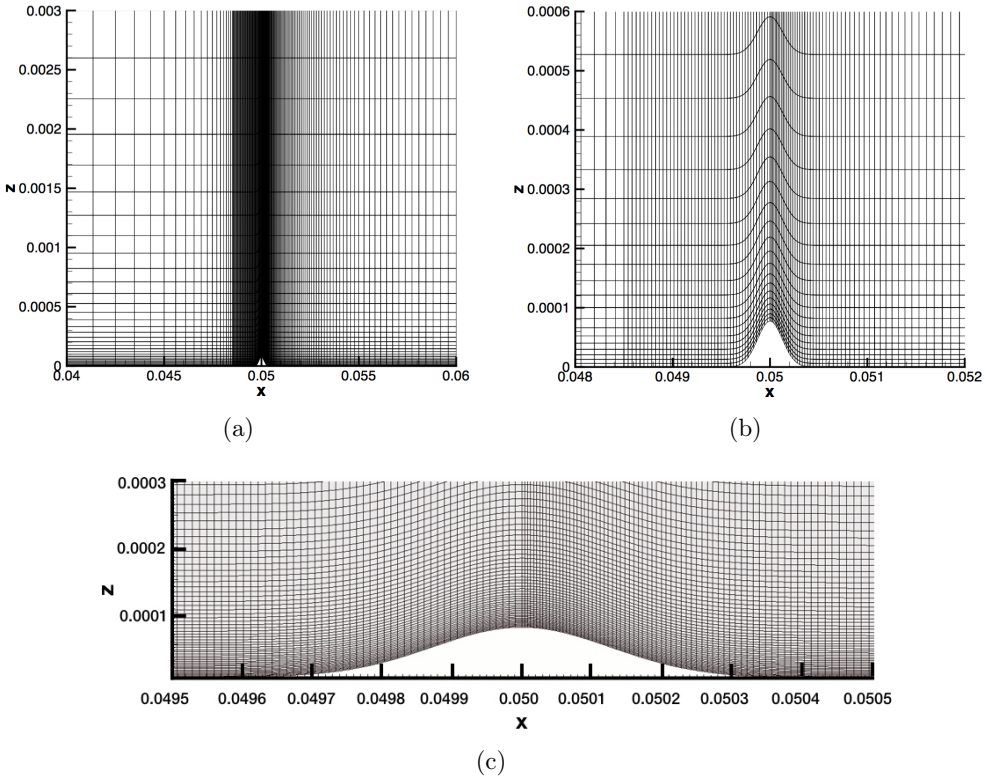


Figure 5.3: Streamwise view of the mesh adopted for the DNS simulation. Figure (a) shows the entire mesh, figure (b) shows the details of the mesh around the obstacle, figure (c) shows the resolution of the solution points around the hump.

roughness element was equal to $4 \times 10^{-6}m$ which corresponds approximately to $1.5y^+$, while in the spanwise direction was equal to $2 \times 10^{-5}m$ which corresponds approximately to $7.5y^+$. The description of the roughness element geometry was obtained using a 4th-order spline to achieve an accurate representation. For more details on the curvilinear boundary-layer mesh adopted one can refer to Moxey (2015).

The simulations were run until convergence and the steady-state criterion adopted is the following:

$$\frac{||g^t - g^{t+\Delta t}||}{g_\infty} < 1 \times 10^{-8}, \tag{5.5}$$

where g is a generic variable and g_∞ is the related free-stream value. The DNS

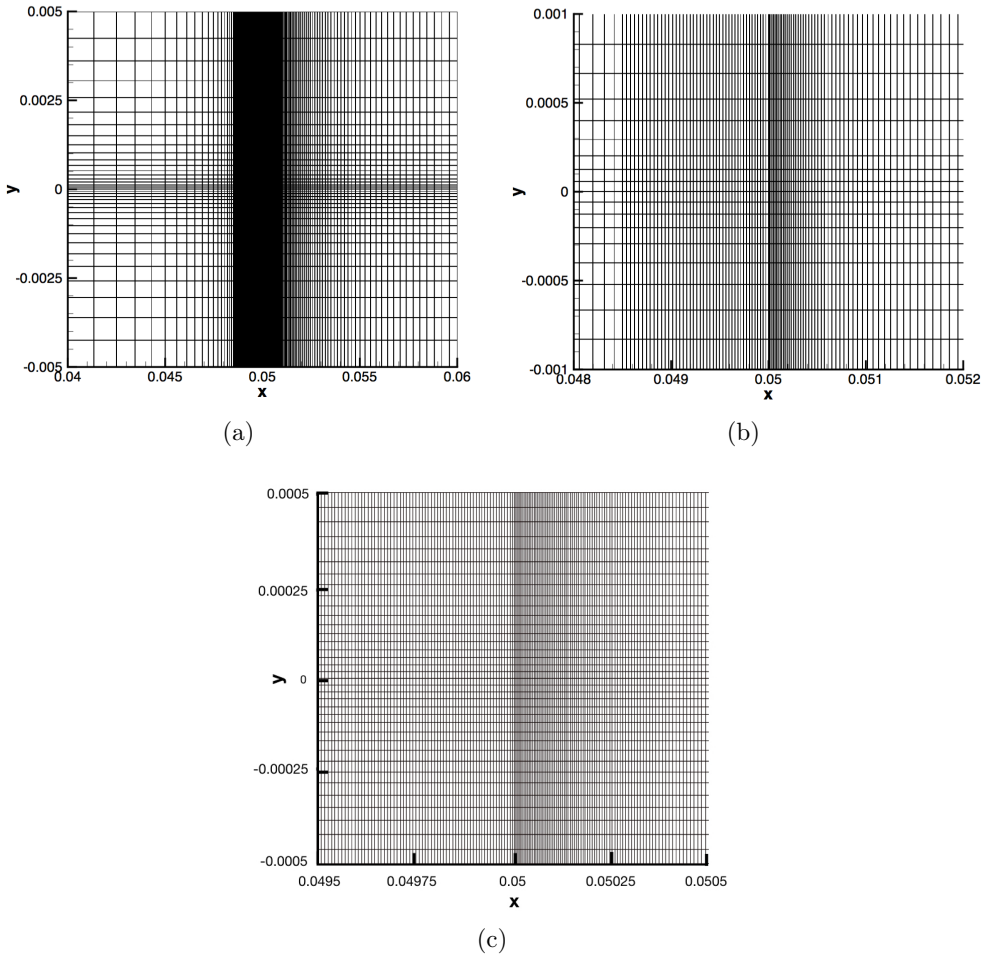


Figure 5.4: Spanwise view of the mesh adopted for the DNS simulation. Figure (a) shows the entire mesh, figure (b) shows the details of the mesh around the obstacle, figure (c) shows the resolution of the solution points around the hump.

process described made the simulations feasible in terms of computational costs and represents an efficient way for performing DNSs on the configurations investigated.

Similarity solution

For compressible boundary-layer flow, an exact solution to the Navier-Stokes equations does not exist. However, it is possible to simulate the solution based on

the boundary-layer approximation. In the present study of two-dimensional compressible boundary layer developing over a flat surface encountering a roughness element, the Reynolds number is sufficiently high that a thin boundary layer will develop along the wall. The compressible boundary-layer equations can be used as initial conditions and as inflow conditions. Although the initial boundary-layer solution can deviate significantly from the flow at a later time in large parts of the flow domain (e.g. when separation occurs), it gives a good approximation to the solution at the inflow boundary, provided that the flow is undisturbed in the inflow region. A correct inflow condition keeps the computational extent to a minimum, while a good approximation of the initial flow shortens the route to the (statistically) stationary state. Based on the Navier-Stokes coordinate scaling, the gradients in the streamwise and normal directions within the boundary layer are quite different.
We can introduce a new scaling of the streamwise and normal coordinates as follows:

$$x' = \frac{x}{L}, \qquad z' = \frac{Re^{\frac{1}{2}}}{L} z, \tag{5.6}$$

where L is the horizontal length scale. An equivalent scaling can be introduced for the time:

$$t' = t\frac{U_\infty}{L}, \tag{5.7}$$

and for the flow variables:

$$\rho' = \frac{\rho}{\rho_\infty}, \quad u' = \frac{u}{U_\infty}, \quad w' = \frac{Re^{\frac{1}{2}}}{U_\infty} w, \quad p' = \frac{p}{\rho_\infty U_\infty^2}, \quad T' = \frac{T}{T_\infty}, \tag{5.8}$$

where u and w are the velocity components in the streamwise and normal directions, whilst ρ, p and T are the thermodynamic quantities: density, pressure and temperature respectively.

Taking the dominant terms in case $Re \gg 1$ from the Navier-Stokes equations based on the new scaling, we come to the non-dimensionalised compressible boundary-layer equations:

$$\begin{cases} \dfrac{\partial \rho' u'}{\partial x'} + \dfrac{\partial \rho' w'}{\partial z'} = 0 \\ \rho' u' \dfrac{\partial u'}{\partial x'} + \rho' w' \dfrac{\partial w'}{\partial z'} = -\dfrac{\partial p'}{\partial x'} + \dfrac{\partial}{\partial z'}\left(\mu' \dfrac{\partial u}{\partial z'}\right) \\ \dfrac{\partial p'}{\partial z'} = 0 \\ \rho' u' \dfrac{\partial T'}{\partial x'} + \rho' w' \dfrac{\partial T'}{\partial z'} = A\left[u' \dfrac{\partial p'}{\partial x'} + w' \dfrac{\partial p'}{\partial z'} + \mu' \left(\dfrac{\partial u}{\partial z'}\right)^2\right] + B \dfrac{\partial}{\partial z'}\left(\mu' \dfrac{\partial T}{\partial z'}\right) \end{cases} \tag{5.9}$$

where $A = (\gamma - 1)Ma^2$ and $B = Pr^{-1}$.
We can now introduce the Dorodnitsyn-Stewartson transformation:

$$\xi = x', \qquad \eta = \frac{1}{(2x')^{\frac{1}{2}}} \int_0^{z'} \rho'(x', s)ds = \left(\frac{Re}{2x}\right)^{\frac{1}{2}} \int_0^z \rho(x, s)ds. \tag{5.10}$$

We also define a stream function Ψ such that:

$$\rho' u' = \frac{\partial}{\partial z'}\Psi, \qquad \rho' w' = -\frac{\partial}{\partial x'}\Psi, \tag{5.11}$$

which implies that the continuity equation is identically satisfied. Using the following representation:

$$\Psi = (2\xi)^{\frac{1}{2}} F(\xi, \eta) \tag{5.12}$$

and substituting Eq. 5.12 into the system 5.9 we find that:

$$\begin{cases} \dfrac{\partial}{\partial \eta}\left(\chi \dfrac{\partial^2 F}{\eta^2}\right) + F\dfrac{\partial^2 F}{\partial \eta^2} = \dfrac{2\xi}{\rho'}\dfrac{\partial p'}{\partial \xi} + 2\xi\left(\dfrac{\partial F}{\partial \eta}\dfrac{\partial^2 F}{\partial \xi \partial \eta} - \dfrac{\partial F}{\partial \xi}\dfrac{\partial^2 F}{\partial \eta^2}\right) \\ \dfrac{\partial p'}{\partial \eta} = 0 \\ \dfrac{\partial}{\partial \eta}\left(\dfrac{\chi}{Pr}\dfrac{\partial T'}{\partial \eta}\right) + F\dfrac{\partial T'}{\partial \eta} + (\gamma - 1)Ma^2\chi\left(\dfrac{\partial^2 F}{\partial \eta^2}\right)^2 = \\ -(\gamma - 1)Ma^2\dfrac{2\xi\dfrac{\partial F}{\partial \eta}}{\rho'}\dfrac{\partial p'}{\partial \xi} + 2\xi\left(\dfrac{\partial F}{\partial \eta}\dfrac{\partial T'}{\partial \xi} - \dfrac{\partial F}{\partial \xi}\dfrac{\partial T'}{\partial \eta}\right) \end{cases} \tag{5.13}$$

where $\chi = \rho'\mu'$. This system can be greatly simplified if F and T' depend on η only. In addition, the pressure is constant in the boundary-layer approximation. It can be written as:

$$p' = \frac{1}{\gamma Ma^2} \tag{5.14}$$

and, hence, we finally obtain a system of two ordinary differential equations:

$$\begin{cases} \dfrac{d}{d\eta}\left(\chi\dfrac{d^2 F}{d\eta^2}\right) + F\dfrac{d^2 F}{d\eta^2} = 0 \\ \dfrac{d}{d\eta}\left(\dfrac{\chi}{Pr}\dfrac{dT'}{d\eta}\right) + F\dfrac{dT'}{d\eta} + (\gamma - 1)Ma^2\chi\left(\dfrac{d^2 F}{d\eta^2}\right)^2 = 0 \end{cases} \tag{5.15}$$

with the following boundary conditions for F:

$$\left.\frac{dF}{d\eta}\right|_{z=0} = 0, \qquad F|_{z=0} = 0, \qquad \left.\frac{dF}{d\eta}\right|_{z\to\infty} = 1, \tag{5.16}$$

while for the temperature one can choose either:

$$\begin{aligned} T'|_{z=0} &= \frac{T_w}{T_\infty} \qquad T'\Big|_{z\to\infty} = 1, \qquad \textit{isothermal wall}, \\ \frac{dT'}{d\eta}\Big|_{z=0} &= 0, \qquad T'|_{z\to\infty} = 1, \qquad \textit{adiabatic wall}. \end{aligned} \tag{5.17}$$

These boundary conditions fully specify the similarity solution. The solution to this

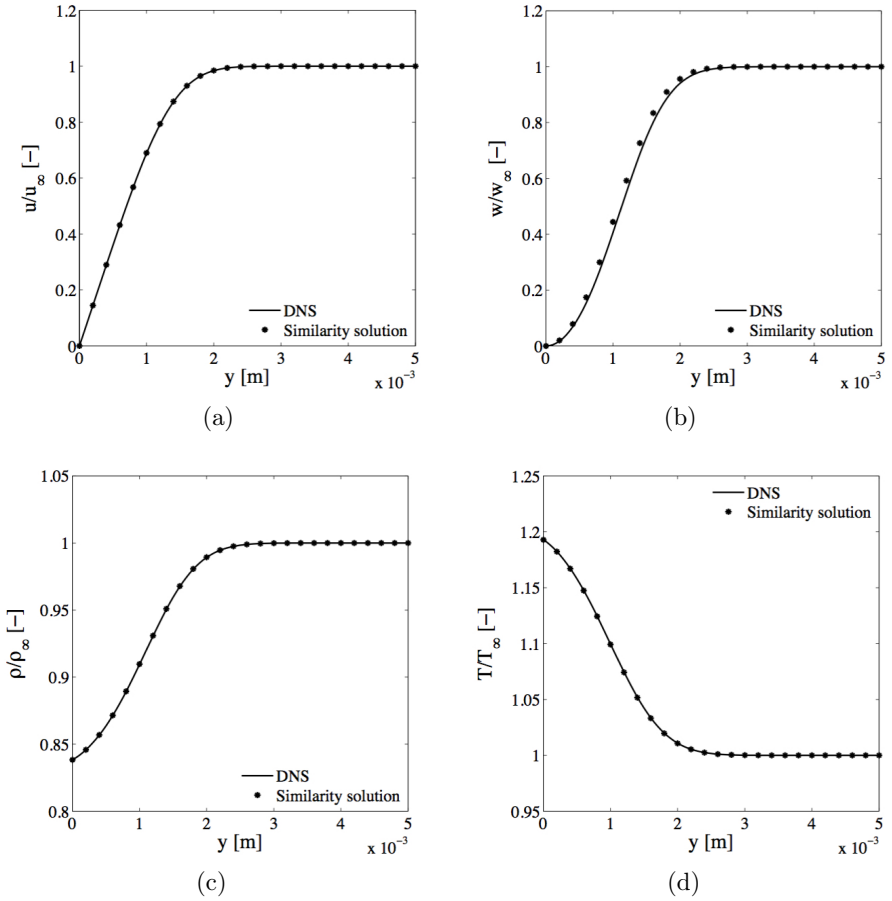


Figure 5.5: Streamwise velocity (a) and normal (b) velocity components, density (c) and temperature (d) at $x = 0.05\,m$, normalized by free-stream values, obtained from similarity solution (dotted) and Navier-Stokes computation (solid).

system of equations can be used to reconstruct the desired flow field since use can be made of the relations $u' = dF/d\eta$ and $\rho' = 1/T'$ in view of the constant pressure. This results in u', ρ', T' and p'. The non-dimensional vertical component of the velocity w' is then calculated by using the continuity equation. As a result the field is now described as a function of η and we can then determine the dependence on the original coordinates x and z. To examine the performance of the similarity transformation, and therefore the quality of the initial condition, we compare the similarity solution of the compressible boundary-layer equations and a converged Navier-Stokes solution. Specifically we considered a flow over a flat plate at $Ma = 0.87$ and $Re = (\rho_\infty U_\infty L)/\mu_\infty = 4\times10^5$ with $L = 0.05\,m$ identical to that we aimed to simulate. The mesh adopted for the Navier-Stokes simulation is the same used in the previous chapter (see Figure 3.10). Figure 5.5 shows the quantities normalised by their value at the edge of the boundary layer at $x = L = 0.05\,m$. We can appreciate a very good agreement between the similarity solution and the solution of Navier-Stokes equations.

5.4 Numerical results

In this section, the DNS results of the boundary-layer flow studied are shown and described. Specifically we first present comparisons between two- and three-dimensional cases of different hump heights for $Ma_\infty = 0.87$. We successively show a comparison between two different flow regimes; transonic with $Ma_\infty = 0.87$ and subsonic with $Ma_\infty = 0.5$[4]. Then we focus on the results of the the three-dimensional hump case for the transonic regime in terms of wall shear stress, shape of separation bubble and streamwise vortices. We then draw the conclusion explaining the main findings.

5.4.1 Two-dimensional vs three-dimensional geometry

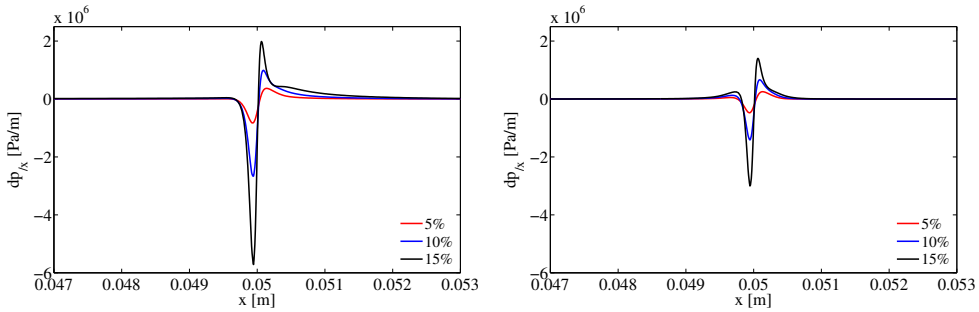
In Figures 5.6 and 5.7 we show dp/dx and τ_{xz} at the wall for the three smallest hump heights studied in the transonic regime ($Ma_\infty = 0.87$) for two- and three-dimensional cases. The three-dimensional results represent the flow quantities in the centerline of the domain.

We can see that the effects of the hump on the boundary-layer flow are larger in the two-dimensional case than in the three-dimensional, both in terms of peaks and longitudinal extension of the perturbation. It is evident that, in order to produce the same perturbation effect of a two-dimensional hump, the equivalent three-

[4]These are the only results where the author considered a flow regime with a different Mach number rather than $Ma_\infty = 0.87$. In this study all the other flow variables are the same as presented in Table 5.1.

dimensional hump must be higher. This trend was expected and can be explained with the smaller blockage effect that the flow around the three-dimensional shape encounters. In all the figures it is possible to see that the distortion of the boundary layer increases as the hump height increases.

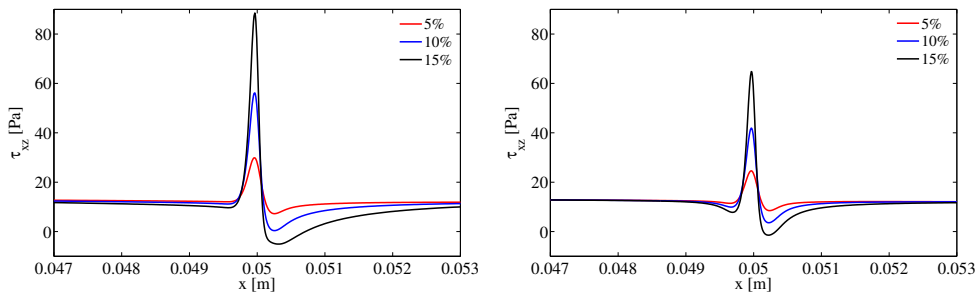
Specifically, in Figure 5.6(a), we can appreciate that the distortion of dp/dx



(a) 2D (b) 3D

Figure 5.6: Comparison of $\frac{dp}{dx}$ (dimensional values) at the wall between two- and three-dimensional case varying the hump heights.

behind the highest hump assumes a slightly different shape with a small change in curvature than the other humps.



(a) 2D (b) 3D

Figure 5.7: Comparison of τ_{xz} (dimensional values) at the wall between two- and three-dimensional case varying the hump heights.

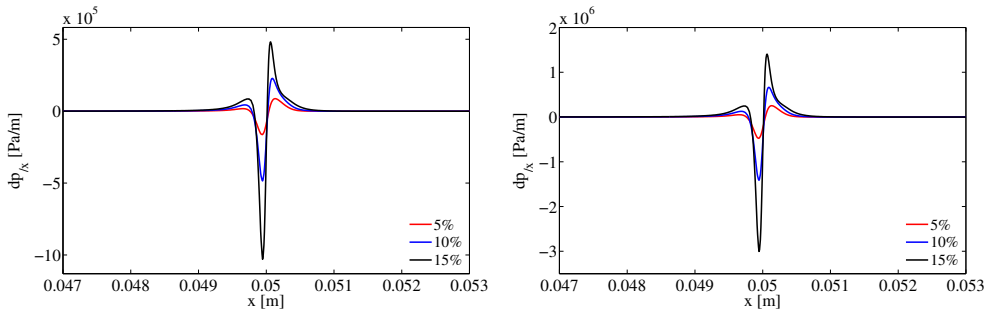
This can indicate that, by increasing the hump height further, we may encounter the development of additional nonlinear effects which could potentially

lead to the unsteadiness of the separation bubble. It is likely that a similar shape will appear also for the three-dimensional case for a bigger hump height.
In Figure 5.7(a) and 5.7(b) we can see that the flow develops a small separation bubble behind the roughness element (the wall shear stress τ_{xz} becomes in fact negative in a small region behind the hump). We also note that the region of separated flow behind the hump is larger in the two-dimensional case than in three-dimensional. It is also clear that the two-dimensional flow separates at a smaller height than the three-dimensional (note the differences in the minimum value of the 10% humps).

5.4.2 Mach number effect

In Figures 5.8 and 5.9 we show dp/dx and τ_{xz} at the wall for the three smallest hump heights studied for two different flow regimes: subsonic with $Ma_\infty = 0.5$ and transonic with $Ma_\infty = 0.87$. These results represent the flow quantities at the centerline of the domain.

In both the figures we can see that the pressure gradient and the skin friction have a very similar behaviour and the main difference is represented by the peaks which are higher in the transonic case (note that the scales of the subfigures are different). As already mentioned the distortion of the boundary-layer solution increases as the hump height increases for both the parameters.

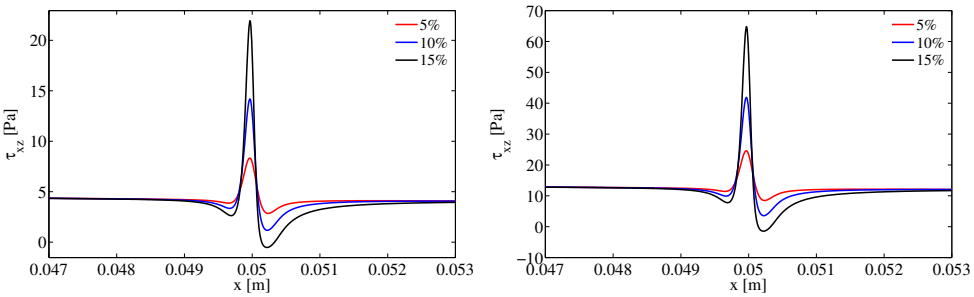


(a) $Ma_\infty = 0.50$ (b) $Ma_\infty = 0.87$

Figure 5.8: Comparison of $\frac{dp}{dx}$ (dimensional values) at the wall for different Mach numbers varying the hump heights.

Specifically, in Figures 5.8(a) and 5.8(b), we can appreciate that the behaviour of dp/dx is very similar in terms of extension of the perturbation. The same comment can be made of 5.9(a) and 5.9(b). In particular, the separation seems to appear at a similar hump height for both the subsonic and the transonic cases. We

can note that the average shear stress is bigger in the transonic case than in the subsonic. This implies that the separation in the transonic case requires a bigger drop in the shear stress. Therefore the transonic effect of the higher shear stress and higher peaks compensate each other and are one of the reasons for which the separation in both transonic and subsonic cases appear at a similar height of the obstacle.



(a) $Ma_\infty = 0.50$ (b) $Ma_\infty = 0.87$

Figure 5.9: Comparison of τ_{xz} (dimensional values) at the wall for different Mach numbers varying the hump heights.

In addition, another factor of compensation is represented by the fact that the boundary layer thickness increases as the free-stream Mach number increases and, therefore, the same height of the hump represents a smaller obstacle in the transonic case than in the subsonic.

5.4.3 Skin friction and pressure distributions

In Figure 5.10, we show the distribution of τ_{xz} and p at the wall in a plane along the centerline of the domain for the different hump heights simulated. It is possible to note that the perturbations increase as the hump height increases and we can observe that starting with the 15% hump the flow develops a separation bubble behind the roughness. The extension of this separation region increases with the height of the hump.
In addition we can observe that the shape of the skin friction distribution behind the 25% hump is different from that of the smaller humps (see Figure 5.10(b)) and this difference is even more evident with the 30% hump. In the region behind the hump, the pressure starts to rise causing flow deceleration. The skin friction decreases, crossing zero at the separation point. It then reaches a minimum and starts to rise slowly. While it remains negative, the fluid in the separation region

is moving in the opposite direction to the rest of the flow. The motion in this area is slow and, for this reason the pressure develops a "plateau", which can be clearly seen in Figure 5.10(d). Then the skin friction starts to decrease again reaching a second minimum and, further downstream it crosses zero at the reattachment point. Through the reattachment point the pressure monotonically increases and further downstream returns to its unperturbed value. Increasing the height of the hump new features are likely to occur in the flow development like those seen by Korolev et al. (2001) in their analysis of the supersonic flow separation near a corner. In this study they observed the skin friction becoming positive again after the first minimum, and after that a secondary region of separation inside the primary region before the second minimum.

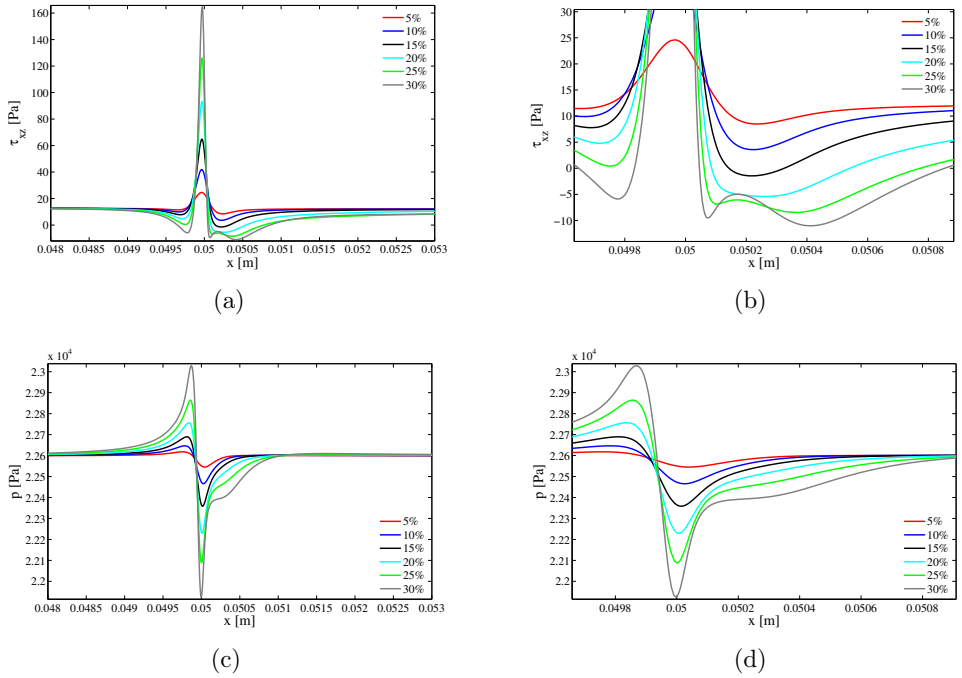
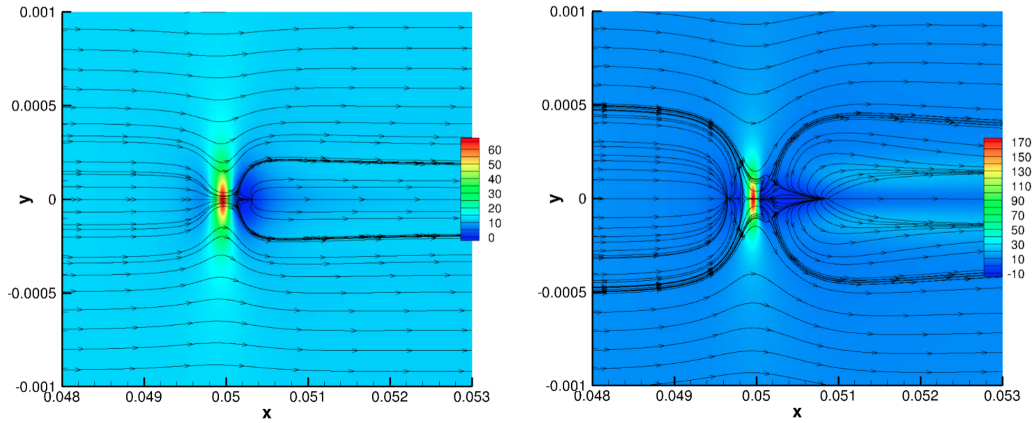


Figure 5.10: Skin friction and pressure distribution along the centerline of the domain varying the hump heights. On the right a zoom around the hump location.

In Figures 5.10(a) and 5.10(b) we can also see that, when the hump becomes high enough, a small region of separation starts to develop before the roughness element, for example observe the 30% hump. This can be better appreciated in Figure 5.11 where the contours of the skin friction at the wall are depicted for the

15% hump and the 30% hump. In both Figures 5.11(a) and 5.11(b) we can note the separation regions (darkest blue zones) behind the roughness element which start from the separation point and end with the node point of attachment where all the skin friction lines are directed outward away from the node. In Figure 5.11(b) there is also evidence of a second smaller separation region before the hump.



(a) 15 %

(b) 30 %

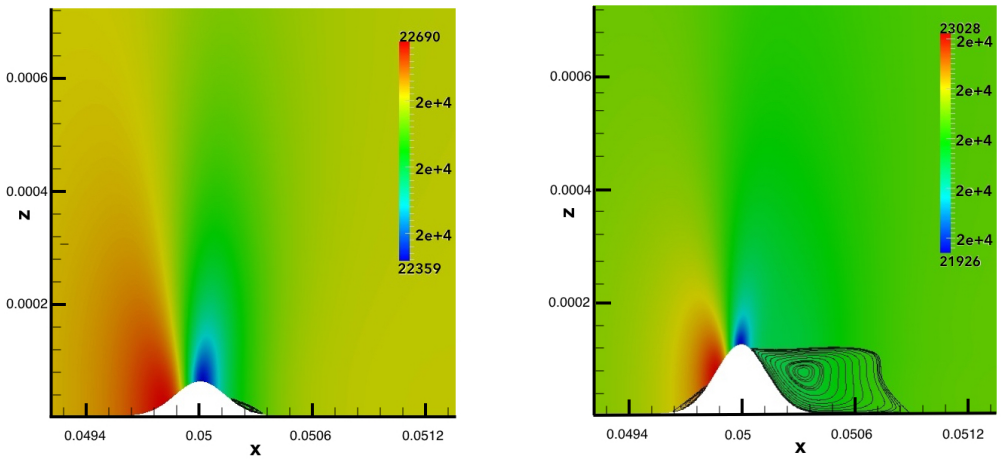
Figure 5.11: Skin friction contours and limit streamlines at the wall varying the hump heights.

The separation regions and the difference of their extension is more evident in Figure 5.12 which shows the pressure distribution over the same humps, 15% and 30% in a plane along the centerline of the domain.

It is possible to appreciate that the disturbance generated by the hump in Figure 5.12(b) is much bigger than that in Figure 5.12(a) both in terms of pressure and separation bubbles. We can also note the unusual shape of the separation bubble for the 30% hump that is caused by the streamwise vortices forming downstream of the hump.

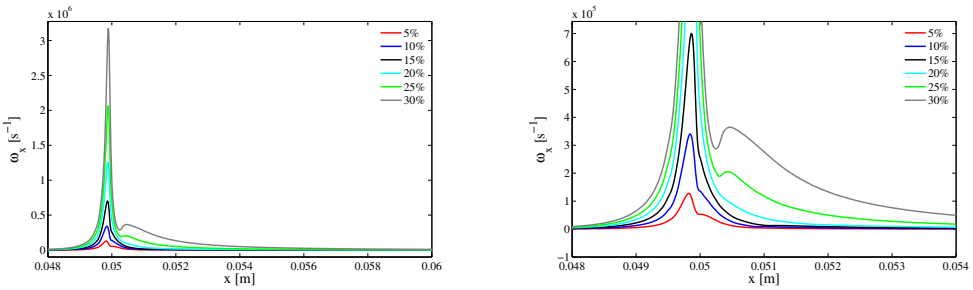
5.4.4 Streamwise vortices

In this section we analyse the generation of streamwise vortices caused by the different humps. The existence of these vortices is the main difference between the flow with a two-dimensional and a three-dimensional hump. In fact streamwise vortices are created only by three-dimensional roughness elements. The main feature of the flow considered is the wake behind the roughness, which is composed of two symmetric counter-rotating vortices.



(a) 15% (b) 30%

Figure 5.12: Pressure distribution and recirculation regions on a plane along the centerline of the domain varying the hump heights.



(a) (b)

Figure 5.13: Magnitude of maximum streamwise vorticity. On the right a zoom around the hump location.

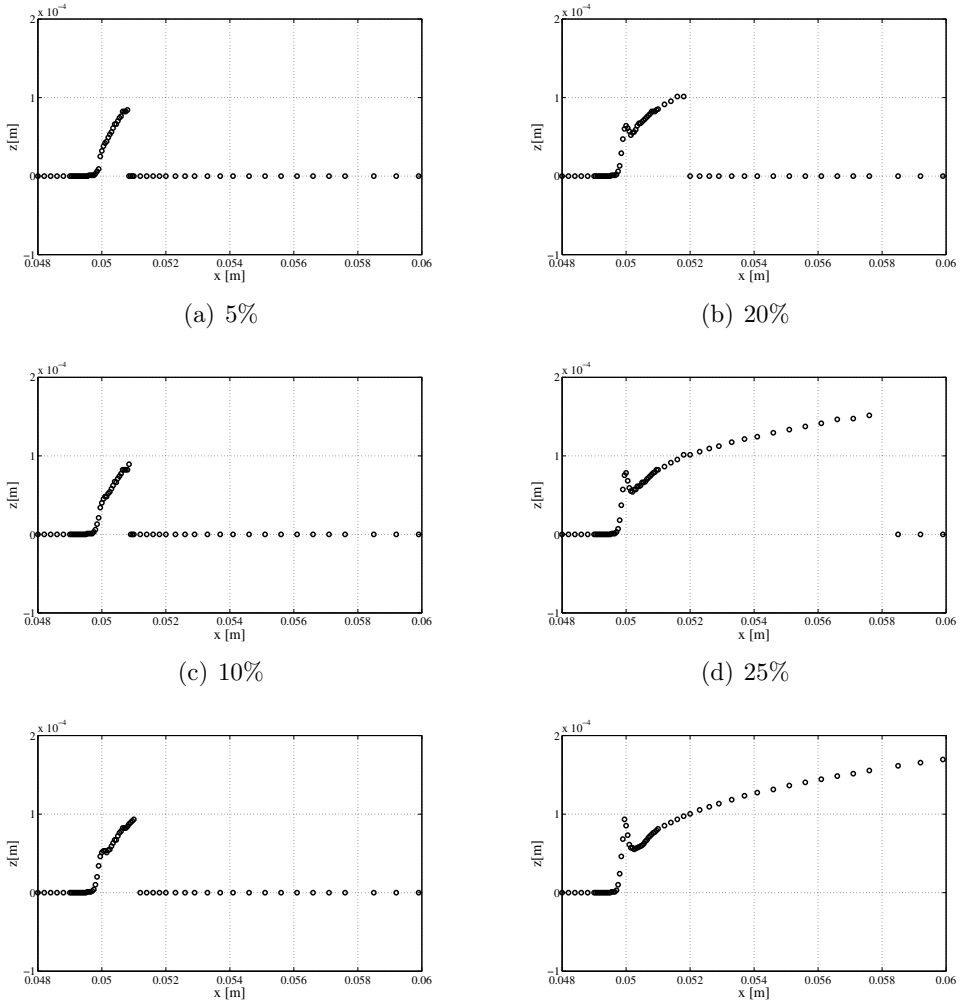
Figure 5.13 shows the maximum of the streamwise vorticity at different locations in the streamwise direction. We can see that the distortion increases as the height of the roughness increases. The maximum value is in proximity of the hump location. After the maximum the streamwise vorticity starts to decrease monotonically for the 5%, 10%, 15%, 20% humps approaching quickly the unperturbed value. As already seen for the pressure and shear stress, the vorticity of the 25% and 30%

humps is different and this is evident in Figure 5.13(b). The streamwise vorticity reaches a minimum inside the separation region, after this mimimum it starts to slightly increase. Then we can observe a second maximum in the separation region and after this the vorticity decreases monotonically returning to the unperturbed value.

Figure 5.14 shows the location of the streamwise vortices in the wall-normal direction z at different values of x. In particular we can see that the generation of streamwise vortices appear evident from the 20% hump and the length of these vortices downstream of the roughness increases with the hump height. Considering the height hump as reference length the streamwise vortices of 25% and 30% humps extend for different lenghts after the hump locations, with the last reaching the end of the domain. We see that the vortices grow downstream of the hump and move away from the wall.

The vertical motion of the vortices can be explained by their rotation visible from the streamlines of Figure 5.15, which shows the streamwise vorticity and the pressure of the 25% hump in a plane perpendicular to the streamwise velocity at $x = x_h + 4h$, being x_h the location of the hump centre and h the hump height. Specifically, observing the in-plane streamlines we can see that the rotation of each vortex induces a positive vertical motion on the other and this results in a central upwash. As seen in Mason and Morton (1987), the sense of rotation of the vortices depends on both the shape of the obstacle and its depth relative to the boundary-layer height. Obstacles that divide the stream laterally produce vortices with a central downwash, whereas those lifting the flow predominantly over their crests, such as the humps considered in this study, produce vortices with a central upwash. In particular the direction of the streamlines can be explained through the pressure distribution. In this plane, inside the separation bubble and close to the centre of the hump the pressure away from the symmetry plane is larger than the pressure in the region close to this. This causes a motion of the flow towards the plane $y = 0$ as we can see in the bottom part of Figure 5.15 which shows the instantaneous streamwise vorticity and pressure.

When the flow approaches the symmetry plane both from the left and right there is a collision of the fluid particles which causes a positive wall-normal velocity. The flow motion in the plane $y = 0$ is towards the top the domain, but, the pressure in the upper part of the domain increases and is close to the unperturbed value. This results in a change in the flow direction visible from the streamlines that start to diverge towards negative and positive values of y. The separation characterised by the collision of the flow in the symmetry plane results in this typical mushroom-shaped plume that we can well appreciate in the streamlines of Figure 5.15. The information in Figure 5.15 are related to the instantaneous values of the flow quantities and the dynamic of the flow needs to be investigated. In particular to characterize the roughness effect on the flow it can be useful to look



(a) 5%

(b) 20%

(c) 10%

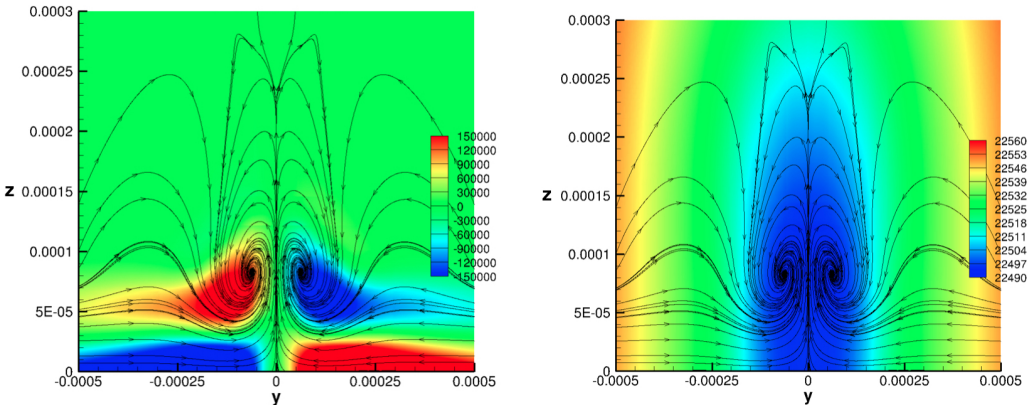
(d) 25%

(e) 15%

(f) 30%

Figure 5.14: Z-location of maximum streamwise vorticity for the different hump heights.

at the frequency of the vortex shedding and the associated Strouhal number. This type of information, together with the study the frequency spectra of the energy and velocity, can enable a further understanding of the vortex mechanism which, in case of larger heights, propagating downstream will lead to flow breakdown.



(a) Streamwise vorticity

(b) Pressure

Figure 5.15: Streamwise vorticity and pressure for the hump 25% at $x = x_{hump} + 4h_r$ after the hump location.

5.5 Conclusions

In this chapter the DNS results of a two-dimensional boundary-layer flow encountering a three-dimensional Gaussian-shaped roughness element are presented. The flow outside of the boundary layers is a high-speed subsonic flow at $Ma = 0.87$. To the knowledge of the author, this regime has remained unexplored so far in direct numerical simulations. The Reynolds number considered based on the distance from the leading edge of the flat plate is $Re = 4 \times 10^5$ and is sufficiently high to be considered in the regime of aeronautical applications.

We first showed a comparison between the results of a three-dimensional hump and of an identical shaped two-dimensional hump in terms of dp/dx and τ_{xz}. From the calculation it is possible to appreciate that the perturbations produced by the two-dimensional humps are bigger than those produced by the equivalent three-dimensional humps, both in terms of peaks and extension. This can be explained by the smaller blockage effect generated by the three-dimensional roughness elements.

We then studied the effect that the Mach number can have on a similar flow comparing the transonic regime of $Ma = 0.87$ with a subsonic of $Ma = 0.5$. The results obtained for dp/dx and τ_{xz} show very similar qualitative distributions of the quantities considered. The separation behind the hump seems to appear at similar hump heights because, even if the peaks of perturbation are larger in the transonic regime, these are compensated by the bigger mean value of the shear stress than in the subsonic regime.

After these numerical comparisons we focussed on the transonic regime $Ma =$

0.87 and we studied different hump heights. Specifically we found, as expected, that the distortion of the boundary layer increases with the increase in roughness height. After a critical hump height we start to see a separation zone downstream of the roughness whose extension becomes larger with larger values of the height. With a further increase of the hump height a separation zone begins to develop upstream of the roughness as well. In addition the calculations show that when the roughness height is sufficiently large, two counter-rotating vortices are found to form in the wake behind the roughness. They move slowly away from the wall and their longitudinal extension increases with the height of the hump and becomes comparable with the dimension of the domain. The pressure perturbations instead decay rather fast and this suggests that the pressure is unlikely to play a decisive role in supporting the growth of the vortices far from the hump.

This work constitutes the first step to close the gap between the DNS results in high speed subsonic regimes and the data present in the literature for similar analyses in supersonic and hypersonic regimes.

Finally, the results presented in this chapter show that the numerical approaches and techniques developed in this thesis work together with the compressible aerodynamic solver represent a potentially robust, stable, reliable, accurate and efficient tool for simulating this class of problems.

Conclusions and future work

In this research the author developed and analysed two types of discontinuous spectral/hp element methods and their application to compressible Navier-Stokes equations.

The first step of this work was focussed on the implementation of FR into the spectral/hp element library Nektar++. In this library an implementation of DG method was already present and this allowed the exploration of the links between DG method and high-order FR schemes. After that the connections between the two approaches have been investigated considering multidimensional systems of conservation laws on tensor product grids. Finding the connections between the DG and FR approaches has been crucial for understanding the advantages we can attain by using one or the other method and has helped to form a better contextualisation of the FR approach within the framework of discontinuous spectral/hp element methods. Specifically, we examined the connections between three nodal versions of tensor product discontinuous Galerkin spectral element approximations and two types of collocated flux reconstruction schemes. The different types of discontinuous Galerkin approximations arise from the choice of the solution nodes of the Lagrange basis representing the solution and from the quadrature approximation used to integrate the mass matrix and the other terms of the discretisation. In chapter 2 we have demonstrated the $\mathrm{DG_{SEM}}$ method with an exact mass matrix ($\mathrm{DG_{SEM}}$-EMM) and the $\mathrm{FR_{DG}}$ scheme are equivalent for both linear and nonlinear equations on quadrilateral grids. In addition it has been proved that, if we use GLL points as solution points, the $\mathrm{DG_{SEM}}$ method with a lumped mass matrix ($\mathrm{DG_{SEM}}$-LMM) is equivalent to the $\mathrm{FR_{g2}}$ scheme.

All the equivalences have been found by considering a collocated version of DG and FR which allows a significant reduction in computational costs but can lead to aliasing-driven instability when solving nonlinear equations due to under-integration errors. These errors can lead eventually to the failure of the simulation and this lack of numerical stability is one aspect that has crucial implications on the applicability of these methods to challenging industrial problems with complex geometries and high Reynolds number flows. When solving nonlinear systems the aliasing errors of FR schemes have been proved to be identical to their DG counterpart and they can be recast in the wider class of aliasing issues affecting

discontinuous spectral/hp element methods. This suggested that, if we treat the nonlinear flux of the FR and DG formulations with an identical strategy, the equivalences obtained can be extended to non collocated versions as well. Therefore the techniques adopted in the more established context of continuous Galerkin (CG) and DG can be also applied to the FR approach providing the same effect in alleviating the aliasing issues. In chapter 4 two strategies for reducing the aliasing errors, both based on the role of consistent integration of the nonlinearities, have been proposed and investigated.

The first technique presented, namely the *Local* dealiasing approach, targets the PDE-aliasing sources which arise from the nonlinearities contained in the flux function and they are therefore related to the physics of the problem. This approach exploits the tensor product structure of the elements to reduce the number of floating point operations, making it computationally efficient at higher polynomial orders. The application of *Local* dealiasing is particularly useful when we have localised nonlinearities (such as in the incompressible Navier-Stokes equations) and geometric aliasing is not significant.

The second technique, namely the *Global* dealiasing approach, involves the use of a richer quadrature order than is typically necessary for linear PDEs for all the terms constituting the right-hand side of the discretised problem and, therefore, it targets both PDE- and geometrical-aliasing sources. The latter arises when deformed or curved elements are used, which is often the case in industrially-relevant problems, where complex geometries are a fairly common feature. In contrast to the first approach, this technique is more computationally intensive (especially from a floating point operation point of view), but can address all the aliasing sources arising in any spectral/hp element method when coupled with a sufficiently large number of quadrature points to tackle the nonlinearities within the problem.

In discontinuous spectral/hp element methods the aliasing issues due to both volumetric and interface terms have been explored and the importance of these two sources of aliasing has been compared. We established that interface dealiasing may play a bigger role as the order of the interface flux function increases. It was also found that a full control (i.e. up to machine precision) of all aliasing effects is not possible, since the boundary terms (through the numerical fluxes at the interfaces between the elements) introduce non-polynomial functions into the discretised problem. Finally, we showed that the two dealiasing strategies can be used in a complementary manner and, when applied to challenging applications, have proven effective and have increased the numerical stability of the simulations.

After having developed and analysed the underlying numerics of the discontinuous spectral/hp element methods, and having described possible techniques to alleviate the aliasing issues that can lead the simulations to instability, we have applied the numerical approaches presented in the thesis to a compressible boundary-

layer flow. Specifically, in chapter 5 we performed a DNS study of a high-speed subsonic boundary-layer flow encountering a three-dimensional Gaussian-shaped roughness element. This class of simulations is of particular interest for the LFC-UK group and was motivated by the lack of DNS data of boundary-layer flows past roughness elements in a similar regime. The high speed subsonic regime is typical of civil aircraft and the Reynolds number and the sizes of the roughness considered are typical of aeronautical applications as well. In particular the high Reynolds number considered required a highly demanding resolution to produce accurate and reliable results. Nevertheless the efficiency of the compressible aerodynamics solver developed permitted to study different roughness heights. Specifically, the efficiency of the $\mathrm{DG_{SEM}}$ method with diagonal mass matrix was extensively exploited when the stability of the simulation permitted it. In case of aliasing-driven instabilities these were addressed by a *Global* dealiasing technique, which enhanced the numerical stability of the DG scheme and prevented the simulations to fail. The results obtained on this flow configuration look promising and indicate that the numerical framework of spectral/hp element methods is attractive for high Reynolds number boundary-layer flow problems and the techniques described during this work can be applied to overcome one the main issues of this class of schemes that is numerical stability.

Ultimately, we have developed an efficient and effective numerical tool for compressible flow simulations that have been thoroughly tested for external aerodynamics studies with particular focus to boundary-layer flows. This work helps to provide a more comprehensive view of the DG and FR approaches. Some of the connections that exist between the schemes have been analysed and exploited attaining critical advantages in terms of time-step restrictions (the use of the $\mathrm{FR_{g2}}$ or the $\mathrm{DG_{SEM}}$-LMM scheme allows the bigger time-step across the methods investigated) and computational costs. The aliasing properties of the two schemes are identical and the strategies described for tackling this issue have proven to be effective also in challenging simulations over complex geometries. This can be extremely beneficial for future developments of this class of methods in order to solve one of the issues that has limited the adoption of high-order techniques in both academia and industry.

Other points have remained unaddressed in this research. These include both a further numerical analysis of the spectral/hp element methods as well as other applications to boundary-layer flows or to different compressible aerodynamic simulations. From the first perspective it could be of interest exploring possible connections between DG and FR in other two- and three-dimensional types of grids. In addition other strategies for alleviating the aliasing issues may be investigated and these include more stable forms of the nonlinear terms, a more optimal distribution of quadrature points, polynomial filtering taking into account both the benefits and the computational cost. Together with its accuracy and robustness

the efficiency of a CFD solver is a crucial aspect especially in industrial applications. Exploring implicit or semi-implicit time-integration schemes for increasing the CFL limit and reducing the simulation time is another study that we plan to do. From the latter perspective the next step would be to do a comparison between the DNS data of the boundary-layer study performed with other approaches, such as the triple-deck theory. The numerical simulation of the roughness element can be also considered as a first step of an analysis of roughness-induced wake instability and transition behaviour in a high-speed subsonic regime, where other geometries of the roughness can also be considered.

One of the consequences of this work is that with the development of the compressible aerodynamic solver and many numerical techniques, Nektar++ has been extended to enable the study of a wide range of practical CFD problems.

Bibliography

ABGRALL, R. 1994 *On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation.* Journal of Computational Physics **114** (1), 45-58.

GRINSTEIN, F.F., MARGOLIN, L.G. & RIDER, W.J. 2011 *Implicit Large Eddy Simulation.* Cambridge.

ALLANEAU, Y. & JAMESON, A. 2011 *Connections between the filtered discontinuous Galerkin method and the flux reconstruction approach to high order discretizations.* Computational Methods in Applied Mechanics and Engineering **200**, 3628-3636.

ARNOLD, D.N. 1982 *An interior penalty finite element method with discontinuous elements.* Journal on Numerical Analysis **19** (4), 742-760.

ARNOLD, D.N., BREZZI, F., COCKBURN, B. & MARINI, L.D. 2001 *Unified analysis of discontinuous Galerkin methods for elliptic problems.* SIAM Journal on Numerical Analysis **39**(5), 1749-1779.

BARTH, T.J. & FREDERICKSON, P.O. 1990 *Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction.* AIAA paper **90-0013**.

BARTH, T.J. & DECONINCK, H. 1999 *High-order methods for computational physics.* **9**, Springer Verlag.
T.J. Barth and H. Deconinck. High-order methods for computational physics, volume 9. Springer Verlag, 1999.

BASSI, F. & REBAY, S. 1997 *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations.* Journal of Computational Physics **131** (2), 267-279.

BERNARDINI, M.,PIROZZOLI, S., & ORLANDI, P. 2012 *Compressibility effects on roughness-induced boundary-layer transition.* International Journal of Heat and Fluid Flow **35**, 45-51.

BERNARDINI, M.,PIROZZOLI, S., ORLANDI, P. & LELE, S,K, 2012 *Compressible boundary-layer transition induced by isolated roughness elements.* Center for Turbulence Research,Proceedings of the Summer Program 2012.

BERNARDINI, M.,PIROZZOLI, S., ORLANDI, P. & LELE, S,K, 2014 *Parameterisation of Boundary-Layer Transition Induced by Isolated Roughness Elements.* AIAA Journal **52**(10), 2261-2269.

BLAISDEKK, G. A., SPYROPOULOS, E.T. & QIN, J. H. 1996 *The effect of the formulation of nonlinear terms on aliasing errors in spectral methods.* Applied Numerical Mathematics, **21** (3), 207-219.

BOCHEV, P.B. & GUNZBURGER, M.D. 1998 *Finite element methods of least-squares type.* SIAM, Review **40** (4), 789-837.

BODONY, D.J. 2006 *Analysis of sponge zones for computational fluid mechanics.* Journal of Computational Physics **212** (4), 681-702.

BOGOPOLEV, V.V. & NEILAND, V.Y. 1971 *Supersonic flow over a surface with local distortions.* TsAGI Trans., **1363**, 1-12.

BOGOPOLEV, V.V. & NEILAND, V.Y. 1985 *Locally three-dimensional boundary flows.* Tekh. Fiz, **1**, 28-36.

BOLEMANN, T., BECK, A., FLAD, D., FRANK, H., MAYR, V. & MUNZ, C.,-D. 2015 *High-order discontinuous galerkin schemes for large-eddy simulations of moderate reynolds number flows.* In IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach, 435 -456. Springer.

BROOKS, A.N. & HUGHES, T.J.R. 1982 *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations.* Computer Methods in Applied Mechanics and Engineering **32** (1), 199-259.

BUTCHER, J.C. 1987 *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods.* Wiley-Interscience.

CANUTO, C. & QUARTERONI, A. 1982 *Approximation results for orthogonal polynomials in sobolev spaces* Mathematics of Computation **38** (157), 67-86.

CANUTO, C., HUSSAINI, M.Y., QUARTERONI, A. & ZANG, T. 1987 *Spectral methods in fluid dynamics.* Springer, New York.

CANTWELL, C.D., MOXEY, D., COMERFORD, A., BOLIS, A., ROCCO, G., MENGALDO,G., DE GRAZIA, D., YAKOLEV, S., LOMBARD, J.-E., EKELSHOT, D. et al. 2015 *Nektar++: An open-source spectral/hp element framework.* Computer Physics Communications **192**, 205-219

CASTONGUAY, P., VINCENT, P.E. & JAMESON, A. 2011 *Application of high-order energy stable flux reconstruction schemes to the Euler equations.* AIAA Aerospace Sciences Meeting, Orlando FL, 4-7 January.

CASTONGUAY, P., VINCENT, P.E. & JAMESON, A. 2011 *A new class of high-order energy stable flux reconstruction schemes for triangular elements.* Journal of Scientific Computing.

CHANG, C.-L., VENKATACHARI, B. & CHENG, C. 2013 *Time-Accurate Local Time Stepping and High-Order Space-Time CESE Methods for Multi-Dimensional Flows with Unstructured Meshes.* AIAA paper.

CHEN, M.-H., COCKBURN, B. & REITICH, F. 2013 *High-order RKDG Methods for Computational Electromagnetics.* Journal of Scientific Computing, **22**(1), 205-226

CHOUDARI, M., LI, F., WU, M., CHANG, C.L., EDWARDS, J., KEGERISE, M. & KING, R. 2010 *Laminar-Turbulent Transition behind Discrete Roughness Elements in a High-Speed Boundary Layer.* AIAA paper **2010-1575**.

CHOUDARI, M., NORRIS A., LI, F., CHANG, C.L. & EDWARDS, J. 2013 *Wake Instabilities behind Discrete Roughness Elements in High Speed Boundary Layers.* 51st AIAA Aerospace Sciences Meeting, Reston, Virginia, 1-17,

CHOW, J.S., ZILLIAC, G.C. & Bradshaw, P.B. 1997 *Mean and turbulence measurements in the near field of a wingtip vortex.* AIAA Journal **35**, 1561-1567.

COCKBURN, B., LIN, S.-Y., & SHU, C.-W. 1989 *Tvb Runge-Kutta local projection discontinuous galerkin finite element method for conservation laws iii: one-dimensional systems.* Journal of Computational Physics **84** (1), 90-113.

COCKBURN, B. & SHU, C.-W. 1990 *Tvb Runge-Kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework.* Mathematics of Computation **52** (186), 41-435.

COCKBURN, B., SHU, C.-W. & LIN, S.-Y. 1989 *The Runge-Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws.* Institute for Mathematics and its Applications (USA).

COCKBURN, B., HOU, S., & SHU, C.-W. 1990 *The Runge-Kutta local projection discontinuous galerkin finite element method for conservation laws. iv. the multidimensional case.* Mathematics of Computation **54** (190), 545-581.

COCKBURN, B. & SHU, C.-W. 1998 *The local discontinuous galerkin method for time-dependent convection-diffusion systems.* SIAM, Journal on Numerical Analysis **35** (6), 2440-2463.

COCKBURN, B. & SHU, C.-W. 1998 *The Runge-Kutta discontinuous galerkin method for conservation laws v: multidimensional systems.* Journal of Computational Physics **141** (2), 199-224.

COCKBURN, B., KARNIADAKIS G.E., & Shu, C.-W. 2000 *The development of discontinuous Galerkin methods..* Springer.

COCKBURN, B. & Shu, C.-W. 2001 *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems.* Journal of Scientific Computing **16**.

COCKBURN, B., GOPALAKRISHNAN, J., & LAZAROV, R. 2009 *Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems* SIAM J. Numer. Anal., **47**(2) 319-1365.

COLONIUS, T. 2004 *Modelling artificial boundary conditions for compressible flow.* Annual Review of Fluid Mechanics **36**, 315-345.

COURANT, R., FRIEDRICHS, K. & Lewy, H. 1967 *On the partial difference equations of mathematical physics.* IBM, Journal of Research and Development **11** (2), 215-234.

DAVIS, S.F. 1988 *implified SecondÐOrder GodunovÐType Methods.* SIAM J. Sci. Stat. Comput., **9**, 445-473.

DE GRAZIA, D. 2013 *Flux reconstruction on unstructured hexahedral grids for compressible boundary-layers transition.* Internal Report, Imperial College London.

DE GRAZIA, D., MENGALDO, G., MOXEY, D., VINCENT, P.E. & SHERWIN, S.J 2014 *Connections between the discontinuous Galerkin method and high-order flux reconstruction schemes.* International Journal for Numerical Methods in Fluids **75** (12), 860-877.

DELANAYE, M. & LIU, Y. 1999 *Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids.* AIAA Paper 1999-3259.

DE TULLIO, N., PAREDES, P., SANDHAM, N.D. & THEOFILIS, V. 2013 *Laminar-turbulent transition induced by a discreet roughness element in a supersonic boundary layer.* Journal of Fluid Mechanics **735**, 613-646.

DE TULLIO, N. & SANDHAM, N.D. 2012 *Direct numerical simulations of roughness receptivity and transitional shock-wave/boundary-layer interactions.* Technical report, NATO RTO-MP-AVT-200.

DE TULLIO, N. & SANDHAM, N.D. 2015 *Influence of boundary-layer disturbances on the instability of a roughness wake in a high-speed boundary layer.* Journal of Fluid Mechanics, **763**, 136-165.

DEVILLE, M.O., MUND, E.H. & FISCHER, P.F. 2002 *High Order Methods for Incompressible Fluid Flow.* Cambridge University Press.

EINFELDT, B. 1988 *On Godunov-Type Methods for Gas Dynamics.* SIAM J. Numer. Anal., **25**(2), 294-318.

FARHAT, C., RAJASEKHARAN, A. & KOOBUS, B. 2006 *A dynamic variational multiscale method for large eddy simulations on unstructured meshes.* Computational Methods for Applied Mechanics and Engineering **195**, 1667-1691.

FINLAYSON, B.A. 1972 *The method of weighted residuals and variational principles.* Academic Press, New York.

FISCHER, P. & MULLEN, J. 2001 *Filter-based stabilization of spectral element methods.* Comptes Rendus de l'Academie des Sciences - Series I - MathematicsVol. **332** (3), 265-270

FISCHER, P.F., KRUSE, G.W. & LOTH, F. 2002 *Spectral element methods for transitional flows.* Journal of Scientific Computing **17**, 81-98.

FLETCHER, C. A. J. 1984 *Computational Galerkin methods.* Springer, Berlin.

GASSNER, G.J. 2013 *A kinetic energy preserving nodal discontinuous Galerkin spectral element method* International Journal for Numerical Methods in Fluids **27** (3-4), 221–237.

GASSNER, G. J. & BECK, A.D. 2013 *On the accuracy of high-order discretizations for underresolved turbulence simulations.* Theoretical and Computational Fluid Dynamics **76**, 28-50.

GASSNER, G.J. 2013 *A Skew-Symmetric Discontinuous Galerkin Spectral Element Discretization and its Relation to SBP-SAT Finite Difference Methods* Journal of Scientific Computing **35** (3), 233–1253.

GASSNER, G.J. & KOPRIVA, D.A. 2010 *On the Quadrature and Weak Form Choices in Collocation Type Discontinuous Galerkin Spectral Element Methods.* Journal of Scientific Computing **44** (2).

GASSNER, G.J. & KOPRIVA, D.A. 2011 *A Comparison of the Dispersion and Dissipation Errors of Gauss and GaussÐLobatto Discontinuous Galerkin Spectral Element Methods.* SIAM, JOURNAL OF SCIENTIFIC COMPUTING **33** (5), 2560-2579.

GILES, M.B. 1990 *Nonreflecting Boundary Conditions for Euler Equation Calculations.* AIAA Journal **28** (12).

GODUNOV, S.K. 1959 *A Finite Difference Method for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics.*, Mat. Sb., **47**, 357-393.

GODUNOV, S.K. 1976 *Numerical Solution of Multi-Dimensional Problems in Gas Dynamics.* Nauka Press, Moscow.

GOLDSTEIN, M E., SESCU, A., DUCK, P. W. & CHOUDARI, M. 2010 *The long range persistence of wakes behind arrow of roughness elements.* Journal of Fluid Mechanics **644**, 123-163, .

GOLDSTEIN, M E., SESCU, A., DUCK, P. W. & CHOUDARI, M. 2011 *Algebraic/transcendental disturbance growth behind a row of roughness elements.* Journal of Fluid Mechanics **688**, 236-266.

GOTTLIEB, J.J. & GROTH, C.P.T. 1988 *Assessment of Riemann Solvers for Unsteady OneÐDimensional Inviscid Flows of Perfect Gases.* J. Comput. Phys., **78**, 437-458.

GOTTLIEB, D. & HESTHAVEN, J.S. 2001 *Spectral methods for hyperbolic problems.* Journal of Computational and Applied Mathematics **128** (1-2), 83-131.

GOTTLIEB, D. & ORSZAG, S. A. 1977 *Numerical Analysis of Spectral Methods: Theory and Applications.* SIAM-CMBS.

GROSKOPF, G. KLOKER, M.J. & MARXEN, O. 2008 *Numerical Analysis of Spectral Methods: Theory and Applications.* SIAM-CMBS.

HARTEN, A., ENGQUIST, B., OSHER, S. & CHAKRAVARTHY, S.R. 1987 *Uniformly high order accurate essentially non-oscillatory schemes, III.* Journal of Computational Physics, **71** (2), 231-303.

HARTEN, A., LAX, P.D., LEER, B. V. & CHAKRAVARTHY, S.R. 1983 *On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws.* SIAM Review, **25**(1), 35Ð61.

HEINRICH, J.C. & ZIENKIEWICZ, O.C. 1977 *Quadratic finite element schemes for two-dimensional convective-transport problems.* International Journal for Numerical Methods in Engineering **11** (12), 1831-1844.

HESTHAVEN, J.S. & KIRBY, R.M. 2008 *Filtering in Legendre spectral methods.* Mathemathics of Computation **77**, 1425-1452.

HESTHAVEN, J.S. & WARTBURTON, T. 2008 *Nodal Discontinuous Galerkin Methods—Algorithms, Analysis, and Applications.* Springer, Berlin.

HU, C. & SHU, C.-W. 1999 *Weighted essentially non-oscillatory schemes on triangular meshes.* Journal of Computational Physics **150** (1), 97-127.

HUGHES, T.J.R. & BROOKS, A.N. 1979 *A multidimensional upwind scheme with no crosswind diffusion.* In T. J. R. Hughes, editor, Finite element methods for convection dominated flows, ASME, New York, 19-35.

HUGHES, T.J.R. & BROOKS, A.N. 1982 *A new finite element formulation for computational fluid dynamics: V. circumventing the Babuska-Brezzi condtion: A stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations.* Computer Methods in Applied Mechanics and Engineering **59** (1), 85-99.

HUGHES, T.J.R., FRANCA, L.P. & BALESTRA, M. 1986 *A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions: Application to the streamline upwind procedure.* In R. H. Gallagher, D. H. Norrie, J. T. Oden, and O. C. Zienkiewicz, editors, Finite elements in fluids, volume IV, Wiley, London, 46Ð65.

HUGHES, T.J.R., FEIJO, G.R., MAZZEI, L. & QUINCY, J. 1998 *The variational multiscale method – a paradigm for computational mechanics.* Computer methods in applied mechanics and engineering,**166** (1), 3-24.

HUGHES, T.J.R., OBERAI, A.A. & MAZZEI, L. 2001 *Large eddy simulation of turbulent channel flows by the variational multiscale method.* Physics of Fluids **13**, 1784-1799.

HUYNH, H.T. 2007 *A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods.* 18$^{\text{th}}$ AIAA CFD Conference, 25-28 June, Miami, FL.

ISRAELI, M. & OORSZAG, S.A. 1981 *Approximation of radiation boundary conditions.* Journal of Computational Physics **41** (1), 115-135.

JAMESON, A., VINCENT, P.E. & CASTONGUAY, P. 2012 *On the nonlinear stability of the flux reconstruction schemes.* (2nd edn). Journal of Scientific Computing **50**, 434-445.

JIANG, B.-N. 1998 *The least-squares finite element method: theory and applications in computational fluid dynamics and electromagnetics.* Springer Science & Business Media.

KARAMOS, G-S. & KARNIADAKIS, G.E. 2000 *A Spectral Vanishing Viscosity Method for Large-Eddy Simulations.* Journal of Computational Physics **163** (1), 22-50.

KARNIADAKIS, G.E., ISRAELI, M. & Orszag, S.A. 1991 *High-order splitting methods for the incompressible Navier-Stokes equations* Journal of Computational Physics **97**, 414-443.

KARNIADAKIS, G.E. & SHERWIN, S.J. 2005 *Spectral/hp element methods for computational fluid dynamics.* Oxford University Press.

KIM, J.W. & LEE, D.J. 1996 *Optimized compact finite difference schemes with maximum resolution* AIAA journal, **34**(5):887-893.

KIRBY, R.M. & KARNIADAKIS, G.M. 2002 *Coarse Resolution Turbulence Simulations With Spectral Vanishing Viscosity—Large-Eddy Simulations (SVV-LES).* Journal of Fluids Engineering **124** (4), 886-891.

KIRBY, R.M. & KARNIADAKIS, G.M. 2003 *De-aliasing on non-uniform grids: algorithms and applications.* Journal of Computational Physics **191**, 249-263.

KIRBY, R.M. & SHERWIN, S.J. 2006 *Stabilisation of spectral h/p element methods through spectral vanishing viscosity: Application to fluid mechanics modelling.* Computational Methods for Applied Mechanics in Engineering **195**, 1667-1691.

KIRBY, R.M. & SHERWIN, S.J. 2006 *Aliasing errors due to quadratic nonlinearities on triangular spectral/hp element discretisations. Journal of engineering mathematics.* Journal of engineering mathematics **56** (3), 273-288.

KLEBANOFF, P.S. & TIDSTROM, K.D. 1972 *Mechanism by which a two-dimensional roughness element induces boundary-layer transition.* Physics of Fluids (1958-1978) **15**(7), 1173-1188.

KOPRIVA, D.A. 2006 *Metric Identities and the Discontinuous Galerkin Spectral Element Method on Curvilinear Meshes* Journal of Scientific **26** (3), 301-327.

KOPRIVA, D.A. 2009 *Implementing spectral methods for partial differential equations – Algorithms for scientists and engineers.* Springer.

KOPRIVA, D.A. & KOLIAS, J.H. 1996 *A conservative staggered-grid Chebyshev multidomain method for compressible flows.* Journal of Computational Physics **125**.

KOPRIVA, D.A., WOODRUFF S.L. & HUSSAINI, M.Y. 2000 *Discontinuous spectral element approximation of Maxwells equations.* Discontinuous Galerkin Methods, Springer.

KOROLEV, G.L., GAJJAR, J.S.B. & RUBAN, A.I. 2001 *Once again on the supersonic flow separation near a corner.* Journal of Fluid Mechanics, **463** , 173-199.

KOROLEV, G.L., 2007 *Numerical method for solving three-dimensional viscous-inviscid interaction equations.* J. Vych. Mat. Mat. Fiz, **47** (3), 506-529.

KRAVCHENKO, A.G. & MOIN, P., 2000 *Numerical studies of flow over a circular cylinder at* $Re_d = 3900$. Physics of Fluids **12** 403-417.

LANDMANN, B., KESSLER, M., WAGNER, S. & KRAMER, E. 2008 *A parallel, high-order discontinuous galerkin code for laminar and turbulent flows.* Computers & Fluids **37**(4), 427-438.

LEHMKUHL, O., RODRIGUEZ, I., BORRELL, O. & OLIVA, A. 2013 *Low-frequency unsteadiness in the vortex formation region of a circular cylinder.* Physics of Fluids **25**(8), 3165-8.

LELE, S. K. 1989 *Direct numerical simulation of compressible free shear flows.* In Annual Research Briefs, **1**, 79-98.

LELE, S. K. 1992 *Compact finite diference schemes with spectral-like resolution.* Journal of Computational Physics **103**, 16-42.

LEVEQUE, R.J. 1996 *High-Resolution Conservative Algorithms for Advection in Incompressible Flow.* SIAM Journal on Numerical Analysis **33** (2), 627-665.

LEVEQUE, R.J. 2002 *Finite volume methods for hyperbolic problems.* **31**, Cambridge University Press.

LEVEQUE, R.J. 2007 *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems.* SIAM **98**.

LIANG, C., COX, C. & PLESNIAK, M. 2013 *A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction.* Journal of Computational Physics **239**, 138-146.

LIANG, C., MIYAJI, K., ZHANG, B. 2013 *An Efficient Correction Procedure via Reconstruction for Simulation of Viscous Flow on Moving and Deforming Domains.* AIAA **2013-0998**.

LIU, X.-D., OSHER, S. & CHAN, T. 1994 *Weighted essentially non-oscillatory schemes. Journal of Computational Physics.* Journal of Computational Physics **115** (1), 200-212.

LIU, Y., VINOKUR, M. & WANG, Z.J. 2006 *Spectral difference method for unstructured grids I: basic formulation.* Journal of Computational Physics, **216**.

MA, X., KARAMOS, G.-S. & KARNIADAKIS, G. E. 2000 *Dynamics and low dimensionality of a turbulent near wake.* Journal of Fluid Mechanics **410**, 29-65.

MADAY, Y. & RØNQUIST, F. 1990 *Optimal error analysis of spectral methods with emphasis on non-constant coefficients and deformed geometries.* Computer methods in applied mechanics and engineering **80** (1-3), 80-115.

MADAY, Y. & TADMOR, E. 1989 *Analysis of the Spectral Vanishing Viscosity Method for Periodic Conservation Laws.* SIAM J. SCI. COMPUT., **26** (4), 854-870.

MALM, J., SCHLATTER P., FISCHER P.F. & HENNINGSON, D.S. 2013 *Stabilization of the Spectral Element Method in Convection Dominated Flows by Recovery of Skew-Symmetry.* Journal of Scientific Computing **77**, 254-277.

MANI, A. 2012 *Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment.* Journal of Computational Physics **231** (2), 704-716.

MARRAS S., KELLY, J.F., GIRLADO, F.X. & VAZQUEZ, M. 2012 *Variational multiscale stabilization of high-order spectral elements for the advection-diffusion equation.* Journal of Computational Physics **231** (21), 7187-7213.

MARXEN, O. & IACCARINO, G. 2008 *Numerical simulation of the effect of a roughness element on high-speed boundary-layer instability.* 38th Fluid Dynamics Conference, AIAA paper **2008-4400** , 1-17.

MASON, P.J. & MORTON, B.R. 1987 *Trailing vortices in the wakes of surface-mounted obstacles.* Journal of Fluid Mechanics **175** , 247-293.

MENGALDO, G. 2012 *Spectral h/p flux reconstruction approach for compressible boundary-layer transition.* Internal Report, Imperial College London.

MENGALDO, G., DE GRAZIA, D., PEIRO, J., FARRINGTON, A., WITHERDEN, F., VINCENT, P.E. & SHERWIN, S.J. 2014 *A guide to the implementation of boundary conditions in compact high-order methods for compressible aerodynamics.* 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation, Atlanta, GA.

MENGALDO, G., DE GRAZIA, D., MOXEY, D., VINCENT, P.E. & SHERWIN, S.J 2015 *Dealiasing techniques for high-order spectral element methods on regular and irregular grids.* Journal of Computational Physics **299**, 56-81.

MENGALDO, G., KRAVTSOVA, M., RUBAN A.I. & SHERWIN, S.J. 2015 *Triple-deck and direct numerical simulation analyses of high-speed subsonic flow over a roughness element.* Journal of Fluid Mechanics **774**, 311-323.

MENGALDO, G., DE GRAZIA, D., VINCENT, P.E. & SHERWIN, S.J 2015 *On the Connections Between Discontinuous Galerkin and Flux Reconstruction Schemes: Extension to Curvilinear Meshes.* Technical Note, Journal of Scientific Computing, 1-21.

MESSITER, A. F. 1970 *Boundary-layer flow near the trailing edge of a flat plate.* SIAM Journal, Applied Mathematics **18**, 241-257.

MOIN, P. & MAHESH, K. 1998 *Direct numerical simulation: a tool in turbulence research.* Annual review of fluid mechanics **30** (1), 539-578.

MOXEY, D., HAZAN, M., SHERWIN, S.J. & PEIRO', J. 2015 *An isoparametric approach to high-order curvilinear boundary-layer meshing.* Computer Methods In Applied Mechanics and Engineering **283**, 636-650.

NAGARAJAN, S., LELE S.K. & FERZIGER, J.H. 2003 *A robust high-order compact method for large eddy simulation.* Journal of Computational Physics, **191**(2), 392-419. S. Nagarajan, S.K. Lele, and J.H. Ferziger.

NEILAND, V. Y. 1969 *Theory of laminar boundary-layer separation in supersonic flow.* Mekh. Zhid. Gaza **4**, 53-57.

NGUYEN, N.C., PERAIRE, J. & COCKBURN, C. 2009 *An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations.* J. Comput. Phys., **228**(9), 3232-3254.

NGUYEN, N.C., PERAIRE, J. & COCKBURN, C. 2009 *An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations.* J. Comput. Phys., **228**(23), 8841-8855..

NGUYEN, N.C., PERAIRE, J. & COCKBURN, C. 2010 *An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations.* J. Comput. Phys., **230**(4), 53-57, 1147-1170

ODEN, J.T. 1994 *Optimal h-p finite element methods.* Computer methods in applied mechanics and engineering **112** (1), 309-331.

OLLIVIER-GOOCH, J.T. 1997 *Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction.* Journal of Computational Physics, **133**(1), 6-17.

ORSZAG, S.A. 1971 *On the elimination of aliasing in finite-difference schemes by fltering high-wavenumber components.* Journal of the Atmospheric Sciences **28** (6), 1074-1074.

ORSZAG, S.A. 1972 *A Comparison of pseudospectral and spectral approximations.* Studies in Applied Mathematics **51**, 253-259.

ORSZAG, S.A. & ISRAELI, M. 1974 *Numerical simulation of viscous incompressible flows.* Annual Review of Fluid Mechanics **6**, 281-318.

PARNAUDEAU, P., CARLIER, J., HEITZ, D. & LAMBALLAIS, E. 2008 *Experimental and numerical studies of the flow overa circular cylinder at reynolds number 3900.* Physics of Fluids **20**(8), 085101Ð085101-14.

PASQUETTI, R. 2006 *Spectral Vanishing Viscosity Method for Large-Eddy Simulation of Turbulent Flows.* Journal of Scientific Computing **27** (1-3), 365-375.

PATERA, A. 1984 *A spectral element method for fluid dynamics: Laminar flow in a channel expansion.* Journal of Computational Physics **54** (3), 468-488.

PEANO A. 1976 *Hierarchies of conforming finite elements for plane elasticity and plate bending.* Computers & Mathematics with Application **2** (3), 211-224.

PERAIRE J. & PERSONN, P.-O. 2008 *The compact discontinuous Galerkin (CDG) method for elliptic problems..* SIAM, Journal on Scientific Computing **30** (4), 1806-1824.

PERSONN, P.-O. 2013 *A Sparse and High-Order Accurate Line-Based Discontinuous Galerkin Method for Unstructured Meshes.* Journal of Computational Physics, **233**, 414-429.

PROOT, M.M.J. & GERRTISMA, M.I. 2002 *Least-squares spectral elements applied to the stokes problem.* Journal of Computational Physics **181** (2), 445-457.

REDFORD, J.A., SANDHAM, N.D. & ROBERTS, G.T. 2010 *Boundary-Layer Transition Induced by an Isolated Roughness Element.* AIAA journal **48**(12), 2010-2830.

REED, H.L., SARIC, W.S. & ARNALD, D. 1996 *Linear Stability Theory Applied to Boundary Layers.* Annual Review of Fluid Mechanics **28**, 389-428.

REED, W.H. & HILL, T.R. 1973 *Triangular mesh methods for the neutron transport equation.* Technical Report LA-UR-73-479, Los Alamos National Laboratory, Los Alamos, New Mexico, USA.

RIZZETTA, D.P., BURGGRAF, O.F. & JENSON, R. 1978 *Triple-deck solutions for viscous supersonic and hypersonic flow past corners.* Journal of Fluid Mechanics **89**(3), 535-552.

RIZZETTA, D.P. & VISBAL, M.R. 2007 *Direct numerical simulations of flow past an array of distributed roughness elements.* AIAA journal **45**(8), 1967-1976.

RIZZETTA, D.P. & VISBAL, M.R. 1976 *Numerical simulation of excrescence generated transition.* AIAA journal **52**(2), 385-397.

RIZZETTA, D.P., VISBAL, M.R., REED, H.L. & SARIC, W.S. 2010 *Direct numerical simulation of discrete roughness on a swept-wing leading edge.* AIAA journal **48**(11), 2660-2673.

ROE, P.L. 1981 *Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes.* J. Comput. Phys., **43**, 357-372.

RUBAN, A. I. &. KRAVTSOVA, M.A. 2013 *Generation of steady longitudinal vortices in hypersonic boundary layer.* Journal of Fluid Mechanics **729**, 702-731.

SCHLICHTING, H., 1979 *Boundary-layer theory. In Boundary-layer theory.* McGraw-Hill.

SHERWIN, S.J & KARNIADAKIS, G.E. 1995 *A new triangular and tetrahedral basis for high-order (hp) finite element methods.* Int. J. Numer. Meth. Eng., **38**(22), 3775-3802.

SHERWIN, S.J, WARTBURTON, T.C.E. & KARNIADAKIS, G.E. 1998 *Spectral/hp methods for elliptic problems on hybrid grids.* Contemp. Math., **218**, 191Ð216.

SHU, C.-W, 1998 *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations.* Ed. A. Quarteroni, Lecture Notes in Mathematics, **1697**, (Springer:Berlin).

SHU, C.-W & OSHER, S. 1988 *Efficient implementation of essentially non-oscillatory shock capturing schemes.* Journal of Computational Physics, **77**(2), 439-471.

SHU, C.-W & OSHER, S. 1989 *Efficient implementation of essentially non-oscillatory shock capturing schemes, II..* Journal of Computational Physics, **83**(1), 32-78.

SMITH, F.T. 1973 *Laminar flow over a small hump on a flat plate.* Journal of Fluid Mechanics **57**(4), 803-824.

SMITH, F.T. 1976 *Pipeflows distorted by non-symmetric indentation or branching.* Mathematika **23**, 62-83.

SMITH, F.T. 1980 *A three-dimensional boundary-layer separation.* Journal of Fluid Mechanics **99**, 165-186.

SMITH, F.T., SYKES, R. I. & Brighton P. W. M. 1977 *A two-dimensional boundary layer encountering a three-dimensional hump.* Journal of Fluid Mechanics **83** (1), 163-176.

STEWARTSON, K. 1969 *On the flow near the trailing edge of a flat plate ii.* Mathematika **16**, 106-121.

STEWARTSON, K., & WILLIAMS, P.G. 1969 *Self-induced separation.* Proc. R. Soc. Lond. A **312**, 181-206.

SWEBY, P.K. 1984 *High resolution schemes using flux limiters for hyperbolic conservation laws..* SIAM J. Num. Anal. **21**, 995Ð1011.

SYCHEV, V.V., RUBAN, A.I. & KOROLEV, G.L., 1998 *Asymptotic theory of separated flows.* Cambridge University Press.

SYKES, R. I. 1978 *Stratification effects in boundary-layer flows over hills.* Proc. R. Soc. A **361**, 225-243.

SYKES, R. I. 1980 *On three-dimensional boundary-layer flows over surface irregularities.* Proc. R. Soc. A **373**, 311-329.

SZABO, B.A. & BABUSKA, I. 1991 *Finite element analysis.* John Wiley & Sons.

THOMPSON, W. 1987 *Time Dependent Boundary Conditions for Hyperbolic Systems*, Journal of Computational Physics **68** (1)

THOMPSON, W. 1990 *Time-Dependent Boundary Conditions for Hyperbolic Systems, II*, Journal of Computational Physics **89** (1)

THORWALD, H., 1997 *PArabolized stability equations.* Annual Review of Fluid Mechanics **29**, 245-283.

TORO, E.F. 1989 *A Fast Riemann Solver with Constant Covolume Applied to the Random Choice Method.* Phil. Trans. Roy. Soc. London Series A: Physical Sciences and Engineering, Int. J. Numer. Meth. Fluids, **9**, 1145Ð1164.

TORO, E.F. 1992 *The Weighted Average Flux Method Applied to the Euler Equations.* Phil. Trans. Roy. Soc. London Series A: Physical Sciences and Engineering, **A341**, 499-530.

TORO, E.F. 2009 *Riemann solvers and numerical methods for fluid dynamics.* Third edition, Springer.

TORO, E.F., SPRUCE, M. & SPEARES, W. 1992 *Restoration of the Contact Surface in the HLLÐRiemann Solver.* Technical Report CoAÐ9204, Department of Aerospace Science, Collegue of Aeronautics, Cranfield Institute of Technology, UK.

TORO, E.F., SPRUCE, M. & SPEARES, W. 1994 *Restoration of the Contact Surface in the HLLÐRiemann Solver*. Shock Waves, **4**, 25-34.

URANGA, A., PERSONN, P,-O.. DRELA, M. & PERAIRE, J. 2011 *Immplicit large eddy simulation of transition to turbulence at low reynolds numbers using a discontinuous galerkin method*. nternational Journal for Numerical Methods in Engineering **87** (1-5), 232-261

VAN DEN EYNDE, J. 2015 *Stability and Transition of the Flow behind Isolated Roughness Elements in Hypersonic Boundary Layers.* PhD thesis, University of Southampton.

VAN LEER, B. 1974 *Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme.* J. Comput. Phys. **14**, 361-370.

VISBAL, M.R. & GAITONDE, D.V. 2002 *TOn the use of higher-order finite-difference schemes on curvilinear and deforming meshes* Journal of Computational Physics, **181**(1), 155-185.

VINCENT, P.E., CASTONGUAY, P. & JAMESON, A. 2011 *A new class of high-order energy stable flux reconstruction schemes.* Journal of Scientific Computing, **47**, 50-72.

VINCENT, P.E., CASTONGUAY, P. & JAMESON, A. 2011 *Insights from Von Neumann analysis of high-order flux reconstruction schemes.* Journal of Computational Physics **230** (22), 8134 -8154.

WARBURTON, T. 2013 *A Low Storage Curvilinear Discontinuous Galerkin Method for Wave Problems.* SIAM Journal on Scientific Computing, **35** (4), 1987-2012.

WASBERG, C.E., GJESDAL, T., REIF, P. B. A. & ANDREESSEN, O. 2009 *Variational multiscale turbulence modelling in a high order spectral element method.* Journal of Computational Physics, **228** (19), 7333-7356.

WITHERDEN, F.D., VERMEIRE, B.C. & Vincent, P.E. 2015 *Heterogeneous computing on mixed unstructured grids with PyFR.* Computers and Fluids **120** 173-189.

WILLIAMS, D.M., CASTONGUAY, P., VINCENT, P.E. & JAMESON, A. 2011 *An extension of energy stable flux reconstruction to instead, nonlinear, viscous problem on mixed grids.* AIAA CFD Conference, Honolulu Hawaii, 27-30 June.

XU, C. 2006 *Stabilization Methods for Spectral Element Computations of Incompressible Flows.* Journal of Scientific Computing **27** (1-3), 495-505.

YU, M. & WANG, Z.J. 2013 *On the accuracy and efficiency of several discontinuous high-order formulations.* AIAA **2013-855**.

Appendix A

Dealiasing within triangular and tetrahedral regions

Throughout this paper, most of the applications demonstrate how the local and global dealiasing techniques can be applied in the setting of quadrilateral and hexahedral elements. In this appendix, we show that the same arguments are also applicable in the case of tensor product triangular elements. An extension of this argument to prismatic and tetrahedral elements follows easily.

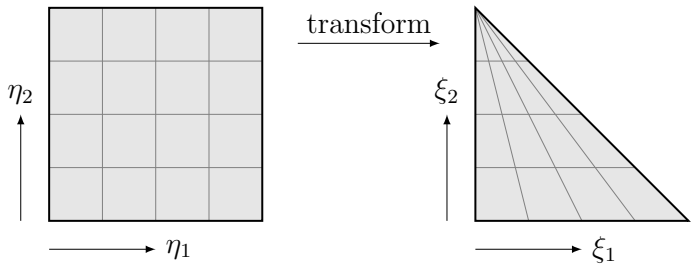


Figure A.1: Illustrative diagram describing Duffy transformation between collapsed coordinates $(\eta_1, \eta_2) \in [-1, 1]^2$ and Cartesian coordinates $(\xi_1, \xi_2) \in \Omega_{\text{st}}$.

We consider a tensor product of two one-dimensional hierarchical *hp* expansion basis functions ψ^a and ψ^b, so that in the standard triangle $\Omega_{\text{st}} = \{(\xi_1, \xi_2) \mid \xi_1 \in [-1, 1], \xi_1 + \xi_2 \leq 0\}$, an approximate solution u^δ may be written as an expansion

$$u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{P} \sum_{q=0}^{P-p} \hat{u}_{pq} \psi_p^a(\eta_1) \psi_q^b(\eta_2)$$

where the collapsed coordinates $(\eta_1, \eta_2) \in [-1, 1]^2$ are given by the Duffy transformation

$$\eta_1 = 2\frac{1+\xi_1}{1-\xi_2} - 1, \qquad \eta_2 = \xi_2.$$

The use of this collapsed coordinate system within the standard quadrilateral region $[-1, 1]^2$ leads to a tensor product of quadrature points within the standard triangular region, as shown in Figure A.1.

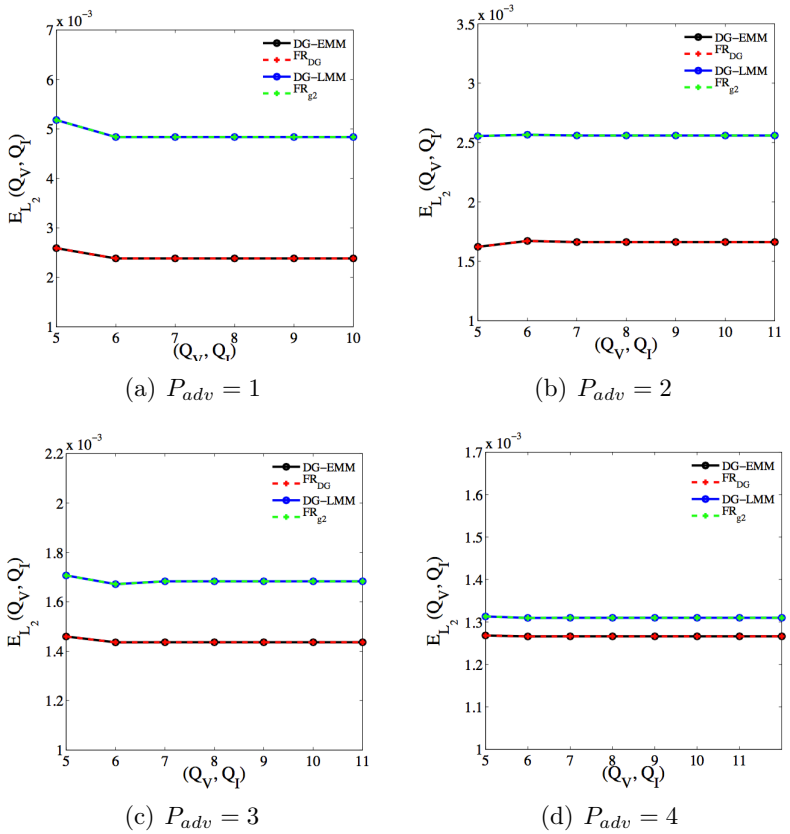
Although this transformation is singular at the top vertex of the triangle, the use of Gauss-Radau quadrature in the η_2 direction, in which the top vertex is excluded, leads to a formulation without geometric singularities. Additionally, Gauss-Radau points with a choice of $\alpha = 1$ and $\beta = 0$ naturally incorporate the Jacobian term which appears in integrals over the standard triangular region when weighted by a constant factor of $\frac{1}{2}$ (Karniadakis and Sherwin (2005)).

In this setting, the sum-factorised dealiasing methods described in section 4.3 can be utilised for triangular elements by applying the stated tensor product logic to the grid of (η_1, η_2) quadrature points, without applying the Duffy transformation to obtain the desired dealiasing effect in the simplex region. Mathematically this approach projects the nonlinear terms down to the tensor product space spanned by the (η_1, η_2) space which is typically richer than the space spanned by the triangular expansion in the (ξ_1, ξ_2) space. However since the inner product using the collapsed coordinates spans the (η_1, η_2) space, this is sufficient to ensure the non-linear product is correctly integrated. This logic extends to three dimensions for both prismatic elements and tetrahedra when coupled with suitable extensions of the Duffy transformation to a hexahedral region.

Appendix B

Tabulated L_2 errors

Role of dealiasing for higher order advection velocities



(a) $P_{adv} = 1$ (b) $P_{adv} = 2$

(c) $P_{adv} = 3$ (d) $P_{adv} = 4$

Figure B.1: L_2 errors vs. (Q_I, Q_V) using the *Local* dealiasing technique.

	$\mathbf{DG_{SEM}}$-EMM	$\mathbf{DG_{SEM}}$-LMM	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00258809184895	0.00517893325001	0.00258809184895	0.00517893325001
$Q = 5, Q_V = 6, Q_I = 5$	0.00245321968401	0.00484792022346	0.00245321968401	0.00484792022346
$Q = 5, Q_V = 7, Q_I = 5$	0.00245321968401	0.00484792022346	0.00245321968401	0.00484792022346
$Q = 5, Q_V = 5, Q_I = 6$	0.00274779163543	0.00520769016821	0.00274779163543	0.00520769016821
$Q = 5, Q_V = 5, Q_I = 7$	0.00274779163543	0.00520769016821	0.00274779163543	0.00520769016821
$Q = 5, Q_V = 6, Q_I = 6$	0.00238095562953	0.00483526988468	0.00238095562953	0.00483526988468
$Q = 5, Q_V = 7, Q_I = 7$	0.00238095562953	0.00483526988468	0.00238095562953	0.00483526988468

Table B.1: L_2 errors for *Local* strategy applied to the DG and FR formulations: Case A mesh as in Figure 4.5(a) for $P_{adv} = 1$.

	$\mathbf{DG_{SEM}}$-EMM	$\mathbf{DG_{SEM}}$-LMM	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00162175780389	0.00255493783002	0.00162175780389	0.00255493783002
$Q = 5, Q_V = 6, Q_I = 5$	0.00200593990694	0.00273091929262	0.00200593990694	0.00273091929261
$Q = 5, Q_V = 7, Q_I = 5$	0.00195899758668	0.00270213303084	0.00195899758668	0.00270213303084
$Q = 5, Q_V = 8, Q_I = 5$	0.00195899758668	0.00270213303084	0.00195899758668	0.00270213303084
$Q = 5, Q_V = 5, Q_I = 6$	0.00192599663912	0.00264115695244	0.00192599663912	0.00264115695244
$Q = 5, Q_V = 5, Q_I = 7$	0.00192322522336	0.00264249943410	0.00192322522336	0.00264249943410
$Q = 5, Q_V = 5, Q_I = 8$	0.00192322522336	0.00264249943410	0.00192322522336	0.00264249943410
$Q = 5, Q_V = 6, Q_I = 6$	0.00167138551874	0.00256594327336	0.00167138551874	0.00256594327336
$Q = 5, Q_V = 7, Q_I = 7$	0.00166192034989	0.00255904806344	0.00166192034989	0.00255904806344
$Q = 5, Q_V = 8, Q_I = 8$	0.00166192034989	0.00255904806344	0.00166192034989	0.00255904806344

Table B.2: L_2 errors for *Local* strategy applied to the DG and FR formulations: Case A mesh as in Figure 4.5(a) for $P_{adv} = 2$.

	$\mathbf{DG_{SEM}}$-EMM	$\mathbf{DG_{SEM}}$-LMM	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00145932994542	0.00170667104239	0.00145932994542	0.00170667104239
$Q = 5, Q_V = 6, Q_I = 5$	0.00180851746188	0.00195005809316	0.00180851746188	0.00195005809316
$Q = 5, Q_V = 7, Q_I = 5$	0.00171296996980	0.00185789594691	0.00171296996980	0.00185789594691
$Q = 5, Q_V = 8, Q_I = 5$	0.00171296996980	0.00185789594691	0.00171296996980	0.00185789594691
$Q = 5, Q_V = 5, Q_I = 6$	0.00171031388518	0.00188971296917	0.00171031388518	0.00188971296917
$Q = 5, Q_V = 5, Q_I = 7$	0.00171086707088	0.00189247015965	0.00171086707088	0.00189247015965
$Q = 5, Q_V = 5, Q_I = 8$	0.00171086707088	0.00189247015965	0.00171086707088	0.00189247015965
$Q = 5, Q_V = 6, Q_I = 6$	0.00143565622567	0.00167138551874	0.00143565622567	0.00167138551874
$Q = 5, Q_V = 7, Q_I = 7$	0.00143604324582	0.00168273243243	0.00143604324582	0.00168273243243
$Q = 5, Q_V = 8, Q_I = 8$	0.00143604324582	0.00168273243243	0.00143604324582	0.00168273243243

Table B.3: L_2 errors for *Local* strategy applied to the DG and FR formulations: Case A mesh as in Figure 4.5(a) for $P_{adv} = 3$.

	$\mathbf{DG_{SEM}}$-EMM	$\mathbf{DG_{SEM}}$-LMM	$\mathbf{FR_{DG}}$	$\mathbf{FR_{g2}}$
$Q = 5$	0.00126816984360	0.00131289321292	0.00126816984360	0.00131289321292
$Q = 5, Q_V = 6, Q_I = 5$	0.00140946714795	0.00141416909853	0.00140946714795	0.00141416909853
$Q = 5, Q_V = 7, Q_I = 5$	0.00136256229206	0.00138729451635	0.00136256229206	0.00138729451635
$Q = 5, Q_V = 8, Q_I = 5$	0.00136245047508	0.00138724135413	0.00136245047508	0.00138724135413
$Q = 5, Q_V = 9, Q_I = 5$	0.00136245047508	0.00138724135413	0.00136245047508	0.00138724135413
$Q = 5, Q_V = 5, Q_I = 6$	0.00134288748995	0.00134093637296	0.00134288748995	0.00134093637296
$Q = 5, Q_V = 5, Q_I = 7$	0.00134727708510	0.00135650237293	0.00134727708510	0.00135650237293
$Q = 5, Q_V = 5, Q_I = 8$	0.00134727957789	0.00135657503799	0.00134727957789	0.00135657503799
$Q = 5, Q_V = 5, Q_I = 9$	0.00134727957789	0.00135657503799	0.00134727957789	0.00135657503799
$Q = 5, Q_V = 6, Q_I = 6$	0.00126591291017	0.00130947841396	0.00126591291017	0.00130947841396
$Q = 5, Q_V = 7, Q_I = 7$	0.00126617520357	0.00130988788959	0.00126617520357	0.00130988788959
$Q = 5, Q_V = 8, Q_I = 8$	0.00126617189683	0.00130988726422	0.00126617189683	0.00130988726422
$Q = 5, Q_V = 9, Q_I = 9$	0.00126617189683	0.00130988726422	0.00126617189683	0.00130988726422

Table B.4: L_2 errors for *Local* strategy applied to the DG and FR formulations: Case A mesh as in Figure 4.5(a) for $P_{adv} = 4$.

Link between geometrical- and PDE-aliasing

	Case A: $[\mathbf{M}(J)_{Q+1}$ - $\mathbf{M}(J)_Q]_{L_2}$			
	$J \in \mathbb{P}^1$	$J \in \mathbb{P}^2$	$J \in \mathbb{P}^3$	$J \in \mathbb{P}^4$
Q_{min} - $Q_{min}+1$,	0.7244	0.3144	0.1773	0.1176
Global = 4	0.0160			
***Global* = 5**	**4.4222e-16**			
Global = 6		0.0162		
Global = 7		0.0055		
***Global* = 8**		**3.6038e-15**	0.0135	
Global = 9			0.0079	
Global = 10			0.0027	0.0105
***Global* = 11**			**1.2322e-12**	0.0080
Global = 12				0.0047
Global = 13				0.0016
***Global* = 14**				**3.4482e-15**

	Case B: $[\mathbf{M}(J)_{Q+1}$ - $\mathbf{M}(J)_Q]_{L_2}$			
	$J \in \mathbb{P}^2$	$J \in \mathbb{P}^4$	$J \in \mathbb{P}^6$	$J \in \mathbb{P}^8$
Q_{min} - $Q_{min}+1$,	0.8368	0.3472	-	-
Global = 4	0.0210			
***Global* = 5**	**1.1943e-15**			
Global = 6		0.0351		
Global = 7		0.0096		
***Global* = 8**		**4.2100e-15**		

Table B.5: L_2 norm of the difference between mass matrices calculated with Q and $Q+1$ GLL points for the four orders considered and using both the meshes in Fig.(4.8).

Appendix C

Riemann invariants of the Euler system

In this appendix, we describe a characteristic approach based on the Riemann invariants which can be used alternatively to the approach described in section 3.4.1 and produces identical results. This approach is particularly effective in damping spurious reflections from the boundaries and is also referred to as non-reflective farfield BCs. According to the hyperbolic nature of the Euler equations the flux evaluated at the boundary is a combination of the information coming from inside and outside the domain. In order to explain how to effectively apply characteristic boundary conditions, we start by defining the Riemann invariants which are derived through the characteristic analysis of the Euler equations. Specifically, we consider a normal frame of reference with respect to a given interface between two elements through Eq. 3.20 where we can define the normal velocity associated to the interface as $u_n = \boldsymbol{v} \cdot \mathbf{n}$. We additionally can derive the vector of the eigenvalues of the Euler equations, $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = \begin{bmatrix} u_n + c \\ u_n \\ u_n \\ u_n \\ u_n - c \end{bmatrix} \tag{C.1}$$

The eigenvalues $\boldsymbol{\lambda}$ defined in Eq. C.1 are the velocities at which the information travels along the characteristic lines $d\boldsymbol{x}/dt$. In particular, the characteristic variables being propagated along the characteristic lines are defined as:

$$\boldsymbol{r} = \mathbf{L}\boldsymbol{u}, \tag{C.2}$$

where $\mathbf{L}$ is the matrix of the left eigenvectors and they satisfy the following advec-

tion equation:

$$\frac{\partial \boldsymbol{r}}{\partial t} + diag(\boldsymbol{\lambda}) \cdot \nabla \boldsymbol{r} = 0, \tag{C.3}$$

where $diag(\boldsymbol{\lambda})$ is the diagonal matrix of the eigenvalues. The first and the last equations, associated to the smallest and to the largest eigenvalues, are two genuinely nonlinear fields and they are either shock or rarefaction waves.

These equations can be integrated using the isentropic assumption $p/\rho^{\gamma} = const$, producing the following two expressions:

$$r^{-} = u_n - \frac{2c}{\gamma - 1}, \qquad r^{+} = u_n + \frac{2c}{\gamma - 1}, \tag{C.4}$$

which are known as Riemann invariants associated to the characteristic waves r^{-} and r^{+}. Having defined the Riemann invariants, we can now define the farfield non-reflective BCs which include two cases: infow ($u_n < 0$) and outflow ($u_n > 0$).

Inflow ($u_n < 0$)

In the case of a subsonic inflow region, where the normal Mach number at the inflow boundary is defined as $M_n = |u_n|/c < 1$ the incoming Riemann invariant r^{-} is associated to the free-stream values of the conserved variables - i.e. $\boldsymbol{u}_{\infty} = [\rho_{\infty}, (\rho u)_{\infty}, (\rho v)_{\infty}, (\rho w)_{\infty}, E_{\infty}]^T$, while the outgoing Riemann invariant r^{+} is associated to values of the conserved variables inside the domain $\boldsymbol{u}_{in} = [\rho_{in}, (\rho u)_{in}, (\rho v)_{in}, (\rho w)_{in}, E_{in}]^T$. Note that the subscript "in" here indicates variables in the interior of computational domain. So the appropriate Riemann invariants to consistently solve the hyperbolic problem are:

$$r^{-} = u_{n,\infty} - \frac{2c_{\infty}}{\gamma - 1}, \quad r^{+} = u_{n,in} + \frac{2c_{in}}{\gamma - 1}, \tag{C.5}$$

where:

$$u_{n,in} = \boldsymbol{u}_{in} \cdot \mathbf{n}, \quad u_{\infty} = \boldsymbol{v}_{\infty} \cdot \mathbf{n}, \quad c_{in}^2 = \frac{\gamma p_{in}}{\rho_{in}}, \quad c_{\infty}^2 = \frac{\gamma p_{\infty}}{\rho_{\infty}}, \tag{C.6}$$

with c_{in} and c_{∞} being the free-stream and interior values of the speed of sound, respectively.

At the boundary "b" we know that the two Riemann invariants assume a given value that depends on the boundary quantities and we can write the following relations:

$$r_b^{-} = u_{n,b} - \frac{2c_b}{\gamma - 1} = r^{-}, \quad r_b^{+} = u_{n,b} + \frac{2c_b}{\gamma - 1} = r^{+}. \tag{C.7}$$

From Eq. (C.7) we can therefore calculate a velocity and a speed of sound at the boundary as follows:

$$u_{n,b} = \frac{r^{-} + r^{+}}{2}, \quad c_b = \frac{(\gamma - 1)(r^{+} - r^{-})}{4}. \tag{C.8}$$

The normal velocity $u_{n,b}$ can then be used to calculate the three Cartesian components of the velocity $\boldsymbol{v}_b = [u_b, v_b, w_b]^T$ at the boundary:

$$\boldsymbol{v}_b = \boldsymbol{v}_\infty + (u_{n,b} - \boldsymbol{v}_\infty \cdot \mathbf{n})\mathbf{n}. \tag{C.9}$$

Since the flow is entering the domain and under the assumption that there are no discontinuities in the solution (e.g. shock waves), the entropy at the boundary is equal to the free-stream entropy and, therefore, we can write the following relation:

$$s_b = \frac{c_\infty^2}{\gamma \rho_\infty^{\gamma-1}}. \tag{C.10}$$

Now, using Eq. C.8 for the speed of sound and Eq. C.10 for the entropy, the density and the pressure at the boundary can be computed as follows:

$$\rho_b = \left(\frac{c_b^2}{\gamma s_b}\right), \quad p_b = \frac{\rho_b c_b^2}{\gamma}. \tag{C.11}$$

Having fully defined the thermodynamic state (ρ_b, p_b) and the velocities $\boldsymbol{v}_b = [u_b, v_b, w_b]^T$ at the boundary (based on the characteristic analysis of the Euler equations) it is possible to calculate the characteristic boundary state as:

$$\boldsymbol{u}_{BC} = \left\{ \begin{array}{c} \rho_{BC} \\ (\rho u)_{BC} \\ (\rho v)_{BC} \\ (\rho w)_{BC} \\ E_{BC} \end{array} \right\} = \left\{ \begin{array}{c} \rho_b \\ \rho_b u_b \\ \rho_b v_b \\ \rho_b w_b \\ E_b \end{array} \right\}. \tag{C.12}$$

where the total energy E_b is calculated through Eq. 3.4 . The subsonic in ow BCs are finally obtained by using Eq. C.12 an the the boundary numerical flux can be calculated through Eq. 3.48.

If the flow at the inflow boundary is supersonic, $M_n = |u_n|/c \geq 1$, the above derivation holds, with the only exception that there is no outgoing characteristic wave, thus the outgoing Riemann invariant assumes the following expression:

$$r^+ = u_{n,\infty} + \frac{2c_\infty}{\gamma - 1}. \tag{C.13}$$

Outflow ($u_n > 0$)

In the case of a subsonic outflow region, where the normal Mach number at the outflow boundary is defined as $M_n = |u_n|/c < 1$ the incoming Riemann invariant r^- is associated to the free-stream values of the conserved variables - i.e. $\boldsymbol{u}_\infty = [\rho_\infty, (\rho u)_\infty, (\rho v)_\infty, (\rho w)_\infty, E_\infty]^T$, while the outgoing Riemann invariant r^+ is associated to values of the conserved variables inside the domain

$\boldsymbol{u}_{in} = [\rho_{in}, (\rho u)_{in}, (\rho v)_{in}, (\rho w)_{in}, E_{in}]^T$. Note that the subscript "in" here indicates variables in the interior of computational domain. So the appropriate Riemann invariants to consistently solve the hyperbolic problem are:

$$r^- = u_{n,\infty} - \frac{2c_\infty}{\gamma - 1}, \quad r^+ = u_{n,in} + \frac{2c_{in}}{\gamma - 1}. \tag{C.14}$$

At the boundary "b" we can write the same relations as in Eq. C.7 and compute the normal velocity $u_{n,b}$ and the speed of sound c_b cvia Eq. (C.8). The normal velocity $u_{n,b}$ can then be used to calculate the the three Cartesian components of the velocity $\boldsymbol{v}_b = [u_b, v_b, w_b]^T$ at the boundary using Eq. C.9.

Since the flow is exiting the domain and under the same assumption made for the inflow case (i.e. no discontinuities), the entropy at the boundary is extrapolated from the interior and is equal to:

$$s_b = \frac{c_{in}^2}{\gamma \rho_{in}^{\gamma-1}}. \tag{C.15}$$

The density and the pressure at the boundary can be calculated with Eq. (C.11). Therefore we can evaluate all the conserved variables as in Eq. (C.12). The flux is then evaluated Eq. 3.48.

Note that, if the flow at the ouflow boundary is supersonic, $M_n = |u_n|/c \geq 1$, the above derivation holds, with the only exception that there is no incoming characteristic wave, thus the incoming Riemann invariant assumes the following expression:

$$r^- = u_{n,in} - \frac{2c_{in}}{\gamma - 1}. \tag{C.16}$$